Sensors and Materials, Vol. 35, No. 7 (2023) 2241–2264
MYU Tokyo

S & M 3320

A Variational AutoEncoder (VAE)-based Deep Learning Anomaly Detection Model for Industrial Products with Dynamic Weights Assigned to Loss Function

Shunta Nakata,[1] Takehiro Kasahara,[2] and Hidetaka Nambo[1*]

[1]Division of Electrical and Computer Science, Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa, Ishikawa 920-1192, Japan
[2]Industrial Research Institute of Ishikawa, Kanazawa, Ishikawa 92−8203, Japan

(Received January 16, 2023; accepted June 6, 2023)

Keywords: anomaly detection, industrial product, variational autoencoder, deep learning, generative model, loss function, unsupervised learning

In the industrial field, deep-learning-based image anomaly detections are attracting attention because of some of their advantages. The deep-learning-based models can overcome the shortcomings of traditional methods, such as human eye detection and rule-based machine detection. When using deep learning, which has many advantages, one of the limitations is that anomalous products are difficult to obtain. Since most industrial products do not have defects, unsupervised learning detection models are strongly required. We propose a new model based on the variational autoencoder (VAE), which is a generative model applicable to detection by unsupervised learning. VAE is a model for optimizing parameters or latency based on a loss function that is the sum of several terms, and in our proposed method, original weights are given to these terms. In addition, our model dynamically and adaptively explores a ratio of weights. We have developed a dynamic weighted VAE adapted to area under the receiver operating characteristic curve (AUROC, AUC) using validation data. We have already reported the efficiency of the AUC-adapted VAE; however, this method is not unsupervised learning, and a method that does not use validation data was desired. In this paper, we discuss the previous method in more detail and describe the new method, which is fully unsupervised learning, by conducting additional experiments. The results of several experiments show that the proposed method is potentially effective for some actual industrial product image datasets while maintaining unsupervised learning.

1. Introduction

In modern society, parts and products are mass-produced in various fields including the industrial field. In the production process, some products have defects, so it is necessary to identify and remove anomalous products. At present, for example, inspectors are often hired to detect defects visually. However, this visual inspection has several problems. The first is the training of inspectors. Inspectors make judgments mainly on the basis of their experience, and it

*Corresponding author: e-mail: hide_nambo@staff.kanazawa-u.ac.jp
https://doi.org/10.18494/SAM4237

ISSN 0914-4935
https://myukk.org/

takes time for them to gain that experience. In addition, since the judgment criteria depend on each person, when there are multiple inspectors, there may be variations in judgment.

Therefore, automated inspection using product images is in considerable demand in the industrial field. Until now, rule-based image inspection, which is set by engineers who are familiar with the product, has been widely used. However, it cannot cope with unknown defects or anomalies for which rules are difficult to establish. There are high expectations for image anomaly detection based on deep learning, which has the potential to solve these problems.

Anomaly detection models based on deep learning include, for example, those based on convolutional neural networks (CNNs). A discriminator can be constructed by training a large number of images with normal and anomaly labels associated with them. However, since the number of anomalies is generally much smaller in industrial products than in normal products, it is difficult to collect a large amount of anomaly data. This means that supervised classification learning is not suitable as an anomaly detection model for industrial products. To solve this problem, methods using image generation by autoencoder (AE),[(1)] variational autoencoder (VAE),[(2)] or generative adversarial network (GAN)[(3)] have been proposed. AE and VAE are deep learning models that can be trained using only normal data.

In our previous work, we mainly constructed anomaly detectors for the Resin Products dataset described in Sect. 4. Our GAN-based model requires about 3 s per image of a dataset for detection, so it was necessary to develop a model that enables faster detection for actual implementation in the production process. On the other hand, our VAE-based model requires about 0.1 s or shorter per image and delivers high performance with AUC = 90% or more. However, for practical use, we believe that the AUC should be as close to 100% as possible. In addition, we aim to create a model that is capable of detecting any industrial product with a high degree of accuracy.

In this paper, we focus on the anomaly detection method for industrial products using VAE and propose a new detection model based on VAE. The purpose of this study is to compare the results between the new model and the conventional VAE through several experiments using actual industrial product images. The main contribution of this paper is that our new model may provide greater performance for any image dataset than the conventional VAE by learning adaptively. In particular, it is significant that VAE adapted to loss improves accuracy with fully unsupervised learning.

We have already reported on similar research;[(4)] however, in this paper, we include newly obtained and more detailed results and a more elaborate discussion by conducting additional experiments with a new actual industrial product dataset. These elements enhanced the potential to demonstrate the superiority of our proposed method. Therefore, in this paper, we show that our method is approaching a competent anomaly detection system using artificial intelligence, which is highly desired in the industrial field.

2. Literature Review

Because of the advantages of unsupervised learning, various types of generative models have been built by many researchers. Figure 1 shows the generative models classified by a tree

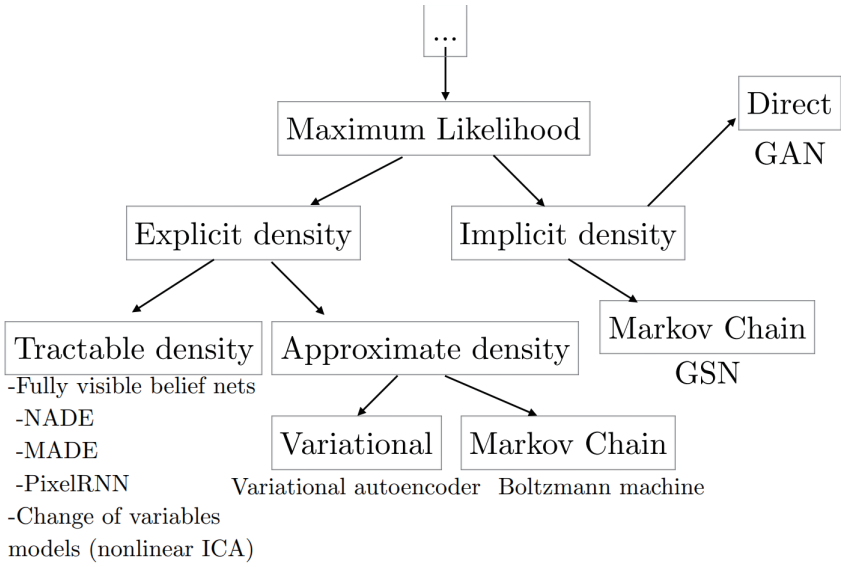


Fig. 1. Classification of generative models.[5]

structure. The models are generally optimized by maximizing the likelihood. On the left branch of this tree, models construct an explicit density and thus an explicit likelihood that can be maximized. Among these explicit density models, the density may be computationally tractable. The models belonging to tractable density are neural autoregressive distribution estimation (NADE),[6] masked autoencoder for distribution estimation (MADE),[7] and so forth. Or it may be intractable, meaning that to maximize the likelihood, it is necessary to make either variational approximations or Monte Carlo approximations or both. On the right branch, the model provides some way of interacting less directly with the probability distribution. Typically, the indirect means of interacting with the probability distribution is the ability to draw samples from it. The one that uses a Markov chain is called generative stochastic networks (GSNs).[8]

Among them, VAE and GAN are often used for anomaly detection. Since VAE and GAN have their advantages and disadvantages, it is necessary to consider the data and conditions.

VAE is a combination of two neural networks. The first is called the encoder, which learns hidden latent representation from input data and converts input into an encoding vector following a normal distribution. The second is called the decoder, which generates data as output from the latent vector. As does VAE, GAN consists of two types of neural network. One is called the generator, which takes random noise as input and generates data. The other is called the discriminator, which distinguishes whether the generated data is the pre-supplied supervised data or the data output from the generator. The generator is trained to generate data that the discriminator misjudges as the supervised data. On the other hand, the discriminator is trained not to misjudge data output from the generator.[3,9] GAN can generate images more clearly than VAE; however, GAN has several problems. One of the most important problems is the instability of training. It may focus on only a few patterns or specific objects and generate them repeatedly. This phenomenon is called mode collapse, which occurs when the generator learns to associate similar outputs with several different inputs and fails to learn a rich feature representation.[10] In

addition, owing to the complexity of the network structure, GAN requires greater computational resources than VAE. Therefore, we decided to use VAE because of the stability of its training process and fast detection.

