

# PNI : Industrial Anomaly Detection using Position and Neighborhood Information

Jaehyeok Bae <sup>\* 1,2</sup> Jae-Han Lee <sup>\* 1</sup> Seyun Kim <sup>† 1</sup>

<sup>1</sup>Gauss Labs Inc. <sup>2</sup>Seoul National University

wogur110@snu.ac.kr, {jaehan.lee, seyun.kim}@gausslabs.ai

## Abstract

Because anomalous samples cannot be used for training, many anomaly detection and localization methods use pre-trained networks and non-parametric modeling to estimate encoded feature distribution. However, these methods neglect the impact of position and neighborhood information on the distribution of normal features. To overcome this, we propose a new algorithm, **PNI**, which estimates the normal distribution using conditional probability given neighborhood features, modeled with a multi-layer perceptron network. Moreover, position information is utilized by creating a histogram of representative features at each position. Instead of simply resizing the anomaly map, the proposed method employs an additional refine network trained on synthetic anomaly images to better interpolate and account for the shape and edge of the input image. We conducted experiments on the MVTec AD benchmark dataset and achieved state-of-the-art performance, with **99.56%** and **98.98%** AUROC scores in anomaly detection and localization, respectively. Code is available at [https://github.com/wogur110/PNI\\_Anomaly\\_Detection](https://github.com/wogur110/PNI_Anomaly_Detection).

## 1. Introduction

In industrial inspection [1], delivering defective products to customers due to detection failure can be costly, and false detection can increase manufacturing costs. Therefore, high prediction accuracy alone is insufficient, and low false positive rate (FPR) and false negative rate (FNR) are preferred. Additionally, collecting abnormal samples can be difficult, making it almost impossible to build a supervised model for the task. Thus, anomaly detection methods that use only normal samples are adopted. Anomaly localization quantifies the anomaly of each pixel in an input image, allowing users to identify where the defect is located and improve

<sup>\*</sup>Equal contribution.

<sup>†</sup>Corresponding author.

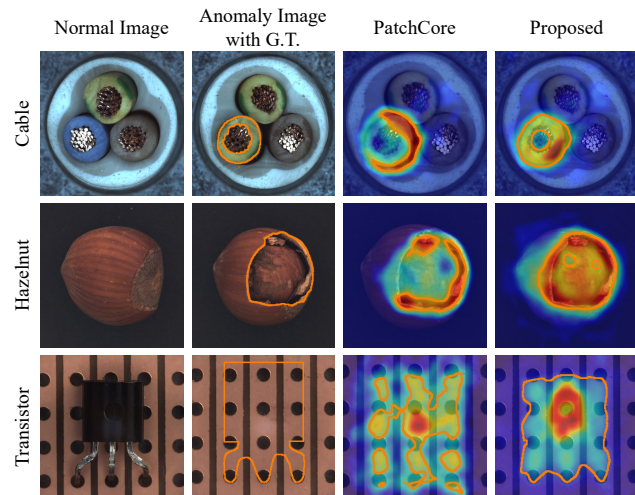


Figure 1: Examples from MVTec AD [1]. The normal images (left) and the anomalous images (second column) overlaid by ground truth mask are followed by the anomaly maps from PatchCore [21] (third column) and our proposed approach (right). The contours overlaid on anomaly maps are from thresholds optimizing F1 scores of anomaly localization.

manufacturing processes. Figure 1 displays example images and results of an existing method and our proposed approach from the MVTec AD benchmark dataset.

Since there are only normal samples available for training, conventional classification methods cannot be used. One approach is to generate defective samples to train classifiers with supervised learning [29, 28, 22]. CutPaste [17], for instance, used masks of rectangular and scar shapes to learn representation in a self-supervised manner. However, these methods show relatively low performance compared to other recent methods due to the lack of realism in abnormal patterns. To overcome this difficulty, many recently proposed methods [6, 21] adopt a pre-trained network such as pre-trained ResNet [11] using ImageNet [8], which intensively learned low-level image features from a super large-

sized dataset.

Various methods are utilized to model the distribution of normal features that are transformed by a pre-trained network. For example, PatchCore [21] sub-samples representative features from the extracted normal features to achieve efficient non-parametric modeling of nominal features. Similarly, CFLOW-AD [9] uses normalizing flow to model the normal feature distribution. However, these methods quantify the anomalies of input feature vectors independently, without considering the correlation between neighboring features. Additionally, normal features can be abnormal if they are in the wrong position. For instance, in the first row of Figure 1, the top-view cables are normal only when the color order is correct, as shown in the first column image. However, if the color order is incorrect, the product is defective, even though all the local features are normal. Existing representation-based approaches, such as PatchCore, cannot capture this type of abnormality. Although CFLOW-AD adopted position encoding blocks, the implicit method appears insufficient to model positional information and overlooks the correlation between normal features.

To address this problem, this paper utilizes position and neighborhood information in simple yet effective ways. At each position of the encoded feature dimension, a histogram of all the training features is constructed to model a conditional probability distribution given the positional information. Meanwhile, an MLP (multi-layer perceptron) network models the probability distribution of normal features conditioned by neighboring features, where the input is the concatenated neighboring features. Through this process, the MLP network observes a large support region, while the features remain local, allowing the proposed method to produce a finely detailed localization map. These two distributions are combined to estimate the likelihood and anomaly score of an input image and its pixels during testing. While PatchCore serves as a baseline to demonstrate the validity of the proposed ideas, they can be applied to any representation-based method that uses a pre-trained network to generate input features and non-parametric modeling of normal features.

Representation-based approaches have a limitation in depicting detailed anomaly maps because local features are extracted from image patches of moderate size. When the patch size is small, enough information may not be extracted, leading to degraded detection performance. On the other hand, a large patch size may result in a blurred localization map. To overcome this problem, we trained an additional refinement network with synthetic abnormal images, which improves the detail of the localization map. While the simulation of synthetic abnormalities was explored in previous works [17, 32, 34], the proposed approach is differentiated by its design of refinement network architec-

ture and loss function aimed at correcting representation-based distances, and by introducing various defect generation methods to enhance refinement robustness. It’s important to note that synthetic images are not used to encode input images or to estimate anomaly scores. They are used only to train the refinement network. This is different from existing methods. By using synthetic images and corresponding anomaly maps generated by the above-mentioned method, the refinement network learns how to revise the anomaly map to look like the ground truth mask.

The proposed method resulted in a decrease in FNR from 1.83% to 0.95% compared to the current state-of-the-art method [21]. This reduction means that customers receive 48% fewer defective products. Additionally, The FPR was reduced from 4.07% to 1.50%, which means that 63% fewer good products are wasted. Although the improvement in the area under the receiver operating characteristic (AUROC) metric, 0.46%, may seem small, it can provide significant benefits to industrial manufacturing.

