Spatial Contrastive Learning for Anomaly Detection and Localization

DAEHWAN KIM, DAUN JEONG, HYUNGMIN KIM, KIBONG CHONG, SEUNGRYONG KIM, AND HANSANG CHO

1Samsung Electro-Mechanics, Suwon, Gyeonggido 16674, South Korea
2Department of Computer Science and Engineering, Korea University, Seongbuk-gu, Seoul 02841, South Korea
3School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Hansang Cho (hansang.cho@samsung.com)

ABSTRACT With the development of deep learning, abnormal detection methods have been widely presented to improve performances in various applications, including visual inspection systems. However, there remains difficult to be directly applied to real-world applications, which often include the lack of abnormal samples and diversity. This paper proposes contra embedding that adopts progressive autoencoder with contrastive learning to address these difficulties. The autoencoder is trained progressively to reproduce the details of the original images, and modified CutPaste augmentation helps to learn to recover normal images. Especially, contrastive learning based on normal embedding vectors effectively reduces false positives caused by the autoencoder. The proposed method is also helpful when normal data have complex shapes, sizes, and colors. In experiments, MVTec AD dataset is used to show the generalization ability of the proposed method in various real-world applications. It achieves over 98.0% AUROCs in detection and 97.7% AUROCs in the localization, respectively, without using the ImageNet pre-trained model as in previous methods.

INDEX TERMS Deep learning, anomaly detection, progressive autoencoder, contrastive learning.

I. INTRODUCTION

Recently, huge performance improvement has been achieved in the unsupervised anomaly detection task, which aims to detect unusual events of test data by only training unlabeled data [1], [2]. Unsupervised anomaly detection is assumed that the training dataset does not usually have abnormal samples, and it can be used when normal samples occur more frequently than abnormal ones. This method can be applied to cases that are difficult to collect abnormal data, such as medical and industrial applications [3]–[6].

In anomaly detection and localization, there exist basically two categorized methods such as a reconstruction-based method [7]–[11], [12], and an embedding similarity-based one [13]–[15]. An Autoencoder (AE) structure is one of the reconstruction-based methods. It finds the differences between the input image and the reconstructed one. The AE is an intuitive and interpretable, and effective solution to extract important features of images that have various characteristics [16], [17]. In anomaly detection fields, the reconstructed image quality is one of the key issues to improve performances because generated images usually seem blurred due to the bottleneck structure of the model. To avoid this...
drawback, Gong et al. [18] used AE with the memory module to correct the reconfiguration error of the abnormal sample. On the other hand, Chung [19] solved the problem by restoring the image using knowledge distillation and outlier exposure regularization.

Recently, embedding similarity-based methods in [1], [13]–[15] showed good performance using feature vectors of a pre-trained network to detect the abnormality. However, they require an appropriate pre-trained model trained using extra data such as ImageNet. The used model does not guarantee that it will properly extract the feature vectors of all datasets.

Our approach, based on the reconstruction-based method is inspired by the recent success of self-supervised models such as SimCLR [20] and MoCo [21]. Spatial contrastive learning is applied to learn local features for fine-grained datasets. The method is more applicable for extracting and adjusting features when sufficient normal data exists. This paper proposes a novel method called contra embedding for unsupervised anomaly detection trained on normal data, including various characteristics such as shape, size, and color. The proposed contra embedding combines a Progressive Autoencoder (PAE) and spatial contrastive learning approach to perform outstandingly. The PAE is used to detect subtle abnormalities in detail, and it is robust to small changes in size and shape of normal instances. Additionally, modified CutPaste augmentation reduces the parameters dependencies of the autoencoder. The augmentation helps autoencoder learn denoising and reduce over-detection when autoencoder has large compression vectors. The patch-level contrastive learning is proposed to extract local embedding vectors, which compensates for the anomaly map of the PAE.

Figure 1 shows the anomaly map that is generated from the proposed method. Structural Similarity Index (SSIM) from autoencoder detects abnormalities effectively but detects high-frequency regions of normal images as defects. This problem is supplemented with a generated mask through contrastive learning. The last anomaly map shows the effectiveness of our proposed method.

The contribution of our method can be summarized:

- We address the drawback of over-detection by the reconstruction-based methods through patch-level contrastive learning.
- The proposed approach is more useful in realistic conditions like non-aligned datasets.
- The method improves the reconstruction performance of the autoencoder by using progressive learning and modified CutPaste augmentation.

II. RELATED WORK

A. RECONSTRUCTION-BASED METHODS

Reconstruction-based methods encode or reconstruct normal images using neural network architectures like Autoencoder (AE) [7], [8], Variational Autoencoder (VAE) [9], or Generative Adversarial Networks (GAN) [10]–[12], [3]. The autoencoder is based on the idea that the model could not reconstruct the anomalous image properly since it concentrates only on the frequent occurrence of the normal one. The anomaly score measures the difference between input images and reconstructed ones.