Researchers have proposed VAEs with various innovations depending on the purpose. For instance, in β-VAE,[11] the independence of each component of the latent variable is increased by adjusting the constraint using the Kullback–Leibler distance (KLD) with the variable β (>>1). This allows components such as the angle of each part in images to correspond to specific components of the latent variable by obtaining greater interpretability that comes with disentangled representation. When using β-VAE for anomaly detection, it is expected to be able to richly represent even small normal areas. However, we believe that this is not so simple in practice because focusing on the KLD may neglect the reconstruction error, and the reconstructed images become blurred more easily than with a normal VAE. There is a trade-off relationship between the KLD distance and the reconstruction error; therefore, β needs to be fine-tuned.

3. Methods

3.1 VAE

VAE aims to find the parameters θ and ϕ that best represent inputs $\boldsymbol{x}$ is sing the maximum likelihood method for the generative model $p_\theta(\boldsymbol{x})$ represented by the parameters ϕ in the encoder and θ in the decoder. Figure 2 shows the network structure of VAE. The latent variable $\boldsymbol{z}$ is sampled following a normal distribution $N(0,1)$ with the mean vector $\boldsymbol{\mu}_z$ and the standard deviation vector $\boldsymbol{\sigma}_z$. Therefore, the encoder outputs $\boldsymbol{\mu}_z$ and $\boldsymbol{\sigma}_z$ in the model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ of the latent variable $\boldsymbol{z}$ after receiving the input $\boldsymbol{x}$. The decoder outputs the mean vector $\boldsymbol{\mu}_{x'}$ and the standard deviation vector $\boldsymbol{\sigma}_{x'}$ in the model $p_\theta(\boldsymbol{x}'|\boldsymbol{z})$ of the output $\boldsymbol{x}'$. Because of the difficulty of maximizing

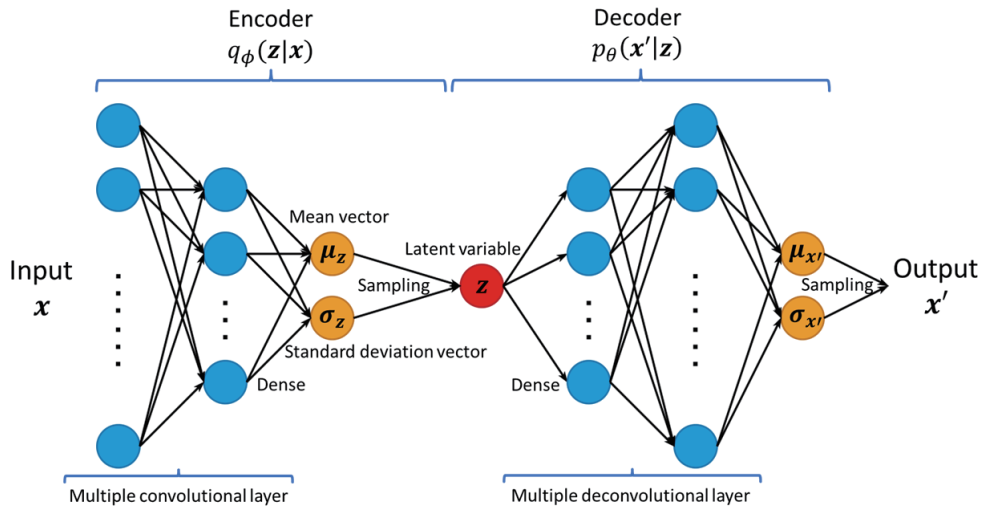


Fig. 2. (Color online) Network structure of VAE.

$p_\theta(\boldsymbol{x})$ or the integral of its log-likelihood $\log(p_\theta(\boldsymbol{x}))$, VAE optimizes the parameters θ and ϕ by minimizing the evidence lower bound (ELBO) $\mathcal{L}_{VAE}(\boldsymbol{x})$ instead.

$$\mathcal{L}_{VAE}(\boldsymbol{x}) = R_{VAE}(\boldsymbol{x}) + D_{VAE}(\boldsymbol{x}) \tag{1}$$

The term $R_{VAE}(\boldsymbol{x})$ shows how well the VAE can reconstruct the input sequence from the latent space, while $D_{VAE}(\boldsymbol{x})$ measures how similar the two data distributions are to each other. Here, when VAE trains data, $\mathcal{L}_{VAE}(\boldsymbol{x})$ is used as the loss function. The regularized term $D_{VAE}(\boldsymbol{x})$ is the Kullback–Leibler divergence between $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ when $\boldsymbol{z}$ follows the normal distribution and can be transformed into Eq. (2).

$$\begin{aligned} D_{VAE}(\boldsymbol{x}) &= D_{KL}\left(q_\phi\left(\boldsymbol{z}|\boldsymbol{x}\right)\|p\left(\boldsymbol{z}\right)\right) \\ &= \frac{1}{2}\sum_{j=1}^{N_z}\left\{-\log\left(\sigma_{z_j}^2\right) - 1 + \sigma_{z_j}^2 + \mu_{z_j}^2\right\} \end{aligned} \tag{2}$$

Here, j is the index of $\boldsymbol{z}$. The term $R_{VAE}(\boldsymbol{x})$ is called a reconstruction error.

$$R_{VAE}(\boldsymbol{x}) = -\log p_\theta(\boldsymbol{x}|\boldsymbol{\mu}_z) \tag{3}$$

The approach to transforming Eq. (3) differs depending on what distribution is assumed for the reconstruction model $p_\theta(\boldsymbol{x}'|\boldsymbol{z})$ of the latent variable $\boldsymbol{z}$ using a decoder. If $p_\theta(\boldsymbol{x}'|\boldsymbol{z})$ is assumed to follow the Bernoulli distribution, $R_{VAE}(\boldsymbol{x})$ is expressed as Eq. (4), or if the normal distribution, it is expressed as Eq. (5).

$$R_{VAE_{2terms}}(\boldsymbol{x}) = \sum_{i=1}^{D}\left\{x_i \log x'_i + (1-x_i)\log(1-x'_i)\right\} = E_{BC}(\boldsymbol{x}, \boldsymbol{x'}), \tag{4}$$

where E_{BC} means the binary cross entropy.

$$R_{VAE_{3terms}}(\boldsymbol{x}) = A_{VAE}(\boldsymbol{x}) + M_{VAE}(\boldsymbol{x}), \tag{5}$$

where

$$A_{VAE}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{N_x}\log(2\pi\sigma_{x_i}^2), \tag{6}$$

$$M_{VAE}(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{N_x}\frac{(\mu_{x_i} - x_i)^2}{\sigma_{x_i}^2}. \tag{7}$$

3.2 Anomaly detection using VAE

VAE can be used for anomaly detection because of its ability to reconstruct images. The anomaly detection is carried out in five steps as follows.

First, data preprocessing is conducted to make them suitable for learning. In general, image data are normalized or resized to fit the model structure. Then, VAE trains the network using only images without defects. It is desirable to prepare as much data as possible for training. After the training, regardless of whether the input data is normal or abnormal, the VAE outputs an image. If the input data is normal, the VAE tries to output an image, which is the same as the input, and reconstructs the unique parts of each product that does not have defects. If not, it outputs an image like a normal product image or an unfamiliar image (Fig. 3). In the 4th step, the anomaly scores of each input image are calculated. In this calculation, for instance, the loss functions of VAE [$\mathcal{L}_{VAE}(\boldsymbol{x})$] mean squared error (MSE), etc. between input and output are used as anomaly scores. This allows us to generate a score histogram that indicates the relationships between the anomaly scores and the number of images for which the scores are calculated. Finally, a threshold to classify either normal or anomaly is determined. The threshold is often established using the maximum of the Youden index J as in Eq. (8).

$$\begin{aligned} J &= Sensitivity + Specificity - 1 \\ &= \frac{N_{\mathrm{TruePositive}}}{N_{\mathrm{TruePositive}} + N_{\mathrm{FalseNegative}}} + \frac{N_{\mathrm{TrueNegative}}}{N_{\mathrm{TrueNegative}} + N_{\mathrm{FalsePositive}}} - 1 \end{aligned} \tag{8}$$

Here, N_{Class} is the number of data points belonging to that class. Sensitivity is the percentage of the data that are predicted to be abnormal by the discriminant out of the data that are abnormal, and specificity is the percentage of the data that are predicted to be normal by the discriminant out of the data that are normal. If the objective is to detect anomalies at all costs, the threshold should be set at a sensitivity of 100%. However, we used the Youden index as the threshold setting index because we would like normal images to be discriminated to some extent. To

