Anomaly Detection Based on Semi-Formula Driven Pre-training Dataset to Represent Subtle Difference and Anomaly Score

Hiroki Kobayashi
kobayashi@isl.sist.chukyo-u.ac.jp
Naoki Murakami
murakami@asmi.sist.chukyo-u.ac.jp
Naoto Hiramatsu
hiramatsu@asmi.sist.chukyo-u.ac.jp
Takahiro Suzuki
suzuki@isl.sist.chukyo-u.ac.jp
Manabu Hashimoto
mana@isl.sist.chukyo-u.ac.jp

Graduate School of Engineering
Chukyo University
Aichi, Japan

Abstract

The goal of a surface anomaly detection task is to classify an inspection image and pixel as normal/anomaly with high precision. A typical conventional method called PaDiM pre-trains convolutional neural networks with the ImageNet dataset for 1000-class classification and detects the anomaly images from deviance on the feature space. However, since a single class in ImageNet has a wide range of meanings, it is difficult to represent subtle difference between normal and anomaly images as different features. Moreover, PaDiM assumes that two images with similar anomaly scores have features with similar values. However, the feature space is made to classify the ImageNet class, it is not designed to assign two images with similar anomaly scores to relatively similar features. Therefore, we propose an anomaly detection method based on pre-training using novely semi-formula driven image dataset to represent “subtle difference” between two images as different features and two images with similar “anomaly score” as similar features. An image dataset for pre-training is generated by adding pseudo-defects with random Gaussian Mixture Model (GMM) parameters to an existing image dataset. GMM parameters have a different value for each parameter, but the appearances of the generated images have only subtle difference. Next, the regression network is pre-trained to estimate GMM parameters that represent the anomaly score of generated anomaly images. In the experiments with MVTecAD, the proposed method achieved high precision anomaly detection for categories where ImageNet performed poorly.

1 Introduction

In recent years, there has been an increasing need for reliable automatic visual inspection at manufacturing sites to reduce the burden on workers and prevent human errors. The goal

of this task is to classify an image and a pixel to be inspected into either the "normal" or "anomaly" class with high accuracy. Additionally, visual inspection using machine learning has been attracting attention. This method generally requires a large number of normal and anomaly images for training. However, the frequency of anomaly occurrence is extremely low at manufacturing sites. Therefore, sufficient anomaly images for training often cannot be obtained. Therefore, in situations where anomaly images are difficult to obtain, normal and anomaly images need to be classified with high accuracy.

To solve this problem, various methods have been proposed. Specifically, many methods use only normal images for training. Statistical methods [1, 2] approximate the distribution of normal images with a simple function. Feature extraction-based methods train a feature extractor so that normal images are centered [3, 4], or estimate the distribution of normal images with pre-trained Convolutional Neural Networks (CNN) feature representations [5, 6, 7, 8]. Density-based methods estimate the density of normal images using Gaussian Mixture Model (GMM) [9] or Normalizing Flow [10, 11, 12]. Image generation-based methods detect anomalies based on image difference by generating normal image from anomaly image with an AutoEncoder (AE) [13, 14, 15, 16], Generative Adversarial Network (GAN) [17, 18, 19, 20], or Diffusion Model [21].

Among these conventional methods, a method that uses pre-trained CNN feature representations and approximates the distribution of normal images in the feature space has been shown to have particularly high performance. In particular, Patch Distribution Modeling (PaDiM) has been attracting attention as a typical method for anomaly detection. This method is based on the simple idea of approximating the normal features obtained by inputting them into a pre-trained network to a multivariate normal distribution. Moreover, PaDiM achieved high anomaly detection performance in the MVTecAD [22] dataset. Since the deviation of the distribution is evaluated for each patch of the image, another advantage is to determine normal or anomaly at not only the image level but also the pixel level.

First, PaDiM pre-trains a CNN such as ResNet [23] or WideResNet [24] for 1000-class classification using an image dataset called ImageNet [25] that has various image variations. Next, without performing additional training for the network, target normal images are input to the network, and the normal features are approximated by a normal distribution on the feature space of the network. Next, in the same way, the test image is input to the network, and the features are acquired. At this time, if the feature value deviates from the approximated normal distribution, it is determined to be an anomaly.

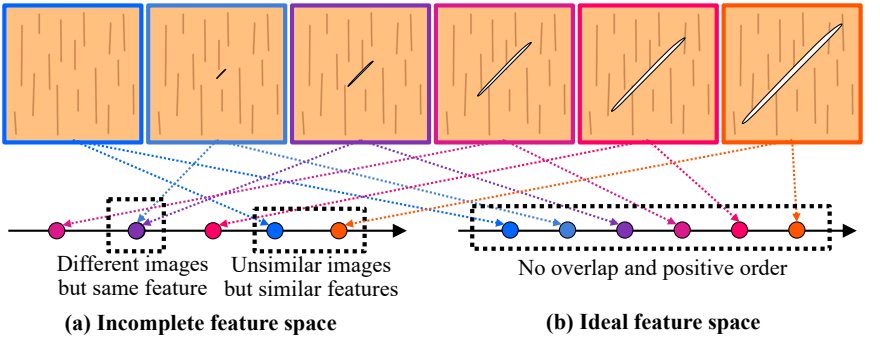

Figure 1: Suitable positional relationship of each image in feature space.

