IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 4, 2020, accepted November 25, 2020, date of publication December 1, 2020,
date of current version December 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041790

Spatially Variant Convolutional Autoencoder Based on Patch Division for Pill Defect Detection

SORA KIM[1], YOUNGJAE JO[2], JUNGCHAN CHO[3], (Member, IEEE), JIWOO SONG[1], YOUNYOUNG LEE[4], AND MINSIK LEE[1], (Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Hanyang University, Ansan 15588, South Korea
[2]ADAS Platform Cell Team, Hyundai Mobis Company Ltd., Yongin 16891, South Korea
[3]College of Information Technology, Gachon University, Seongnam 13120, South Korea
[4]Advanced Technology Center, Daekhon Corporation, Seoul 08381, South Korea

Corresponding author: Minsik Lee (mleepaper@hanyang.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2020R1C1C1012479, and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Grand Information & Communication Technology Research Center Support Program (Supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP)) under Grant IITP-2020-0-101741.

ABSTRACT Detecting pill defection remains challenging, despite recent extensive studies, because of the lack of defective data. In this paper, we propose a pipeline composed of a pill detection module and an autoencoder-based defect detection module to detect defective pills in pill packages. Furthermore, we created a new dataset to test our model. The pill detection module segments pills in an aluminum-plastic package into individual pills. To segment pills, we used a shallow segmentation network that is then divided into individual pills using the watershed algorithm. The defect detection module identifies defects in individual pills. It is trained only on the normal data. Thus, it is expected that the module will be unable to reconstruct defective data correctly. However, in reality, the conventional autoencoder reconstructs defective data better than expected, even if the network is trained only on normal data. Hence, we introduce a patch division method to prevent this problem. The patch division involves dividing the output of the convolutional encoder network into patch-wise features, and then applying patch-wise encoder layer. In this process, each latent patch has its independent weight and bias. This can be interpreted as reconstructing the input image using multiple local autoencoders. The patch division makes the network concentrate only on reconstructing local regions, thereby reducing the overall capacity. This prohibits the proposed network reconstructing unseen data well. Experiments show that the proposed patch division technique indeed improves the defect detection performance and outperforms existing deep learning based anomaly detection methods. The ablation study shows the efficacy of patch division and compression following the concatenation of patch-wise features.

INDEX TERMS Pill defect detection, deep neural network, convolutional autoencoder, unsupervised learning.

I. INTRODUCTION

In the manufacturing process, there can be defects in products that should be detected before they are packaged. Defective products on the market can cause problems that can lead to human casualties. Until recently, many companies relied on human inspectors that can induce excessive labor cost. With the development of deep learning, some companies are attempting to replace human inspectors with automatic testing systems. Defect detection is a task for detecting defects on data. Defect detection can be applied to various products such as fabric, metallic surface, pill, and so on. This is quite different from object detection which is a task for detecting and localizing predefined classes of objects. First of all, collecting defective data is very challenging, because we cannot anticipate all the types of defects in advance. Moreover, defects mostly appear as a part of the object of interest, which makes it hard to distinguish between a normal sample and a defective one. Accordingly, defect detection methods require an ability to handle the problem under the scarcity of annotated data. This paper studies the defect detection algorithm for pills using deep learning, to decrease the waste of human resources and increase the accuracy of defect detection. There are mostly three types of pill packages, i.e., bottle, bagged, and

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

aluminum-plastic packages. In this paper, we focus on pills in aluminum-plastic packages, which are easily accessible in pharmacies. Due to their complicated production process, defective pills are inevitably produced in aluminum-plastic packages [15].

There are various defect detection algorithms [15], [17], [35] based on Bayes classifier, support-vector machine, and mixtures of dynamic textures [5], [8], [9]. However, these are based on conventional machine learning techniques and their performance is rather limited. Recently, some deep learning-based methods have been proposed in various products, such as the detecting defects on fabric, metallic surfaces [19], [34], images and videos [4], [12], [23], [24], [30], [31], [37]. Especially, [19], [34] proposed autoencoder-based anomaly detection methods. An and Cho [2] introduced a variational autoencoder (VAE)-based anomaly detection method. Unlike the autoencoder-based anomaly detection method, which identifies anomalies with reconstruction errors, the VAE-based method identifies anomalies using reconstruction probabilities. There are also the generative adversarial network (GAN) based [3], [13], [25] anomaly detection methods [1], [18], [32]. Schlegl *et al.* introduced anomaly GAN (AnoGAN) using anomaly scores to detect anomalies in medical images. Following AnoGAN [33], Zenati *et al.* [36] introduced a model based on BiGAN [10] that learns an encoder network, along with a generator and a discriminator during training, that maps the input sample to a latent representation and evaluated the detection performance. Furthermore, there are adversarial learning-based anomaly detection methods that use the adversarial loss of GAN to detect anomalies without a generator [28], [29]. Above methods have proposed a new way of detecting defects or anomalies in a data, however, they are not appropriate for pill data. The main reason is that these methods are not suitable for training low variance data, which the pill data is. Furthermore, adversarial learning-based methods need to train an additional network to detect and localize the anomalies and have too high capacity for pill data. On the other hand, Du *et al.* have proposed a change detection algorithm for remote sensing images [11]. The problem that they deal with is somewhat similar to the anomaly/defect detection problem but has a fundamental difference: Their problem is to find the difference between two inputs of data samples, while ours is to distinguish the defective samples from a set of non-defective samples. In other words, the non-defective samples are not unique and these samples also form a distribution in the image space.

Although various studies have been conducted on defect detection, detecting pill defects remains a challenge and has many issues to handle. As mentioned earlier, the defective pills should be detected before they are released into the market. In the process, each pill should be inspected. However, in the manufacturing system, several pills are packaged together in a single package. To detect defects in pills, a cropped image containing a single pill should be extracted from the image of a package. A naïve method would be to simply segment the pills by thresholding the pixel values and applying mathematical morphology to the result. However, this process might not be successful when the color of the pill is similar to that of the package. To resolve this problem, we introduce a pill detection module in this paper.

After separating the images of individual pills, a defect detection method must be applied to find defects. However, as mentioned earlier, existing autoencoder-based defect detection networks is not the best choice for this purpose. For more accurate results, we introduce the patch division method in this paper. If we align a cropped pill image accurately, the aligned images have low variance. However, most autoencoders are usually too complex for reconstructing these data, which end up being capable of reconstructing unseen (defective) data. Hence, we propose a spatially variant convolutional autoencoder based on the newly introduced patch division method that is designed to be trained only on normal data so that it can only reconstruct the normal pills. We apply the patch division method on the patch-wise features extracted from the output of the convolutional encoder network. These features are encoded using the respective patch-wise encoders, which have independent weights and biases for different patches, hence the name spatially variant autoencoder. The proposed network can have lower capacity due to this patch-wise structure, which enforces the network to largely focus on local information. Furthermore, the computational complexity of the patch division method is comparable to a regular convolution. Therefore, adding the patch division method to an autoencoder or a VAE is not too much of a burden. The spatially variant autoencoder learns normal data with the patch division method and detects pill defects successfully.