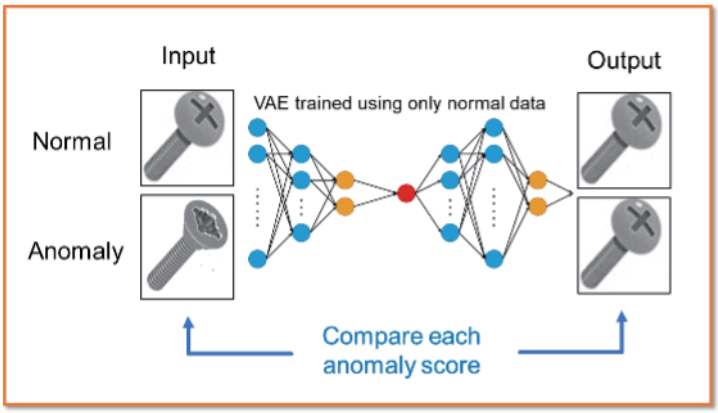


Fig. 3. (Color online) Reconstruction of normal or anomaly images.

measure the performance of the detector, the AUROC, which is the area of the ROC curve, is used. The ROC curve plots the changes in the true positive rate (TPR) and false positive rate (FPR) for different classification thresholds for determining the predicted probability of a class as positive.

3.3 Proposed algorithm

As stated in Sect. 3.1, VAE trains by optimizing the following loss functions:

$$\mathcal{L}_{VAE_{2terms}}(\boldsymbol{x}) = E_{BC}(\boldsymbol{x}, \boldsymbol{x'}) + D_{VAE}(\boldsymbol{x}), \tag{9}$$

or

$$\mathcal{L}_{VAE_{3terms}}(\boldsymbol{x}) = M_{VAE}(\boldsymbol{x}) + A_{VAE}(\boldsymbol{x}) + D_{VAE}(\boldsymbol{x}). \tag{10}$$

To obtain greater interpretability that comes with disentangled representation, Higgins *et al.* proposed β-VAE.[11] By referring to β-VAE, the weights (α, β) or (w_M, w_A, w_D) of each term in $\mathcal{L}_{VAE}$ as Eqs. (11) and (12) are added.

$$\mathcal{L}_{VAE_{2terms}}(\boldsymbol{x}) = \alpha E_{BC}(\boldsymbol{x}, \boldsymbol{x'}) + \beta D_{VAE}(\boldsymbol{x}) \tag{11}$$

$$\mathcal{L}_{VAE_{3terms}}(\boldsymbol{x}) = w_M M_{VAE}(\boldsymbol{x}) + w_A A_{VAE}(\boldsymbol{x}) + w_D D_{VAE}(\boldsymbol{x}) \tag{12}$$

Here, (w_M, w_A, w_D, α, β) $\in (0, \infty)$. These weights can be regarded as hyperparameters; therefore, it may be possible to make the VAEs learn to be more expressive than convention by tuning them in some way.

Moreover, we considered changing the values of these parameters at specific learning steps. We named the loss function with these parameters the adaptive weighted loss (AWL). For example, the weights can be set as a function for epochs, loss, etc.:

$$w = f(Epochs), \tag{13}$$

$$w = f(\mathcal{L}_{VAE}), \tag{14}$$

where $w \in (w_M, w_A, w_D, \alpha, \beta)$.

3.3.1 Example 1: Adaptive to AUC

Here, an algorithm using adaptive weights related to AUC is built as shown in Fig. 4, and patterns of weights (α, β) and (w_M, w_A, w_D) are set as shown in Tables 1 and 2, respectively. The key point of this algorithm is that it sets up a function for the AUCs calculated at each epoch during training. It watches the AUC values and decides whether to change the pattern of the weights depending on whether it is improving or not. At the set update frequency F_{Update}, it compares the previous AUC mean with the AUC mean before that and changes the weight pattern (α, β) or (w_M, w_A, w_D) if the AUC has not improved as shown in Figs. 5 and 6. This pattern of changes goes from I to II, then II to III, and finally back to pattern I, and so on. Here, in Tables 1 and 2, the weight of the term to be disabled is set to 0; however, it should be given a very small value to prevent loss divergence in practice.

F_{Update} : Epoch frequency of updating weight pattern,
AUC_{before} : Average AUC on ($Epoch_{current} - 2F_{Update} + 1$) to ($Epoch_{current} - F_{Update}$) epochs,
AUC_{after} :Average AUC on ($Epoch_{current} - F_{Update} + 1$) to $Epoch_{current}$ epochs.

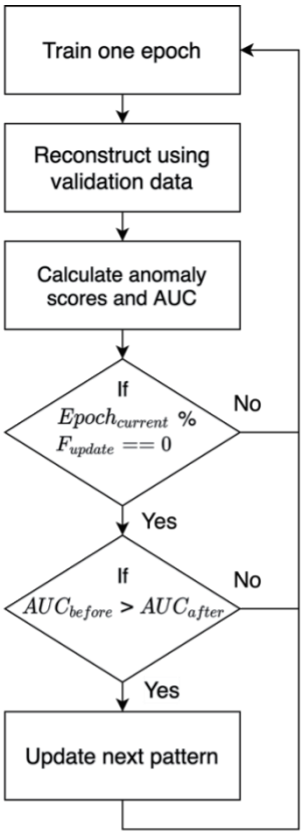


Table 1
An example of the pattern (α, β).

Pattern	α	β
I	1	1
II	1	0
III	0	1

Table 2
An example of the pattern (w_M, w_A ,w_D).

Pattern	w_M	w_A	w_D
I	1	1	1
II	1	0	0
III	0	1	0
IV	0	0	1
V	1	1	0
VI	1	0	1
VII	0	1	1

Fig. 4. Flowchart of an example of the AWL algorithm.

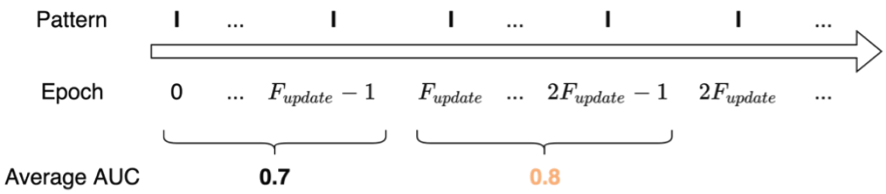


Fig. 5. (Color online) Patterns when AUC improves.

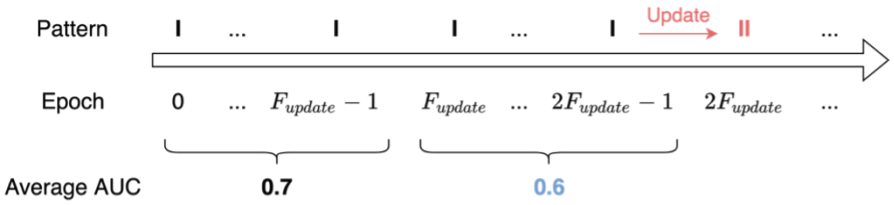


Fig. 6. (Color online) Patterns when AUC does NOT improve.

3.3.2 Example II: Adaptive to loss

To improve accuracy in fully unsupervised learning, we built an algorithm of the AWL related to $\mathcal{L}_{VAE}$ as shown in Fig. 7 and set patterns of weights as shown in Table 3. The terms (α, β) are the weights to be set before learning, and α_1, β_1 are the weights to be set after learning between certain epochs. The term $w_* \in (\alpha_*, \beta_*)$ is a specific value ranging from 0 to 1. Since it is considered undesirable to change the pattern of weights abruptly at a given epoch, the weights are changed as gradually and as continuously as possible between arbitrary epochs using an appropriate function [Eq.(14)]. The function should be monotonically increasing or decreasing. This means, for example, that in Pattern II, α is fixed at 1 and β is gradually lowered to 0 between certain epochs.