In summary, our contributions are threefold. Firstly, we demonstrate the effectiveness of using conditional normal feature distribution based on position and neighborhood information for anomaly detection and localization. Secondly, we validate that training a refinement network with synthetic datasets can significantly enhance performance. Finally, we provide insight into the factors that contribute to the noticeable improvement with the ablation study.

## 2. Related Work

We selected PatchCore [21] as our baseline because it employs a generic representation-based structure using non-parametric modeling and exhibits state-of-the-art performance. PatchCore aggregates local patch features from normal training data and selects a representative subset through greedy coresets subsampling [23]. During testing, the anomaly score for each patch feature is calculated pixel-wise by performing a nearest neighbor search from the coreset. Our proposed method utilizes the same process for feature vector creation, which we have summarized for completeness in this paper. However, it’s worth noting that any other feature extraction process can be used because our proposed method is independent of the process used.

With a pre-trained network  $\phi$ , an input image is converted into hierarchical features  $\phi_{i,j} = \phi_j(x_i)$ , where  $j$  denotes the hierarchy level of  $\phi$ . For instance, in ResNet-50 [11],  $j \in 1, 2, 3, 4$  represents the final output of each resolution block. We denote the feature map  $\phi_{i,j} \in \mathbb{R}^{c^j \times h^j \times w^j}$  as a three-dimensional tensor, where  $c^j$ ,  $h^j$ , and  $w^j$  are the number of channels, height, and width, respectively. To avoid using too high or low-level features, the intermediate features  $\phi_{i,2}$  and  $\phi_{i,3}$  are concatenated and used. As the spatial sizes  $(h, w)$  of these features are different, the smaller one is resized to be the same size as the larger one:  $(h^*, w^*)$ ,

where  $h^* = \max(h_2, h_3)$  and  $w^* = \max(w_2, w_3)$ . Then, they are concatenated to obtain  $\phi_i^* \in \mathbb{R}^{c^* \times h^* \times w^*}$ , where  $c^* = c_2 + c_3$ . Furthermore, to increase the receptive field of feature maps, the pixel-level feature  $\Phi_i(h, w)$  is extended to incorporate neighborhood features within a specific patch size  $p$ . Adaptive average pooling is performed to output a single feature of dimension  $d$  at  $(h, w)$ . Through this process, the input image is converted into a set of local patch-level features  $\Phi_i \in \mathbb{R}^{d \times h^* \times w^*}$ , where  $d$  denotes the dimension of the feature vector.

### 3. Method

#### 3.1. Overview

Assume that  $\mathbf{x}$  represents the spatial coordinates  $(h, w)$  in the patch-level feature  $\Phi_i$ . In most existing representation-based methods [5, 6, 21], the anomaly score of the patch-level feature  $S(\mathbf{x})$  is estimated as the negative log-likelihood of  $p(\Phi_i(\mathbf{x}))$ , given by

$$S(\mathbf{x}) = -\log p(\Phi_i(\mathbf{x})), \quad (1)$$

where  $p(\Phi_i(\mathbf{x}))$  represents the probability that  $\Phi_i(\mathbf{x})$  is normal and is modeled using trained normal features.

In this paper, we argue that the probability should be modeled based on the position and neighboring features. As shown in Figure 1, electric wires are located within their sheaths (as seen in the left image of the first row), while transistors in the normal dataset are typically located at the center of the images (as seen in the left image of the third row). Denoting position and neighborhood information as  $\Omega$ , the anomaly score  $S(\mathbf{x})$  is represented as negative log-likelihood of conditional probability of  $\Phi_i(\mathbf{x})$  given  $\Omega$ :

$$S(\mathbf{x}) = -\log p(\Phi_i(\mathbf{x})|\Omega). \quad (2)$$

To model the conditional probability from training features, we introduce the embedding coreset  $C_{\text{emb}}$ . The feature vectors of  $C_{\text{emb}}$  are sub-sampled from all normal features in all training images using a greedy coreset sub-sampling method [23]. Each element of  $C_{\text{emb}}$  delegates a group of similar normal features. In a given circumstance, the number of occurrences of normal features associated with an embedding coreset vector  $c \in C_{\text{emb}}$  is proportional to the probability that  $c$  is normal in that condition  $p(c|\Omega)$ . The normal probability of a patch  $p(\Phi_i(\mathbf{x})|\Omega)$  is expressed with  $C_{\text{emb}}$  as follows:

$$p(\Phi_i(\mathbf{x})|\Omega) = \sum_{c \in C_{\text{emb}}} p(\Phi_i(\mathbf{x})|c, \Omega) p(c|\Omega). \quad (3)$$

While the computation is challenging with a large size of  $C_{\text{emb}}$ , it has been observed that  $p(c|\Omega)$  is a sparse distribution with many small values that can be ignored. To take advantage of this property, (3) is approximated as follows:

$$p(\Phi_i(\mathbf{x})|\Omega) \approx \max_{c \in C_{\text{emb}}} p(\Phi_i(\mathbf{x})|c, \Omega) T_\tau(p(c|\Omega)), \quad (4)$$

where  $T_\tau(x)$  is defined as:

$$T_\tau(x) = \begin{cases} 1, & \text{if } x > \tau \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

To stop considering insignificant values of  $p(c|\Omega)$ , the threshold function is applied. Applying  $T_\tau(x)$ ,  $p(c|\Omega)$  with moderate probability becomes one. Using this thresholding technique and the maximum operation in equation (4), it is possible to identify the coreset feature that is most similar to  $\Phi_i(\mathbf{x})$  while rejecting improbable  $c_{\text{emb}}$  features. While this approximation may not be intuitive, it can significantly reduce computation time with only a small decrease in performance.  $\tau$  lower than  $1/|C_{\text{emb}}|$  guarantees at least one of  $c$  in  $C_{\text{emb}}$  be a normal feature. In this paper, we set  $\tau = 1/(2|C_{\text{emb}}|)$  without optimizing.

To generate an anomaly score map, scores are estimated for all features in an input image. However, the resolution of the score map may differ from that of the original input, so it is resized using bi-linear interpolation and smoothed with a Gaussian kernel of  $\sigma = 8$  as described in [21]. Note that the parameter  $\sigma$  is not extensively optimized. While Gaussian smoothing is performed to eliminate noisy values, it may damage the detailed information of the score map. Therefore, an additional pixelwise refinement step is performed to enhance the resized score map and make it more consistent with the edges, textures, and shapes of defects and objects in the input image.

