IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 8, 2021, accepted April 30, 2021, date of publication May 14, 2021, date of current version May 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3080330

Anomaly Detection Using Autoencoder With Feature Vector Frequency Map

YOUNG-GYU KIM[1] AND TAE-HYOUNG PARK[2], (Member, IEEE)
[1]Industrial AI Research Center, Chungbuk National University, Cheongju 28644, South Korea
[2]Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju 28644, South Korea

Corresponding author: Tae-Hyoung Park (taehpark@cbnu.ac.kr)

This work was supported by the Ministry of Science and Information and Communications Technology (ICT) (MSIT), South Korea, through the Grand Information Technology Research Center support program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-2020-0-01462.

ABSTRACT Anomaly detection uses various machine learning techniques to identify and classify defective data on the production line. The autoencoder-based anomaly detection method is an unsupervised method that classifies abnormal samples using an autoencoder trained only from normal samples and is useful in environments where it is difficult to obtain abnormal samples. This method uses an abnormal score based on the reconstruction loss function, making it difficult to detect defects, such as stains, having a similar texture to a normal sample. To solve this problem, we propose an anomaly detection method using a vector quantized variational autoencoder and a feature vector frequency map. We use the prototype vector histogram and its frequency for anomaly detection instead of the reconstruction loss function. The prototype vector histogram is obtained from the vector quantized variational autoencoder's codebook in the training stage. The feature vector frequency map of the input image is generated using the prototype vector histogram in the inference stage. We calculated the abnormal score using the generated frequency map and classified the abnormal samples. The experimental results showed that the proposed method has a higher Area Under Receiver Operating Characteristics (AUROC) than the previous method in stain and scratch defects.

INDEX TERMS Anomaly detection, automatic optical inspection, deep learning.

I. INTRODUCTION

The method of distinguishing abnormal samples in a data set is known as anomaly detection. It uses various modeling techniques to detect and identify abnormal samples or data. Common examples of this method are the detection of people with specific features using closed-circuit television (CCTV) and defect detection in the manufacturing industry. They generally use feature-based classifier, generative adversarial network and autoencoder.

The classifier -based anomaly detection uses a classification model that detect abnormal samples based on features extracted from input samples. The classifier-based anomaly detection method creates a classifier that divide input data into normal samples and other samples using features extracted from input data [1]–[10], [23]–[28]. The One-Class Support Vector Machine (OC-SVM) [1] method assumes that the abnormal sample's feature vector exists at the origin. It learns a hyperplane that distinguishes abnormal samples from the feature vector of the normal sample. Taking one stage further from the OC-SVM method, the Support Vector Data Description (SVDD) method [2] classifies normal and abnormal samples by defining hyperspheres with a minimum volume. The deep-SVDD method [3] changes the feature vector extraction method in the SVDD method from a handcraft-based to a convolutional neural network. In addition, transfer learning [6], metric learning [7], voting [8], geometric transform [9], and adversarial training [10] based method is used to detect abnormal samples. However, these methods only classify normal and abnormal samples and cannot obtain information on the defect's location.

The Generative Adversarial Network (GAN)-based anomaly detection method uses a generator and a discriminator learned from normal samples [11]– 14]. The generator creates a normal sample from the probability distribution of normal samples, and the discriminator classifies the generated sample and the real-world normal sample. The sample generated by the generator is similar to the normal sample, but there is a slight difference, which can be viewed as a kind of abnormal sample. The discriminator is trained to classify the

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

sample generated by the generator and the normal sample. This discriminator is used to classify normal and abnormal samples. There is a method that uses the Deep Convolutional GAN (DCGAN) loss function as an anomaly score [11], and a method that uses a discriminator to classify anomaly and normal samples [14]. However, the GAN-based method is difficult to train using complex images due to model collapse problem.

The autoencoder-based anomaly detection uses the autoencoder trained by the dataset which only include normal samples. It uses a data set consisting only of normal samples, which is its deep learning network training data instead of an abnormal sample to construct a detection model. This method classifies an abnormal sample using the difference in a loss function or mutual information between normal sample and abnormal sample. The autoencoder-based method is an unsupervised-based method composed of an encoder and a decoder [15]–[22]. The autoencoder-based anomaly detection method uses the image difference between input and output images [16], [18], [20], a latent space-based score [15], [17], a loss-based score [19], and the method using the autoencoder-based GAN structure [20]. The autoencoder-based anomaly detection method has higher learning stability than the GAN-based method, making it possible to classify abnormal samples simply by reconstructing the input image and the output image. In addition, autoencoder-based anomaly detection method can identify the defect location by using the difference image between input and output image. However, the reconstruction loss-based anomaly detection method has difficulty detecting stains and scratches defect. In the autoencoder-based anomaly detection method, the L1 and L2 reconstruction loss values in the 2D image are compared with the normal sample's values to determine an abnormal sample. However, stain and scratch defects have a subtle difference compared with normal samples and making it difficult to distinguish using the L1 and L2 reconstruction loss–based methods.

This paper proposes an anomaly detection method based on a prototype vector histogram and a feature vector frequency map. We extracted the prototype vector histogram from the vector quantized variational autoencoder's prototype vector on codebook and used it to extract the input image's feature vector frequency map. Then, we calculate abnormal score from feature vector frequency map of the input image in the inspection stage and use it as a measure for anomaly detection.

The contribution of this paper can be defined as follows:

1. We propose a novel method using the frequency of latent vector for anomaly detection. To determine abnormal samples, the previous method uses a decoder result, but the proposed method uses a frequency map derived from latent space.
2. We propose the feature vector frequency map as a new anomaly detection measure. The prototype vector histogram extracted in the training stage were used in the inspection stage. Using this method, we achieved a higher Area Under Receiver Operating Characteristics (AUROC) compared to the reconstruction loss method using the feature vector frequency map-based defect detection method.