The overall structure of the proposed method consists of the pill detection module and the defect detection module, i.e., the spatially variant autoencoder. The pill detection module estimates the pill segments using a deep network. The distance transform [22] and the watershed algorithm [20] are then used to divide the package image into individual pill images. The defect detection module, i.e., the proposed spatially variant autoencoder, is based on conventional autoencoder-like structures but has additional layers for patch division, which make the network largely focus on reconstructing local regions. The proposed networks are based either on a convolutional autoencoder or a variational autoencoder (VAE); however, we expect that other generative models can also be used here. Our pill defect detection method does not need annotated data. Moreover, the proposed network doesn't need any additional network to train the overall algorithm, unlike the adversarial methods. Figure 1 shows the overall pipeline of the proposed method. Furthermore, in this paper, we also introduce a new dataset to evaluate our networks. The experiments show that the proposed networks have better performance than the other autoencoder-based baseline methods. Furthermore, we compared our method with the existing deep-learning based anomaly detection methods and show that the proposed methods outperform

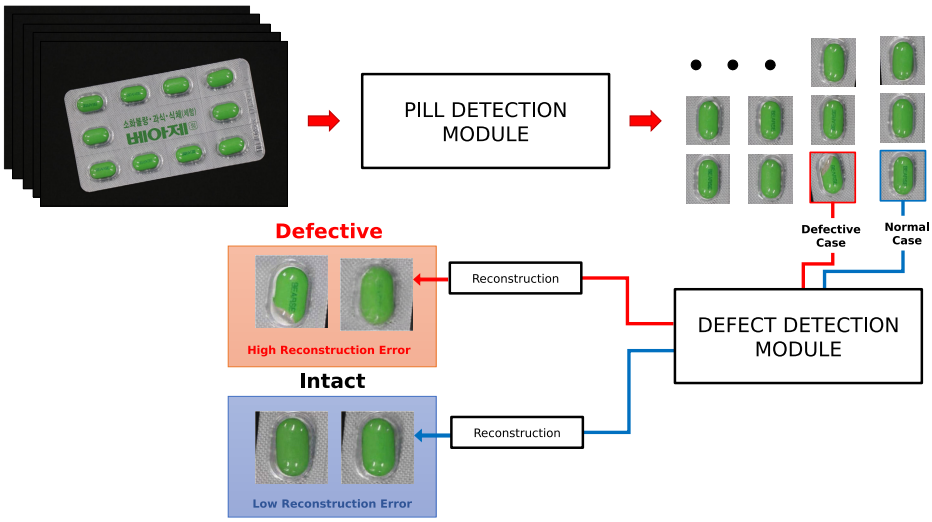


FIGURE 1. Overall pipeline of the proposed method.

the anomaly detection methods. We also conducted ablation studies to demonstrates that the patch division method indeed improves the defect detection performance.

Our contributions can be summarized as follows:

- We propose a pipeline that automates the pill defect detection process based on the pill detection module and defect detection module.
- We propose the patch division method to lower the capacity of a defect detection network, because the low variance of the data poses challenges in learning only the information of normal data. As a result, it improves the defect detection performance.

The remainder of this paper is organized as follows: In Section II, we present the background of this work, i.e., the autoencoder and VAE. In Section III, we explain the pill detection module. In Section IV, we introduce the architecture of the defect detection module. Section V presents the implementation details of our model. Section VI and Section VII detail the experimental results and ablation studies, respectively. The conclusion follows in Section VIII.

II. BACKGROUND

The proposed method is based on autoencoders and VAEs. In Sections II-A and II-B, we explain the concepts of an autoencoder and a VAE, respectively.

A. AUTOENCODER

In this paper, we use autoencoders to differentiate defective data from normal data. The goal of using autoencoders is to train them so that they are only capable of reconstructing the normal data.

Autoencoders are a particular structure of neural networks that are used in many problems such as image reconstruction and information encoding. They are often used to learn a meaningful latent representation from input data. An autoencoder consists of an encoder network and a decoder network. The encoder learns the mapping between the input data and a multi-dimensional latent space. On the other hand, the decoder network learns to reconstruct the original input image from the learned feature map of the encoder network. The loss function of an autoencoder is the difference between the original input data and the reconstructed data.

$$z = h(x) \tag{1}$$
$$\hat{x} = g(z) = g(h(x)) \tag{2}$$
$$L = ||x - \hat{x}||^2 \tag{3}$$

The equation (1) maps an input vector x to a latent variable h using the encoder network. The equation (2) maps the latent variable h to the reconstructed vector $\hat{x}$. An autoencoder is usually trained to minimize (3), which is called reconstruction error. No label is required in this learning process, so it is called unsupervised learning.

B. VARIATIONAL AUTOENCODER

A VAE is a generative model that approximates a posterior density using variational inference [16]. It is based on an autoencoder-like structure and the latent variable is assumed to be a random variable. Let us consider a dataset $X = \{x^{(i)}\}_{i=1}^{N}$ consisting of N i.i.d. samples of some continuous or discrete random variable x. The marginal likelihood of data to be maximized is the sum of the marginal likelihoods of individual data:

$$\log p_\theta(x^{(1)}, \ldots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}) \tag{4}$$

Let us introduce a known model $q_\phi(z|x)$, i.e., an approximation to the unknown model $p_\theta(z|x)$. The marginal likelihood of each data can be represented as follows:

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) + \mathbb{E}_{q_\phi(z|x)}[-\log q_\phi(z|x) + \log p_\theta(x, z)] \tag{5}$$