However, these methods have two problems (Fig. 1(a)). The first is that a single class in ImageNet has a wide range of meanings. Images in a single class are assigned to same ground truth, so it has risk of representing two images with subtle difference as same feature. The second is that ImageNet feature representations are not designed to assign two images with similar anomaly scores to relatively similar features. In pre-training with ImageNet, the feature space is made to classify the ImageNet class, so it has the risk of assigning unsimilar two images (i.e., two images with completely different anomaly score) to similar features.

Therefore, we propose an anomaly detection method using pre-training with semi-formula driven image dataset that represents “subtle difference” between two images as different features and two images with similar “anomaly score” as similar features (Fig. 1(b)). The image dataset for pre-training is generated by adding pseudo-defects with random Gaussian Mixture Model (GMM) parameters to the existing image dataset. Also, although the GMM parameters have different for each generated image, there are only subtle difference in the appearance of the generated images. Next, we pre-train a regressive CNN to estimate GMM parameters that represent the anomaly score of generated anomaly images. This realizes to represent “subtle difference” and “anomaly score” for high precision anomaly detection.

2 Basic Idea

PaDiM has achieved high anomaly detection performance (AUROC) on MVTecAD, a representative dataset for anomaly detection. However, when examining each object of MVTecAD, not all objects have high performance, and there are objects with low performance such as Metal Nut, Pill, Tile, and Wood. In this study, we considered the reasons the anomaly detection performance of these categories is low and devised an idea on the basis of them.

2.1 Representing Subtle Difference

PaDiM uses ImageNet to pre-train the network. Here, in ImageNet, one class has a wide range of meaning for images. For example, ImageNet has a class called “lens cap”, and all images of caps used to protect camera lenses are defined as this class. Therefore, everything from items with scratches or peeled labels to new items are treated as one class. In contrast, the images used in actual inspections have small defects (e.g., scratches) in local area of the texture and object (e.g., metal plate). Therefore, to determine whether it is normal or anomaly, the network needs to pay close attention to the details of the textures and objects in the image and distinguish the subtle difference in their surface conditions. However, in ImageNet, “lens cap” is defined as one class without considering such surface conditions, and the classes are not divided to distinguish the subtle difference. This means that it is difficult to generate the differences in features between normal and anomaly images.

Therefore, in this research, we prepare a real image dataset and add pseudo-defects to it by formula, thereby the difference in appearance between images is subtle. In addition, by training a neural network to predict different values for each generated image, images with subtle difference in image appearance can have different values in the feature space.

2.2 Representing Anomaly Score

Conventional PaDiM assumes that the features of normal images follow a normal distribution and two images with similar anomaly scores have features with similar values. This means

that the ranking of the features in each inspection image must correspond to the “anomaly score” on a specific axis. For example, as shown in Fig. 2, in the Pill category of MVTecAD, it is considered that normal or anomaly labels are assigned depending on the anomaly score of chipping. Specifically, the second image from the left is a normal limit sample, and those with smaller and larger scores of chipping are labeled as normal and anomaly, respectively. This “limit” differs depending on each manufacturing site. Therefore, a feature space in which the score (order) of defects is maintained needs to be created so that an anomaly can be detected even if the limits differ. If this can be achieved, anomaly features will no longer be plotted inside the distribution of normal features estimated by PaDiM. However, the feature space of conventional methods was constructed to classify ImageNet classes, and it is not suitable for representing a regressive feature space with the anomaly score.

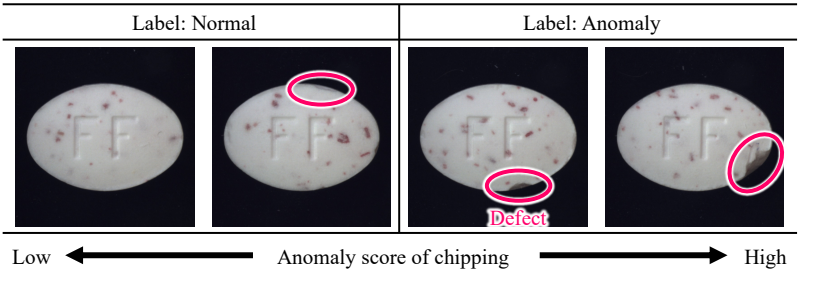

Figure 2: Relationship between label and anomaly score in MVTecAD [22].

In this research, CNN is pre-trained so that subtle difference between normal and anomaly images is assigned to different features, and the order of the feature values of each test image corresponds to the anomaly score of an image on a specific axis in the feature space.