The thesis is organized as follows. Chapter I describes the background description, necessity, and proposal method for the proposed method. Chapter II describes the autoencoder-based anomaly detection problem. Chapter III describes the vector quantized variational autoencoder, and Chapter IV describes the proposed vector quantized variational autoencoder and anomaly detection using the feature vector frequency map. In Chapter V, the performance of the existing method and the proposed method is compared based on the experimental results in an actual defective dataset. Chapter VI describes the conclusion of this paper.

II. AUTOENCODER–BASED ANOMALY DETECTION PROBLEM

The autoencoder-based anomaly detection problem involves classifying the input image into normal and abnormal samples based on the abnormal score calculated by the autoencoder. This score is computed using a score function based on the autoencoder loss function, such as reconstruction loss and latent vector distance. As the learning progresses, the output image of the autoencoder becomes the same as the input image, the abnormal score uses this characteristic. In the training stage, the autoencoder is trained using only normal samples. The autoencoder's output image for the trained normal sample is reconstructed almost similar to the input image, so the reconstruction loss for the normal sample is small. In contrast, the autoencoder's output image for the abnormal sample has much difference from the input image, because it is not learned in the training stage. The reconstruction loss for the abnormal sample is high. The autoencoder-based anomaly detection method determines the samples using the reconstruction loss difference score. Fig. 1 shows the structure of an anomaly detection in defect detection problem based on an autoencoder. The input image is converted into a latent vector through the encoder, then the decoder converts it, generating an output image. After calculating the abnormal score using the input image and the output image, it is classified into normal samples and abnormal samples based on abnormal score.

The anomaly detection problem consists of a training stage and an inspection stage. In the training stage, the autoencoder adjusts the weight of the network so that the output image is the same as the input image. In the inspection stage, the input image's abnormal score is calculated with the trained autoencoder, and the abnormal score is used to determine whether the input image is a normal sample or an abnormal sample.

A. TRAINING STAGE

In the training stage, the output image of autoencoder is the same as the normal sample by updating the weight of autoencoder. Since the training data set consist of only normal samples, the autoencoder only learns the reconstruction task

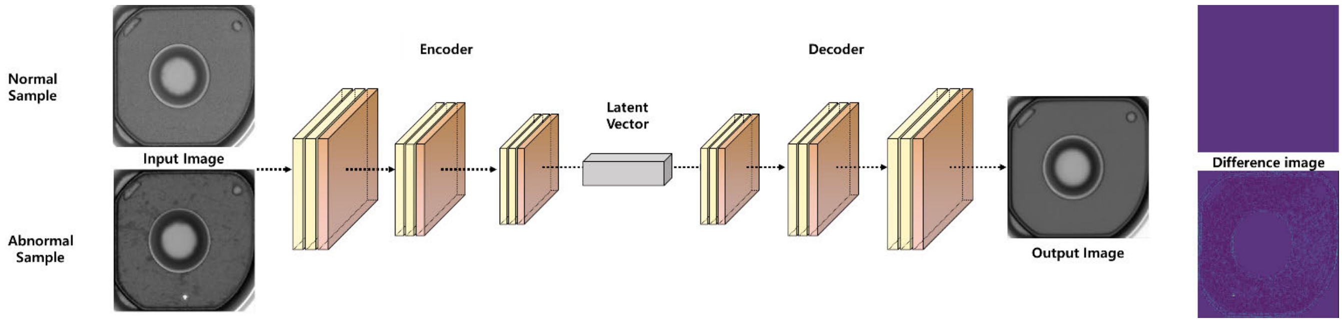

FIGURE 1. Structure of autoencoder based anomaly detection.

for the normal samples. The autoencoder consists of encoder network $E(\cdot)$ and decoder network $D(\cdot)$. Encoder E($\cdot$) converts the input image to a latent vector, and decoder D($\cdot$) restores the latent vector to the same size as the input image. The autoencoder uses a reconstruction loss function to learn so that the decoder's output result is the same as the input image.

In the training stage, the sample of training data set x is converted into a latent vector $\mathbf{z}$ through E($\cdot$). The encoding process can be expressed as follows.

$$z = E(x) \tag{1}$$

The latent vector $\mathbf{z}$ output from the encoder network is converted to an output image $\hat{x}$ of the same size as the input image through D($\cdot$). This process can be expressed as the following equation.

$$\hat{x} = D(z) \tag{2}$$

The autoencoder output should be the same as the input image. Therefore, reconstruction loss functions such as L1/L2-distance and structural similarity index measure (SSIM) for the input image and output image are used as the loss function. We used the autoencoder using L2-distance as the reconstruction loss as an example. Let $x\,(r, c)$ and $\hat{x}\,(r, c)$ are the brightness values of x_n and $\hat{x}_n$ at the pixel (r, c) position, H and W is height and width of image, respectively. The L2-distance loss function for input image x and output image $\hat{x}$ can be expressed as

$$L_{l2} = \sum\nolimits_{r=0}^{H} \sum\nolimits_{c=0}^{W} \left(x_n\,(r, c) - \hat{x}_n\,(r, c)\right)^2 \tag{3}$$

B. INSPECTION STAGE

The inspection stage identifies defects in the input image using the autoencoder. In the inspection stage, the autoencoder extracts the output image for the input image. Then, by calculating the abnormal score function for the output image and the input image, the input image is classified as a normal sample or an abnormal sample.

As described above, the abnormal score function is mostly based on reconstruction loss. In more detail, it varies according to each method, such as a method using L1-distance of the output image and input image [16], [19], method of using the distance of probability distribution in latent space [15], a method using SSIM [18]. The threshold value is defined based on the normal sample's abnormal score value in the training stage.