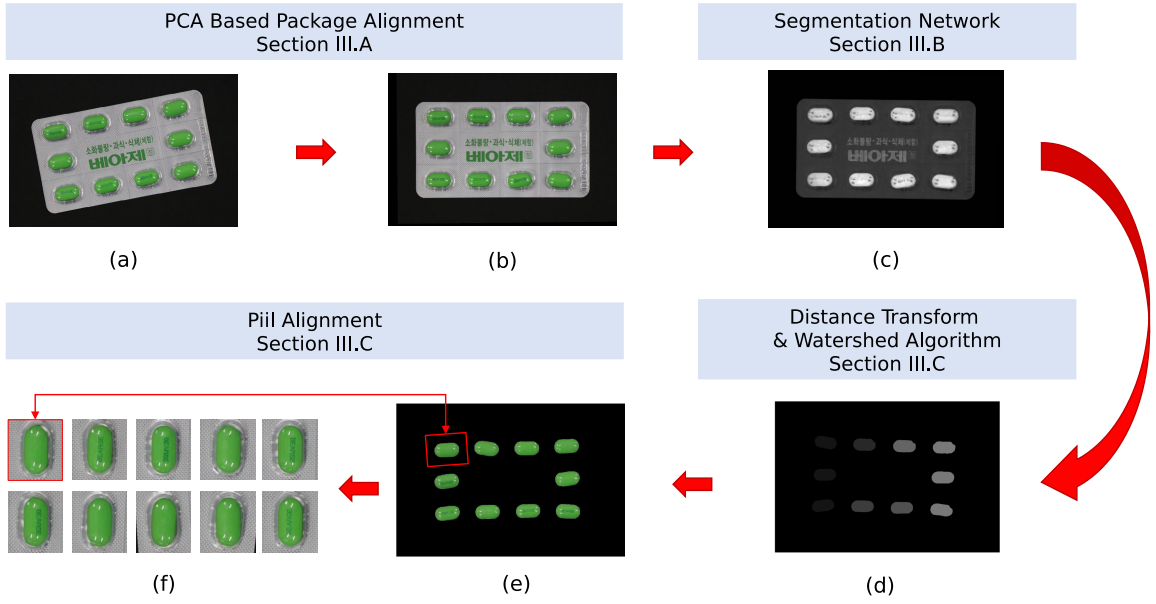


FIGURE 2. Data pre-processing pipeline.

Because the KL divergence term is greater than or equal to zero, the equation (5) can be rewritten as follows:

$$\begin{aligned}\log p_\theta(x^{(i)}) &\geq L(\theta, \phi; x^{(i)}) \\ &= \mathbb{E}_{q_\phi(z|x)}[-\log q_\phi(z|x) + \log p_\theta(x, z)] \\ &= -D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \\ &\quad + \mathbb{E}_{q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] \end{aligned} \tag{6}$$

Now, the optimal parameter can be found by solving the problem:

$$(\theta^*, \phi^*) = \arg\max_{\theta,\phi} L(\theta, \phi; x^{(i)}), \tag{7}$$

To train a VAE, the loss function should be differentiable. However, the last term of the right-hand side in the equation (6) is not differentiable, because the sampling is not a differentiable operation. Kingma and Welling [16] introduced the reparameterization trick to make the model to be deterministic (differentiable) by making $q_\phi(z|x) = \mathcal{N}(\mu(x), \sigma(x))$ if we choose Gaussian. Then, z can be expressed as $z = \mu(x) + \Sigma(x)^{1/2} \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Finally, z becomes differentiable with respect to (w.r.t.) the parameter (μ, Σ).

III. PILL DETECTION MODULE

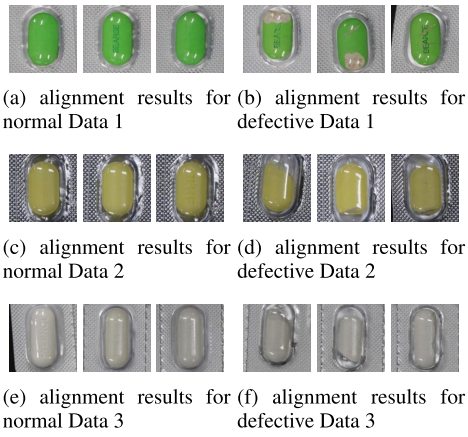
In the manufacturing process, several pills are packaged in a single aluminum package. Then, many of these pill packages are carried on conveyor belts before it is packed in paper boxes. Unfortunately, there can be defects in these products. Thus, before they are on the market, it should be examined if there is a defective pill in each pill package. Even if there is only one defective pill in an aluminum package, the entire package should be abandoned. The images of the package on the conveyor belts can be easily obtained with a camera, so using computer vision and deep learning techniques to detect those defects can be a cost-efficient way for handling this problem. In order to realize this system, we first divide the image of a single package into cropped images containing individual pills so that we can concentrate on each pill.

The pre-processing procedure for preparing the training data is three-fold. The pipeline of the data pre-processing part is shown in Figure 2. First, we align the pill package using the principal component analysis (PCA) algorithm (Section III-A). Second, we annotate the pills in the aligned results. Then, we train a segmentation network to segment the pills (Section III-B). Finally, we separate individual pills using the segmentation result (Section III-C).

A. PACKAGE ALIGNMENT WITH PCA

We used similarity transforms to align packages, because the manufacturing environment restricts the experiment conditions such as camera angle and location of the camera. The package alignment makes the pill segmentation easier. It has been a popular technique to apply PCA to the coordinates of positive points in mask images in order to find similarity transformations and align images. For example, Mudrová and Procházka described two applications of PCA in image processing. One is image compression and the other is image rotation [21]. Recently, Rehman and Lee used PCA to align medical images [26].

We applied PCA to the coordinates of the edges detected by the Canny edge detector [7] to find the principal axes. Then, the input images are aligned based on the principal components of the coordinates. Furthermore, to obtain more accurate principal components, we applied a median filter to denoise the detected edges. Then, edges inside of the package region are discarded by examining each edge pixel whether it



(a) alignment results for normal Data 1 (b) alignment results for defective Data 1

(c) alignment results for normal Data 2 (d) alignment results for defective Data 2

(e) alignment results for normal Data 3 (f) alignment results for defective Data 3

FIGURE 3. Alignment results for the three data.

is the left-most, right-most, top-most, or bottom-most among the edge pixels in the same row or column.

B. SEGMENTATION NETWORK

We built a simple segmentation network to separate a package into individual pills. The training data are the alignment result of the pill packages and the labels have been annotated manually. The network is composed of two convolutional layers and a ReLU layer. The output of the segmentation network is the mask of the detected pills in the input image. Because the data is quite simple, two convolutional layers are sufficient to detect pills in package images. The dimensions of the convolutional layers are 128 and 1, respectively, and the kernel size is 3.

C. DISTANCE TRANSFORM AND WATERSHED ALGORITHM

To divide a package image into individual pill images, we applied the distance transform [22] and the watershed algorithm [20] to the mask. Then, the mask is separated for each pill. We again applied PCA to the coordinates of the mask of each pill to find their principal axes to align each pill image. From the center of each pill, we crop the $160 \times 160 \times 3$ pill image, considering the principal axes.

Figure 3 shows the result of the pill alignment. After the alignment, the normal pills are aligned with the center. However, the defective pills with cracks or peeled surfaces are often not perfectly aligned, because the masks of the defective pills are usually unbalanced.