#### 3.2. Modeling Normal Feature Distribution

To model the normal feature distribution  $p(c|\Omega)$ , it is approximated as an average of the two probabilities as:

$$p(c|\Omega) \approx \frac{p(c|N_p(\mathbf{x})) + p(c|\mathbf{x})}{2}, \quad (6)$$

where  $p(c|N_p(\mathbf{x}))$  is the normal feature distribution in neighborhood information and it is modeled using an MLP. To model  $p(c|\mathbf{x})$ , the normal feature distribution in position information, histograms are constructed by counting the normal training features at every position  $\mathbf{x}$  as shown in Figure 2. To train the MLP and create the histograms, using a small size for  $C_{\text{emb}}$  is preferable. However, reducing the size of  $C_{\text{emb}}$  can lead to a decrease in the accuracy of the normal probability of the input vector,  $p(\Phi_i(\mathbf{x})|c, \Omega)$ . To address this issue, we introduce the distribution coreset,  $C_{\text{dist}}$ , which is sub-sampled from the embedding coreset  $C_{\text{emb}}$  using the same method as in [23]. In our implementation, both coresets are calculated simultaneously because  $C_{\text{dist}}$  is a subset of  $C_{\text{emb}}$ , and the mapping from  $c_{\text{emb}}$  vectors to  $c_{\text{dist}}$  vectors is calculated at the beginning. Therefore,  $p(c_{\text{emb}}|\Omega)$  in equation (4) is changed to  $p(c_{\text{dist}}|\Omega)$  according to the corresponding  $c_{\text{dist}}$  vectors.

To model  $p(c_{\text{dist}}|\Omega)$ , a simple MLP network is trained with neighboring features  $N_p(\mathbf{x})$  of input feature  $\Phi_i(\mathbf{x})$ .

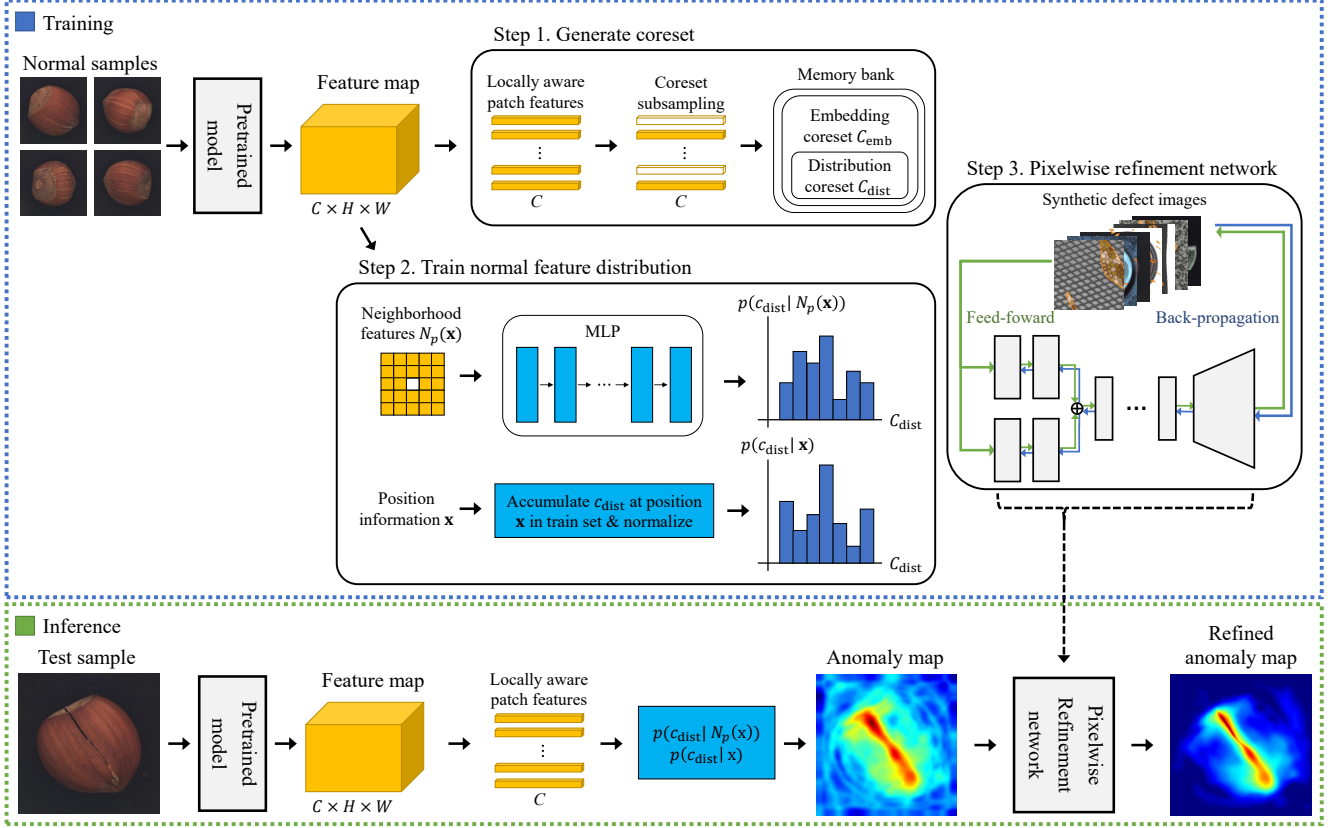


Figure 2: Overview of PNI algorithm. At train time, normal samples are converted to feature map  $\Phi_i$  using ImageNet pre-trained model  $\phi$ . Aggregated patch-level features are subsampled to generate embedding coreset  $C_{emb}$  and distribution coreset  $C_{dist}$  using the greedy subsampling method. After storing the coresets, normal feature distribution given neighborhood and position information is trained with MLP and histogram respectively. A pixelwise refinement network is trained separately using synthetic defect images. At inference time, anomaly score for local test feature is evaluated using the trained normal feature models. At last, the refinement step is performed to improve the anomaly map considering the input image.

$N_p(\mathbf{x})$  is defined as a set of features that are within a  $p \times p$  patch, excluding  $\mathbf{x}$  itself as follows:

$$N_p(\mathbf{x}) = \{\Phi_i(m, n) \mid |m - h| \leq p/2, |n - w| \leq p/2, (m, n) \neq (h, w)\}, \quad (7)$$

where  $\Phi_i(m, n)$  is feature vector at position  $(m, n)$  in the feature map  $\Phi_i$ . The MLP takes an input of a 1-dimensional vector obtained by concatenating all features in  $N_p(\mathbf{x})$  and has  $N_{MLP}$  sequential layers with  $c_{MLP}$  channels. Batch normalization and ReLU activation functions are used between layers. The output of the MLP has  $|C_{dist}|$  nodes, with the value of each node representing the probability of the corresponding distribution coreset feature. The ground truth used for training is a one-hot vector, where the distribution coreset index closest to the true center feature vector is one, and the cross-entropy loss is calculated with the MLP output. To address the overconfidence of the trained deep neural network models, temperature scaling [10] with temperature  $T = 2$  is applied to make the confidence values

more realistic.