III. VECTOR QUANTIZED VARIATIONAL AUTOENCODER BASED ANOMALY DETECTION

The vector quantized variational autoencoder [29] is an autoencoder that selects latent vectors from a codebook consist of prototype vectors. The codebook is a latent embedding space $e \in \mathbb{R}^{K \times D}$ where K is the number of the latent vector and D is the dimensionality of each latent vector [29]. This codebook is randomly initialized for first, and update by loss function. The variational autoencoder assumes that the latent vector follows a normal distribution and generates a latent vector by resampling the encoder feature map into the latent space. The encoder feature map is defined as a set of feature vectors extracted from the encoder. The vector quantized variational autoencoder selects the prototype vector that is the closest to the encoder feature vector of encoder feature map as the latent vector. The latent vectors are a compressed feature of the input image, and the diversity of latent vector is limited in real-world data. Vector quantized variational autoencoder design latent vectors using prototype vectors instead of resampling method which generate a latent vector using the mean and variance. Vector quantization is a series of processes that replace each feature vector in the encoder feature map with the prototype vector. Fig. 2 shows the structure of vector quantized variational autoencoder.

The vector quantization module converts the feature vector of the encoder feature map into a prototype vector. The vector quantization module creates a quantized feature map from the encoder feature map. Vector quantization module consists of latent vector distance calculation, nearest prototype vector extraction, and quantized feature map generation. In the stage of calculating the latent vector distance, the distance between each feature vector of the encoder feature map and the prototype vector of the codebook is calculated. Next, the prototype vector that is the closest to the feature vector is selected as the nearest prototype vector. Let the feature vector of the encoder feature map be $z_E\,(x)$, and the prototype vector of the codebook as $\boldsymbol{e}_j (j = 1, 2, \ldots, N)$. The nearest prototype

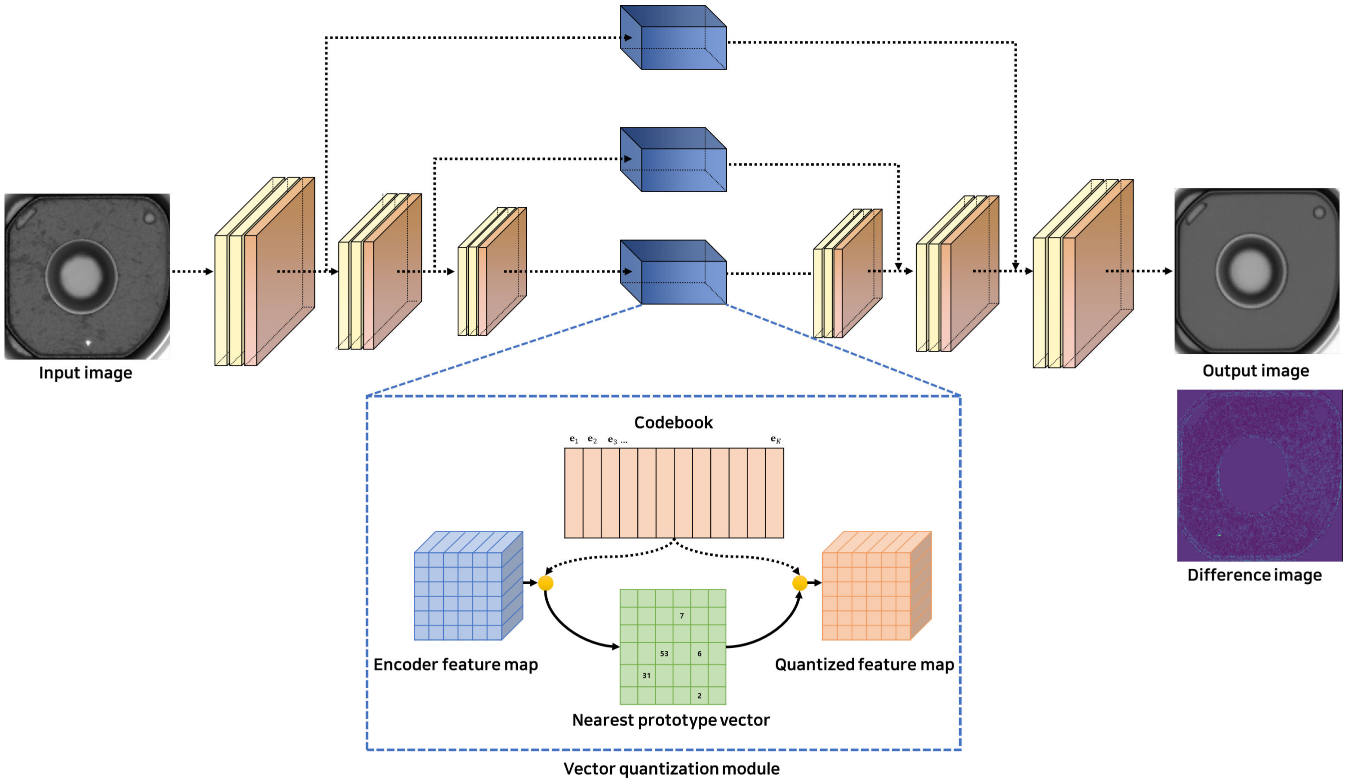

FIGURE 2. Vector quantized variational autoencoder.

vector $z_q(x)$ of the vector quantized variational autoencoder is defined as follows.

$$z_q(x) = \boldsymbol{e}_k \tag{4}$$

where,

$$k = \underset{j}{argmin} \left\| z_E(x) - \boldsymbol{e}_j \right\|^2 \tag{5}$$

The nearest prototype vector $z_q(x)$ extracted from each feature vector of the encoder feature map is combined into one feature map corresponding to each original encoder feature map position. We define this feature map as a quantized feature map. The quantized feature map is converted into an output of the same size as the input image through a decoder.