In the method shown in Fig. 7, training is initially performed with the conventional weight pattern until the E_1 epoch, and then, from the E_1 to E_2 epochs, training is performed with a pattern of decreasing α or β (Patterns I–III) using the saved model. Then, several loss values including the loss obtained by the conventional method at the E_2 epoch are compared, and the pattern with the smallest loss is regarded as the pattern with the most advanced learning. However, when implemented, they should not be set as in the values of α_2 and β_2 in Table 3. This is because we understood the weakness of this setting method from the results of our pre-experiment: when the weight approaches 0, the term becomes indifferent and consequently approaches divergence as shown in Fig. 8. To prevent this phenomenon, the algorithm stops decreasing the weight when the slope of the value of the term pertaining to the weight becomes positive. The weight at the point at which the positive or negative slope switches is saved as w_*. From the $E_2 + 1$ to E_3 epochs, exploration is performed again using the model with the lowest loss obtained at the E_2 epoch. In order not to change the weights abruptly, training is continued

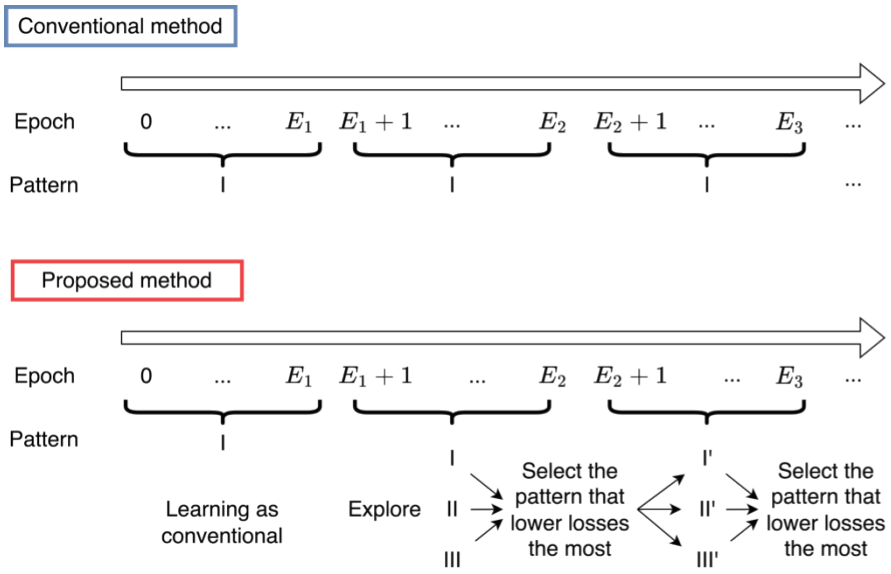


Fig. 7. (Color online) Difference between conventional and proposed methods.

Table 3
An example of the vector (α, β) patterns.

Pattern	(α_1, β_1)	(α_2, β_2)
I	(1, 1)	(1, 1)
II	(1, 1)	(1, 0)
III	(1, 1)	(0, 1)
I'	(α_*, β_*)	(α_*, β_*)
II'	(α_*, β_*)	(α_*, 0)
III'	(α_*, β_*)	(0, β_*)

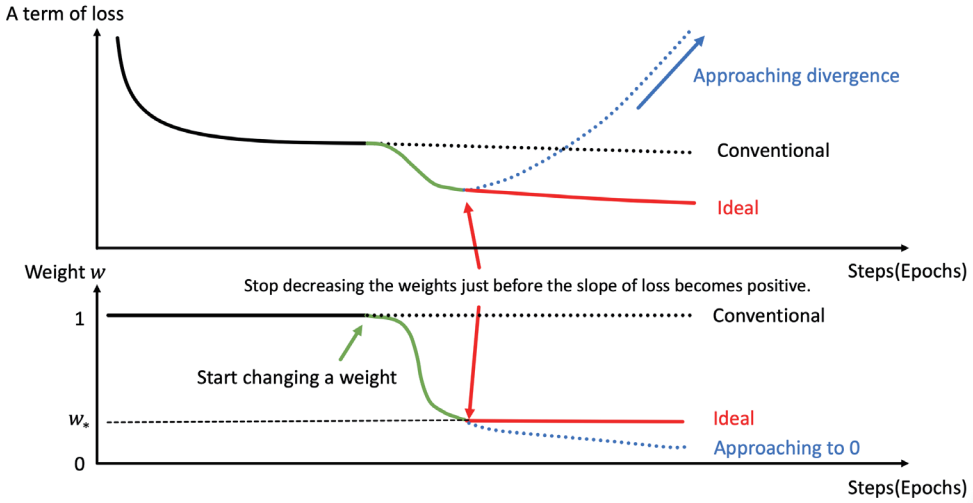


Fig. 8. (Color online) Relationships between transitions of a term of loss and a weight w using the proposed method.

using the w_* values of the optimal model at the $E_1 + 1$ to E_2 epochs as the initial values of the weights. Then, the pattern with the lowest loss is selected at the E_3 epoch. As shown in Fig. 9, the resulting model is expected to have lower losses than the conventional model, which may result in improved detection performance.

The description for the case where three terms (w_M, w_A, w_D) are used is omitted; however, it is similar to what it has been presented so far.

4. Experiment I: VAE Adapted to AUC

To evaluate the effectiveness of the method, we compared the performances of the conventional and AUC-adapted VAE following the procedure described in Sects. 3.2 and 3.3.1.

4.1 Setup

4.1.1 Datasets

In this study, we use actual industrial product datasets from two companies as shown in Table 4 and Figs. 10–12. All of these datasets consist of 8-bit, grayscale images. Images of Bellows are characterized by a regular jagged structure in the horizontal direction, and there are noises that are not defects, depending on the light exposure in the shooting environment. Although images of the Resin Product datasets cannot be published due to confidentiality, they can be presented in the illustrations shown in Figs. 11 and 12. The outermost white square is the main body of the square-shaped product, whereas the inner circle is the device used to take the photograph. Therefore, only defects that are burrs or short shots that appear on the outside square part are considered for the square-shaped Resin Product. Images of the comb-shaped Resin Product dataset were copied eight times and set to 10816 copies for accuracy. Note that Figs. 11 and 12 are not actual data for confidentiality reasons.

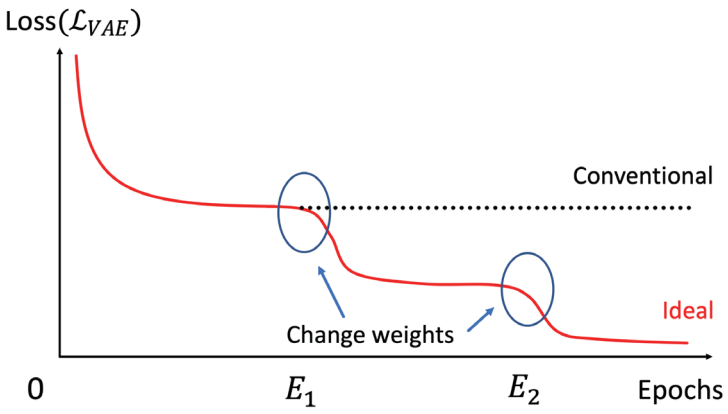
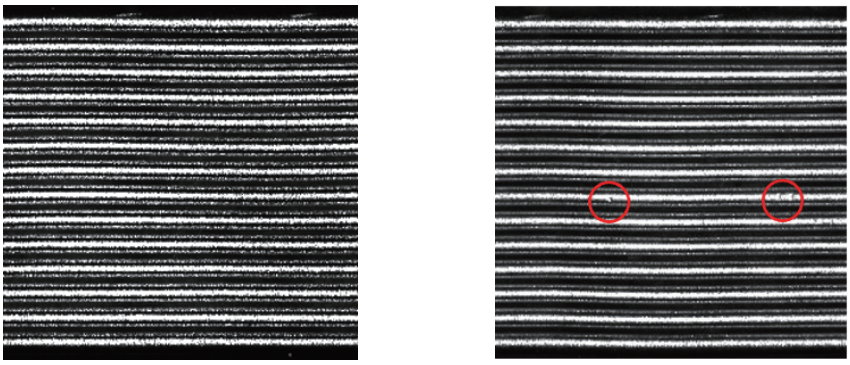


Fig. 9. (Color online) Ideal loss reduction using the proposed method.