Position information  $\mathbf{x}$  is also crucial and can significantly affect the probability of  $\Phi_i(\mathbf{x})$  being a normal feature, especially in object-type images. To capture the position information, we generate  $p(c_{dist} | \mathbf{x})$  by accumulating the indices of  $C_{dist}$  for each position  $\mathbf{x}$  in all training images  $\forall x_i$ , using Algorithm 1. In this process, features in the  $p \times p$  neighborhood are accumulated in the histogram for robust estimation.

To calculate  $p(\Phi_i(\mathbf{x}) | c_{emb}, \Omega)$  in (4), we assume that  $p(\Phi_i(\mathbf{x}) | c_{emb})$  is independent of  $\Omega$ , since  $c_{emb}$  contains all the information in  $\Omega$  related to  $\Phi_i(\mathbf{x})$ . Then,  $p(\Phi_i(\mathbf{x}) | c_{emb})$  is expressed in terms of an exponent of the distance between  $\Phi_i(\mathbf{x})$  and  $c_{emb}$  as in most existing methods:

$$p(\Phi_i(\mathbf{x}) | c_{emb}, \Omega) \approx p(\Phi_i(\mathbf{x}) | c_{emb}) \approx e^{-\lambda \|\Phi_i(\mathbf{x}) - c_{emb}\|_2}, \quad (8)$$

where  $\lambda$  is a hyperparameter of an exponential function, and we set  $\lambda = 1$  without optimizing.

---

**Algorithm 1** Calculation of  $p(c_{\text{dist}}|\mathbf{x})$ 

---

- 1: Initialize  $\text{hist}(\cdot|\mathbf{x})$  as a zero vector of  $\mathbb{R}^{|c_{\text{dist}}|}$  for all  $\mathbf{x}$
  - 2: **for** all training images  $x_i$  **do**
  - 3:   **for** all coordinates  $\mathbf{x}$  **do**
  - 4:      $\text{idx} \leftarrow \text{find an index of nearest } c_{\text{dist}} \text{ to } \Phi_i(\mathbf{x})$
  - 5:      $\text{hist}(\text{idx}|\mathbf{x}) \leftarrow \text{hist}(\text{idx}|\mathbf{x}) + 1$
  - 6:   **end for**
  - 7: **end for**
  - 8:  $p(c_{\text{dist}}|\mathbf{x}) \leftarrow \text{normalize}(\text{hist}(\cdot|\mathbf{x}))$
- 

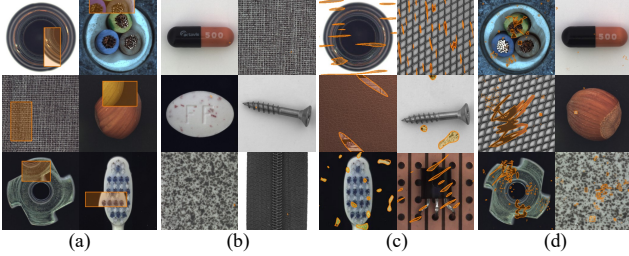


Figure 3: Examples of defect images generated by (a) Cut-Paste, (b) CutPaste (scar), (c) DRÆM, and (d) manual drawing. The area corresponding to the defects are highlighted.

### 3.3. Pixelwise Refinement

We further improve the reliability of the anomaly map by using a refinement network  $f$ , trained in a supervised manner using an artificially created defect image dataset  $\mathcal{D}$ . Let  $\theta$  be parameters of  $f$ . We aim to train optimal parameters

$$\theta^* = \arg \min_{\theta} \sum_{(I, \hat{A}, A) \in \mathcal{D}} \ell(f(I, \hat{A}; \theta), A). \quad (9)$$

$\mathcal{D}$  is composed of  $(I, \hat{A}, A)$  pairs.  $I$  is an artificially generated anomaly image, and  $A$  represents the ground-truth anomaly map of  $I$ , with 1 assigned to defect regions and 0 assigned to others.  $\hat{A}$  is an anomaly map estimated from the proposed algorithm. We normalize each map into  $[0, 1]$ .  $\ell$  is a loss function between the refine  $\tilde{A} \triangleq f(I, \hat{A}; \theta)$  and the ground-truth  $A$ .

we create four different types of data for the dataset  $\mathcal{D}$  with the same ratio. These four methods include Cut-Paste [17], CutPaste (scar) [17], DRÆM [32], and manual drawing. As pointed out in CutPaste, training with defects of varying sizes and shapes together prevents the network from optimizing in a naive direction and enables better generalization performance. This is a significant advantage in cases where real abnormal data is unknown. Figure 3 shows the defect image examples generated by 4 methods from normal MVTEC AD training data. Defects generated by each method have distinct characteristics. CutPaste creates rectangular defects in larger areas, while CutPaste (scar) produces more detailed and thinner defects. DRÆM and

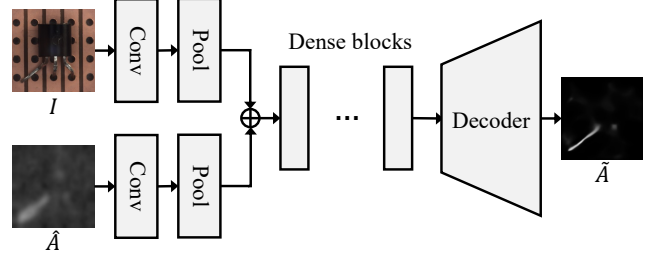


Figure 4: Schematic structure of the refinement network.

manual methods generate a more complex variety of defect patterns.

We adopt the encoder-decoder architecture for  $f$ . The network structure is based on [16] that uses DenseNet161 [13] as the backbone, but we introduce two modifications to it. First, the refinement network takes 4-channel inputs of an RGB image  $I$  and an anomaly map  $\hat{A}$ . Second, we apply the early fusion method [14] and fuse the features of  $I$  and  $\hat{A}$  after the first convolution layer. A schematic structure of the pixelwise refinement network is presented in Figure 4.