To train the encoder, decoder, and vector quantization module simultaneously, we use a loss function that can consider all three modules on reconstruction, codebook, and commitment losses. Reconstruction loss is the difference between the input image and the output image, and the codebook loss represents the similarity between the prototype vector and the feature vector of encoder feature map. The commitment loss induces the feature vector of encoder feature map to be kept close to the prototype vector to reduce the wrong mapping of the same signal with another prototype vector during the training process.

$$\mathcal{L}_{VQ-VAE} = \|x - D(e)\|_2^2 + \|sg(E(x)) - e\|_2^2 + \beta \|sg(e) - E(x)\|_2^2 \tag{6}$$

$sg(*)$ means 'stop gradient' operator. The stop gradient operator is defined as identity at forward stage and has zero partial derivatives at backward stage. $sg(*)$ blocks inter-layer interference factors in the calculation of the codebook loss function and commitment loss function.

IV. VECTOR QUANTIZED VARIATIONAL AUTOENCODER BASED ANOMALY DETECTION WITH FEATURE VECTOR FREQUENCY MAP

This chapter describes an anomaly detection method using a vector quantized variational autoencoder and a feature vector frequency map. First, we define a prototype vector histogram extracted from the vector quantization module and explain the extraction process. We define a feature vector frequency map based on the prototype vector histogram. Finally, the method of detecting abnormalities by the extracted feature vector frequency map is described.

A. MOTIVATION

In the anomaly detection method based on reconstruction loss, the autoencoder uses a data set containing only normal samples as a training data set. It allows the autoencoder to reconstruct a normal sample regardless of the input image. However, since the reconstruction loss only considers the difference between the input image and the output image, modeling a normal sample is not appropriately performed.

An abnormal sample is used as an input in the inspection stage causes a problem: the input image and the output image become the same. As a result, the anomaly detection method's detection accuracy using reconstruction loss as an abnormal score decreases. Especially, the detection accuracy decreases in stain defects and scratch defects that are slightly different from normal samples.

We focused on the prototype vector of codebook, the latent vector of the vector quantized variational autoencoder, to solve this problem. In the training stage, the vector quantized variational autoencoder is trained using only normal samples, and the prototype vector is updated according to the loss function of equation (6). After the learning of the vector quantized variational autoencoder is complete, the prototype vector can be divided into two categories: the prototype vector selected as the "frequently" nearest prototype vector in the training stage and other prototype vectors. The prototype vector selected as the nearest prototype vector with a high frequency in the training stage means that it is a prototype vector commonly extracted from the normal sample, which means that the prototype vector is a major feature vector of the normal sample. Therefore, the region in which the corresponding prototype vector is selected as the nearest prototype vector has a high probability of being a normal region. In contrast, a prototype vector that is rarely selected as the nearest prototype vector is likely to be an abnormal region that is not found in the normal region. It means that the prototype vector is a minor feature vector of the normal sample.

Focusing on the above facts, we propose a prototype vector histogram extracted based on the number of times the prototype vector is selected as the nearest prototype vector and a feature vector frequency map extracted based on prototype vector histogram. By replacing the nearest prototype vector with the prototype vector histogram value, we created a feature vector frequency map based on the extracted prototype vector histogram. Since the vector quantized variational autoencoder's latent vector is defined as a 3D feature map, a 2D feature map can be obtained by replacing each nearest prototype vector with their corresponding prototype vector histogram value. This 2D feature map is a feature vector frequency map. The anomaly score is calculated based on the generated feature vector frequency map, and the abnormal sample is detected by comparing it with the anomaly score in the training stage.

Since the proposed method detects abnormal samples using a feature vector frequency map, it improves the detection accuracy of defects with similar textures to normal, such as stains and scratches, which were difficult to identify and detect using previous methods. There is a slight difference in brightness values for stain and scratch defects compared to the normal sample. Therefore, it is difficult to detect stain and scratch defects in a reconstruction loss-based anomaly detection method that identify defects using differences in brightness values. The prototype vector histogram records whether prototype vector of codebook is frequently observed during the training stage and determines abnormal samples based on this histogram value.

B. SYSTEM STRUCTURE

The system structure of the proposed method is shown in Fig. 3. The proposed method consists of an encoder, a vector quantization module, and a decoder. The vector quantization module consists of a codebook, a prototype vector histogram, and a nearest prototype vector. The encoder converts the input image into an encoder feature map consisting of feature vectors, and it converted into a quantitated feature map by the vector quantization module. The decoder then generates an output image from the quantitated feature map. The prototype vector histogram of the prototype vector is extracted from the vector quantized variational autoencoder using the training dataset. We calculate abnormal score from this frequency map. In the inspection step, a feature vector frequency map of the input image is generated using the prototype vector histogram, and the abnormal score for the input image is calculated based on this.

C. PROTOTYPE VECTOR HISTOGRAM

The prototype vector histogram records the number of times the prototype vector was used as the nearest prototype vector at the training stage. The vector quantization module selects the nearest prototype vector by calculating the distance between each feature vector of the encoder feature map and the prototype vector. Let $z_e(r, c)$ is the feature vector of the encoder feature map z_e. The prototype vector histogram $H = \{h_1, h_2, \ldots, h_N\}$ is defined as follows:

$$k = \underset{j}{argmin} \left\| z_e(r, c) - \boldsymbol{e}_j \right\|_2 \tag{7}$$

$$H(k) += \gamma \tag{8}$$

D. FEATURE VECTOR FREQUENCY MAP

The feature vector frequency map is a feature map extracted by prototype vector histogram and encoder feature map. In the step selecting the nearest prototype vector, a one-dimensional feature map is obtained by replacing the nearest prototype vector with corresponding the prototype vector histogram value of the nearest prototype vector. We define a feature map consisting of prototype vector histogram values corresponding to each prototype vector as a feature vector frequency map for the input image. In this paper, we used the reciprocal of the prototype vector histogram value for feature vector frequency map to simplify the abnormal score calculation.