3 Pre-training with Semi-Formula Driven Dataset

In this chapter, we propose an anomaly detection method based on pre-training with semi-formula driven image dataset that represents “subtle difference” between two images as different features and two images with similar “anomaly score” as relatively similar features. First, an image dataset for pre-training is generated by adding the pseudo-defects with random GMM parameters to an existing image dataset (Sec. 3.1). GMM parameters have a different value for each parameter, but the appearances of the generated images have only subtle difference. Since this dataset is created with real normal image and formula defect pattern, it is called as semi-formula driven image dataset. Details of the proposed dataset are provided in the supplementary material. Unlike some formula-driven image datasets [29, 30, 31], the proposed method is specialized and has high performance for anomaly detection. Next, the regressive CNN is pre-trained to estimate GMM parameters that represent the anomaly score of anomaly images (Sec. 3.2). Finally, PaDiM [7] with CNN pre-trained by the proposed method is performed.

3.1 Generating Image Dataset with Gaussian Mixture Model

As shown in Fig. 3, inspired by [26], GMM is used to represent defects. The reason for choosing GMM is as follows. The adjacent pixel in the defect region does not have random

values but a nearly uniform value, and a simple elliptical defect can be approximated by one Gaussian distribution. In addition, complex defects can also be represented by combining many Gaussian distributions. Moreover, this method can control the position by changing the mean, the size by changing the standard deviation, the shape by changing the correlation coefficient, and the transparency by changing the density. Therefore, GMM is adopted. This section describes how to create an anomaly image $\boldsymbol{I}_{\text{ano}}$ from a normal image $\boldsymbol{I}_{\text{norm}}$ and GMM Parameter $\boldsymbol{\theta}_{\text{GMM}}$ with function $f_{\text{ano}}(\boldsymbol{I}_{\text{norm}}, \boldsymbol{\theta}_{\text{GMM}})$.

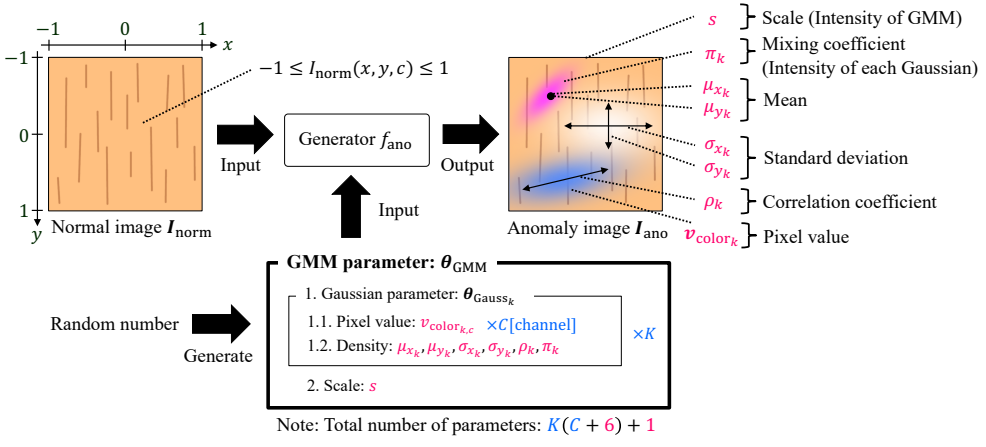

Figure 3: Anomaly image generation with random GMM parameter.

First, GMM parameter $\boldsymbol{\theta}_{\text{GMM}}$ is generated with a random number. One GMM parameter $\boldsymbol{\theta}_{\text{GMM}}$ has K instances for Gaussian parameter $\boldsymbol{\theta}_{\text{Gauss}_k}$ and one instance for scale s, where $\boldsymbol{\theta}_{\text{Gauss}_k}$ is $C+6$-dimensional vector and s is scalar (Eq. 1). And, k-th Gaussian parameter $\boldsymbol{\theta}_{\text{Gauss}_k}$ has $\boldsymbol{v}_{\text{color}_k} = (v_{\text{color}_{k,1}}, ..., v_{\text{color}_{k,C}})$ as pixel value information with C elements, and mean (μ_{x_k}, μ_{y_k}), standard deviation $(\sigma_{x_k}, \sigma_{y_k})$, correlation coefficient ρ_k, and mixing coefficient π_k as density information with 6 elements (Eq. 2).

$$\boldsymbol{\theta}_{\text{GMM}} := (\{\boldsymbol{\theta}_{\text{Gauss}_k}\}_{k=1}^{K},\ s) \tag{1}$$

$$\boldsymbol{\theta}_{\text{Gauss}_k} := (v_{\text{color}_{k,1}},\ ...,\ v_{\text{color}_{k,C}},\ \mu_{x_k},\ \mu_{y_k},\ \sigma_{x_k},\ \sigma_{y_k},\ \rho_k,\ \pi_k) \tag{2}$$