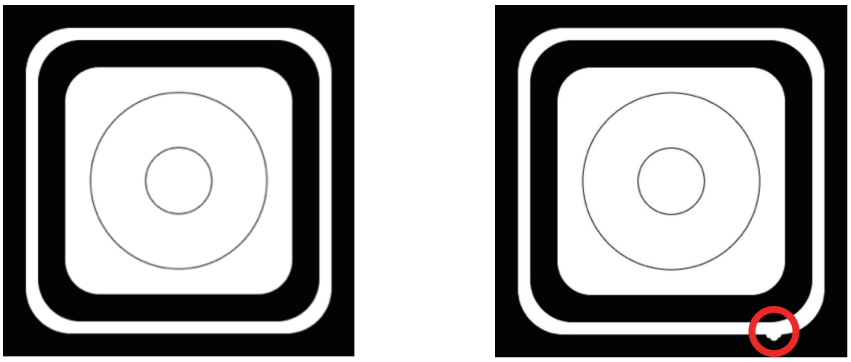
Table 4
Dataset detail.

Dataset	Number of training data sets (Only normal)	Number of evaluation data sets (Normal + Anomaly)	Size (pix)
Bellows	28000	90 + 104	256 × 256
Square-shaped Resin Product	10000	1000 + 1000	512 × 512
Comb-shaped Resin Product	1352 (×8)	72 + 100	320 × 320



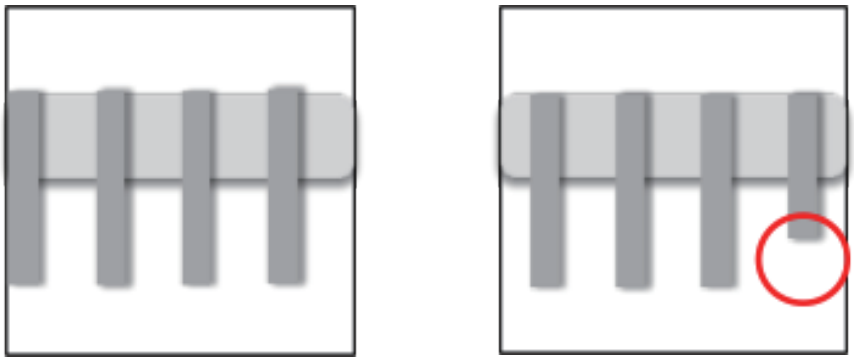
(a) (b)

Fig. 10. (Color online) Examples of the Bellows dataset: (a) normal and (b) anomaly.



(a) (b)

Fig. 11. (Color online) Examples of the square-shaped Resin Product dataset: (a) normal and (b) anomaly.



(a) (b)

Fig. 12. (Color online) Examples of the comb-shaped Resin Product dataset: (a) normal and (b) anomaly.

4.1.2 Conditions

We built the structure of AWL VAE adapted to AUC as Fig. 13. Since the parameter F_{Update} can be variable, the AWL VAE can be assumed as conventional VAE when $F_{Update} = N_{Epoch}$. This allows us to compare their performances easily under the same condition of parameters except for F_{Update}. Table 5 shows the parameter setup. The adaptive weights for the terms to be disabled were given as 10^{-10}. We set $F_{Update} = 100$ for the conventional VAE and $F_{Update} = 3$ for the AWL VAE. In this process, it is conceivable that overfitting may occur. In this experiment, the number of epochs is set as a constant for comparison with conventional methods; however, in actual implementation, it should be optimized appropriately to prevent overfitting. For example, the model at the epoch with the highest AUC should be used as the optimized model, or training should be stopped when the decrease in loss becomes relatively slow.

When calculating anomaly scores, MSE scores [Eq. (15)] are used for the Bellows and comb-shaped Resin Product datasets. For the Resin Products dataset, the Max1p score [Eq. (16)] is used, which is the maximum value of each pixel difference between input and output, because we had a better result with Max1p than MSE for this dataset using the conventional VAE.

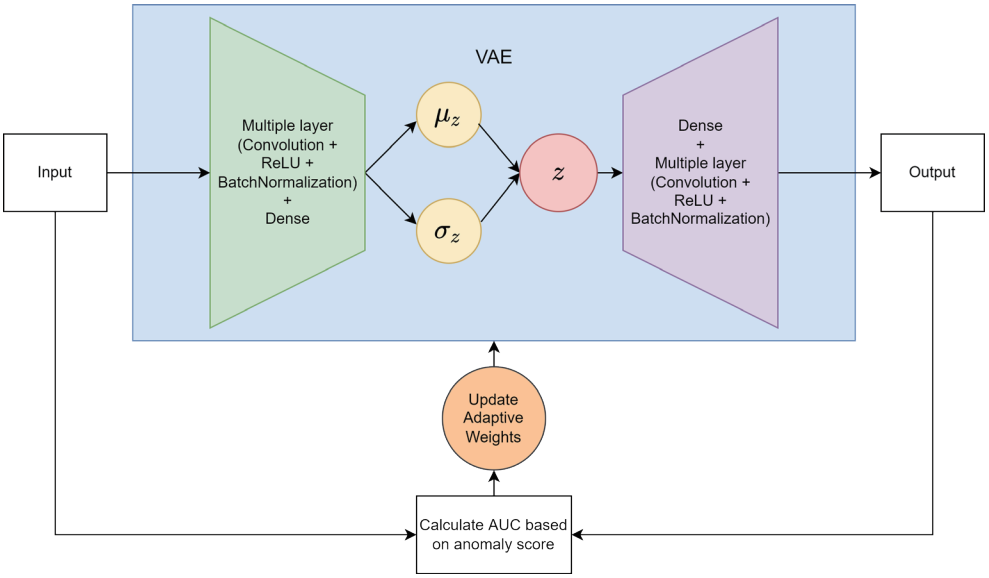


Fig. 13. (Color online) Structure of AWL VAE adapted to AUC.

Table 5
Parameter setup.

Parameters	Value
Number of epochs (N_{Epoch})	100
Rank of latent variable z	1000
Batch size	32 (for Bellows) 64 (for square- or comb-shaped Resin Products)
Optimizer	Adam
Learning rate	10^{-4}

$$\text{MSE score} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x'_i)^2 \tag{15}$$

$$\text{Max1p score} = \max|\boldsymbol{x} - \boldsymbol{x}'| \tag{16}$$

Considering subtle differences in performance caused by random seeds of Python, Tensorflow, and Numpy, training is attempted five times on fixed seeds = {1, 10, 20, 50, 100}. After finishing the training, the max values of AUC for each epoch, sensitivity, and specificity are recorded as assessment values. The rank of the latent variable $\boldsymbol{z}$ was chosen from {100, 500, 1000, 2000, 5000} to be 1000, which showed the highest value of AUC in the conventional VAE. The learning rate was 10^{-4}, which is the default value in Tensorflow. This experiment is conducted on a workstation with Intel Xeon W-2133 CPU, 128 GB of RAM, and GeForce TITAN RTX GPU, with Ubuntu 18.04, Python 3.7.5, and Tensorflow 2.4.1 installed. The reconstruction model $p_\theta(\boldsymbol{x}'|\boldsymbol{z})$ is assumed to follow the Bernoulli distribution for the Bellows dataset and the normal distribution for the square-shaped Resin Product and comb-shaped Resin Product datasets. In other words, Eq. (11) is used as the loss function, and the parameters α and β are varied for the Bellows dataset, whereas Eq. (12) is used as the loss function and the parameters w_M, w_A, and w_D) are varied for the square-shaped Resin Product and comb-shaped Resin Product datasets. We chose distributional assumptions for which better results had been obtained by conventional methods.