Let the feature vector of encoder feature map in the input image be $z_e^i(r, c)$. The feature vector frequency map $F(r, c)$ is defined as follows.

$$F(r, c) = \frac{1}{H(k_i)} \tag{9}$$

where

$$k_i = \underset{j}{argmin} \left\| z_e^i(r, c) - \boldsymbol{e}_j \right\|_2 \tag{10}$$

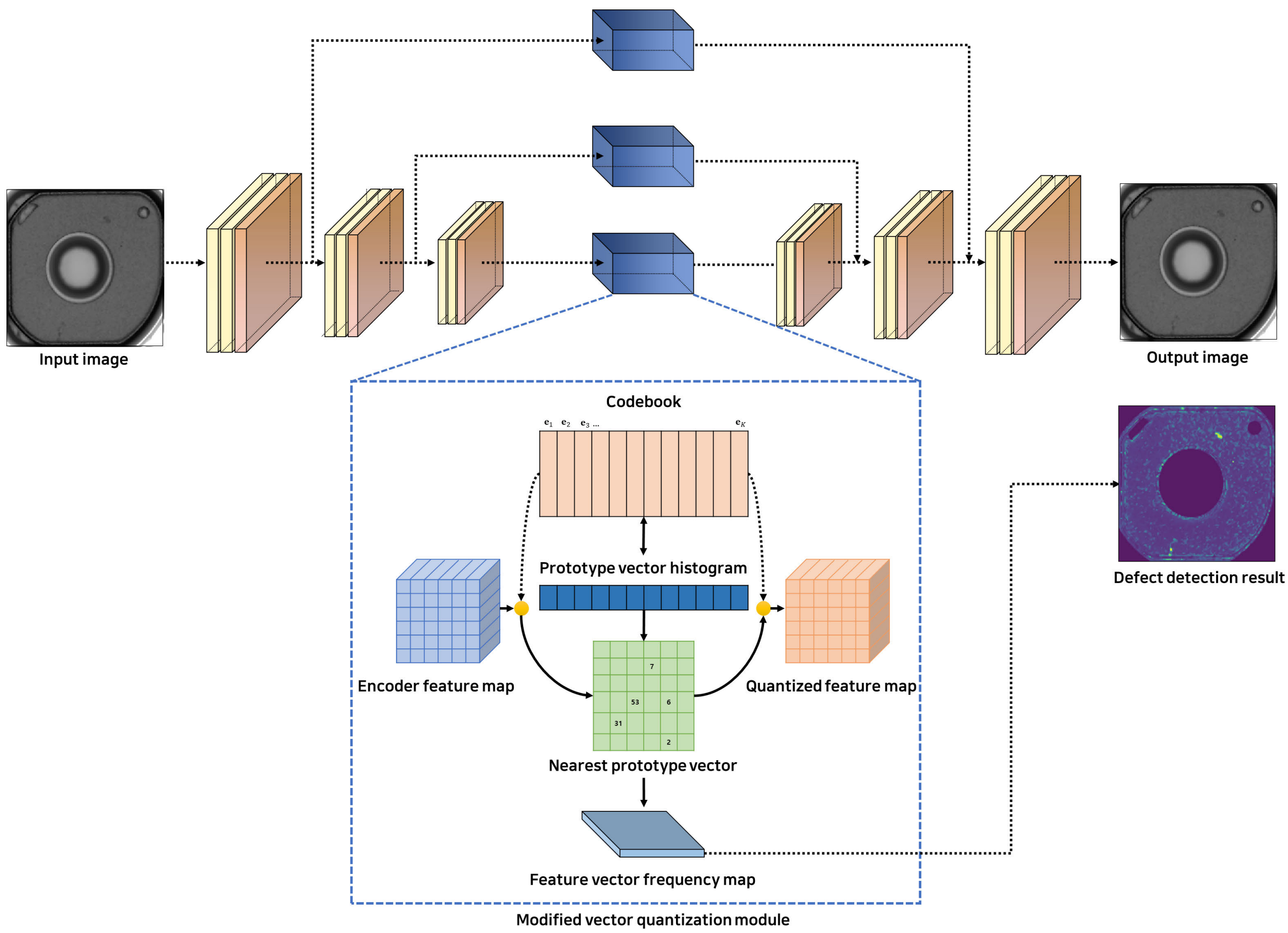

FIGURE 3. Structure of proposed method.

E. ANORMALY DETECTION WITH FEATURE VECTOR FREQUENCY MAP

For anomaly detection, a feature vector frequency map for a normal sample is required. Even in the normal sample, there is an area with a high prototype vector histogram value. The image edge region typically has a high prototype vector histogram value. The convolution operation is specialized in image texture analysis. Unlike the image texture region, which has a low nearest prototype vector diversity, the image edge region has a higher nearest prototype vector diversity than the texture region. In the sentence above, diversity means how many types of prototype vectors are selected as the nearest prototype vector. We used the inverse of the reference frequency map which is the mean of feature vector frequency map of training data set to apply the diversity according to the domain to the abnormal score. Let N is the number of training data set, $F_i (i = 1, 2, \ldots, N)$ is the feature vector frequency map of each samples in training data set, respectively. The reference frequency map F_{ref} is defined as follows.

$$F_{ref} = \frac{\sum_{i=0}^{N} F_i}{N} \tag{11}$$

In the inspection step, the feature vector frequency map is used to detect abnormal samples. For anomaly detection, we defined the abnormal score function using the extracted feature vector frequency map. Since the abnormal sample include the prototype vector with a low frequency value, the value of feature vector frequency map in the defect region is higher than that of the normal sample. As a result, the abnormal sample's feature vector frequency map has a higher value than that of the normal sample. We determine whether the test image is abnormal or normal based on the average of the abnormal score calculated from the training sample. When the abnormal score of the test image exceeds the threshold, we determine the test image as abnormal.

