

Article

FR-PatchCore: An Industrial Anomaly Detection Method for Improving Generalization

Zhiqian Jiang, Yu Zhang ^{*}, Yong Wang, Jinlong Li  and Xiaorong Gao

School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 611756, China; 2022201064@my.swjtu.edu.cn (Z.J.); wangyonga@swjtu.edu.cn (Y.W.); jinlong_lee@126.com (J.L.); gxrr@vip.163.com (X.G.)

^{*} Correspondence: zhangyu2@swjtu.edu.cn

Abstract: In recent years, a multitude of self-supervised anomaly detection algorithms have been proposed. Among them, PatchCore has emerged as one of the state-of-the-art methods on the widely used MVTec AD benchmark due to its efficient detection capabilities and cost-saving advantages in terms of labeled data. However, we have identified that the PatchCore similarity principal approach faces significant limitations in accurately locating anomalies when there are positional relationships between similar samples, such as rotation, flipping, or misaligned pixels. In real-world industrial scenarios, it is common for samples of the same class to be found in different positions. To address this challenge comprehensively, we introduce Feature-Level Registration PatchCore (FR-PatchCore), which serves as an extension of the PatchCore method. FR-PatchCore constructs a feature matrix that is extracted into the memory bank and continually updated using the optimal negative cosine similarity loss. Extensive evaluations conducted on the MVTec AD benchmark demonstrate that FR-PatchCore achieves an impressive image-level anomaly detection AUROC score of up to 98.81%. Additionally, we propose a novel method for computing the mask threshold that enables the model to scientifically determine the optimal threshold and accurately partition anomalous masks. Our results highlight not only the high generalizability but also substantial potential for industrial anomaly detection offered by FR-PatchCore.

Keywords: image anomaly detection; self-supervised learning; feature processing



Citation: Jiang, Z.; Zhang, Y.; Wang, Y.; Li, J.; Gao, X. FR-PatchCore: An Industrial Anomaly Detection Method for Improving Generalization. *Sensors* **2024**, *24*, 1368. <https://doi.org/10.3390/s24051368>

Academic Editors: Haitao Wang, Fei Fei and Yongkai Zhu

Received: 28 December 2023

Revised: 18 February 2024

Accepted: 19 February 2024

Published: 20 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The issue of data imbalance is widely observed in industrial anomaly detection, further exacerbated by the scarcity of valuable anomalous data available for this task. Anomalies can arise from various unknown external influences [1], and they can also originate from the objects themselves. It is impractical to account for every possible exception. Unsupervised approaches address this challenge by leveraging unlabeled samples to enable the model to learn the distribution properties and features of both normal and abnormal samples. This enables effective anomaly detection while avoiding the high cost associated with labelled data. In the field of industrial anomaly detection, training autoencoders [2,3] and methods based on generative adversarial networks (GANs) [4–7] are commonly employed. Lyu et al. [8] fused GAN with a deep convolutional neural network (DCNN) to build an anomaly location model, which can find the mapping relationship between images and high-dimensional features. Son et al. [9] computed anomaly scores by training encoder-decoder-based long short-term memory (LSTM) networks to evaluate data based on the time the anomaly existed. J. Mitra et al. [10] used generative adversarial networks to identify a manifold of normal samples and, at the same time, identify abnormal patterns that fall outside the model. Xiao, K., Cao, J., Zeng, Z., and Ling, W.-K. [11] explore sample distribution by utilizing the fast-labelling characteristics of a graph structure and use an autoencoder to blur low-level features while retaining local features to achieve better

anomaly detection performance. However, these methods often rely on the availability of a substantial amount of data. When the number of anomalous samples is limited, the model may struggle to distinguish between normal and anomalous images accurately.

Consequently, self-supervised approaches are highly favored in the field. Several state-of-the-art self-supervised methods, such as PatchCore [12], have achieved remarkable detection results on the widely recognized MVTec AD benchmark [13]. However, it has been observed that the PatchCore method imposes stringent requirements for image alignment. It is worth noting that in the MVTec dataset, most categories satisfy the condition of “similarity”, where samples within the same category share common characteristics and can be approximately aligned at identical positions. These samples do not exhibit significant spatial differences, such as rotation or flipping. Hence, they demonstrate excellent test results and are referred to as “similar categories”. In contrast, categories that fail to meet these similarity conditions are termed “spatial transformation categories”. The effectiveness of their anomaly detection may be compromised as we have observed that these methods are not proficient in handling the latter type of category.