IV. DEFECT DETECTION MODULE BASED ON AN AUTOENCODER WITH PATCH-WISE FEATURES

The proposed spatially variant autoencoder is trained only on normal images. We assume that the network is only trained on normal data and is therefore unable to reconstruct defective pills correctly. However, if the network has a good generalization performance, even if the network is trained only on the normal images, the defective pills would be reconstructed correctly as well. This may be attributed to the high capacity of a deep neural network (DNN). The proposed method effectively minimizes the capacity of the network while maintaining its ability to represent the normal samples accurately, which makes it a good fit for unsupervised defect detection.

In the proposed method, the convolved features are divided into patch-wise features and then they are applied to the patch-wise encoders. This method is hereafter referred to the patch division method. The patch division method effectively suppresses the capacity of the network, compared to when only the plain fully connected layers are used, because the patch division method reduces the number of parameters by approximately N_p^2 times where N_p is the number of patches. The patch division method enforces the network to concentrate on local areas rather than the global area, thereby effectively degrading the generalization ability of the linear layers in the network.

Figure 4 shows our network's architecture. The network is basically an autoencoder. The encoder consists of three parts, i.e., the convolutional part, the patch-wise part, and the global part. Likewise, the decoder has corresponding parts that are inverted versions of the above parts.

The input size of the proposed network is $160 \times 160 \times 3$, the same as the pill image extracted in Section III. An input image passes through five convolution layers to become a $80 \times 80 \times 16$ feature map. These convolution layers are the convolutional part mentioned above. The convolutional part learns the low-level features of the normal data. Then, we divide the convolved features into 400 disjoint patches. Let us consider a feature map $F \in \mathbb{R}^{W \times H \times C}$ that is divided into patches $\{P_{ij} \in \mathbb{R}^{m \times n}\}$. The pixels of P_i can be a derived from F as follows:

$$P_{ij}(x, y) = F(x + n(j-1), y + m(i-1)) \quad \text{for } 1 \le x \le n,\ 1 \le y \le m. \tag{8}$$

In the patch-wise encoding layer, each patch has its own weight and bias. This can be viewed as applying a fully connected layer to each patch. The encoded patch-wise features are then vectorized and concatenated. The concatenated feature vector passes through the global encoding layer to further compress the concatenated features into latent features. Note that, because the patch-wise features have been already compressed by individual patch-wise encoding layers, this global encoding layer has to only deal with the dimension-reduced versions of the features, which can greatly reduce the capacity of the overall encoder structure.

The decoder has a similar structure to the encoder. The latent features pass through the global decoding layer, and they are reshaped again into patch-wise features to which the patch-wise decoding layer is applied. The decoded patch-wise features are reshaped to form a $80 \times 80 \times 16$ feature map and pass through five deconvolution layers, i.e., the convolutional decoding layers, and one sigmoid layer.

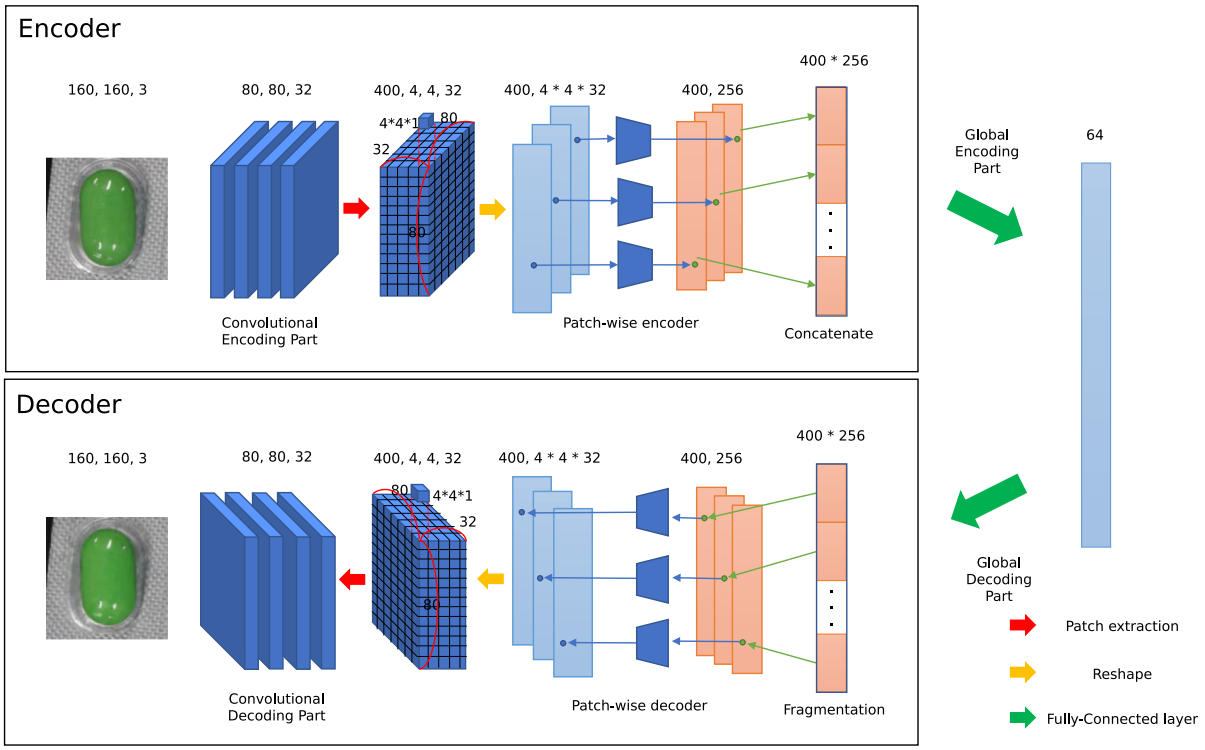


FIGURE 4. Architecture of the proposed defect detection module based on the pill reconstruction network.

TABLE 1. Kernel sizes and strides of the layers in the proposed autoencoder (Autoencoder with patch division). The VAE with patch division has a similar structure with the autoencoder except for an additional Gaussian inference. We refer to the spatially variant fully connected layer as SVFC. SVFC 1 corresponds the patch-wise encoder and SVFC 2 the patch-wise decoder.

	size	stride		size	stride
input	$160 \times 160 \times 3$	-	FC layer 2	64×102400	-
Conv 1	$7 \times 7 \times 3 \times 16$	1	SVFC 2	$400 \times 256 \times 512$	-
Conv 2	$7 \times 7 \times 16 \times 16$	1	Deconv 1	$7 \times 7 \times 16 \times 32$	2
Conv 3	$7 \times 7 \times 16 \times 16$	1	Deconv 2	$7 \times 7 \times 16 \times 16$	1
Conv 4	$7 \times 7 \times 16 \times 16$	1	Deconv 3	$7 \times 7 \times 16 \times 16$	1
Conv 5	$7 \times 7 \times 16 \times 32$	2	Deconv 4	$7 \times 7 \times 16 \times 16$	1
SVFC 1	$400 \times 512 \times 256$	-	Deconv 5	$7 \times 7 \times 3 \times 16$	1
FC layer 1	102400×64	-	Sigmoid	$160 \times 160 \times 3$	-