To improve the quality of reconstruction images, SSIM-AE [7], [8] propose to train an autoencoder with the structural similarity loss for comparing luminance, contrast and structural information between local image regions. MemAE [18] and OE-SDN [19] use an additional method like memory module or knowledge distillation to complement reconstruction images. Two methods are mainly focused on a classification task of anomaly detection such as MNIST and CIFAR-10. Recently, InTra [22] suggests a Transformer model with masked patches. The method tried to improve performances using global attentions.

B. EMBEDDING SIMILARITY-BASED METHODS

Embedding similarity-based methods extract meaningful vectors for anomaly detection. Patch-SVDD [23] extends Support Vector Data Description (SVDD) to the patch levels. The anomaly score is measured using Nearest Neighbor Search (NNS) based on the distance between an embedding vector from a test image and all embedding vectors from training images. Patch-SVDD improved the performances using the patch-level features, but the inference time increases linearly with the size of the training data. Recently, the other authors in [1], [13]–[15] suggested using feature vectors of a pre-trained network to detect the abnormality. Especially, PaDiM [1] proposed a patch-based approach that preserves the coordinate from image space to feature maps. The feature vectors are extracted in each patch according to the pre-trained network using ResNet [24], WideResNet [25], or EfficientNet [26]. Although the method is simple, it shows outstanding performance. However, the performance is limited by the pre-trained data and the approach is hard to learn additional features when performance is not enough.

C. SELF-SUPERVISED LEARNING

Self-Supervised Learning (SSL) has been remarkably developed to reduce the labeled data dependencies in the past few years. Learning of a representation is a core issue of SSL in computer vision. A series of methods have been proposed to learn a representation of an image that does not have labels [20], [21], [27], [28], [29], [30]. These methods help neural networks learn meaningful features by training with a pretext task from unlabeled data. Several related tasks have been suggested such as predicting rotations of images, invariance to image transformations, and context prediction. Contrastive learning is one of the most recent and effective works to use mutual information for training [20], [31]–[33].

In recent research experiments, the SSL in computer vision [34]–[37] has shown good performance for anomaly detection. These methods extract features using geometrical changes such as a rotation or a flip of the entire image. However, the methods are helpful for discriminative features from the overall information of an image, such as CIFAR10
D. PROGRESSIVE LEARNING
In general, the image reconstruction in an AE requires non-negligible computational costs for extracting features, and the performance is highly affected by hyperparameters such as the learning rate. SinGAN [38], Progressive GAN [39], and Pioneer [40] have been proposed to address this problem by gradually learning the images from low resolution to high resolution. These methods used progressive learning sequences in generators and discriminators to increase the resolution of the generated images. We apply this concept of progressively increasing the size to the autoencoder.

III. METHOD
In this section, we present the anomaly detection algorithm. As shown in Figure 2, the method uses two neural networks, where the one is the PAE which reconstructs normal images, reducing parameters dependencies of the autoencoder and improving detection performances, and the other network is a convolutional network for feature extraction. The feature extractor uses the self-supervised method that contains two tasks which are context prediction and comparing similarity. In inference, two anomaly maps are multiplied to construct the final map. The anomaly map of the PAE is calculated based on the Structural Similarity Index (SSIM) [41] to get pixel-size features.

A. PROGRESSIVE AUTOENCODER
Figure 3 presents the learning process of PAE used in our method. At the initial step, the autoencoder is trained using small images of $14 \times 14$ size, and the trained network is

or ImageNet, not suitable for fine-grained datasets such as MVTec AD [7].

CutPaste [2] learns self-supervised representations by classifying augmented images as abnormal ones. GradCAM is used to make an anomaly map. The method shows the similarities between features of abnormal data and augmented normal data in classification, but the outputs of localization are limited with CAMs, which are mainly highlight the most discriminative points of the images.
modified to add the convolution and transpose layers to the encoder and the decoder. Through transfer learning, the modified network is trained repeatedly using larger size images than the before step. For this training, the $L_2$ loss is adopted to compare input images and outputs at a pixel level, and the equation is as follows.

$$L = \| f_{AE}(X) - X \|_2. \tag{1}$$

Here, $X$ denotes a training image, and $f_{AE}()$ denotes the autoencoder.

The quality of restoring images is the key issue to the autoencoder to detect abnormalities. The restoring quality, such as shape or noise, depends on the number of compressed vectors. If the number of vectors is too large, the autoencoder reconstructs too detailed images, including unwanted factors, such as defects. To address the issue, we adopt an autoencoder with augmentation, modified CutPaste (M-CutPaste), which leads to utilizing sufficiently large vectors and preventing training of defective objects, as shown in Figure 4. A detailed procedure of M-CutPaste is described in Algorithm 1. First, random augmentations such as contrast, brightness, hue, 90° rotation, and resize are applied as $I_{aug}$ for a patch image. Next, $I_{aug}$ is generated by an inversed cutout mask which is a random size white square. The cutout mask $M$ is applied to the input image to generate $I_{cutout}$. Finally, a modified cutpaste image $I_{mcup}$ is made by combining $I_{cutout}$ and $I_{aug}$.