We defined the abnormal score as the sum of all pixel values of the feature vector frequency map. Let $F_i(r, c)$ is the value of feature vector frequency map of the input image. The abnormal score S_{AD} is defined as follows.

$$S_{AD} = \sum_{r=0}^{H} \sum_{c=0}^{W} \frac{F_i(r, c)}{F_{ref}(r, c)} \tag{12}$$

V. EXPERIMENTAL RESULT

A. EXPERIMENTAL SETUP

We used the defect image of a compact camera module to evaluate the performance of the proposed method and previous method. The size of the input image is 256 × 256 with a single channel image. We used 368 normal samples to

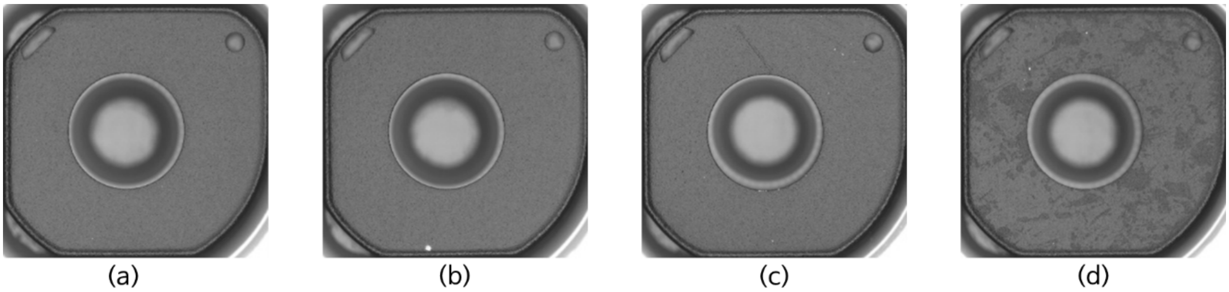

FIGURE 4. Defect of compact camera module. (a) Normal sample (b) Coating defect (c) Scratch defect (d) Stain defect.

train vector quantized variational autoencoder, and 42 normal samples, 26 coating defects, 68 scratch defects, and 82 stain defects for inspection. Fig. 4 shows defect images of compact camera module.

We used a PC with an Intel Core i7 processor and NVIDIA GeForce RTX 2080 Ti graphics card. We used Pytorch library for training and testing neural network. We used 0.001 as the learning rate of the model, and the mini-batch size was fixed to 16 for both the previous method and the proposed method. We trained the proposed method and the previous method until the neural network repeated 300 epochs of the entire training dataset.

AUROC value was used as a performance measure for anomaly detection. The AUROC value refers to the width of the bottom of the Receiver Operating Characteristics (ROC) curve. The ROC curve is a graph consisting of TPR (True Positive Rate) on the y-axis and False Positive Rate (FPR) on the x-axis. TPR is the ratio of predicting the actual normal sample as normal, and FPR is the ratio of predicting the defect sample as normal. TPR and FPR are defined as follows.

$$TPR = \frac{TP}{TP + FN} \tag{13}$$

$$FPR = \frac{FP}{TP + FP} \tag{14}$$

If the threshold value is set high to improve the classification performance for normal samples, both TPR and FPR rise. On the other hand, if the threshold is set low to improve the classification performance for defective samples, the FPR decreases, but the TPR decreases simultaneously. As such, the ROC curve can evaluate the performance of the model regardless of the threshold.

We compared the L2 autoencoder [18], SSIM autoencoder [18], variational autoencoder [16] and vector quantized autoencoder [29] with the proposed method. L2 and SSIM denote the reconstruction loss function of the autoencoder, respectively. The L2 and SSIM autoencoders use L2 distance and SSIM as the reconstruction loss functions, respectively. The variational autoencoder generates a latent vector by sampling with a normal distribution. The vector quantizaed variational autoencoder was used to compare the difference between the reconstrucion loss-based method and the feature vector frequency map.

TABLE 1. Number of dataset.

Step / Defect	Training	Testing
Normal	368	42
Coating	-	26
Scratch	-	68
Stain	-	82

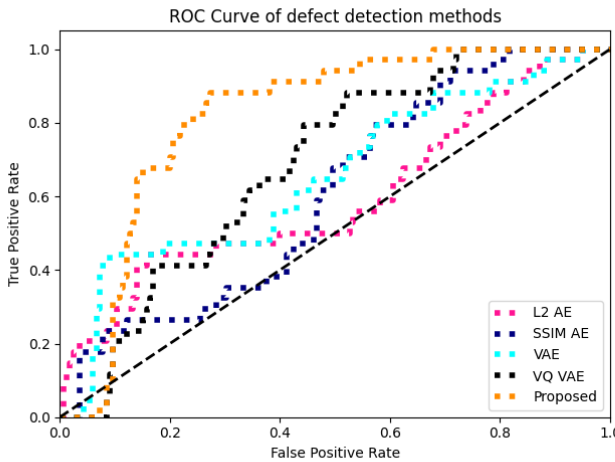

FIGURE 5. ROC curve.

B. RESULT

Table 2 shows the defect detection results for the compact camera module. Fig. 5 shows the AUROC of each method. The AUROC of the proposed method shows the highest performance at 78.5%. Since the L2 and SSIM autoencoders differ only in the loss function within the same autoencoder structure, they performed similarly. The variational autoencoder has better performance than the L2 and SSIM autoencoders but has a higher neural network weight than other methods because the fully connected layer is used in the latent vector extraction process. Among the comparison methods, the proposed method has the highest AUROC performance.

TABLE 2. Performance of defect detection methods.