4.2 Results

4.2.1 AUC and weight transition

In this experiment, the reconstruction model $p_\theta(\boldsymbol{x}'|\boldsymbol{z})$ is assumed to follow the Bernoulli distribution. In other words, Eq. (11) is used as the loss function and the parameters α and β are varied in the pattern shown in Table 1 for this training. Figure 14 shows the AUC transition for each epoch using the conventional VAE on a fixed seed of 1. In this and the next section, we targeted the Bellows dataset. Although the AUC was about 0.8 at the beginning of this training,

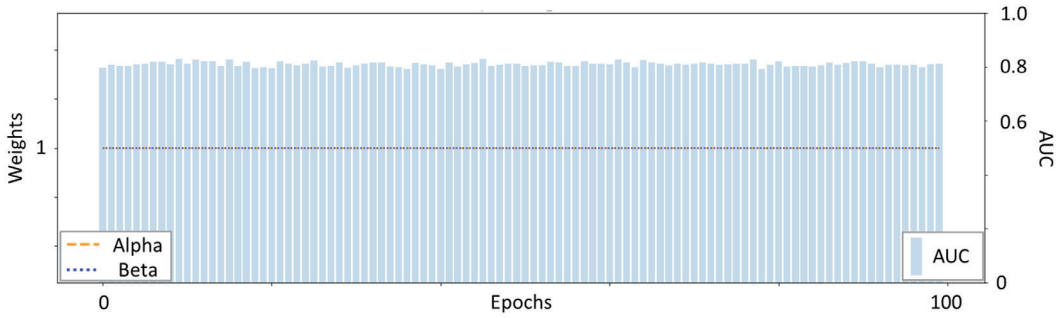


Fig. 14. (Color online) AUC (blue bar) for each epoch of conventional VAE.

no significant improvement was observed as the training progressed. The maximum AUC in this process was 83.09%, which was observed at the 9th epoch. In contrast, Fig. 15 shows that using AWL VAE. Compared with the training shown in Fig. 14, the parameters other than F_{Update}, the seed values, and the target dataset have not been changed. Here, the AUC transition is depicted as a blue bar, the parameter α as a yellow line, and the parameter β as a blue line. It can be seen that the value of α or β is switched to 0 or 1 when the condition of no improvement in AUC in the last three epochs is satisfied. The value of AUC may or may not be better after being switched. However, improvement of the AUC compared with the conventional VAE was observed at the specific epoch. The maximum AUC in this process was 88.24%, which was observed at the 63rd epoch.

4.2.2 Reconstructed images

As in the previous section, we are focusing on the Bellows dataset to provide detailed results in this section again. To visually confirm that anomalies are detected, reconstructed and squared error images for each input image are compared. The squared error image is obtained by squaring the difference in pixel values between the original and reconstructed images. Pixels or areas with a small degree of squared error can be regarded as normal areas. Since it is not possible to describe all images, we are focusing on one normal image and one anomaly image. Therefore, the explanation here is not for all images.

Figures 16 and 17 show the original, reconstructed, and squared error images using the conventional or proposed methods, respectively, for the normal case. Both reconstructed and squared error images were obtained at the epoch where the maximum AUC was recorded as described in Sect. 4.2.1. The images shown here are obtained from the same input images for the conventional and proposed methods. Comparing only the reconstructed images by the conventional and proposed methods, we do not perceive any difference visually. However, upon checking the squared error images, it appears that AWL VAE is trying to reproduce non-anomaly noise due to light exposure. To confirm the difference quantitatively, the MSEs were calculated to be 0.029917 and 0.029297 for the conventional and proposed methods, respectively. The fact that the anomaly score in the normal image is smaller by the proposed method than by the conventional method indicates that the proposed method is effective, at least for this image.

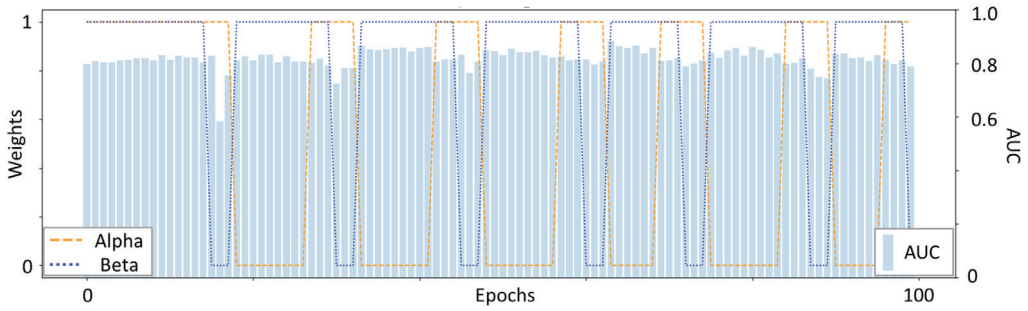


Fig. 15. (Color online) AUC (blue bar), α (yellow line), and β (blue line) for each epoch of AWL VAE.

(a) (b) (c)

Fig. 16. An example of inputting a NORMAL image using CONVENTIONAL VAE at the epoch the maximum AUC is recorded (9th epoch): (a) original image, (b) reconstructed image, and (c) squared error image. *MSE* = 0.029917.

(a) (b) (c)

Fig. 17. An example of inputting a NORMAL image using AWL VAE at the epoch the maximum AUC is recorded (63rd epoch): (a) original image, (b) reconstructed image, and (c) squared error image. *MSE* = 0.029297.

Similarly, Figs. 18 and 19 show the results of reconstruction when the same anomaly image is input to the conventional and proposed VAE. Upon checking the squared error images, it appears that the non-anomaly noise in the proposed method is not reconstructed as well as that in the conventional method, and this appears to be the reason for the higher degree of anomaly. Nevertheless, the MSE scores increased significantly from 0.030345 to 0.061639.

4.2.3 AUC, sensitivity, and specificity

The maximum AUCs obtained by the method explained in Sect. 4.2.1 were recorded as the performance using that VAE while the fixed seed value was changed five times. Table 6 shows the maximum AUC means at five fixed seed values for each dataset. The anomaly detectors using AWL VAE give better results in AUC than the conventional one for the Bellows and comb-shaped Resin Product datasets; the results are almost the same for the square-shaped Resin Product dataset. In particular, the improvement of the average AUC is 4.77% for the Bellows dataset. Tables 7 and 8 show the mean values at five fixed seed values of sensitivity and specificity, respectively, in the epochs where the maximum AUC was observed. Depending on the dataset, some results are better by using AWL, and some are worse. The remarkable result is the sensitivity for the Bellows dataset: it was improved by 10.39% compared with that using the conventional VAE. Although there was a 4.89% decrease in specificity, the benefit of the large

(a) (b) (c)

Fig. 18. An example of inputting an ANOMALY image using CONVENTIONAL VAE at the epoch the maximum AUC is recorded (9th epoch): (a) original image, (b) reconstructed image, and (c) squared error image. *MSE* = 0.030345.

(a) (b) (c)

Fig. 19. An example of inputting an ANOMALY image using AWL VAE at the epoch the maximum AUC is recorded (63rd epoch): (a) original image, (b) reconstructed image, and (c) squared error image. *MSE* = 0.061639.

Table 6
Maximum AUC means (%).

Dataset	Conventional VAE	AUC-adapted VAE
Bellows	83.64 ± 0.34	88.41 ± 1.13
Square-shaped Resin Product	99.73 ± 0.002	99.72 ± 0.01
Comb-shaped Resin Product	94.18 ± 2.10	97.41 ± 0.90

Table 7
Sensitivity means at the epoch the maximum AUC (%) was recorded.

Dataset	Conventional VAE	AUC-adapted VAE
Bellows	67.88 ± 1.87	78.27 ± 8.00
Square-shaped Resin Product	97.14 ± 0.48	97.14 ± 0.29
Comb-shaped Resin Product	87.00 ± 9.62	92.00 ± 4.95

Table 8
Specificity means at the epoch the maximum AUC (%) was recorded.

Dataset	Conventional VAE	AUC-adapted VAE
Bellows	89.56 ± 1.49	84.67 ± 7.30
Square-shaped Resin Product	99.48 ± 0.19	99.66 ± 0.11
Comb-shaped Resin Product	86.67 ± 4.87	91.94 ± 2.48

increase in AUC and sensitivity was significant. Another notable result was obtained using the comb-shaped Resin Product dataset. Although there was a large variance in the sensitivity, the means of the AUC, sensitivity, and specificity were improved by 3.23, 5.00, and 5.27%, respectively. This was the only dataset in which all three indicators improved; however, it made a significant contribution to the objective of achieving the highest AUC possible for several industrial product image datasets.

5. Experiment II: VAE Adapted to Loss

To evaluate the effectiveness of the method, we compared the performances of the conventional and loss-adapted VAE following the procedure described in Sects. 3.2 and 3.3.2.

5.1 Setup

The dataset used is the same as that described in Sect. 4.1.1. The parameter setting is basically also the same as that shown in Sect. 4.1.1, except that the trials are three times on fixed seeds = {1, 10, 20}. The numbers of epochs in Fig. 7 were set as (E_1, E_2, E_3) = (100, 150, 200). The sigmoid was used as the function to vary the weights $w \in (w_M, w_A, w_D, \alpha, \beta)$. We chose distributional assumptions for which better results had been obtained by conventional methods. In the proposed method, the determination of whether the loss value is increasing or not is simply based on the magnitude of the value between the previous epoch and the current epoch.