Here, K is the number of Gaussian distributions on one GMM, and if K is large, it can generate defects of complex shapes and color scheme. C is image channel, and it is 1 for grayscale image and 3 for RGB image. The range of image coordinates is defined as $-1 \leq x, y \leq 1$. Moreover, each parameter is randomly generated so that the range is defined as $-1 \leq \mu_{x_k}, \mu_{y_k} \leq 1$ for mean, $\sigma_{x_k}, \sigma_{y_k} > 0$ for standard deviation, $-1 < \rho_k < 1$ for correlation coefficient, $0 \leq \pi_k \leq 1$ (s.t. $\sum_{k=1}^{K} \pi_k = 1$) for mixing coefficient, $-1 \leq v_{\text{color}_{k,c}} \leq 1$ for pixel value, and $0 \leq s \leq 1$ for scale. In Fig. 3, the red values are different for each GMM parameter, while the blue values are always fixed.

Next, the detail of anomaly generator f_{ano} is described as follows. A mask $M_{\text{GMT}}(x, y)$ is created to indicate the mixing ratio of the normal image $\boldsymbol{I}_{\text{norm}}$ and the pixel value $\boldsymbol{v}_{\text{color}}$ of GMM indicating defect color. This mask has the value from 0.0 to 1.0 and the role of reflecting the pixel value of the normal image as it becomes closer to 0.0; otherwise, it reflects the pixel value of GMM. The coordinate (x, y) in this mask is calculated with the

following equations, where $\mathcal{N}$ is the probability density function of the normal distribution.

$$p_{\text{Gauss}_k}(x,y) = \pi_k \mathcal{N}(x,y|\mu_{x_k},\mu_{y_k},\sigma_{x_k},\sigma_{y_k},\rho_k) \tag{3}$$

$$p_{\text{GMM}}(x,y) = \sum_{k=1}^{K} p_{\text{Gauss}_k}(x,y) \tag{4}$$

$$M_{\text{GMT}}(x,y) = s \cdot \frac{p_{\text{GMM}}(x,y)}{\max\limits_{x,y} p_{\text{GMM}}(x,y)} \tag{5}$$

Finally, the pixel value $I_{\text{ano}}(x,y,c)$ of coordinate (x,y) and the c-th channel in the anomaly image $\boldsymbol{I}_{\text{ano}}$ are calculated with the pixel value $I_{\text{norm}}(x,y,c)$ of the normal image $\boldsymbol{I}_{\text{norm}}$ and the pixel value $v_{\text{def}}(x,y,c)$ obtained from the pixel value $\boldsymbol{v}_{\text{color}}$ of GMM as follows:

$$v_{\text{def}}(x,y,c) = \sum_{k=1}^{K} s \cdot \frac{p_{\text{Gauss}_k}(x,y)}{\max\limits_{x,y} p_{\text{GMM}}(x,y)} \cdot v_{\text{color}_{k,c}} \tag{6}$$

$$I_{\text{ano}}(x,y,c) = I_{\text{norm}}(x,y,c)(1 - M_{\text{GMT}}(x,y)) + v_{\text{def}}(x,y,c) \tag{7}$$

Through the above operations, one anomaly image $\boldsymbol{I}_{\text{ano}}$ is generated from one normal image $\boldsymbol{I}_{\text{norm}}$ and one GMM parameter $\boldsymbol{\theta}_{\text{GMM}}$. This operation is repeated many times to generate a large number of GMM parameters and their corresponding anomaly images. Here, the difference between the generated anomaly images is subtle. By training a neural network to predict different values for each generated image, images with subtle difference in image appearance can have different values in the feature space.

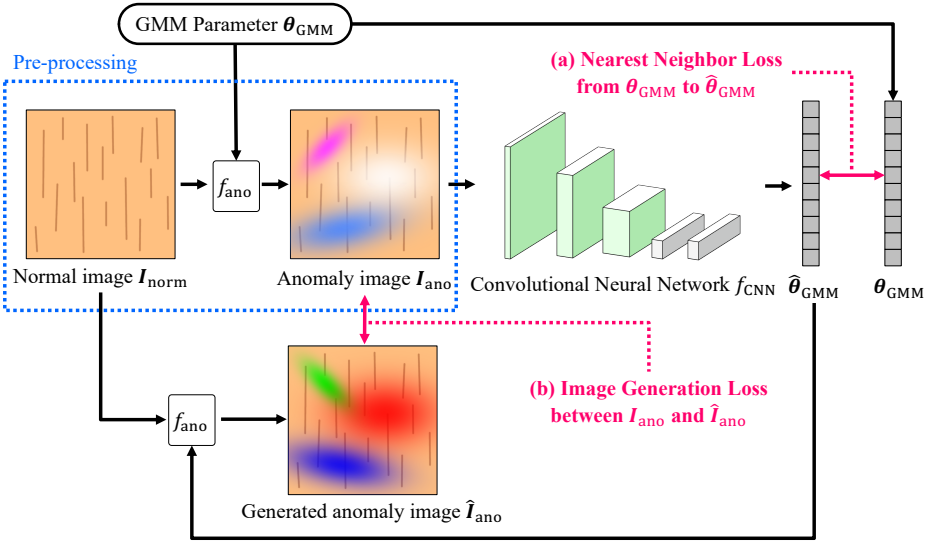

Figure 4: Pre-training based on estimation of GMM parameter from anomaly image.