The autoencoder is trained to generate $I_{aug}$ with the generated image $I_{mcup}$ as input. This method assumes the M-CutPaste area as noise and helps learn denoising.

### B. CONTRA EMBEDDING LEARNING

Contrastive learning is generally used as a positive pair by applying different augmentation to one image, and the other images are used as a negative pair. We utilize the process for a patch level, not an image level. In addition, instead of using images with different applications of augmentation as positive, we use the exact same location as the original image restored by autoencoder as positive and different locations as negative.

Figure 5 shows the training process of the proposed contrastive learning. The batch size is $N$, and we define the contrastive prediction task on same position pairs of input patches $P_i(i = 1, \ldots, N)$ and reconstructed patches $Q_j (j = 1, \ldots, N)$ where $P_i$ and $Q_j$ have the same position when $i = j$. Let the feature extractor network as $g(f_{con}())$ where $g()$ is the projection head which constructs with two dense layers and $f_{con}()$ is encoder network which constructs with

---

**Algorithm 1** Autoencoder Learning Algorithm With Modified CutPaste

**Data:** sampled minibatch $\{x_k\}_{k=1}^{N}$

**Result:** autoencoder $f_{AE}$

1: for all $k \in \{1, \ldots, N\}$ do
2: $I_{in} = augment(x_k)$
3: $I_{mcup} = modCutPaste(I_{in})$
4: $L = \| f_{AE}(I_{mcup}) - I_{in} \|_2$
5: update networks $f_{AE}$ to minimize $L$
6: end for

7: def modCutPaste($I_{in}$):
9: $I_{aug} = augment(I_{in})$
10: generate a random size cutout mask $M$
11: $I_{aug} = I_{aug} \otimes inv(M)$
12: $I_{cutout} = I_{in} \otimes M$
13: $I_{mcup} = I_{cutout} + I_{aug}$
14: return $I_{cut}$
loss is calculated as follows:

\[ L_{cl}(z_i, z'_i) = -\log \frac{\exp(\text{sim}(z_i, z'_i)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(z_i, z'_k)/\tau)} \]  

(2)

where \( \tau \) denotes a temperature parameter, \( \text{sim}(\cdot) \) represents the cosine similarity function between two vectors. Equation 2 is calculated for all batch size \( N \), and represents the cosine similarity between two vectors.

The second anomaly map, \( M_{con} \) is calculated using feature network \( f_{con}(\cdot) \) which is trained with contrastive learning. First, an input set consists of \( m \) cropped images with a patch size \( p_n \times p_w \) and half stride from the original image \( I_0 \times I_w \) like Figure 2. The patch set of the original image \( p_i(i = 1, \ldots, m) \), and the patch set of the image restored through the autoencoder \( q_j(j = 1, \ldots, m) \). Then, two embedding vector sets are generated as \( f_{con}(p_i) \) and \( f_{con}(q_j) \). Here, \( f_{con}(q_j) \) becomes a query set and a given query vector, the minimum value is defined as one anomaly score by measuring the distance of \( L_2 \) from all vector sets \( f_{con}(p_i) \). The equation of anomaly score for a patch is as follows.

\[ M_{con}(j) = \min_{i \in [1, \ldots, m]} \| f_{con}(p_i) - f_{con}(q_j) \|_2, \]  

(7)

\( M_{con}(j) = \min_{i \in [1, \ldots, m]} \| f_{con}(p_i) - f_{con}(q_j) \|_2 \) makes small anomaly map and resize to same size as the original image. The final anomaly map \( M \) is constructed by pixel-wise multiplication of two maps.

\[ M = M_{ssim} \odot M_{con}. \]  

(8)

The anomaly score is defined as the maximum value of the final anomaly map.

### IV. EXPERIMENTS

#### A. IMPLEMENTATION DETAILS

The training process of the proposed method is performed in two steps. First, the input images are trained using PAE, and reconstructed images are generated. Second, contrastive learning is performed at the patch level using the reconstructed and input images. During the second step, trained PAE weights are fixed and feature extractor \( f_{con} \), classifier \( C \), and projector \( g \) are trained simultaneously using equation 5.

The PAE was trained with the following augmentation operations: rotation(−10, +10), brightness(0.1), contrast(0.8, 1.2), hue(0.1), saturation(0.9, 1.1) and M-CutPaste. M-CutPaste used same hyper-parameters for brightness, contrast, hue and saturation. Additionally, 90° rotation, zoom in(1.1), and mask size of cutout in Table 2 are used. Adam optimizer and learning rate of four steps \([0.0005, 0.00025, 0.000125, 0.00006] \ were used with a batch size of 128. MvTec AD consists of two datasets as normal train data and test data. We randomly choose 80% of images from the train data set which is used in training. The remaining 20% of the image is used to check the training loss. The image size of training data is selected differently depending on the types of data, such as texture and object. For the zipper image set, the image size is set the same as the texture instead of the size of the object. Detailed information about the autoencoder structure, the size of training images, and the epochs of PAE are explained in Table 1 and Table 2.