5.2 Results

5.2.1 Loss and weight transitions

First, the recorded loss transitions are compared between the conventional and proposed methods. In this section, for illustrative purposes, the examples are limited to the square-shaped Resin Product dataset with a seed value of 1. Figure 20 shows the transition from the start of learning to 100 epochs, which is the same for the conventional and proposed methods. It can be confirmed that the loss is reduced to some extent up to the 100th epoch. Similarly, Fig. 21 shows the transition from the 100th to the 150th epochs. An exploration for the pattern with the lowest loss is performed here. In Fig. 21, the blue lines depict conventional losses, and other reducing patterns are depicted in different colors. Of all the patterns, including conventional losses, the transition of the pattern with the most loss reduction at the 100th epoch is shown in red. It was found that there were several patterns with lower losses than the conventional losses, and pattern III was selected as the optimal pattern among them. The selected patterns were saved and explored again up to the 200th epoch, and the results are shown in Fig. 22. As in Fig. 21, the conventional loss is shown in blue and the loss at the 200th epoch is shown in red. Again, as in Fig. 21, there is a slight improvement in loss compared with the conventional method, although there are temporary increases.

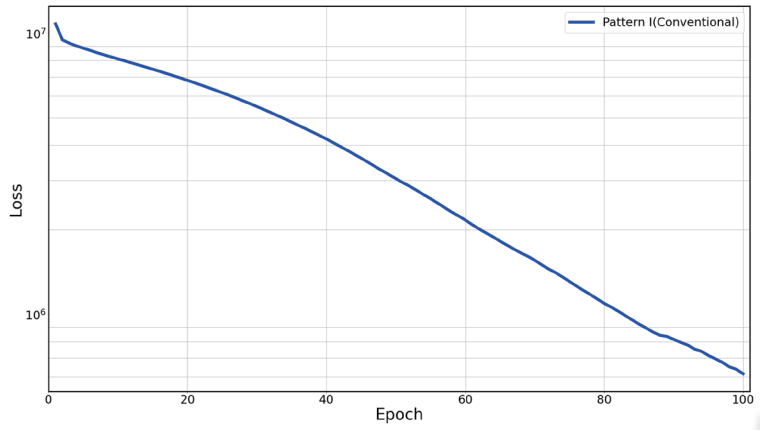


Fig. 20. (Color online) An example of loss transition from 0th to 100th epochs.

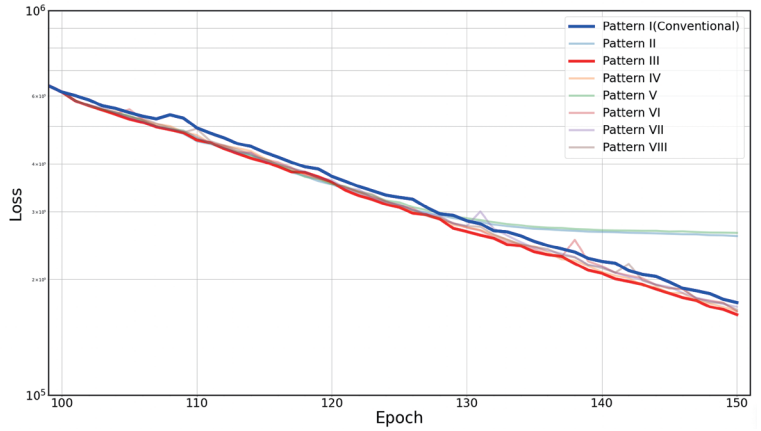


Fig. 21. (Color online) An example of loss transition from 100th to 150th epochs.

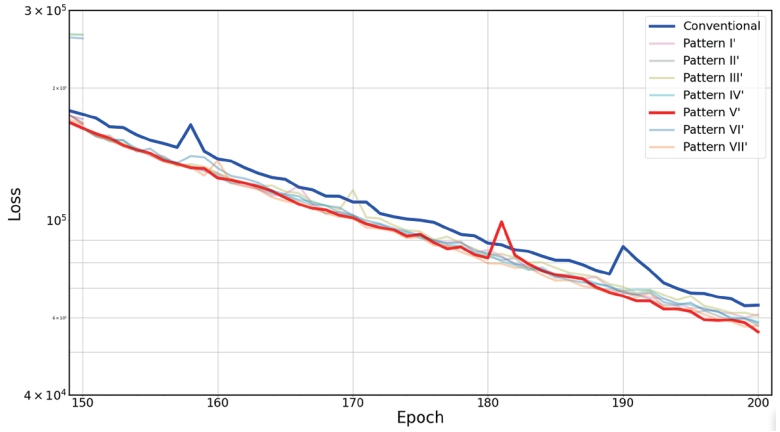


Fig. 22. (Color online) An example of loss transition from 150th to 200th epochs.

Next, the recorded weight transitions for the proposed method is examined. Figure 23 shows the weight transitions for the optimized pattern at the 200th epoch (patterns III to V'). Here, a pattern is applied where only w_D decreases from the 100th epoch, and w_M and w_D decrease from the 150th epoch. The value w_D stopped changing at the fourth epoch after it began to decrease, owing to an increase in the number of detected losses. After that, training proceeded with the stopped weights, and w_D began to decrease again at the 150th epoch and stopped at four epochs. The value w_M decreased for five epochs and then stopped as well. Since the example in Fig. 23 shows only a slight reduction in weights, Fig. 24 presents another example that shows a remarkable reduction in weights, with a loss reduction up to the 150th epoch compared with the conventional method. In this pattern, w_M and w_A began to change simultaneously, with w_M decreasing for 14 epochs and w_A decreasing for 26 epochs. In the example shown in Fig. 24, it can be seen that the weights can be gradually decreased using the sigmoid function.

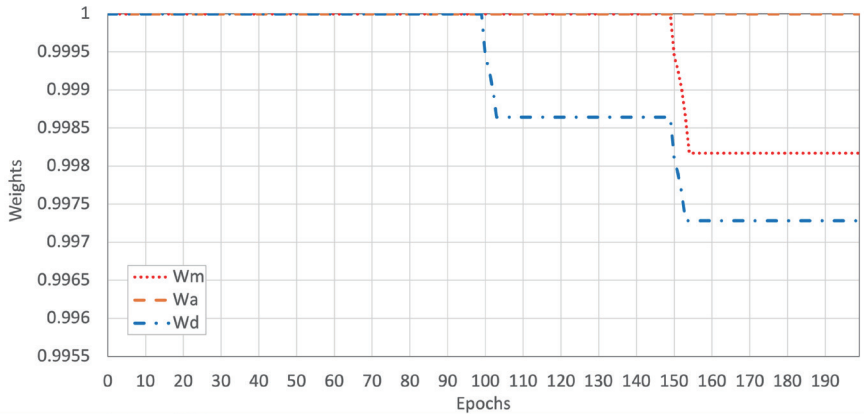


Fig. 23. (Color online) Weight transitions in the pattern with the lowest loss.

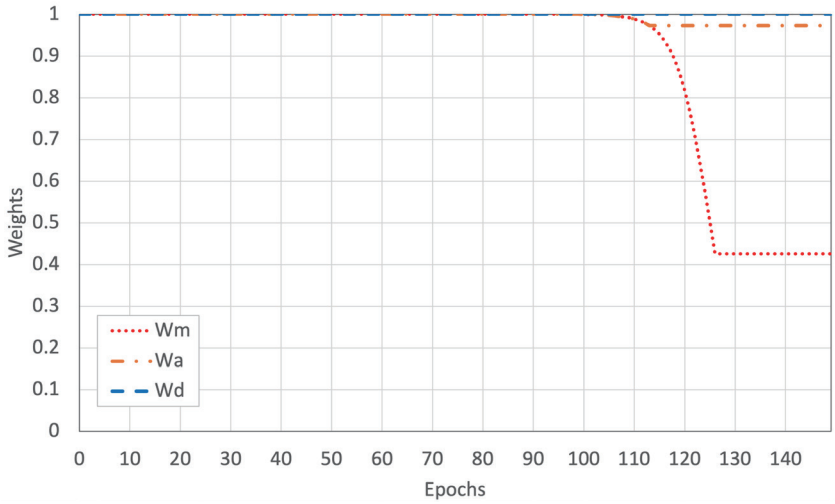


Fig. 24. (Color online) Weight transitions in the other patterns in Fig. 23, which result in lower losses than by the conventional method.