V. IMPLEMENTATION DETAILS

In this section, we will explain the implementation details of our network and experimental settings. We conducted experiments on an Nvidia Titan Xp GPU. We used the binary cross-entropy loss instead of the mean squared error, because the latter can cause blurry outputs. Further, we used the Adam optimizer where the learning rate and the weight decay were both fixed as 10^{-4}. All the activation functions were set to leaky ReLU. The size of the mini-batch was five. The number of training epochs was 200. Table 1 shows our network. The size of the input image is $160 \times 160 \times 3$. By passing through the convolutional encoding layers, the dimension of the input features becomes $80 \times 80 \times 32$. The kernel size was seven for all convolutional layers, and the strides were one for from Conv 1 (Deconv 2) to Conv 4 (Deconv 5) and two for Conv 5 (Deconv 1). The number of patches were 400 and the width and height of each patch were both set to four, which achieved the best performance as shown in Figure 6. After vectorizing each patch-wise feature, there were 512 channels; these were then encoded into 256 channels by the patch-wise encoding layer. These 400 256-channel vectors were then concatenated, and further encoded into a 64-channel vector. The decoder has a similar inverted structure to reconstruct a $160 \times 160 \times 3$ image.

A. COMPUTATIONAL COMPLEXITY AND PROCESSING TIME

In this section, we compare the computational complexities between a regular convolution and the patch division method. Let us consider a feature map $F \in \mathbb{R}^{W \times H \times C}$. If we apply a convolution to the feature with a $k \times k$ kernel whose output channel size is C', the computational complexity of

TABLE 2. Processing time (sec) of the proposed methods.

Method	training time	test time
Autoencoder	4913.44	48.73
VAE	4755.81	47.13
AE with PD	6797.59	67.63
VAE with PD	6839.14	69.19

the convolution is $O(WHCC'k^2)$. On the other hand, if we apply the patch division method with patch size $m \times n$, the number of patches becomes $(WH)/(mn)$, and the fully-connected operation for each patch takes $O(mnCD)$ where D is the dimension of output vectors. Accordingly, the computational complexity of the patch division method becomes $O(WHCD)$. Depending on the value of D, the computational complexity and the output size of the patch-wise layer can be different: If $D = C'k^2$, the complexity is the same as that of the previous convolution. However, the size of the output will become $(WH)/(mn) \times C'k^2 = WHC' \times k^2/(mn)$, which is WHC' for the previous convolution, so the output size of the patch-wise layer is either larger or smaller depending on the ratio between k^2 and mn. If $D = mnC'$, on the other hand, the output size becomes $(WH)/(mn) \times mnC' = WHC'$, which is the same as that of the convolution. In this case, the complexity becomes $O(WHCC'mn)$, which is again either larger or smaller than that of the convolution depending on the ratio between k^2 and mn. k, m, and n are usually small positive integers around three to seven, thus we can say that both the operations have comparable complexities.

Table 2 shows the processing time of the proposed methods. As shown in the table, all the training took around one hour. Here, we refer to the patch division as PD. Note that the plain autoencoder (or VAE) had a similar structure to that described in Table 1, replacing the SVFC layers with convolutional layers with the same output sizes. k was set to seven and both m and n were set to four, so the computational complexities of the SVFC layers were about three times smaller. However, the above table shows that the plain autoencoder (or VAE) was faster, and we conjecture that this has to do with the convolution operation being optimized on GPUs with CUDA.

VI. EXPERIMENTS

A. DATASET AND EVALUATION METRIC

1) DATASET

We controlled the image acquisition process to have identical conditions for illumination, focal length, and the distance between the pills and the camera. For the convenience of segmentation, the background of the data was set to a black-colored paper. We selected three different types of pills to demonstrate the efficacy of our networks for diverse data. Data 1 was the easiest one. The difference between normal and defective data was very distinctive, because the pill was white inside and green outside, as shown in Figure 3. Data 2 was the hardest one in terms of defect detection, as shown in Figure 3, because there was major light reflection on the package unlike the other datasets. Data 3 was the hardest data for the segmentation task, because the colors of the package and pills were very similar. To produce defective pills, we manually repackaged some pills after breaking them with hands. Each dataset had 2000 segmented normal pill images that were used to train the spatially variant autoencoder. Additional 500 normal and 500 defective images were used for validation and test. These sets were randomly sampled and fixed, and cross-validation was not used in the experiments. We compared the performance of our networks with that of a plain autoencoder and a plain VAE. Although the pill dataset is quite small compared to what is usually used in deep learning, but the variance in the pill data is also small due to the restricted data acquisition process. Therefore, it was enough to learn the features of the pill data using the patch division method.

2) EVALUATION METRICS

The receiver operating characteristic (ROC) curve and area under the curve (AUC) were used for evaluation. To find the optimal parameters, such as the number of patches, the number of channels, and the kernel size, we conducted a random search on a few dozen cases. Following the random search, we chose a few parameters that had high detection performance and conducted narrow tuning around them to identify the best parameters.

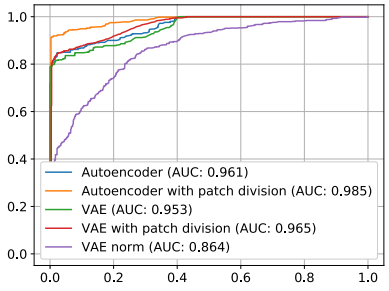
B. QUANTITATIVE RESULTS

Table 3 shows the AUC values for each data and the experiments are conducted on AE, VAE, AE with PD, and VAE with PD. The values in the table are the average AUC values and standard deviations of five trials. Here, (A) indicates that the same hyper-parameters were used for all data, while (O) means that different optimal hyper-parameters were selected for each data. When we compared AE (A) with AE with PD (A), the smallest increase in the AUC value was 2.21% on Data 1, and the greatest increase in the AUC value was 4.1% on Data 2. Moreover, note that AE with PD (A) was not less effective than AE (O), even though AE (O) was much more finely tuned. Similarly, the AUC values of VAE with PD (A) were greater than those of VAE (A). The smallest increase in the AUC value was 0.69% on Data 1, and the greatest increase in the AUC value was 5.07% on Data 2. Furthermore, VAE with PD (A) demonstrated better performance than VAE (O), although VAE (O) was much more finely tuned. The table shows that the proposed method does not show much increase in performance for Data 1 as for the other datasets. This is because Data 1 is the easiest data for detecting defects as mentioned in Section VI-A. Accordingly, defects in Data 1 can be easily detected even though the patch division is not used.