The feature extractor was trained with random crop (12 − 128) augmentation. Adam optimizer and initial learning rate of 0.0002 with cosine decay policy were adopted.
TABLE 1. Architecture for the PAE. All convolution layers have a kernel size of 3 or 4. A Conv block has two convolution layers with the same filter size. Each layer has a kernel size of 3 and 4 and a stride of 1 and 2 respectively. A Transpose block has a transpose layer and a convolution layer with a kernel size of 4 and 3, a stride of 2 and 1, and the same filter size.

| Layer       | Filter | Kernel | Stride | Activation |
|-------------|--------|--------|--------|------------|
| Conv layer  | 32     | 3      | 1      | Relu       |
| Conv block1 | 64     | 3, 4   | 1, 2   | Relu       |
| Conv block2 | 96     | 3, 4   | 1, 2   | Relu       |
| Conv block3 | 128    | 3, 4   | 1, 2   | Relu       |
| Conv block4 | 160    | 3, 4   | 1, 2   | Relu       |
| Conv block5 | 192    | 3, 4   | 1, 2   | Relu       |
| Conv layer  | 512    | 7      | 1      | -          |
| Transpose block1 | 192 | 4, 3   | 2, 1   | Relu       |
| Transpose block2 | 160 | 4, 3   | 2, 1   | Relu       |
| Transpose block3 | 128 | 4, 3   | 2, 1   | Relu       |
| Transpose block4 | 96 | 4, 3   | 2, 1   | Relu       |
| Transpose block5 | 64 | 4, 3   | 2, 1   | Relu       |
| Conv layer  | 3      | 3      | 1      | Sigmoid    |

TABLE 2. The training details of the PAE. The size of images is progressively increased during training. M-CutP means the maximum size of M-CutPaste mask.

| Resize       | Crop Size   | Epoch | M-CutP |
|--------------|-------------|-------|--------|
| Texture + Zipper | 64 × 64  | 14 × 14 | 2000 | 3      |
|              | 128 × 128  | 28 × 28 | 2000 | 27     |
|              | 256 × 256  | 56 × 56 | 2000 | 55     |
|              | 512 × 512  | 112 × 112 | 2000 | 111    |

| Object       | Crop Size   | Epoch | M-CutP |
|--------------|-------------|-------|--------|
|              | 16 × 16     | 14 × 14 | 3000 | 7      |
|              | 32 × 32     | 28 × 28 | 1000 | 14     |
|              | 64 × 64     | 56 × 56 | 1000 | 28     |
|              | 128 × 128   | 112 × 112 | 1000 | 56     |
|              | 256 × 256   | 224 × 224 | 1000 | 112    |

A loss hyper-parameter $\lambda$ is set as 1 for all experiments. The network was trained for $2k$ epochs with a batch size of 128. The feature extractor network structure is described in Table 3.

The experimental results are calculated as the average of Area Under the Receiver Operating Characteristic curve (AUROC) with three different random seeds. In the inference, the SSIM with a window size of $15 \times 15$ is used to calculate the anomaly scores for $M_{SSIM}$. The patch size $p_h \times p_w$ in $M_{con}$ is set as $32 \times 32$ for all evaluations.

B. DATASET DESCRIPTION

Our proposed method is evaluated on the MVTec AD, commonly used to test anomaly detection and localization in manufacturing fields. The dataset consists of five textures and ten objects categories, and each category has different normal data quantities for training from 60 to 391. Most images are well center-aligned except the screw category. The training set has only normal images, while the test data contains normal and abnormal samples.

TABLE 3. Architecture for the feature extractor: The Residual block means ResNet v1 block used in ResNet18 [24]. The two dense layers are called projector and only used for training.

| Layer       | Filter | Kernel | Stride | Activation |
|-------------|--------|--------|--------|------------|
| Conv - Batchnorm | 192   | 3      | 1      | Leaky Relu |
| Residual block 1 | 192   | 3      | 2      | Leaky Relu |
| Residual block 2 | 192   | 3      | 1      | Leaky Relu |
| Residual block 3 | 348   | 3      | 2      | Leaky Relu |
| Residual block 4 | 348   | 3      | 1      | Leaky Relu |
| Residual block 5 | 768   | 3      | 2      | Leaky Relu |
| Residual block 6 | 768   | 3      | 1      | Leaky Relu |
| GAP          | -      | -      | -      | -          |
| Dense layer  | 256    | 1      | -      | Relu       |
| Dense layer  | 256    | 1      | -      | -          |

C. EXPERIMENTAL RESULTS

1) COMPARISON WITH THE STATE-OF-THE-ART

The detection and localization results of MVTec AD are shown in Table 4 and Figure 6. Patch-SVDD [23], CutPaste [2], and InTra [22] have compared performances with our solution when each method is trained only using a given 60 to 391 train data from MvTec AD. Additionally, PaDiM [1] is compared, although it needs additional data. PaDiM is a representative method in embedding similarity-based techniques that have different characteristics. The proposed method presents improved results for the average of all categories in detection and localization without using extra data. In particular, the cable category is difficult to detect abnormalities for the reconstruction-based method, because the image contains many high-frequency components such as stripped wires. Intra, the reconstruction-based method, shows particularly low performance in cable, but our method shows the best performance by overcoming its drawbacks. Although our method shows better performance on average AUROC, it is less effective on specific categories. The localization results of wood and toothbrush show lower performance than the state-of-the-art methods. For these categories, images have a lot of dots as noises. The patch-level contrastive learning detects these noises as abnormal points. It could not compensate the autoencoder adequately. Figure 6 presents success and failure cases. In Figure 6, success cases show accurate localization even when defects are small and have tiny color changes. However, the method is hard to localize defects if abnormalities have too tiny texture variances.