To train  $f$ , we use a loss function consisting of two terms:  $\ell = (\ell_{\text{reg}} + \ell_{\text{grad}})/2$ . The regression loss  $\ell_{\text{reg}}$  is calculated using L2-norm between  $\tilde{A}$  and  $A$ .

$$\ell_{\text{reg}} = \frac{\|\tilde{A} - A\|_2}{HW}, \quad (10)$$

where  $H$  and  $W$  are the width and height of  $A$ . Next, the gradient loss  $\ell_{\text{grad}}$  is

$$\ell_{\text{grad}} = \frac{\|\nabla_{\text{h}} \tilde{A} - \nabla_{\text{h}} A\|_2 + \|\nabla_{\text{w}} \tilde{A} - \nabla_{\text{w}} A\|_2}{2HW}, \quad (11)$$

where  $\nabla_{\text{h}}$  and  $\nabla_{\text{w}}$  are partial derivative operations in the vertical and horizontal directions, respectively.  $\ell_{\text{grad}}$  improves the refinement results by making the network's training more concentrated near the edges of the defect region.

## 4. Experimental Results

### 4.1. Implementation Details

**Datasets** We adopt three popular industrial datasets, MVTEC AD [1], BTAD [19], and VisA[35] to evaluate the proposed PNI. MVTEC AD includes 15 subcategories, consisting of 10 object categories and 5 texture categories. The dataset contains a total of 5,354 color images, including 3,629 defect-free training images and 1,725 test images that include both normal and anomalous images with ground-truth defect masks. Anomalous images are labeled with various types of defects. BTAD is an industrial anomaly detection dataset with 3 subcategories and a total of 2,830 color images. Of these, 1,800 training images are

Table 1: Anomaly detection and localization AUROC scores on MVTec AD [1] are presented. The first and second numbers indicate I-AUROC (image-level detection score) and P-AUROC (pixel-level localization score), respectively. Sub-total averages are provided for object and texture categories. For each category, the best result is **boldfaced**.

	RIAD [33]	InTra [20]	CutPaste [17]	FastFlow [30]	Tsai <i>et al.</i> [24]	CFLOW-AD [9]	PatchCore [21]	<b>PNI</b>
Object	Bottle	99.9 98.4	<b>100</b> 97.1	98.3 97.6	<b>100</b> 97.7	<b>100</b> 98.6	<b>100</b> 98.6	<b>100 98.87</b>
	Cable	81.9 84.2	70.3 91.0	80.6 90.0	<b>100</b> 98.4	98.8 98.2	97.59 97.64	99.76 <b>99.10</b>
	Capsule	88.4 92.8	86.5 97.7	96.2 97.4	<b>100</b> 99.1	97.2 97.9	97.68 98.98	98.1 98.8
	Hazelnut	83.3 96.1	95.7 98.3	97.3 97.3	<b>100</b> 99.1	99.6 97.8	99.98 98.82	<b>100</b> 98.7
	Metal nut	88.5 92.5	96.9 93.3	99.3 93.1	<b>100</b> 98.5	97.8 99.1	99.26 98.56	<b>100</b> 98.4
	Pill	83.8 95.7	90.2 98.3	92.4 95.7	<b>99.4 99.2</b>	97.7 98.8	96.82 98.95	96.6 97.4
	Screw	84.5 98.8	95.7 99.5	86.3 96.7	97.8 99.4	94.1 98.5	91.89 98.10	<b>98.1</b> 99.4
	Toothbrush	<b>100</b> 98.9	<b>100</b> 98.9	98.3 98.1	94.4 98.9	<b>100</b> 99.0	99.65 98.56	<b>100</b> 98.7
	Transistor	90.9 87.7	95.8 96.1	95.5 93.0	99.8 97.3	98.9 97.7	95.21 93.28	<b>100</b> 96.3
	Zipper	98.1 97.8	99.4 99.2	99.4 99.3	99.5 98.7	99.5 98.6	98.48 98.41	99.4 98.8
Average	89.9 94.3	93.0 96.9	94.3 95.8	99.1 98.6	98.4 98.4	97.66 98.01	99.2 98.4	<b>99.55 99.12</b>
Texture	Carpet	84.2 96.3	98.8 99.2	93.1 98.3	<b>100 99.4</b>	93.4 98.4	98.73 99.23	98.7 99.0
	Grid	99.6 98.8	<b>100</b> 98.8	99.9 97.5	99.7 98.3	<b>100</b> 98.5	99.60 96.89	98.2 98.7
	Leather	<b>100</b> 99.4	<b>100</b> 99.5	<b>100</b> 99.5	<b>100</b> 99.5	99.3 99.1	<b>100 99.61</b>	<b>100</b> 99.3
	Tile	98.7 89.1	98.2 94.4	93.4 90.5	<b>100</b> 96.3	96.2 94.4	99.88 97.71	98.7 95.6
	Wood	93.0 85.8	97.5 88.7	98.6 95.5	<b>100</b> 97.0	99.7 <b>97.5</b>	99.12 94.49	99.2 95.0
Average	95.1 93.9	98.9 96.1	97.0 96.3	<b>99.9</b> 98.1	97.7 97.6	99.47 97.59	99.0 97.5	99.59 <b>98.72</b>
Average	91.7 94.2	95.0 96.6	95.2 96.0	99.4 98.5	98.1 98.1	98.26 97.87	99.1 98.1	<b>99.56 98.98</b>

normal, and the remaining test images include both normal and anomalous images with ground-truth masks. VisA includes 12 subcategories, consisting of 4 single instances, 4 multiple instances, and 4 complex structures, and contains a total of 10,821 color images, including 9,621 normal and 1,200 anomalous ones. The anomalous images are evenly distributed with 100 images per subcategory, and ground-truth masks for the anomalous regions are provided. As in [5, 6, 29, 21], images from all datasets are resized to  $512 \times 512$  and center-cropped to  $480 \times 480$  to remove negligible boundary pixels.

**Evaluation Metrics** To access the performance of the proposed PNI, we use two metrics, AUROC (Area Under the Receiver Operator Curve) and AUPRO (per-region-overlap curve), as done in [21, 24, 7]. AUROC is measured at the image level (I-AUROC) for anomaly detection performance and at the pixel level (P-AUROC) for anomaly localization performance. AUPRO evaluates the anomaly localization performance by assigning equal weight to anomalous regions of different sizes in the image. AUPRO addresses the drawback of P-AUROC, where a prediction result in a single large anomalous region may have a greater impact than those in many small anomalous regions. High AUPRO indicates that the algorithm provides good anomaly localization results for both large and small anomalous regions.