Method / Measure	L2 Autoencoder [18]	SSIM Autoencoder [18]	Variational Autoencoder [16]	VQ-VAE [29]	VQ-VAE with Feature Vector Frequency Map (Proposed)
AUROC (%)	59.6	60.4	65.5	66.0	**78.5**
Network Weight (MB)	43.6	43.6	49.9	**5.5**	**5.5**
Inspection Time (ms / sample)	19	**18**	20	45	47

TABLE 3. Performance of defect detection methods.

Method / Defect Type	L2 Autoencoder [18]	SSIM Autoencoder [18]	Variational Autoencoder [16]	VQ-VAE [29]	VQ-VAE with Feature Vector Frequency Map (Proposed)
Coating	60.4	60.7	66.9	57.4	**81.8**
Scratch	57.0	58.1	60.1	62.1	**69.1**
Stain	61.4	62.3	69.6	78.5	**84.7**
Average	59.6	60.4	65.5	66.0	**78.5**

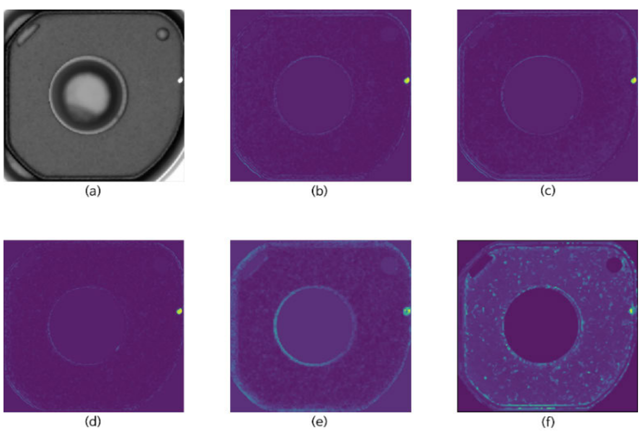

FIGURE 6. Coating defect result. (a) Input image (b) L2 autoencoder (c) SSIM autoencoder (d) Variational autoencoder (e) Vector quantized variational autoencoder (f) Proposed method.

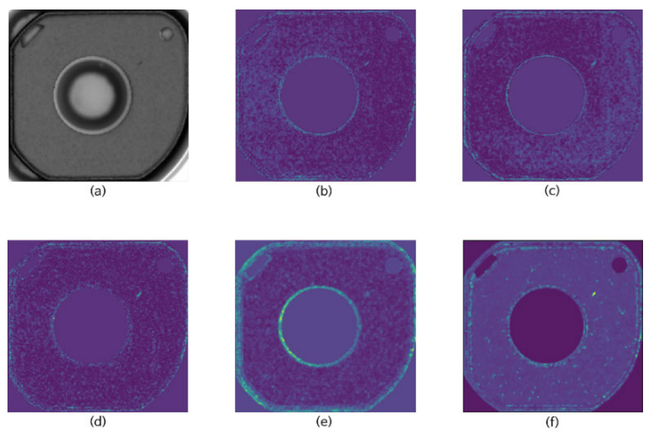

FIGURE 7. Scratch defect result. (a) Input image (b) L2 autoencoder (c) SSIM autoencoder (d) Variational autoencoder (e) Vector quantized variational autoencoder (f) Proposed method.

The proposed method is about 20 ms or slower in terms of inspection speed than the previous method because the vector quantization takes about 20 ms for processing. It takes much time to calculate the distance between the feature vector of encoder feature map and the prototype vector. In addition, the process of selecting the nearest prototype vector after calculating the distance also takes several times.

Table 3 shows the defect detection performance for each defect. Variational autoencoder showed the highest AUROC value among autoencoder-based anomaly detection methods except for the proposed method. The variational autoencoder-based anomaly detection method uses a probability distribution to approximate a normal sample's latent vector, resulting in a more accurate output image. For this reason, variational autoencoder shows higher performance than other autoencoder-based anomaly detection methods.

Figs. 6–8 show the detection results of coating, scratch, and stain defects, respectively. Each figure represents the differences between input and output images as well as the proposed method's feature vector frequency map. The coating defect is different from the normal region, so it is easy to distinguish it from the normal sample. However, some previ-

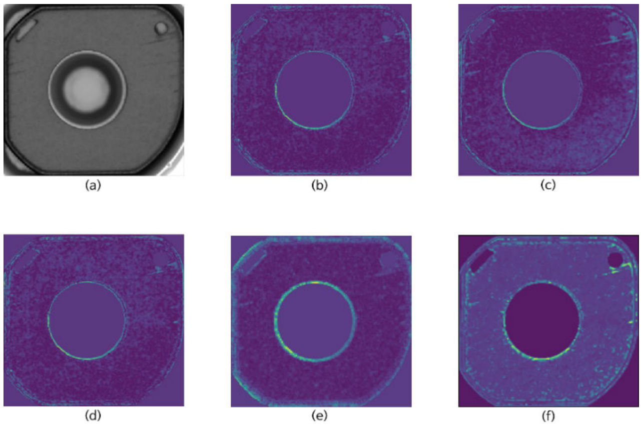

FIGURE 8. Stain defect result. (a) Input image (b) L2 autoencoder (c) SSIM autoencoder (d) Variational autoencoder (e) Vector quantized variational autoencoder (f) Proposed method.

ous methods have poor performance for classifying coating defects because the coating defect's size makes it easily confused with image noise. The proposed method showed high defect detection performance for coating defects by detecting defects with a feature vector frequency map. Figs. 7 and 8 show that scratch and stain defects are difficult to detect as there is no distinct difference from the normal region. It is difficult to detect defects because the previous method restores scratch defects and stain defects with similar textures to normal. The proposed method detects such defects by comparing the feature vector frequency map extracted from the prototype vector histogram.