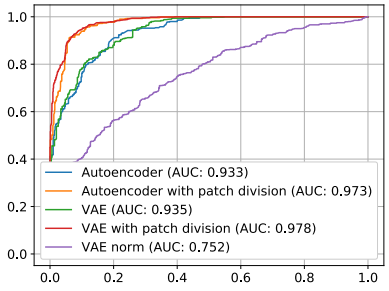
Figure 5 shows the ROC curve for each data on all the proposed methods. The defectiveness of the result obtained using the plain VAE is measured either based on the reconstruction error or the norm of the latent features. If the norm of the latent features is close to 0, it can be interpreted to

TABLE 3. AUC table. (A) means that the same hyperparameters are used for all data. (O) means that different optimal hyperparameters are selected for each data, i.e., hyperparameters are tuned for each data.

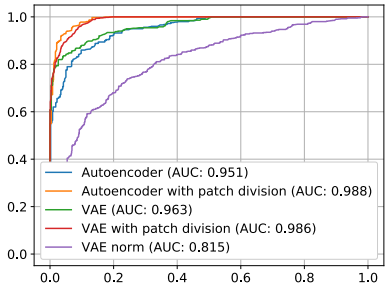
Data	AE		AE with PD		VAE		VAE with PD	
	(A)	(O)	(A)	(O)	(A)	(O)	(A)	(O)
Data 1	96.21±0.67	96.92±0.46	98.42±0.42	**98.54**±0.33	95.95±0.94	96.86±0.53	96.64±0.44	96.70±0.31
Data 2	93.44±1.44	93.44±1.44	97.54±0.62	97.62±0.06	92.79±0.71	92.90±1.44	**97.86**±0.08	**97.86**±0.08
Data 3	95.01±0.88	95.08±0.71	**98.97**±0.11	**98.97**±0.11	96.45±0.82	96.45±0.82	98.39±0.63	98.39±0.63
mean	94.89	95.14	98.31	**98.37**	95.06	95.40	97.63	97.65



(a) ROC curve for Data 1

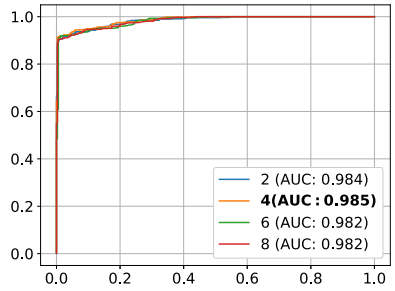


(b) ROC curve for Data 2

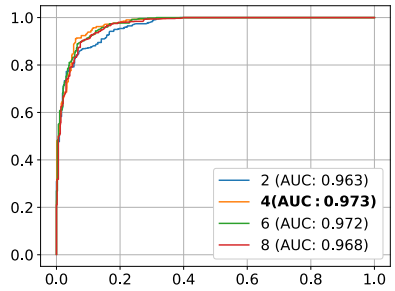


(c) ROC curve for Data 3

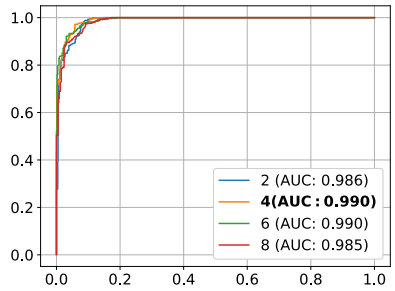
FIGURE 5. ROC curves for the three datasets. The orange, blue, green, red, and violet lines show the performances of the proposed autoencoder with the patch division method, the plain autoencoder, the plain VAE, the VAE with the patch division method, and defect detection based on the norm of the latent variable of the plain VAE, respectively.



(a) ROC curve for different sizes of patches on Data 1



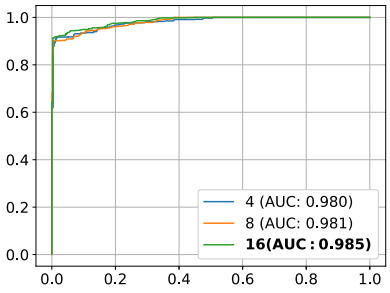
(b) ROC curve for the different sizes of patches on Data 2



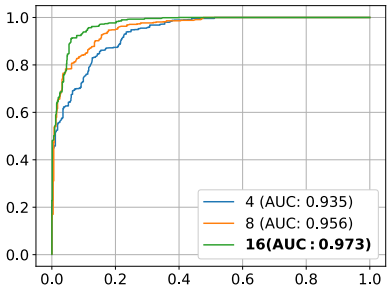
(c) ROC curve for the different sizes of patches on Data 3

FIGURE 6. Performance for different sizes of patches on each dataset. The blue, orange, green, and red lines indicate the ROC curves when the patch sizes are two, four, six, and eight, respectively.

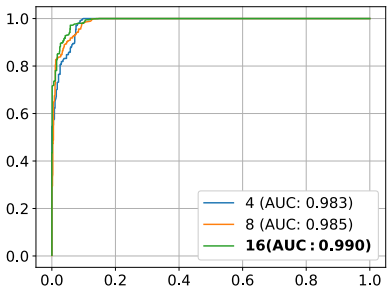
mean that the input image is normal, because the network is trained on normal data and the latent features of a VAE are supposed to be standard Gaussian. For all the other networks, the reconstruction error was used. Figure 5(a) and Figure 5(c) shows that the detection performance of the proposed networks was better, compared to the other networks, on all datasets. The ROC curves on all the data show that using the norm of the latent variable in VAE is also useful for detecting defects. When the latent variable ($\hat{z}$) of the whole training data follows $\mathcal{N}(\hat{\mu}, \hat{\sigma})$, and the latent variable (z) of a given test image follows $\mathcal{N}(\mu, \sigma)$, we can translate the center of the latter distribution to zero by subtracting $\hat{\mu}$. As shown in Figure 5, measuring the norm of VAE can be used



(a) ROC curve for the different numbers of channels on Data 1

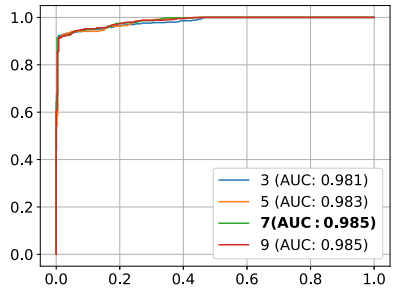


(b) ROC curve for the different numbers of channels on Data 2

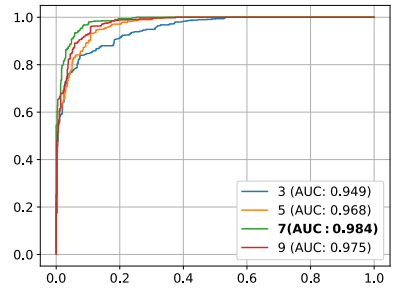


(c) ROC curve for the different numbers of channels on Data 3

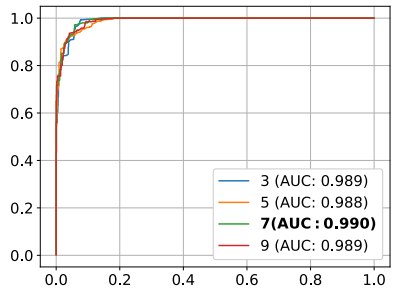
FIGURE 7. Performance for different numbers of channels on each dataset. The blue, orange, and green lines indicate the ROC curves when the channel sizes are four, eight, and 16, respectively.