2) DETECTION AND LOCALIZATION ON A NON-ALIGNED DATASET

To estimate the robustness of the anomaly localization method, non-aligned MVTec AD is used. The dataset is not officially released, but PaDiM proposed it. MVTec AD data is modified with (-10, 10) rotation and randomly cropping with 0.875 proportion (e.g. crop from 256 to 224.). In this case, detection results are not counted because lots of defects are removed during the cropping process.

Table 5 shows that the proposed method is robust to the translation and rotations of the image compared with...
TABLE 4. Comparison of detection and localization results with state-of-the-art methods on the MVTec AD (AUROC %) We report mean and standard error tested with 3 random seeds. The highest AUROC value is represented in bold and the lowest value is underlined. PaDiM only reports mean AUROC for detection.

| Category        | Without extra training data | With extra training data |
|-----------------|-----------------------------|--------------------------|
|                 | Patch-SVDD [23] | CutPaste [2] | InTra [22] | Ours | PaDiM [1] |
| Carpet          | 92.9 ± 0.1 | 92.6 ± 0.1 | 93.1 ± 0.1 | 98.3 ± 0.1 | 96.9 ± 0.1 | 99.3 ± 0.1 | 99.9 ± 0.1 |
| Grid            | 94.6 ± 0.1 | 96.2 ± 0.1 | 99.1 ± 0.1 | 97.5 ± 0.1 | 100.0 ± 0.0 | 98.8 ± 0.1 | 100.0 ± 0.0 | 99.3 ± 0.1 | 97.3 ± 0.1 |
| Leather         | 90.9 ± 0.1 | 97.4 ± 0.1 | 100.0 ± 0.1 | 99.5 ± 0.1 | 100.0 ± 0.0 | 99.5 ± 0.1 | 97.3 ± 0.1 | 98.8 ± 0.1 | 99.2 ± 0.1 |
| Tile            | 97.8 ± 0.1 | 91.4 ± 0.1 | 93.4 ± 0.1 | 90.5 ± 0.1 | 98.2 ± 0.1 | 94.4 ± 0.1 | 98.5 ± 0.1 | 98.3 ± 0.1 | 94.1 ± 0.1 |
| Wood            | 96.5 ± 0.1 | 90.8 ± 0.1 | 98.6 ± 0.1 | 95.5 ± 0.1 | 97.5 ± 0.1 | 88.7 ± 0.1 | 98.7 ± 0.1 | 93.1 ± 0.2 | 94.9 ± 0.1 |
| Avg. textures   | 94.5 ± 0.5 | 93.7 ± 0.5 | 97.0 ± 0.5 | 96.3 ± 0.1 | 98.9 ± 0.1 | 98.7 ± 0.1 | 97.6 ± 0.2 | 99.0 ± 0.5 | 96.9 ± 0.5 |
| Bottle          | 98.6 ± 0.1 | 98.1 ± 0.1 | 98.3 ± 0.1 | 97.6 ± 0.1 | 100.0 ± 0.0 | 98.1 ± 0.1 | 97.6 ± 0.1 | 99.9 ± 0.3 | 97.4 ± 0.3 |
| Cable           | 90.3 ± 0.1 | 96.8 ± 0.1 | 86.0 ± 0.5 | 90.0 ± 0.2 | 70.3 ± 0.1 | 91.0 ± 0.1 | 94.8 ± 0.3 | 97.5 ± 0.1 | 96.7 ± 0.1 |
| Capsule         | 76.7 ± 0.1 | 95.8 ± 0.1 | 96.2 ± 0.5 | 97.4 ± 0.1 | 86.5 ± 0.1 | 97.7 ± 0.1 | 95.6 ± 0.4 | 96.1 ± 0.1 | 98.5 ± 0.4 |
| Hazelnut        | 92.0 ± 0.1 | 97.5 ± 0.1 | 97.3 ± 0.3 | 97.3 ± 0.1 | 95.7 ± 0.1 | 98.3 ± 0.1 | 98.1 ± 0.2 | 98.5 ± 0.0 | 98.2 ± 0.0 |
| Metal Nut       | 94.0 ± 0.1 | 98.0 ± 0.1 | 99.3 ± 0.2 | 93.1 ± 0.4 | 96.9 ± 0.1 | 93.3 ± 0.1 | 96.5 ± 0.2 | 96.2 ± 0.2 | 97.2 ± 0.2 |
| Pill            | 86.1 ± 0.1 | 95.1 ± 0.1 | 92.4 ± 0.3 | 95.7 ± 0.1 | 90.2 ± 0.1 | 98.3 ± 0.1 | 97.5 ± 0.4 | 96.7 ± 0.0 | 95.7 ± 0.0 |
| Screw           | 81.3 ± 0.1 | 95.7 ± 0.1 | 86.3 ± 1.0 | 96.7 ± 0.1 | 95.7 ± 0.1 | 99.5 ± 0.1 | 95.6 ± 0.2 | 99.7 ± 0.1 | 98.5 ± 0.1 |
| Toothbrush      | 100.0 ± 0.0 | 98.1 ± 0.0 | 98.3 ± 0.9 | 98.1 ± 0.0 | 100.0 ± 0.0 | 98.9 ± 0.0 | 100.0 ± 0.0 | 97.7 ± 0.0 | 98.8 ± 0.0 |
| Transistor      | 91.5 ± 0.0 | 97.0 ± 0.0 | 95.5 ± 0.5 | 93.0 ± 0.2 | 95.8 ± 0.1 | 96.1 ± 0.1 | 99.0 ± 0.3 | 98.1 ± 0.1 | 97.5 ± 0.1 |
| Zipper          | 97.9 ± 0.0 | 95.1 ± 0.0 | 99.4 ± 0.2 | 99.3 ± 0.0 | 99.4 ± 0.1 | 99.2 ± 0.1 | 99.1 ± 0.3 | 98.6 ± 0.1 | 98.5 ± 0.1 |
| Avg. objects    | 90.8 ± 0.6 | 96.7 ± 0.6 | 94.3 ± 0.6 | 95.8 ± 0.0 | 93.0 ± 0.1 | 96.9 ± 0.1 | 98.4 ± 0.0 | 97.7 ± 0.0 | 97.2 ± 0.0 |
| Avg. categories | 92.1 ± 0.6 | 95.7 ± 0.6 | 95.2 ± 0.6 | 96.0 ± 0.1 | 95.0 ± 0.1 | 96.6 ± 0.1 | 98.0 ± 0.0 | 97.7 ± 0.0 | 97.9 ± 0.0 |