VI. CONCLUSION

This paper proposed a defect detection method using the prototype vector histogram and the feature vector frequency map. The prototype vector histogram was extracted based on the number of times selected as the nearest prototype vector from the prototype vector of the vector quantization module. The prototype vector histogram obtained a feature vector frequency map from the input image. The abnormal score is calculated from the feature vector frequency map of the input image in the inspection stage to classify whether the input image is a normal or abnormal sample. In the experimental results, the proposed method showed higher AUROC values than other autoencoder-based defect detection methods. The proposed method showed higher accuracy than previous methods in defects with a similar texture to normal, such as stain defects and scratch defects.

However, the proposed method has an AUROC of 78.5%, which is a lower result than that because the actual inspection process requires more than 95% of defect detection accuracy. The proposed method uses a probabilistic model through normal samples. As a result, defect detection accuracy is reduced because it is difficult to extract an accurate model for a normal sample. In addition, the proposed method to which the vector quantization process is added takes more than twice as long as the previous autoencoder-based anomaly detection method.

In future studies, we plan to apply additional modeling techniques to improve the accuracy of defect detection. Even in the normal sample, there is a prototype vector with a low prototype vector histogram value. In particular, there are many false detections at the boundary of the image. To compensate for this, we plan to apply a Virtual Background extractor (ViBe) method to compensate for false detection. In terms of speed enhancement, prototype vector optimization is required. Some prototype vectors have similar values, and these prototype vectors cause redundant calculations of unnecessary distances.

REFERENCES

[1] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1999, pp. 582–588.

[2] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004, doi: 10.1023/B:MACH.0000008084.60811.49.

[3] L. Ruff, R. A. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Long Beach, CA, USA, 2019, pp. 4393–4402.

[4] L. Ruff, R. A. Vandermeulen, N. Goernitz, A. Binder, E. Müller, K. R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, 2019, pp. 4393–4402.

[5] B. Staar, M. Lütjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Procedia CIRP*, vol. 79, no. 1, pp. 484–489, 2019, doi: 10.1016/j.procir.2019.02.123.

[6] J. Andrews, T. Tanay, E. J. Morton, and L. D. Griffin, "Transfer representation-learning for anomaly detection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2016, pp. 1–5.

[7] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez, "Metric learning for novelty and anomaly detection," 2018, *arXiv:1808.05492*. [Online]. Available: http://arxiv.org/abs/1808.05492

[8] A. Roitberg, Z. Al-Halah, and R. Stiefelhagen, "Informed democracy: Voting-based novelty detection for action recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Newcastle, U.K., 2018, pp. 3–6.

[9] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2018, pp. 9781–9791.

[10] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3379–3388.

[11] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: One-class novelty detection using GANs with constrained latent representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2898–2906.

[12] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1517–1528, Aug. 2020, doi: 10.1109/TKDE.2019.2905606.

[13] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag. (IPMI)*, 2017, pp. 146–157.

[14] S. Yarlagadda, D. Gera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. Delp, "Satellite image forgery detection and localization using GAN and one-class classifier," in *Proc. IS&T Int. Symp. Electron. Imag.*, 2018, pp. 1–9.

[15] A. Vasilev, V. Golkov, M. Meissner, I. Lipp, E. Sgarlata, V. Tomassini, D. K. Jones, and D. Cremers, "Q-space novelty detection with variational autoencoders," 2018, *arXiv:1806.02997*. [Online]. Available: http://arxiv.org/abs/1806.02997

[16] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain MR images," in *Proc. Int. MICCAI Brainlesion Workshop*, 2018, pp. 161–169.
[17] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 481–490.
[18] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 372–380.
[19] Y. Lu and P. Xu, "Anomaly detection for skin disease images using variational autoencoder," 2018, *arXiv:1807.01349*. [Online]. Available: http://arxiv.org/abs/1807.01349
[20] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline, "Iterative energy-based projection on a normal data manifold for anomaly localization," in *Proc. Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–7.
[21] C. Huang, F. Ye, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," 2019, *arXiv:1911.10676*. [Online]. Available: http://arxiv.org/abs/1911.10676
[22] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps, "Towards visually explaining variational autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8639–8648.
[23] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," 2018, *arXiv:1805.06725*. [Online]. Available: http://arxiv.org/abs/1805.06725
[24] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5960–5973.
[25] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," 2020, *arXiv:2005.02359*. [Online]. Available: http://arxiv.org/abs/2005.02359
[26] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Seoul Nat. Univ., Seoul, South Korea, Tech. Rep. 2015-03, 2015.
[27] P. Burlina, N. Joshi, and I.-J. Wang, "Where's wally now? Deep generative and discriminative embeddings for novelty detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 11499–11508.
[28] R. Chalapathy, A. Krishna Menon, and S. Chawla, "Anomaly detection using one-class neural networks," 2018, *arXiv:1802.06360*. [Online]. Available: http://arxiv.org/abs/1802.06360
[29] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14866–14876.

YOUNG-GYU KIM received the B.S. degree from the School of Electronics Engineering, Chungbuk National University, South Korea, in 2015, and the M.S. degree from the Department of Control and Robot Engineering, Chungbuk National University, in 2017, where he is currently pursuing the Ph.D. degree. He is currently a Postdoctoral Researcher with the Industrial AI Research Center (IAIRC), South Korea. His research interests include machine vision and deep learning.

TAE-HYOUNG PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the Department of Control and Measurement Engineering, Seoul National University, South Korea, in 1988, 1990, and 1994, respectively. From 1994 to 1997, he worked as a Senior Researcher with Precision Instrument Institute, Samsung Techwin Company Ltd. Since 1997, he has been a Professor with the School of Electronics Engineering, Chungbuk National University. From 2000 to 2001, he was a Visiting Scholar with the University of Toronto. His main research interests include electronic assembly and testing systems, robot motion planning, and optimization algorithms.