(a) ROC curve for the different sizes of kernels on Data 1



(b) ROC curve for the different sizes of kernels on Data 2

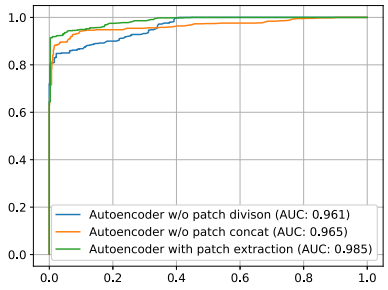


(c) ROC curve for the different sizes of kernels on Data 3

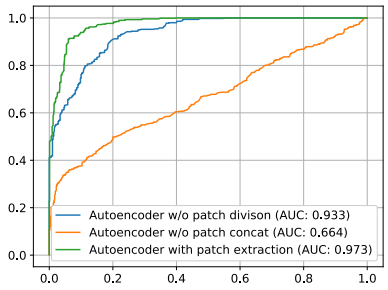
FIGURE 8. Performance for different kernel sizes on each data. The blue, orange, green, and red lines indicate the ROC curves when the kernel sizes are three, five, seven, and nine, respectively.

for detecting defects on pills but shows lower detection performance than measuring the reconstruction error. We conjecture that the distribution of the latent variable may not be exactly Gaussian even if it is enforced during training, because it can be hard to produce a perfect Gaussian distribution.

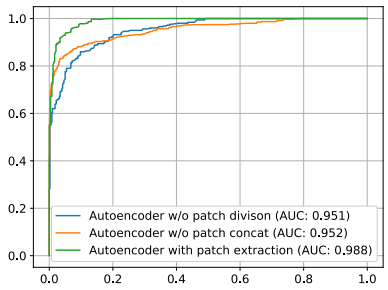
Figure 6 to 8 show the quantitative experiments of hyper-parameter tuning on the patch, channel, and kernel sizes. These experiments were conducted only on the autoencoder with the patch division method. The parameters that are finally selected are shown in bold. Figure 6 shows the performance of the proposed network on different patch sizes. We compared the patch size from 2×2 to 8×8. The performance was similarly good when the patch size was either four or six; however, we chose four, because the average performance was at its best when the patch size was four. Figure 7 shows the performance of the proposed network on different number of channels. Note that the numbers of channels of Conv 5 and Deconv 1 were set as twice as those of the other convolutional layers for all the cases. As shown in the figure, the higher the number of channels, the better the performance become. We tested up to 16 channels due to the memory limitation. Figure 8 shows the performance on different sizes of kernels, which were three, five, seven, and nine. Although the performances were similar for all the kernel sizes on Data 1 and Data 3,



(a) ablation study on Data 1



(b) ablation study on Data 2



(c) ablation study on Data 3

FIGURE 9. Ablation study. The orange, green, and blue lines indicate the ROC curves of the proposed network, that without the global encoding-decoding part, and that without patch division and global encoding-decoding, respectively.

it achieved the best performance when the size of kernels was seven on Data 2.

Table 4 shows a comparison with the existing anomaly detection networks. The defect detection performance of f-AnoGAN [32] is lower than 0.6, which indicates that it is not an appropriate algorithm for detecting defects in pills since detecting defects performance is nearly random (0.5). Adversarially learned one-class classifier for novelty detection method (ALOCC) [28] did somewhat better than f-AnoGAN but shows much lower detection performance than the proposed methods. These generative-model-based methods are designed for general data such as CIFAR-10, SVHN, and KDD99. Accordingly, these methods are not appropriate for learning the pill data which has very low variance. On the other hand, the proposed methods with the patch division method learns spatially variant features from the pill data with minimum capacity, which, as result, succeeds in detecting defective samples.

TABLE 4. Quantitative comparison with existing anomaly detection methods. The unit of AUC values are percentage.

Method	Data1	Data2	Data3
f-AnoGAN [32]	56.19	53.44	50.09
ALOCC [28]	87.42	85.00	70.18
AE with PD	98.42	97.54	98.97
VAE with PD	96.64	97.86	98.39

TABLE 5. Reconstruction results of the proposed autoencoder for three data.

	Normal		Defective	
	Input	Output	Input	Output
Data 1				
Data 2				
Data 3				

TABLE 6. AUC table (without vs with segmentation network).

Data	w/o segmentation network	with segmentation network
Data1	51.51	97.40
Data2	60.43	94.66
Data3	80.97	96.49
mean	64.30	96.18

C. QUALITATIVE RESULTS

Table 5 shows the data samples and reconstruction result of each sample. As mentioned earlier, the proposed network was only trained on normal data; therefore, it was unable to reconstruct the defective data correctly. Because we trained only on normal data, our method yielded almost perfect reconstruction results when the input image was normal. However, when the input image was of a defective pill, the output was quite blurry and more similar to normal data.

VII. ABLATION STUDIES

The most important parts of the spatially variant convolutional autoencoder are the pill detection module and the defect detection module, and the most essential parts of each module are the segmentation network and the patch division method, respectively. In this section, we demonstrate the effects of the segmentation network and the patch division method. Thus, we conducted ablation studies to check the importance of the segmentation network and each part of spatially variant autoencoder i.e., global encoding layer, global decoding layer, and patch division layer.

A. SEGMENTATION NETWORK

In this paper, we used the segmentation network to detect pills in a package. Table 6 shows the defect detection performance for both when the data input packages were segmented with the segmentation network and when they were not. Without the segmentation network, the packages were simply segmented by thresholding the color values of the pills, and then applying mathematical morphology to the result. The detection performance without the segmentation network on Data 1 was 51.51%. However, with the segmentation network, the detection performance was 97.40%. Similarly, the differences in the detection performance between using the segmentation network or not on Data 2 and Data 3 were 34.23% and 15.52%, respectively. The average improvement of the detection performance was 32.19%. Thresholding was not very effective, because it was too simple to detect pills. For example, for Data 3, it was hard to separate the pills from the package, because they had similar colors. The other segmentation methods, such as GrabCut [27] and GraphCut [6], [14], can be used instead of thresholding and applying mathematical morphology; however, they require user input during the test phase. Because the purpose of this paper is to build a complete system for pill defect detection, we consider these as out of its scope.