FIGURE 6. Success(a) and failure(b) cases of the proposed method (contra embedding) in localization. Each column represents each category’s results: carpet, grid, leather, tile, wood, bottle, cable, capsule, hazelnut, metal nut, pill, screw, toothbrush, transistor and zipper. From up to down, each row displays the original image, ground truth mask and anomaly map.

D. ABLATION STUDY

We conduct several experiments to analyze our method, contra embedding. The first experiment is to figure out the importance of each component. The structure of autoencoder, denoising augmentation and contrastive learning parts are analyzed. Secondly, we study the impact of loss and training convergence in contrastive learning. The third experiment shows the validity of the progressive autoencoder by comparing the performance of PAE and SSIM-AE. Additionally, we analyze the effects of modified CutPaste augmentation in PAE.

the PaDiM method. Especially for the texture categories, the results of non-aligned images are as good as aligned images, which shows our method is highly robust to the alignment.
TABLE 5. The localization results on the non-aligned MVTec AD(AUROC %): In MVTec AD, random rotation (−10, +10) and random crop (from 256 × 256 to 224 × 224) applied to both the train and test sets. All results are averages of each category.

| Category | PaDiM [1] | Contra embed. |
|----------|----------|--------------|
|          | Aligned  | Non-Aligned  | Aligned  | Non-Aligned  |
| Textures | 96.9     | 92.4         | 97.6     | 97.3         |
| Objects  | 97.8     | 92.1         | 97.7     | 96.5         |
| Categories | 97.5     | 92.2         | 97.7     | 96.8         |

1) COMPONENT ANALYSIS
We present the impact of different components which we add or remove in our framework. In the structure of autoencoder (AE), SSIM-AE [7] trained using equation 6 and PAE are considered. Denoising augmentation, M-CutPaste and contrastive learning are the other components. In Table 6 last row, removing the contra embedding part shows that the contrastive learning is the essential part of all components. In the case of texture, M-CutPaste is also important to detect and localize the defects. The method using SSIM-AE shows similar performance to PAE with a difference of 0.3-1.3 in detection.

2) EFFECTIVENESS OF CONTRASTIVE LEARNING

a: ANALYSIS OF LOSS FUNCTION
In self-supervised learning, we trained the feature extractor using two losses: contrastive loss and context prediction loss in equation 5. In Table 7, performance by the different loss functions is described. The left column shows the PAE result without contrastive learning, the middle one is the contrastive learning without prediction loss and the right one is the total loss results. In most cases, the contra embedding with the context prediction loss shows good performances. The context prediction loss helps the feature extractor to learn suitable features for anomaly detection. Both results of the grid and screw cases show that the use of the autoencoder, even without other compensation, could represent excellent performances. This outstanding result can be obtained because the normal images are simple and have uncomplicated patterns without large variances. The context prediction could, unfortunately, be ineffective in the average result of texture categories. However, the context prediction loss could be helpful where the input has distinguishable objects like cable and capsule.

b: ANALYSIS OF TRAINING CONVERGENCE
To analyze the convergence of self-supervised learning, we compared the average AUROC depending on the training epochs. The detailed results are different by each category, but most of the training loss is converged after the 2k epoch. Figure 7 shows the performance is saturated after the 2k epoch.