3.2 Pre-training Based on Estimation of GMM Parameters

In this section, the regressive CNN is pre-trained with the generated anomaly image as input such that the network estimates GMM parameters corresponding to this anomaly image

(Fig. 4). And, two versions for pre-training are introduced as follows: direct estimation (Sec. 3.2.1) and indirect estimation (Sec. 3.2.2). These pre-trainings are performed independently and compared. By these pre-trainings, the regressive CNN can represent " subtle difference " as different features and " anomaly score " with the order of the feature value.

3.2.1 Direct Estimation Training of GMM with Nearest Neighbor

First, an anomaly image $\boldsymbol{I}_{\text{ano}}$ is input to CNN f_{CNN}, and then the final output is obtained. Next, this output is separated to each parameter, and then the function is applied to $\hat{\boldsymbol{v}}_{\text{color}}$ with Tanh, $(\hat{\boldsymbol{\mu}}_x, \hat{\boldsymbol{\mu}}_y)$ with Tanh, $(\hat{\boldsymbol{\sigma}}_x, \hat{\boldsymbol{\sigma}}_y)$ with Softplus, $\hat{\boldsymbol{\rho}}$ with Tanh, $\hat{\boldsymbol{\pi}}$ with Softmax, and $\hat{s}$ with Sigmoid. Finally, the estimated parameter $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ is obtained. Here, $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ consists of $\hat{\boldsymbol{\theta}}_{\text{Gauss}_k}$ and $\hat{s}$ by Eq. 8. Similar to Eq. 2, $\hat{\boldsymbol{\theta}}_{\text{Gauss}_k}$ is the vector with elements of $\hat{\boldsymbol{v}}_{\text{color}_k}$, $(\hat{\mu}_{x_k}, \hat{\mu}_{y_k})$, $(\hat{\sigma}_{x_k}, \hat{\sigma}_{y_k})$, $\hat{\rho}_k$, and $\hat{\pi}_k$. However, the network assigns the estimation result of parameters to arbitrary nodes by ignoring the order of Gaussian distribution for the ground truth label. In other words, when $k = l$, $\boldsymbol{\theta}_{\text{Gauss}_k}$ and $\hat{\boldsymbol{\theta}}_{\text{Gauss}_l}$ do not necessarily correspond to each other. Therefore, it is necessary to find the output nodes $\hat{\boldsymbol{\theta}}_{\text{Gauss}_l}$ that correspond to each Gaussian distribution $\boldsymbol{\theta}_{\text{Gauss}_k}$ of ground truth label and calculate the error between the corresponding nodes. In this paper, this is achieved by finding the nearest neighbor from $\boldsymbol{\theta}_{\text{GMM}}$ to $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ with mean squared error as follows:

$$\hat{\boldsymbol{\theta}}_{\text{GMM}} = f_{\text{CNN}}(\boldsymbol{I}_{\text{ano}}), \quad \text{where } \hat{\boldsymbol{\theta}}_{\text{GMM}} \coloneqq (\{\hat{\boldsymbol{\theta}}_{\text{Gauss}_k}\}_{k=1}^{K}, \hat{s}) \tag{8}$$

$$\mathcal{L}_{\text{direct}} = \frac{1}{K+1}\left((s-\hat{s})^2 + \sum_{k=1}^{K} \min_{1\le l\le K} \frac{1}{C+6} \|\boldsymbol{\theta}_{\text{Gauss}_k} - \hat{\boldsymbol{\theta}}_{\text{Gauss}_l}\|_2^2\right) \tag{9}$$

Finally, after $\mathcal{L}_{\text{direct}}$ is calculated by Eq. 9, the CNN is pre-trained so that $\mathcal{L}_{\text{direct}}$ is minimized. And, CNN will directly be able to estimate the parameters of Gaussian distributions in anomaly images.

3.2.2 Indirect Estimation Training of GMM with Image Generation

In case of Sec. 3.2.1, even if the specific π_k has $\pi_k \coloneqq 0$ $(0 \le \pi_k \le 1)$ and the color of this Gaussian distribution with low density doesn't appear in anomaly image, the training to estimate parameters is forced. However, it is difficult for the network to estimate the Gaussian parameter corresponding to this π_k. To estimate the GMM parameters correctly regardless of the value of π_k, GMM is estimated indirectly using image generation (lower part of Fig. 4).