B. PATCH DIVISION

We conducted an ablation study on three cases, to verify the effectiveness of the patch division method. Figure 9 shows the pill defect detection performance under different conditions. Compared to our network, the performance of the network without patch division deteriorated. Furthermore, the performance of the network without global encoding-decoding also deteriorated. This experiment shows that global encoding-decoding is also important as well. With the global encoding-decoding part, our network can learn some additional global information that is helpful for detecting defects. Furthermore, patch division makes the network learn local information, which can reduce the overall capacity of the network.

VIII. CONCLUSION

In this paper, we introduced the pill detection module and the defect detection module based on the newly proposed spatially variant autoencoder. We demonstrated that the proposed patch division method could improve the defect detection performance. We experimented with different parameters, such as kernel, channel, and patch sizes, and selected the best hyper-parameters. Although we conjecture that larger channels would result in better performance, due to the lack of memory space, the largest channel size we could afford was 16. We only tested the patch division method on an autoencoder and a VAE, but we expect that the patch division method can also improve the detection performance on GAN. We will explore this further in future work.

REFERENCES

[1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2018, pp. 622–637.

[2] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lect. IE*, vol. 2, no. 1, pp. 1–18, 2015.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[4] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4183–4192.

[5] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory COLT*, 1992, pp. 144–152.

[6] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Jul. 2001, pp. 105–112.

[7] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[8] A. B. Chan and N. Vasconcelos, "Mixtures of dynamic textures," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 641–647.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[10] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*. [Online]. Available: http://arxiv.org/abs/1605.09782

[11] B. Du, L. Ru, C. Wu, and L. Zhang, "Unsupervised deep slow feature analysis for change detection in multi-temporal remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9976–9992, Dec. 2019.

[12] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[14] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *J. Roy. Statist. Soc. B, (Methodol.)*, vol. 51, no. 2, pp. 271–279, 1989.

[15] Y. Jiang, S. Ma, H. Gao, and F. Xia, "Research on defect detection technology of tablets in aluminum plastic package," *Open Autom. Control Syst. J.*, vol. 6, no. 1, pp. 940–951, Dec. 2014.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[17] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 367–386, Mar. 2015.

[18] J. Liu, C. Wang, H. Su, B. Du, and D. Tao, "Multistage GAN for fabric defect detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3388–3400, 2019.

[19] S. Mei, Y. Wang, and G. Wen, "Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model," *Sensors*, vol. 18, no. 4, p. 1064, Apr. 2018.

[20] F. Meyer, "Topographic distance and watershed lines," *Signal Process.*, vol. 38, no. 1, pp. 113–125, Jul. 1994.

[21] M. Mudrova and A. Procházka, "Principal component analysis in image processing," in *Proc. MATLAB Tech. Comput. Conf.*, Prague, Czech Republic, 2005, pp. 1–4.

[22] D. W. Paglieroni, "Distance transforms: Properties and machine vision applications," *CVGIP: Graph. Models Image Process.*, vol. 54, no. 1, pp. 56–74, Jan. 1992.

[23] G. Pang, C. Yan, C. Shen, A. van den Hengel, and X. Bai, "Self-trained deep ordinal regression for End-to-End video anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12173–12182.

[24] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14372–14381.

[25] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[26] H. Z. Ur Rehman and S. Lee, "Automatic image alignment using principal component analysis," *IEEE Access*, vol. 6, pp. 72063–72072, 2018.
[27] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.
[28] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3379–3388.
[29] M. Sabokrou, M. Pourreza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "AVID: Adversarial visual irregularity detection," in *Proc. Asian Conf. Comput. Vis.* Berlin, Germany: Springer, 2018, pp. 488–505.
[30] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vis. Image Understand.*, vol. 172, pp. 88–97, Jul. 2018.
[31] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, Apr. 2017.
[32] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "F-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Med. Image Anal.*, vol. 54, pp. 30–44, May 2019.
[33] 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.* Berlin, Germany: Springer, 2017, pp. 146–157.
[34] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 8, no. 9, p. 1575, Sep. 2018.
[35] D. Yapi, M. Mejri, M. S. Allili, and N. Baaziz, "A learning-based approach for automatic defect detection in textile images," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 2423–2428, 2015.
[36] H. Zenati, C. Sheng Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," 2018, *arXiv:1802.06222*. [Online]. Available: http://arxiv.org/abs/1802.06222
[37] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, "Anomalynet: An anomaly detection network for video surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2537–2550, Oct. 2019.

SORA KIM received the B.S. degree from the Division of Electrical Engineering, Hanyang University, South Korea, in 2018. She is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Hanyang University. Her research interests include medical image analysis and computer vision.

YOUNGJAE JO received the B.S. degree from the Department of Information and Communication Engineering, Daegu University, South Korea, in 2016, and the M.S. degree from the Department of Electrical and Electronic Engineering, Hanyang University, South Korea, in 2018. He is currently a Research Engineer with Hyundai Mobis Company Ltd. His research interests include deblurring and computer vision.

JUNGCHAN CHO (Member, IEEE) received the B.S. degree from the School of Electrical and Electronics Engineering, Chung-Ang University, Seoul, Republic of Korea, in 2010, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Seoul National University, Seoul, in 2016. From 2016 to 2019, he was a Senior Software Engineer with Samsung Electronics Company Ltd. He is currently an Assistant Professor with the Department of Software, Gachon University, Seongnam, Republic of Korea. His research interests include deep learning, computer vision, and machine learning.

JIWOO SONG received the B.S. degree from the Division of Electrical Engineering, Hanyang University, South Korea, in 2019. He is currently pursuing the M.S. degree with the Department of Electrical and Electronic Engineering, Hanyang University. His research interests include object tracking, detection, segmentation, and their applications.

YOUNYOUNG LEE received the B.S. and M.S. degrees from the Department of Electronic Engineering, Sogang University, South Korea, in 2015 and 2017, respectively. Since 2017, he has been working as a Research Engineer with the Advanced Technology Center, Daekhon Corporation. His research interests include deep learning and software engineering.

MINSIK LEE (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, South Korea, in 2006 and 2012, respectively. From 2012 to 2013, he was a Postdoctoral Researcher with the School of Electrical Engineering and Computer Science. In 2014, he joined Seoul National University as a BK21 Assistant Professor. He is currently an Associate Professor with Hanyang University, Ansan, South Korea. His research interests include shape and motion analysis, deformable models, computer vision, deep learning, pattern recognition, and their applications.