**Parameter Setup** Similar to [17, 21], we trained two models: a single network-based model and an ensemble network-based model. For the single model, WideResNet-101 [31] pre-trained on ImageNet [8] data is used as the

feature extractor. Also, for the ensemble model, ResNext-101 [27] and DenseNet-201 [13] are additionally used as feature extractors. The subsampling ratio to generate the embedding coreset is set to 0.01, and the size of the distribution coreset  $|C_{\text{dist}}|$  is set to 2,048. In the training process of  $p(c_{\text{dist}}|\mathbf{x})$  and  $p(c_{\text{dist}}|N_p(\mathbf{x}))$ , the patch size of the neighborhood  $p$  is set to 9. The MLP network for the normal feature distribution consists of 10 fully-connected layers and each layer includes 2,048 neurons. We train the MLP network using the Adam optimizer [15] for 15 epochs with a  $10^{-3}$  learning rate and batch size 2,048. Also, we adopt step learning rate decaying [12] with  $\gamma = 0.1$  and 5 step size. We don't use any data augmentation since each category has different permissible augmentation based on the characteristics of the images.

In the training process of the refinement network, we use the Adam optimizer for 55,000 iterations with a  $10^{-4}$  learning rate and batch size 8. Also, we perform online data augmentation, including random horizontal flip, rotation, and color change in an online manner. In inference, we fuse  $\tilde{A}$  at 0.1 ratio with  $\hat{A}$  to obtain the final anomaly map.

## 4.2. MVTec AD

**AUROC** Table 1 shows the performance of anomaly detection and localization for 15 subcategories of the MVTec AD dataset. We compare our proposed PNI algorithm with several conventional algorithms [33, 20, 17, 30, 24, 9, 21] in terms of I-AUROC and P-AUROC. Here, the results of single model versions of the proposed PNI algorithm, Cut-Paste, and PatchCore are compared. Also, the results of CFLOW-AD use an evaluation protocol that selects the best

Table 2: Comparison of anomaly detection and localization results on MVTEC AD [1]. The proposed PNI is compared to recent algorithms in terms of I-AUROC, P-AUROC, and AUPRO. For AUPRO, sub-total averages are provided for both object and texture subcategories additionally.

	AUROC		AUPRO		
	Image	Pixel	Object	Texture	Average
Patch SVDD [29]	92.1	95.7	-	-	-
SPADE [5]	85.5	96.0	93.4	88.4	91.7
PaDiM [6]	95.3	97.5	91.6	93.1	92.1
RIAD [33]	91.7	94.2	-	-	-
CutPaste [17]	95.2	96.0	-	-	-
DR-EM [32]	98.0	97.3	-	-	-
FastFlow [30]	99.4	98.5	-	-	-
SOMAD [18]	97.9	97.8	94.1	91.6	93.3
InTra [20]	95.0	96.6	-	-	-
MB-PFM [26]	97.5	97.3	92.3	94.6	93.0
NSA [22]	97.2	96.3	90.4	92.2	91.0
IKD [3]	-	97.81	93.30	91.05	92.55
PatchCore [21]	99.1	98.1	93.3	93.6	93.4
Reverse Distillation [7]	98.5	97.8	93.4	95.0	93.9
Tsai <i>et al.</i> [24]	98.1	98.1	95.7	95.0	95.5
PEFM [25]	-	98.30	95.30	<b>95.95</b>	95.52
CDO [4]	-	98.22	94.57	94.90	94.68
<b>PNI</b>	<b>99.56</b>	<b>98.98</b>	<b>96.34</b>	95.47	<b>96.05</b>
Uniformed Students [2]	-	-	90.8	92.7	91.4
CutPaste (ensemble) [17]	96.1	-	-	-	-
PatchCore (ensemble) [21]	99.6	98.2	-	-	94.9
CFLOW-AD [9]	98.26	98.62	93.58	96.65	94.60
<b>PNI (Ensemble)</b>	<b>99.63</b>	<b>99.06</b>	<b>96.83</b>	<b>96.00</b>	<b>96.55</b>

results from training with various hyperparameters. Some of the conventional algorithms provide multiple models by varying the experimental settings, and we provide detailed information on this in the supplementary document.

The proposed PNI algorithm shows the best anomaly detection performance of 99.56% I-AUROC and anomaly localization performance of 98.98% P-AUROC, surpassing FastFlow by 0.16% and 0.48%, respectively. Although FastFlow with CaiT shows high performance based on a powerful transformer, there is a noticeable performance drop in a few classes, leading to a decrease in the average score. Our motivation for using position and neighborhood information can be more beneficial in object-type images, and the results support this guess. PNI significantly outperforms conventional algorithms in object classes, with considerable improvements in the localization of transistor, metal nut, and bottle classes. Moreover, the average I-AUROC of PNI for object classes is 99.55%, showing a 43.8% reduction in error compared to the second-best PatchCore. Using neighborhood information for each pixel is also effective in measuring the anomaly at that position, even for texture classes. In addition, pixelwise refinement works effectively on texture subcategories. The synthesized defect images used in network training handle various and complex defects occurring in real texture images well.

Table 3: The Ablation Results on MVTEC AD. We analyze the effects of the three main components of the proposed PNI algorithm, namely neighborhood information ‘N’, position information ‘P’, and pixelwise refinement ‘R’. Each experiment is compared in terms of I-AUROC, P-AUROC, and P-F1-max (maximum F1-score at the pixel level).

N	P	R	I-AUROC			P-AUROC			P-F1-max		
			Object	Texture	Avg.	Object	Texture	Avg.	Object	Texture	Avg.
-	-	-	99.01	98.75	98.92	98.70	97.15	98.18	60.06	49.92	57.08
✓	-	-	99.38	99.55	99.44	98.79	98.29	98.62	67.91	58.54	64.79
✓	✓	-	99.46	99.46	99.46	99.04	98.33	98.80	69.30	58.43	65.67
✓	✓	✓	<b>99.55</b>	<b>99.59</b>	<b>99.56</b>	<b>99.12</b>	<b>98.72</b>	<b>98.98</b>	<b>69.90</b>	<b>61.23</b>	<b>67.01</b>

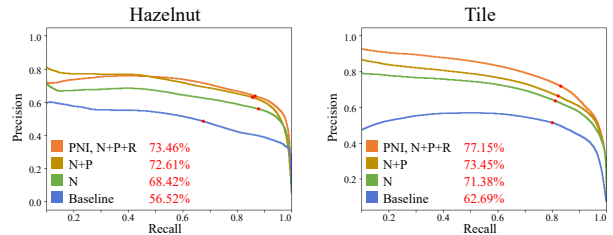


Figure 5: Precision-recall curve at the pixel-level for the hazelnut and tile subcategories of MVTEC AD. The proposed PNI algorithm and three ablation settings in Table 3 are compared. The F1-max scores for each setting are indicated on the right side of the legend.