5.2.2 AUC, sensitivity, and specificity

Performance was compared to determine if the model optimized by the proposed method is superior. The AUCs at the end of the training were recorded as the performance using the VAEs while the fixed seed value was changed three times. Table 9 shows the mean values of AUC at three fixed seed values for each dataset. The anomaly detectors using the proposed VAE give better results in AUC than the conventional one for the square-shaped Resin Product and comb-shaped Resin Product datasets; the results worsened slightly for the Bellows dataset. The remarkable result is the AUC for the comb-shaped Resin Product dataset: it was improved by 2.57% compared with that using the conventional VAE. Tables 10 and 11 show the mean values at three fixed seed values of sensitivity and specificity at the end of training, respectively. Depending on the dataset, some of these indices are better using AWL, and some are worse. However, they show large variances in the results. This is noticeable, especially for the comb-shaped Resin Product dataset, where the variances in sensitivity and specificity are very large.

6. Discussion

6.1 AUC-adapted VAE

The results described in Sect. 4 show that the AUC-adapted VAE causes a significant performance improvement for the Bellows and comb-shaped Resin Products, while it was shown that it may not be a panacea for all image data. One possible reason for this is that the performance of the anomaly detector using the conventional VAE is already excellent. The AWL

Table 9
Average AUC at 200th epoch (%).

Dataset	Conventional VAE	Loss-adapted VAE
Bellows	76.08 ± 0.47	75.95 ± 0.85
Square-shaped Resin Product	99.01 ± 0.11	99.16 ± 0.17
Comb-shaped Resin Product	87.45 ± 2.71	90.02 ± 1.56

Table 10
Average sensitivity at 200th epoch (%).

Dataset	Conventional VAE	Loss-adapted VAE
Bellows	62.18 ± 0.56	61.86 ± 1.11
Square-shaped Resin Product	97.67 ± 0.50	98.03 ± 0.55
Comb-shaped Resin Product	89.67 ± 2.52	88.67 ± 5.77

Table 11
Average specificity at 200th epoch (%).

Dataset	Conventional VAE	Loss-adapted VAE
Bellows	85.56 ± 1.11	85.19 ± 1.70
Square-shaped Resin Product	99.47 ± 0.06	99.47 ± 0.12
Comb-shaped Resin Product	72.69 ± 8.37	79.63 ± 3.50

VAE adapted to AUC may be effective when the conventional performance is not so good. Hence, we need to test it on other datasets.

Factors that enable the AWL VAE to provide better AUC than usual should also be discussed. In Sect. 4.2.2, it is indicated that the AWL VAE can make anomaly scores smaller on normal images and larger on anomaly images than the conventional VAE. Regarding the reconstruction of anomaly images, we stated the fact that non-anomaly noise is not reconstructed as well as in the conventional VAE, which may have caused the anomaly score to increase. However, it is not guaranteed how VAE trained using only normal images will output for anomaly input images. Therefore, although AWL VAE currently provides clearer reconstruction for normal input images, a more comprehensive and detailed investigation of the factors that led to the improved AUC is required.

6.2 Loss-adapted VAE

As a matter of improvement, the rules for determining whether a partial term of loss is headed for divergence should be carefully considered. In this experiment, as described in Sect. 3.1, we simply determined whether the instantaneous slope at one epoch was positive or not as its discriminant rule. However, this function is too sensitive to deal with the occasional temporary rise in losses. For non-dominant loss terms, it is never advisable to use this function, especially since the ups and downs occur so frequently. Therefore, it is necessary to consider various innovations such as discrimination using moving averages.

Furthermore, the type of weight pattern needs to be further considered. In this study, the weights were gradually changed in the patterns shown in Table 3; however, only the decreasing pattern is examined here. To find the optimal ratio of weights, it is better to consider the possibility of increasing them. However, increasing the number of search patterns requires more time to find the optimal model, especially in the case of VAEs whose loss functions have three terms. It will be necessary to find a search method that does not significantly increase the training time.

For further improvement, we need to consider the KL term [$D_{VAE}(\boldsymbol{x})$] vanishing problem. It leads to a representation that is not very variable. One of the ways to mitigate it is to apply an annealing schedule to the KL term. The traditional method is monotonic annealing.[12] Starting with a small beta forces the model to focus on reconstructing the input sequence rather than minimizing the KL loss. As the beta increases, the model gradually emphasizes the shape of the data distribution, eventually reaching a beta of 1 or a pre-specified value. Fu *et al*. proposed a periodic annealing schedule that repeats the traditional annealing multiple times.[13] This schedule may help to build a more well-organized potential space at a small additional cost to the computation. By applying this annealing to our AWL VAE, we may be able to obtain a richer representation capability, which may lead to further performance improvement of anomaly detectors.

7. Conclusions

There is a need for the automated visual inspection of industrial products with improved efficiency and accuracy. In particular, anomaly detectors based on deep learning have attracted considerable attention. Industrial products are generally not suitable for supervised learning such as simply using CNNs because the number of anomaly products is generally small. VAE and GAN are examples of generative models that are capable of unsupervised learning. GAN can produce sharper images than VAE; however, it requires more training parameters and more time for anomaly detection than VAE, making it unsuitable for implementation in industrial settings. Therefore, in this study, we aimed to improve the accuracy of abnormality detection models for industrial products by modifying the probability distribution generation model VAE, which is capable of unsupervised learning.

The new model focuses on the loss function and assigns dynamic weights to each term of the function. The new algorithm also adaptively varies these weights by functionalizing them with respect to AUC, epoch, and so forth. The proposed VAE provides improvements in performance as an anomaly detector for several image datasets of industrial products. In particular, VAE adapted to loss achieves more optimized models and slightly better performance than conventional methods while maintaining unsupervised learning. Although it is not easy to improve performance in unsupervised learning, adjusting the learning method as in this study may help to reduce the burden in industrial settings.

However, it was found that several issues were faced. For example, there is a lack of confidence in the variability of the results, and the performance has not been improved for all images. It has also been found that the AUC does not improve as the loss decreases. Further performance improvement is discussed, however, and to achieve this, a more detailed investigation of the generated images, search patterns, their validity evaluation, and so forth will be necessary. It is also necessary to continue to contribute a great deal to the development of the industrial field through multiple experiments.

Copyright Notice

This article includes materials from "Proposal of VAE-based Deep Learning Anomaly Detection Model for Industrial Products"[4] by the same authors.

Acknowledgments

We wish to thank two companies that do not wish to be identified for providing the datasets. We would like to sincerely acknowledge the funding support provided by one of these companies.

References

1 E. A. Hinton and E. Geoffrey: Science **313** (2006) 504. https://doi.org/10.1126/science.1127647
2 M. W. Kingma and P. Diederik: ICLR (2013). https://doi.org/10.48550/arXiv.1312.6114

3 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio: Commun. ACM **63** (2020) 139. https://doi.org/10.1145/3422622
4 S. Nakata, T. Kasahara, and H. Nambo: ICMSEM **16** (2022) 336. https://doi.org/10.1007/978-3-031-10388-9_24
5 I. Goodfellow: NIPS (2016). https://doi.org/10.48550/arXiv.1701.00160
6 B. Uria, M. Côté, K. Gregor, I. Murray, and H. Larochelle: J. Mach. Learn. Res. **17** (2016) 7184. https://doi.org/10.48550/arXiv.1605.02226
7 M. Germain, K. Gregor, I. Murray, and H. Larochelle: JMLR W&CP **37** (2015) 881. https://doi.org/10.48550/arXiv.1502.03509
8 G. Alain, Y. Bengio, L. Yao, J. Yosinski, E. Thibodeau-Laufer, S. Zhang, and P. Vincent: Inf. Infer. J. IMA **5** (2016) 210. https://doi.org/10.48550/arXiv.1503.05571
9 T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen: NIPS **2016** (2016). https://doi.org/10.48550/arXiv.1606.03498
10 C. Chu, K. Minami, and K. Fukumizu: ICLR (2020). https://doi.org/10.48550/arXiv.2002.04185
11 I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner: ICLR (2016). https://openreview.net/forum?id=Sy2fzU9gl
12 S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio: CoNLL (2015) 10. https://doi.org/10.48550/arXiv.1511.06349
13 H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin: NAACL (2019). https://doi.org/10.48550/arXiv.1903.10145