First, similar to Sec. 3.2.1, the parameter $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ is estimated. Next, the anomaly image $\hat{\boldsymbol{I}}_{\text{ano}}$ is generated with the normal image $\boldsymbol{I}_{\text{norm}}$ and GMM $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ by the anomaly generator f_{ano}. Then, the error between $\boldsymbol{I}_{\text{ano}}$ and $\hat{\boldsymbol{I}}_{\text{ano}}$ is calculated with mean squared error as follows:

$$\hat{\boldsymbol{I}}_{\text{ano}} = f_{\text{ano}}(\boldsymbol{I}_{\text{norm}}, \hat{\boldsymbol{\theta}}_{\text{GMM}}) \tag{10}$$

$$\mathcal{L}_{\text{indirect}} = \|\boldsymbol{I}_{\text{ano}} - \hat{\boldsymbol{I}}_{\text{ano}}\|_2^2 \tag{11}$$

Here, the function f_{ano} is differentiable, then the gradient can be backpropagated from $\mathcal{L}_{\text{indirect}}$ to $\hat{\boldsymbol{\theta}}_{\text{GMM}}$ and f_{CNN}. Finally, after $\mathcal{L}_{\text{indirect}}$ is calculated by Eq. 11, the CNN is pre-trained so that $\mathcal{L}_{\text{indirect}}$ is minimized. And, CNN will indirectly be able to estimate the parameters of Gaussian distributions in anomaly images. By these operations, even if two input images are very similar, if two GMM parameters that represent these images have different values, the feature values and the output values of a network have different values between

two images. This means the CNN is expected to separate normal and anomaly features in the feature space. Moreover, this method can estimate the mean of the Gaussian distribution as the position of pseudo-defects, the standard deviation as the size, the correlation coefficient as the shape, and the density as the transparency. Therefore, the performance is expected to be high when there is an ordinal correlation between "anomaly score" and parameter value (e.g., the defect size in Fig. 3 and the value of the standard deviation).

4 Experiments

4.1 Settings

For generating the pre-training image dataset, we used the beanTech Anomaly Detection (BTAD) dataset [27]. It consists of 2,830 images separated into 3 categories. This has little variation within the normal class on one category, and it is useful to generate anomaly images with "subtle difference". Therefore, all normal images of all categories in BTAD were used as input image $\boldsymbol{I}_{\text{norm}}$ of function f_{ano} for generating an anomaly image $\boldsymbol{I}_{\text{ano}}$. One normal sample is randomly selected from those normal images, one GMM parameter is generated with a random number, and then one anomaly image is generated. This operation was repeated to generate a total of 1,000,000 images. The number C of image channels was 3 to detect the color difference. Additionally, the number K of Gaussian distributions in one GMM parameter was 30 to save the shape and number of various defects. These hyperparameters were decided manually before pre-training, as mentioned above.

For pre-training with the proposed method, we used WideResNet-50 [24] as a CNN. As shown in Fig. 3, the total number of parameters contained in one GMM (i.e., used in one anomaly image) is calculated by $K(C+6)+1$. Therefore, the number of nodes in the final output layer of this network is defined as 271 because of $K = 30$ and $C = 3$.

To evaluate the defect detection performance, we used the MVTec Anomaly Detection (MVTec AD) dataset [22], which is a benchmark dataset for unsupervised defect detection. It consists of 5,354 images separated into 15 categories. Each category contains about 250 training images and 100 test images. In particular, Metal Nut, Pill, Tile, and Wood are labeled as normal/anomaly on a basis of anomaly score from a certain point of view, and PaDiM has low performance (less than 95.0 on Pixel-AUROC such as Sec. 4.2) for these 4 categories. Therefore, we especially focused on these 4 categories. For each category, by using PaDiM, a normal distribution was modeled by using only normal images, and normal/anomaly classification performance was evaluated using both normal and anomaly images in the test. The area under the curve of the receiver operating characteristic (AUROC) was used as a metric to evaluate classification performance at the pixel level (i.e., segmentation performance). The ROC curve was created by utilizing the threshold of the anomaly score. And, five types of random number seed values were used and the mean of five AUROC was calculated.

In this experiment, to evaluate the effectiveness of the proposed pre-training, classification-based pre-training with previous image dataset (ImageNet [25], FractalDB [29], DAGM [28], and BTAD [27]), and regression-based pre-training with proposed GMM-driven image dataset are compared on same PaDiM algorithm. Here, ImageNet has 1,281,167 images and 1,000 classes, FractalDB has 1,000,000 images and 1,000 classes, DAGM has 6,900 images and 12 classes, BTAD has 2,540 images and 6 classes (normal and anomaly images are in different classes), and the proposed dataset has 1,000,000 image pairs and 271-dimensional target variables. For a fair comparison, all methods are pre-trained with 50 [epoch], batch

size of 64, Adam optimizer [32], and image size of 224x224[pixel] on WideResNet-50.