**AUPRO** Table 2 compares the PNI with conventional algorithms, including the AUPRO metric. The performance of each algorithm is compared in terms of the average for object categories, texture categories, and overall. Detailed performance for all subcategories is discussed in the supplementary document. Some algorithms [2, 17, 21, 9] propose models that use multiple networks. At the bottom of Table 2, we compare those results with the ensemble network-based PNI.

PNI also shows superior performance over conventional algorithms in AUPRO. In overall AUPRO, PNI outperforms the second-best PFFM [25] by 0.53%, with a score of 96.05%. As mentioned in the previous analysis, our approach of using position and neighborhood information is more effective for object subcategories, and PNI surpasses the second-best Tsai *et al.* [24] by 0.64%. In the texture subcategories as well, PNI shows the second-best performance. Furthermore, PNI (Ensemble) exhibits the best performance in AUROC and AUPRO, object, and texture subcategories, surpassing the results of all conventional algorithms without exception.

**Ablation Study** We conducted an ablation study to verify the effect of using the three components of our proposed PNI algorithm: neighborhood information, position information, and pixelwise refinement. Table 3 shows the results,

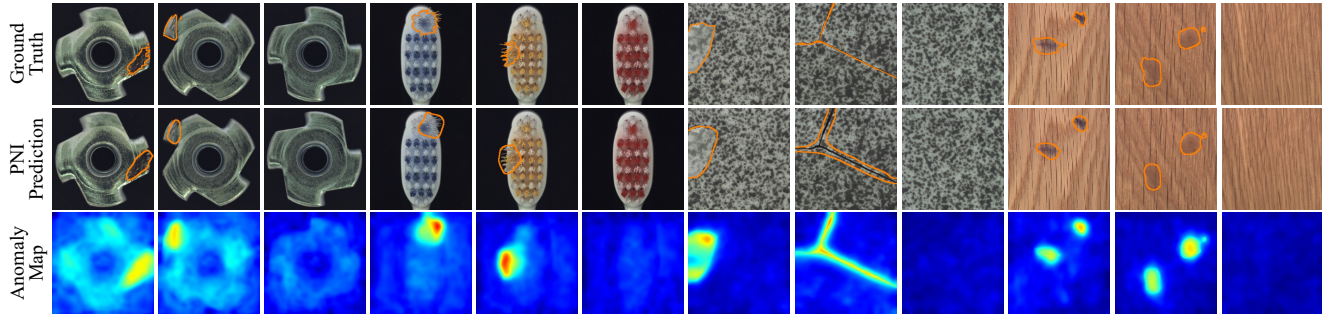


Figure 6: Visualization of anomaly localization results of PNI on the MVTec AD. Input images with ground-truth masks (top), predicted masks (mid), and predicted anomaly maps (bottom) are provided.

Table 4: Comparison of anomaly detection and localization results on MVTec AD [1]. The loss function of the refinement network is replaced with two alternatives  $\ell_{CE}$  and  $\ell_{focal}$ , and then compared.

	AUROC		AUPRO		
	Image	Pixel	Object	Texture	Average
Trained by $\ell_{CE}$	99.56	99.01	96.62	95.72	96.32
Trained by $\ell_{focal}$	99.61	99.03	96.69	96.11	96.50
<b>PNI (Ensemble)</b>	<b>99.63</b>	<b>99.06</b>	<b>96.83</b>	<b>96.00</b>	<b>96.55</b>

and the following observations can be made:

- The baseline without the three components is identical to PatchCore and performs similarly.
- Since the baseline deals with normal features unconditionally, the remaining models in the ablation study outperform the performance of the baseline.
- The use of neighborhood information improves overall performance significantly, enhancing the I-AUROC performance from 98.92% to 99.44%, which reduces the error by approximately 48.1%.
- Position information provides an additional gain in object subcategories in all three metrics, but little improvement is observed in texture subcategories. This result makes sense as the motivation for using position information is irrelevant to texture subcategories.
- Pixelwise refinement is more effective in texture subcategories and has complementary properties with position information.

Figure 5 compares the ablation settings using precision-recall curves and F1-max in two subcategories of MVTec: hazelnut and tile. Consistent with Table 3, significant improvements are observed for neighborhood information and position information in the object category, hazelnut. Also, in the texture category, tile, neighborhood information, and pixelwise refinement demonstrate better efficacy.

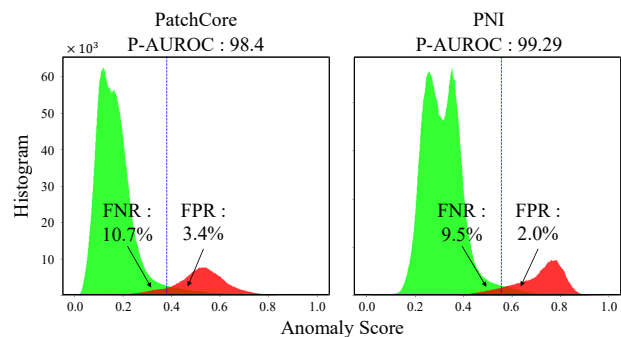


Figure 7: Histograms of anomaly scores for the metal nut subcategory in MVTec AD evaluated by PatchCore (left) and PNI (right) are shown. The green and red indicate the distribution of anomaly scores for normal and anomalous pixels, respectively. The blue vertical line indicates the threshold that maximizes the F1 score.

**Refinement network** Similarly with previous studies that adopted cross-entropy loss  $\ell_{CE}$  [17] or focal loss  $\ell_{focal}$  [34] for network training, we experimented with alternative settings by replacing the loss function of the refinement network. As shown in Table 4, the loss function composed of  $\ell_{reg}$  and  $\ell_{grad}$  outperforms the two alternatives. Since anomaly map  $\hat{A}$  is obtained based on the metric distance between representations, adopting regression-based losses is effective for refining it. On the other hand,  $\ell_{CE}$  and  $\ell_{focal}$  estimate the anomaly map in the form of binary segmentation, which leads to overconfident estimations.

**Qualitative Results** Figure 6 shows the results of anomaly localization performed by the proposed PNI algorithm on the test images of MVTec AD. The first row shows the ground-truth defect mask overlaid on the input image, and the second row shows the prediction results of PNI. We generate the prediction mask using the threshold that maximizes the F1 score. The last row visualizes the anomaly map predicted by PNI. The red parts in each im-



Table 5: Anomaly localization results on BTAD [19] as measured in P-AUROC [%].