3) RECONSTRUCTION PERFORMANCES OF AUTOENCODER
In this ablation, we experiment with the autoencoder with different elements to understand its property. The structure of the autoencoder and different augmentation were tested to compare reconstruction performances.

a: EFFECTIVENESS OF PROGRESSIVE AUTOENCODER
We analyze the effect of the autoencoder structure by comparing its detection and localization performances. In Table 8, PAE shows 5.2% and 3.6% higher performances than SSIM-AE in texture detection and object localization. The reconstruction results through SSIM-AE and PAE in leather and metal nut are presented in Figure 9. The SSIM-AE cannot restore the details of the original image, while the PAE accurately restores the details of the original image. For a more specific analysis, we measure the PSNR between normal and reconstruction images. Figure 8 shows the PSNR measured by SSIM-AE and PAE by category. In all categories except hazelnut and screw, PAE shows higher PSNR than SSIM-AE. As a result of Figures 8, 9 and Table 8, we can clarify that anomaly detection and localization performance are improved because PAE reconstructs normal images more accurately.
TABLE 7. Detection and localization results without SSL and with SSL (contra embed., AUROC%): CP denotes the context prediction loss. The highest result value is shown in bold.

| Category    | Detection      | Localization    |
|-------------|----------------|-----------------|
|             | PAE | Contra embed. | PAE | Contra embed. |
|             |    | -  | + CP |    | -  | + CP |
| Carpet      | 97.1 | 98.5 | 98.9 | 99.0 | 99.1 | 99.3 |
| Grid        | 100.0 | 100.0 | 100.0 | 99.3 | 98.9 | 99.3 |
| Leather     | 91.3 | 99.7 | 97.3 | 98.5 | 98.4 | 98.8 |
| Tile        | 99.8 | 98.9 | 98.5 | 96.5 | 98.4 | 98.3 |
| Wood        | 96.3 | 98.5 | 98.7 | 88.2 | 93.8 | 93.1 |
| Textures    | 96.9 | 99.1 | 98.7 | 96.3 | 97.7 | 97.6 |
| Bottle      | 99.8 | 100.0 | 99.9 | 96.3 | 98.5 | 97.4 |
| Cable       | 66.1 | 93.2 | 94.8 | 89.4 | 96.9 | 97.5 |
| Capsule     | 95.6 | 92.2 | 95.6 | 94.7 | 94.4 | 96.1 |
| Hazelnut    | 95.2 | 97.5 | 98.1 | 98.5 | 98.2 | 98.5 |
| Metal Nut   | 80.4 | 95.6 | 96.5 | 89.6 | 96.3 | 96.2 |
| Pill        | 91.4 | 92.6 | 97.5 | 94.0 | 94.7 | 96.7 |
| Screw       | 97.0 | 96.0 | 95.6 | 99.7 | 99.4 | 99.7 |
| Toothbrush  | 99.7 | 100.0 | 100.0 | 96.9 | 96.1 | 97.7 |
| Transistor  | 96.9 | 97.4 | 99.0 | 97.8 | 97.0 | 98.1 |
| Zipper      | 97.9 | 97.4 | 99.1 | 97.8 | 97.0 | 98.6 |
| Objects     | 92.0 | 96.3 | 98.4 | 95.3 | 96.9 | 97.7 |
| Categories  | 93.6 | 97.2 | 98.0 | 95.6 | 97.2 | 97.7 |

TABLE 8. Detection and localization results of SSIM-AE and PAE without M-CutPaste (AUROC%).

| Category    | SSIM-AE | PAE |
|-------------|---------|-----|
|             | Detection | Localization | Detection | Localization |
| Textures    | 89.7     | 93.1 | 94.9     | 95.5 |
| Objects     | 89.8     | 90.9 | 90.9     | 94.5 |
| Categories  | 89.8     | 92.3 | 93.1     | 94.8 |

FIGURE 8. PSNR between the reconstructed images and the original normal images. Results are shown the average PSNR of each category.

b: EFFECTIVENESS OF MODIFIED CUTPASTE

Table 9 compares the performance according to the use of M-CutPaste. The M-CutPaste case shows 1.3% and 0.8% higher AUROC than the basic ones in detection and localization, respectively. CutPaste and M-CutPaste show similar performances in object categories but M-CutPaste is better in texture ones. Figure 10 shows that the abnormal region of the image (b) remains as it is, whereas as a result of using M-CutPaste, the abnormal region of the image (c) is completely removed. The figures present that M-CutPaste makes the autoencoder learn to recover diverse abnormalities to normal. As a result, if an abnormal image has a huge difference in structure or color, the autoencoder could reconstruct the better normal image than that trained without augmentation.
V. CONCLUSION
This paper has proposed the unsupervised anomaly detection and localization algorithm using PAE combined with self-supervised learning. The autoencoder could be a proper solution when there is a large volume of normal data that has great diversity. In the proposed method, the autoencoder is trained progressively to reproduce the details of the original images and M-CutPaste augmentation helps to learn to recover normal images. Moreover, the contrastive learning is suggested to compensate the autoencoder to overcome the diversity of normal data. Cosine similarity is used to compare to find differences between the recovered images of the autoencoder and the original image in the patch level. The method is helpful for extracting patch level features.