4.2 Results

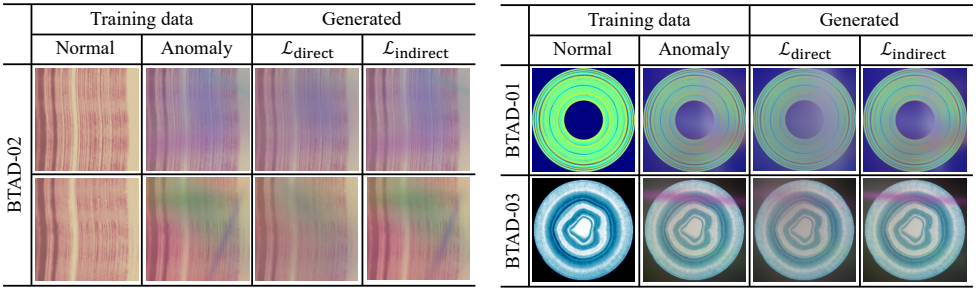

Figure 5: Result of image generation with the proposed method on pre-training.

Figure 5 shows the result of image generation with the proposed method after pre-training. "Normal" and "Anomaly" are normal and anomaly images actually used for pre-training, and the generated image is made by normal image and GMM parameter obtained as the output of CNN with anomaly image as input in each pre-training. The generated image is close to the actual anomaly image, it indicates that the proposed method can estimate GMM parameters.

Table 1: Anomaly detection performance (Pixel-AUROC) in various pre-training.

Category	Previous pre-training				Proposed pre-training	
	ImageNet	FractalDB	DAGM	BTAD	BTAD+GMM w/ $\mathcal{L}_{\text{direct}}$	BTAD+GMM w/ $\mathcal{L}_{\text{indirect}}$
Metal Nut	94.4	91.5	94.5	92.3	95.4	**96.6**
Pill	92.5	71.6	91.0	92.1	93.5	**93.9**
Tile	84.9	72.3	76.8	80.5	90.6	**91.0**
Wood	**90.3**	69.5	85.6	73.7	90.2	89.2
Mean	90.5	76.2	87.0	84.7	92.4	**92.7**
Bottle	**98.1**	90.4	95.1	94.4	97.1	96.6
Cable	95.2	94.4	94.2	87.1	93.5	**95.3**
Capsule	**96.9**	94.4	96.6	96.4	95.6	96.6
Carpet	**98.5**	93.5	78.9	72.3	91.2	92.4
Grid	**95.3**	90.7	74.9	59.8	85.9	89.3
Hazelnut	**97.7**	97.5	97.5	**97.7**	**97.7**	97.6
Leather	98.6	95.0	96.1	93.8	**98.9**	98.6
Screw	**98.3**	97.1	96.4	94.9	93.4	96.3
Toothbrush	98.4	95.8	97.5	**98.5**	97.1	98.0
Transistor	97.3	97.0	97.0	94.5	**97.6**	97.5
Zipper	**97.3**	91.7	94.8	85.1	96.3	97.2
Mean	**97.4**	94.3	92.6	88.6	94.9	95.9
All	**95.6**	89.5	91.1	87.5	94.3	95.1

Table 1 shows the experimental results (Pixel-AUROC) in 15 categories of MVTec AD. In particular, the 4 categories (Metal Nut, Pill, Tile and Wood) where ImageNet performed poorly (less than 95.0 on Pixel-AUROC) are shown in the upper group, and the other 11 categories are shown in the lower group. Overall, PaDiM based on pre-training with proposed GMM-driven image dataset outperformed some previous datasets. In particular, the

proposed method outperformed ImageNet in mean AUROC of 4 categories where ImageNet performed poorly (upper group of table). It is considered that the proposed method has complemented for the weaknesses of ImageNet by pre-training to represent “subtle difference” between two images as different features and two images with similar “anomaly score” as similar features. However, the performance of proposed method was lower than ImageNet in mean AUROC of 11 categories (lower group of table). It is considered that the proposed method had only 3 types of background textures while diversity of ImageNet is large, and it is difficult to adapt various target dataset that differs from pre-training.