Products	VT-ADL [19]	P-SVDD [29]	FastFlow [30]	Tsai <i>et al.</i> [24]	<b>PNI</b>
1	76.3	94.9	95	97.3	<b>97.4</b>
2	88.9	92.7	96	96.8	<b>97.0</b>
3	80.3	91.7	<b>99</b>	<b>99.0</b>	<b>99.0</b>
Average	81.8	93.1	97	97.7	<b>97.8</b>

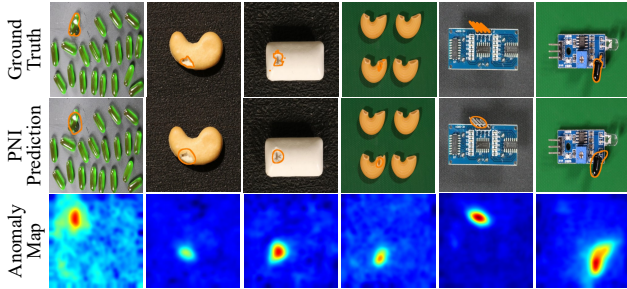


Figure 8: Visualization of anomaly localization results of PNI on the VisA. Input images with ground-truth masks (top), predicted masks (mid), and predicted anomaly maps (bottom) are provided.

age indicate high anomaly while the blue ones indicate low anomaly. Qualitative results show that the predicted mask generally follows the ground truth, leading to performance improvement.

**Analysis** Figure 7 compares the histograms of PatchCore and the proposed PNI algorithm for pixelwise anomaly scores in the metal nut subcategory. The green and red areas represent the distributions of anomaly scores for normal and abnormal pixels, respectively. Additionally, the blue vertical line indicates the threshold that optimizes the F1 score. The red area on the left of the threshold represents misclassified anomaly pixels or false negative pixels, while the green area on the right of the threshold represents false positive pixels. The FPR and FNR of PatchCore are 3.4% and 10.7%, respectively, which decrease to 2.0% and 9.5% in the PNI algorithm. These results correspond to the higher P-AUROC score of the proposed method.

Additionally, we computed image misclassification, false-positive and false-negative samples with the threshold optimizing F1 scores of anomaly detection. Out of the 467 normal test images and 1258 defective test images, a total of 7 false-positive and 12 false-negative detection errors were found, which is a significant improvement compared to 19 false-positive and 23 false-negative errors of PatchCore. We have provided detailed information on this in the supplementary document.

Table 6: Anomaly detection and localization AUROC scores on VisA [35] are presented. Sub-total averages are provided for single instance, multiple instances, and complex structure categories.

	I-AUROC				P-AUROC			
	Single instance	Multiple instances	Complex structure	Avg	Single instance	Multiple instances	Complex structure	Avg
SPD [35]	93.8	78.4	91.3	87.8	91.6	94.6	95.2	93.8
<b>PNI</b>	<b>99.2</b>	<b>87.9</b>	<b>98.5</b>	<b>95.2</b>	<b>98.1</b>	<b>99.2</b>	<b>99.0</b>	<b>98.8</b>

### 4.3. BTAD

The anomaly localization performance of the proposed PNI on the BTAD dataset is shown in Table 5. We compare the anomaly localization performance of PNI with conventional algorithms [19, 29, 30, 24]. As shown in Table 5, the proposed model outperforms other state-of-the-art algorithms in anomaly localization on all product categories in BTAD as well as the average score.

### 4.4. VisA

The anomaly detection and localization performance of the proposed PNI on the VisA dataset is shown in Table 6. The performance is compared with the conventional algorithm, SPD [35]. As shown in Table 6, the proposed model outperforms SPD in both anomaly detection and localization in all three types of categories of VisA as well as the average score. Figure 8 shows the results of anomaly localization performed by the proposed PNI algorithm on the test images of VisA. As Figure 6, each row represents the ground-truth defect mask, prediction results of PNI, and visualized anomaly map, respectively. Qualitative results show that the predicted mask generally follows the ground truth, leading to performance improvement.

## 5. Conclusion

We propose a new algorithm, PNI, for industrial anomaly detection and localization that accurately estimates the distribution of normal features by incorporating position and neighborhood information. PNI models position information using accumulated histograms from normal training images and uses a multi-layer perceptron network to model the normal feature distribution given neighborhood information. Additionally, PNI introduces a pixelwise refinement network using synthesized anomaly images to improve the anomaly map according to the input image, which is the first refinement approach in the field of industrial anomaly detection and localization as far as the authors know. Various experiments demonstrate the overall performance and effectiveness of the proposed PNI algorithm.

## References

- [1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTEC AD—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.
- [2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020.
- [3] Yunkang Cao, Qian Wan, Weiming Shen, and Liang Gao. Informative knowledge distillation for image anomaly segmentation. *Knowledge-Based Systems*, 248:108846, 2022.
- [4] Yunkang Cao, Xiaohao Xu, Zhaoe Liu, and Weiming Shen. Collaborative discrepancy optimization for reliable image anomaly localization. *IEEE Transactions on Industrial Informatics*, 2023.
- [5] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020.
- [6] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, pages 475–489. Springer, 2021.
- [7] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [9] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022.
- [10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019.
- [13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [14] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. Depth coefficients for depth completion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12438–12447. IEEE, 2019.
- [15] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [16] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 785–801. Springer, 2020.
- [17] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021.
- [18] Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Anomaly detection via self-organizing map. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 974–978. IEEE, 2021.
- [19] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Picciarelli, and Gian Luca Foresti. VT-ADL: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021.
- [20] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. In *Image Analysis and Processing—ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part II*, pages 394–406. Springer, 2022.
- [21] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022.
- [22] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 474–489. Springer, 2022.
- [23] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A Core-Set approach. *arXiv preprint arXiv:1708.00489*, 2017.
- [24] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. Multi-scale patch-based representation learning for image anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3992–4000, 2022.
- [25] Qian Wan, Yunkang Cao, Liang Gao, Weiming Shen, and Xinyu Li. Position encoding enhanced feature mapping for image anomaly detection. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 876–881. IEEE, 2022.
- [26] Qian Wan, Liang Gao, Xinyu Li, and Long Wen. Unsupervised image anomaly detection and segmentation based on

- pre-trained feature mapping. *IEEE Transactions on Industrial Informatics*, 2022.
- [27] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.
- [28] Minghui Yang, Peng Wu, and Hui Feng. MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities. *Engineering Applications of Artificial Intelligence*, 119:105835, 2023.
- [29] Jihun Yi and Sungroh Yoon. Patch SVDD: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [30] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. FastFlow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.
- [31] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [32] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRÆM - a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021.
- [33] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021.
- [34] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DSR—a dual subspace re-projection network for surface anomaly detection. In *European conference on computer vision*, pages 539–554. Springer, 2022.
- [35] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022.