To establish that this is not a random occurrence, we also conducted experiments on the MPDD dataset [14], whose six categories also have “spatial location relationships”. These experiments confirmed that the PatchCore method is more sensitive to “similar pictures” and does not pay much attention to the spatial position relationship between samples. In industrial detection, the images to be analyzed are not limited to “similar categories”. They can be captured by sensors from various angles and locations. Moreover, when training data are scarce, employing data augmentation techniques to expand the dataset (e.g., rotation, mirror flipping, left, and right flipping, etc.) can introduce “spatial transformation categories”, which may reduce the effect of anomaly detection, resulting in increased rates of missed detection and false positives.

To enhance the generalization performance of the method and enable it to effectively detect even in the presence of “spatial transformation categories”, we introduce registration as a pretext task. We extract fused features from the feature extraction layer and store them in a database through a series of dimensionality reduction operations. By utilizing a negative cosine similarity loss, we update the features in the memory bank, eliminating overlapping features while incorporating previously neglected ones during training. As the tests adopt a pixel-by-pixel comparison approach, disregarding pixel relationships, we construct a module similar to pyramid pooling to enforce connectivity between patches. Additionally, we devise an innovative approach that enables the model to look for the optimal threshold, resulting in accurate ground truth segmentation. Our proposed method achieves an impressive AUROC performance of 98.81% on the MVTec dataset, significantly enhancing generalization and reducing false positive rates.

2. Related Works

Recently, supervised methods have exerted a significant impact on the field of anomaly detection, resulting in the assimilation of concepts from these approaches into self-supervised methods. Self-supervised anomaly detection methods can be broadly classified into reconstruction-based techniques and feature embedding-based techniques.

2.1. Reconstruction-Based Approaches

Reconstruction-based anomaly detection relies on the assumption that the model can reconstruct normal samples well but struggles with anomalous regions. Early attempts involved using generative adversarial networks (GANs) [15,16]. However, due to the strong generalization of neural networks, abnormal samples could also achieve good reconstruction, rendering the comparison unreliable. More recently, You et al. [17] introduced a new paradigm for reconstruction networks, addressing the issue where popular reconstruction networks may fall into a “same shortcut” scenario, where both normal and abnormal samples can be accurately recovered, leading to difficulties in detecting outliers. This work

tackles the practical challenge of detecting anomalies across different object classes using a unified framework.

2.2. Feature Embedding-Based Approaches

The feature embedding-based approach involves feeding images into the model, extracting features, and constructing interface scoring rules in the feature space. Unlike the reconstruction method that operates in the RGB image space, feature embedding focuses on the high-dimensional feature space for anomaly detection. Notably, Bergman, L., Cohen, N. and Hoshen, Y. [18] encouraged the use of pre-trained models on large-scale datasets for anomaly detection. This recommendation gives great encouragement to such methods.

SPADE [19] initially extracts features of all normal samples (training set) in a network pre-trained on ImageNet. During testing, the K-nearest neighbors (KNN) [20] metric is applied to each pixel-wise feature of the test sample to calculate the anomaly score for each pixel point. However, the more normal images used in training, the more features are stored, resulting in higher KNN complexity during testing. Building upon SPADE, a method named PaDiM [21] improves it by eliminating the construction of a feature bank and performing KNN for anomaly detection. However, the pixels at each position are not strictly aligned, so modelling each pixel position alone may lead to inaccuracies. Y. Zheng et al. [22] performs fine-tuning based on the anomaly localization task and proposed FYD. To address the feature misalignment issue in PaDiM, they introduce the spatial transformer network (STN) [23] module for coarse alignment and use a self-supervised learning paradigm inspired by the SimSiam network (without requiring negative samples). However, certain images may still not be aligned. Furthermore, from a visualization perspective, the detection performance is not optimal, primarily due to the evaluation metric (pixel-wise AUCROC score [24]), which tolerates anomalies in small areas.

Recently, Huang et al. [25] innovated based on PaDiM and FYD and proposed RegAD, which employs registration as a proxy task to train an anomaly detection model with unknown categories. PatchCore addresses the slow testing speed of SPADE. KNN and greedy coresets subsampling are applied to select the most representative feature points for measuring anomaly scores during testing, which reduce the size of the feature bank and finally achieve excellent results.