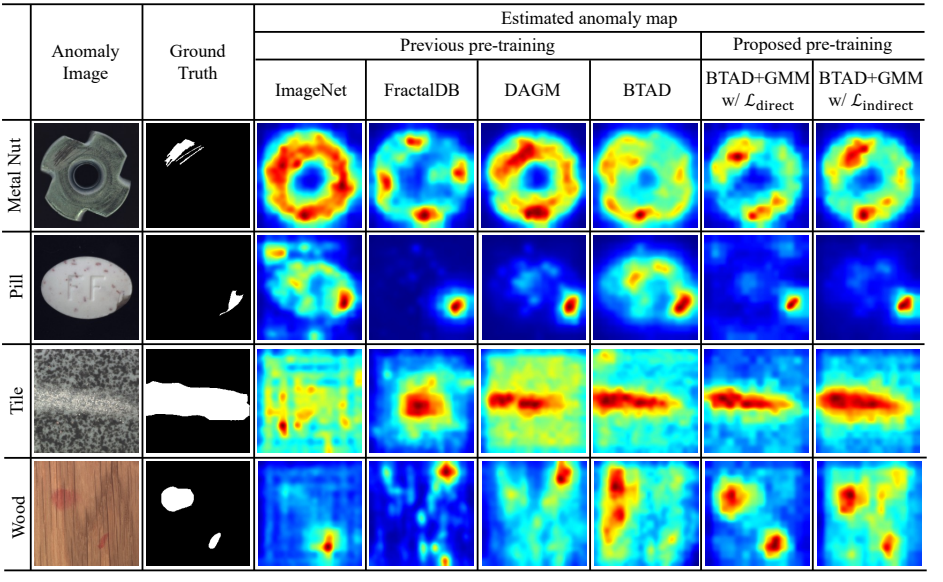

Figure 6: Result of anomaly detection with the estimated anomaly map.

Figure 6 shows the result of anomaly detection with the estimated anomaly map in 4 categories. The heatmap values are linearly normalized for each image. Compared to previous pre-training methods, proposed pre-training methods are able to detect only defect regions.

5 Conclusion

In this paper, we proposed an anomaly detection method by pre-training with semi-formula driven image dataset to represent “subtle difference” between two images as different features and two images with similar “anomaly score” as similar features. An image dataset for pre-training was generated by adding the pseudo-defects with random Gaussian Mixture Model (GMM) parameters to the existing image dataset. GMM parameters have a different value for each parameter, but the appearances of generated images have only subtle difference. And, the regression network was pre-trained to estimate GMM parameters that represent the anomaly score of generated anomaly images. In the experiments with MVTecAD, the proposed method achieved high performance for categories where ImageNet performed poorly.

References

[1] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," Neural Computation, 2001.

[2] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," Machine Learning, 2004.

[3] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep One-Class Classification," ICML, 2018.

[4] J. Yi and S. Yoon, "Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation," ACCV, 2020.

[5] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity," Sensors, 2018.

[6] O. Rippel, P. Mertens, and D. Merhof, "Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection," ICPR, 2021.

[7] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization," Pattern Recognition. ICPR International Workshops and Challenges, 2021.

[8] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards Total Recall in Industrial Anomaly Detection," CVPR, 2022.

[9] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection," ICLR, 2018.

[10] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows," WACV, 2021.

[11] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Fully Convolutional Cross-Scale-Flows for Image-based Defect Detection," WACV, 2022.

[12] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows," arXiv preprint arXiv:2111.07677, 2021.

[13] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving Unsupervised Defect Segmentation by Applying Structural Similarity To Autoencoders," VISAPP, 2019.

[14] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection," ICCV, 2019.

[15] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, and H. Zhou, "Divide-and-Assemble: Learning Block-wise Memory for Unsupervised Anomaly Detection," ICCV, 2021.

[16] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline, "Iterative energy-based projection on a normal data manifold for anomaly localization," ICLR, 2020.

[17] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," IPMI, 2017.

[18] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," ICLR Workshop, 2018.

[19] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training," ACCV, 2018.

[20] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection," IJCNN, 2019.

[21] A. Mousakhan, T. Brox, and J. Tayyub, "Anomaly Detection with Conditioned Denoising Diffusion Models," arXiv preprint arXiv:2305.15956, 2023.

[22] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," IJCV, 2021.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," CVPR, 2016.

[24] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," arXiv preprint arXiv:1605.07146, 2016.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," CVPR, 2009.

[26] H. Kobayashi and M. Hashimoto, "DRepT: Anomaly Detection Based on Transfer of Defect Representation with Transmittance Mask," IJCNN, 2023.

[27] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization," ISIE, 2021.

[28] M. Wieler and T. Hahn, "Weakly Supervised Learning for Industrial Optical Inspection," DAGM, 2007.

[29] H. Kataoka, K. Okayasu, A. Matsumoto, E. Yamagata, R. Yamda, N. Inoue, A. Nakamura, Y. Satoh, "Pre-Training Without Natural Images," IJCV, 2022.

[30] H. Kataoka, R. Hayamizu, R. Yamada, K. Nakashima, S. Takashima, X. Zhang, Edgar Josafat Martinez-Noriega, Nakamasa Inoue, Rio Yokota, "Replacing Labeled Real-Image Datasets with Auto-Generated Contours," CVPR, 2022.

[31] S. Takashima, R. Hayamizu, N. Inoue, H. Kataoka, R. Yokota, "Visual Atoms: Pre-training Vision Transformers with Sinusoidal Waves," CVPR, 2023.

[32] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," ICLR, 2015.