Using this method, the anomaly detection AUROC resulted in 98.0% in the generally verified MV Tec AD dataset and showed high efficiency without a need to use extra training data. Moreover, the methods proved to be effective when input data was diverse in the forms of translation and rotation.

REFERENCES
[1] T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: A patch distribution modeling framework for anomaly detection and localization,” in Proc. ICPP. Cham, Switzerland: Springer, 2021, pp. 475–489.
[2] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-supervised learning for anomaly detection and localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 9664–9674.
[3] 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.
[4] K. Zhou, S. Gao, J. Cheng, Z. Gu, H. Fu, Z. Tu, J. Yang, Y. Zhao, and J. Liu, “SparseGAN: Sparsity-constrained generative adversarial network for anomaly detection in retinal OCT images,” in Proc. IEEE 17th Int. Symp. Biomed. Imag. (ISBI), Apr. 2020, pp. 1227–1231.
[5] P. Napoleano, F. Piccoli, and R. Schettini, “Anomaly detection in nano-fibrous materials by ANN-based self-similarity,” Sensors, vol. 18, no. 2, p. 209, Jan. 2018.
[6] J. Heger, G. Desai, and M. Z. El Abidine, “Anomaly detection in formed sheet metals using convolutional autoencoders,” Proc. CIRP, vol. 93, pp. 1281–1285, 2020.
[7] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVtec AD—A comprehensive real-world dataset for unsupervised anomaly detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9592–9600.
[8] 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.
[9] Y. Lu and P. Xu, “Anomaly detection for skin disease images using variational autoencoder,” 2018, arXiv:1807.01349.
[10] F. Ye, C. Huang, J. Cao, M. Li, Y. Zhang, and C. Lu, “Attribute restoration framework for anomaly detection,” IEEE Trans. Multimedia, vol. 24, pp. 116–127, 2022.
[11] M. Sabokrou, M. Khaleooei, M. Fathy, and E. Adeli, “Adversarially learned one-class classifier for novelty detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3379–3388.
[12] 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), Jun. 2019, pp. 2898–2906.
[13] G. Wang, S. Han, E. Ding, and D. Huang, “Student-teacher feature pyramid matching for anomaly detection,” 2021, arXiv:2103.04257.
[14] O. Rippel, P. Mertens, and D. Mathof, “Modeling the distribution of normal data in pre-trained deep features for anomaly detection,” in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 6726–6733.
DAEHWAN KIM received the B.S. and M.S. degrees from Kookmin University, Seoul, South Korea, in 2011 and 2013, respectively. He is currently a Researcher with Samsung Electro-Mechanics, Suwon, South Korea. His research interest includes deep learning algorithms for visual inspection systems.

DAUN JEONG received the B.S., M.S., and Ph.D. degrees from the Pohang University of Science and Technology, Pohang, South Korea, in 2007, 2009, and 2013, respectively. She is currently a Researcher with Samsung Electro-Mechanics, Suwon, South Korea. Her research interest includes data efficient deep learning for visual inspections.

HYUNGMIN KIM received the B.S. and M.S. degrees in electronic and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2010 and 2011, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. He joined Samsung Electro-Mechanics, South Korea, in 2011. His research interest includes computer vision and deep learning.

KIBONG CHONG received the B.S. and M.S. degrees in electrical and electronic engineering from Kunkuk University, Seoul, South Korea, in 1999 and 2001, respectively. He is currently a Researcher with Samsung Electro-Mechanics, Suwon, South Korea. His research interests include deep learning algorithms and continual learning for visual inspection systems.

SEUNGRYONG KIM received the B.S. and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2012 and 2018, respectively. From 2018 to 2019, he was a Postdoctoral Researcher with Yonsei University. From 2019 to 2020, he was a Postdoctoral Researcher with the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. Since 2020, he has been an Assistant Professor with the Department of Computer Science and Engineering, Korea University, Seoul. His current research interests include 2D/3D computer vision, computational photography, and machine learning.

HANSANG CHO received the B.S. degree in electronics engineering from Kwangwoon University, Seoul, South Korea, in 1994, and the M.S. and Ph.D. degrees from the University of Washington, Seattle, WA, USA, in 2005. From 2005 to 2010, he was a Senior Algorithm Engineer at DRVision Technologies LLC, Bellevue, WA, USA. He is currently the Vice President of Technology at Samsung Electro-Mechanics, Suwon, South Korea. He is leading Software Development Team. His research interests include deep learning algorithms for vision inspection systems and software development for smart factory.

* * *