The method FR-PatchCore (Feature-level Registration PatchCore) of ours preserves the high efficiency of PatchCore. Feature-level registration is employed as a pretext task, and the information stored in the feature database is updated using a negative cosine similarity loss. To enforce pixel relationships, a module similar to pyramid pooling is constructed. During testing, the Euclidean distance from each pixel to its corresponding feature in the database is used for scoring.

3. Methodology

In our methodology, we introduce the concept of feature-level registration and utilize it as a non-cumbersome pretext task during training. The negative cosine similarity loss is employed as the training loss, which we optimize to update the memory bank. The framework of our approach is depicted in Figure 1.

The upper section represents the registration module, while the lower section represents the memory bank module. Both sections operate concurrently. Two images from samples belonging to the same category are randomly selected, and their features are extracted using the convolutional neural network and spatial transformer network (CNN + STN) for registration purposes. The feature registration process is supervised by maximizing the absolute value of the negative cosine similarity loss. After the registration process, the extracted fusion features are stored in a dedicated memory bank and continuously updated with training losses after each registration iteration.

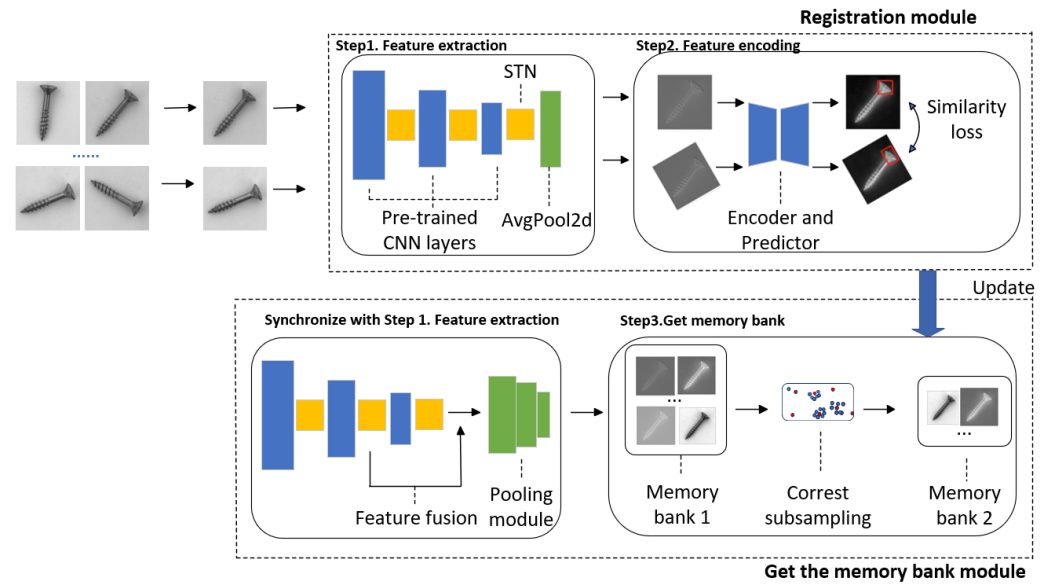


Figure 1. The architecture of the FR-PatchCore model proposed is presented. The entire framework can be divided into three main steps, and the feature extraction processes in both modules are executed simultaneously. In the registration module, randomly selected images are fed into CNN + STN to obtain pre-encoded features. Meanwhile, in the memory bank module, we combine the features passing through the second STN layer and the third STN layer to retain sufficient spatial information of features. Finally, the fusion features are stored in the memory bank.

3.1. Registration Module

Neural network training necessitates task-driven learning. Therefore, the essence of self-supervised learning lies in the thoughtful design of tasks that facilitate effective model learning. Inspired by the work of [25], which obtained a Gaussian distribution model of normal data through feature-level registration training, we leverage the registration task as a pretext task to enhance the model's understanding of features and emphasize spatial and positional differences. Accordingly, we construct the registration module, consisting of a feature extractor, feature encoder, and predictor, as illustrated in Figure 2.

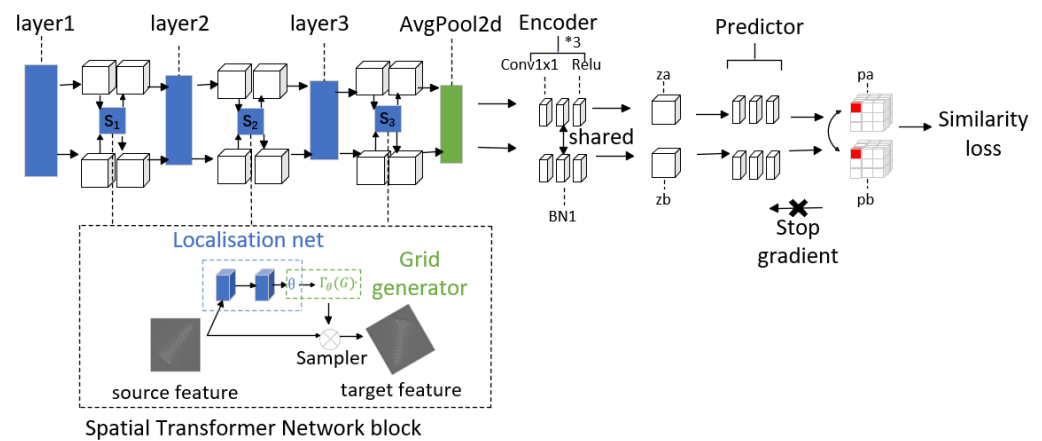


Figure 2. Registration module: two different images of the same category are inputted, and the CNN + STN combination is utilized to extract features. A Siamese structure network is employed for feature encoding and prediction. The objective is to train the registration network in such a way that the predicted features closely resemble the encoded features in the figure.

In feature registration, since spatial transformation can be represented as matrix operations, it is advantageous to allow the network to learn the generation of matrix parameters, thereby acquiring spatial transformation capabilities. Common network frameworks in

deep learning include the CNN and the transformer. Additionally, the spatial transformer network, which plays a pivotal role in our research, can seamlessly integrate into any component of the CNN architecture. To ensure a fair comparison with state-of-the-art methods, we have selected the wide_resnet_50_2 [26] network as the backbone for our experiments among various CNNs commonly employed in anomaly detection tasks. This network has demonstrated exceptional performance on the ImageNet dataset, achieving an accuracy of 78.51% for Top 1 and 94.09% for Top 5. For this specific task addressed in our paper, we conducted an ablation experiment (4.5.1) to compare the feature extraction capabilities of both Resnet and ViT models, where wide_resnet_50_2 exhibited superior performance.

We incorporated the STN module into the wide_resnet_50_2 architecture. The overall structure of the STN module is illustrated in Figure 2 and consists of a localization network, a grid generator, and a sampler. In the first component, a feature map is given as Formula (1):

$$U \in \mathbb{R}^{H \times W \times C} \quad (1)$$

After several convolutions or fully connected layers, a regression layer follows, leading to the output of regression transformation parameter θ . The dimension of θ depends on the specific transformation type chosen by the network.

In the second component, we utilize θ and the specified transformation mode from the localization network output, and the grid generator performs further spatial transformations of the feature map to determine the mapping of $T(\theta)$ between the output and input features. It employs the predicted transformation parameters to create a sampling grid, which represents a set of points where the input map should be sampled to produce the transformed output. In the sampler, it utilizes the sampling grid to determine which points in the input feature map will be utilized for the transformation. It samples the input feature map against the sampling grid to obtain the final output.

In Step 1, assuming that the input RGB image has a resolution of (224,224), Formula (2) is applied.

$$U \in \mathbb{R}^{224 \times 224 \times 3} \quad (2)$$

The fourth layer of the wide_resnet_50_2 network is excluded to preserve more comprehensive spatial information, and the spatial transformer network (STN) is integrated after the initial three layers of the network. The input feature $f_i(x_i^s, y_i^s)$ undergoes a transformation function τ_θ . The mapping relationship between the input and output feature mappings is defined as Formula (3).

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \tau_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \quad (3)$$

In this case, the form of the eigenvector is shown in Formula (4):

$$\begin{cases} U_1 \in \mathbb{R}^{56 \times 56 \times 64} \\ U_2 \in \mathbb{R}^{28 \times 28 \times 128} \\ U_3 \in \mathbb{R}^{14 \times 14 \times 256} \end{cases} \quad (4)$$

When no key points are labeled, the STN allows the neural network to actively transform the feature map based on input features and learn spatial transformation parameters without requiring additional training supervision or modifications to the optimization process. As illustrated in Figure 2, the STN can effectively align input images or learned features during training, thereby mitigating the impact of spatial geometric transformations such as rotation, translation, scale, and distortion on tasks like classification and localization. The STN facilitates the spatial transformation of the input data, thereby enhancing feature classification and enabling the network to achieve rotational invariance dynamically. Moreover, it intelligently selects the most salient region of the image and optimally transforms it into a suitable orientation. Figure 2 illustrates the depiction of feeding an inverted screw

image into the STN module. Through a series of transformations, the input is effectively rectified to face forward.

The employed approach in Figure 2 involves the utilization of a Siamese network for feature encoding, wherein a negative cosine similarity loss is applied as per Formulas (5) and (6).

$$D(p_a, z_b) = -\frac{p_a}{\|p_a\|_2} \cdot \frac{z_b}{\|z_b\|_2} \quad (5)$$

$$D(p_b, z_a) = -\frac{p_b}{\|p_b\|_2} \cdot \frac{z_a}{\|z_a\|_2} \quad (6)$$

The negative cosine similarity loss is an appropriate metric for quantifying the similarity between two vectors. Furthermore, it possesses the capability to map similar vectors to adjacent points and dissimilar vectors to distant points. This characteristic facilitates feature clustering, as described in Section 3.2.3.

The objective is to maximize the similarity between p_a and z_b , as well as between p_b and z_a . To prevent the input data from converging to a constant value after convolution activation, resulting in identical outputs regardless of the input image, we adopt the approach described in [27] by halting the gradient operation on one of the branches to avoid model collapse. Finally, we define Formula (7) as the registration loss for symmetric features.

$$\mathcal{L} = \frac{1}{2}[(\mathcal{D}(p_a, z_b) + (\mathcal{D}(p_b, z_a))] \quad (7)$$

The STN is employed in this stage to perform feature rotation and inversion, facilitating the model's determination of image similarity. Following each training iteration, a negative cosine similarity loss is obtained.

3.2. Memory Bank Acquisition

3.2.1. Feature Extraction

Let $\varphi(h, w, c)$ represent the feature mapping of the second layer, where h denotes the height, w denotes the channel width, and c denotes the number of channels. Like PatchCore, the patch-level features of local features in clustered neighborhoods can be represented as Formula (8):

$$P = f_{agg}(\varphi) \quad (8)$$

Here, f_{agg} represents the aggregation function within the neighborhood. As mentioned in Figure 1, we use a combination of the first three layers of wide_resnet_50_2 and STN to extract features and build the memory library, with each layer followed by the STN. Inspired by the PatchCore method, we likewise did not adopt the last layer of the Resnet network since it lost a lot of the features' spatial information. As shown in Figure 3, the features of the second and third layers of the STN can retain global information while containing more local feature information. However, if the features of the first three or more layers are fused, the features in the input memory library will not contain enough information for accurate detection.

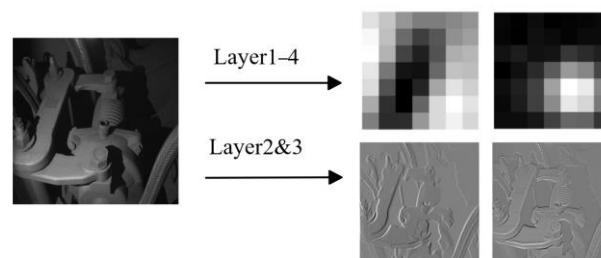


Figure 3. A visualization of features of different levels when fused. When too many levels of the feature are fused, the naked eye can no longer distinguish the outline of the original image, and the features become fine-grained.

3.2.2. Similar to Pyramid Pooling Module (SPPM)

After feature extraction, a 3-dimensional tensor with the shape (C, H, W) is obtained, where C is the sum of the dimensions of the second and third layers. We then process this three-dimensional feature vector by flattening it in three dimensions except for the channels. This results in a two-dimensional tensor with a shape of $(H \times W, C)$, which is randomly projected. Thus, we flatten the original feature vector into a column-wise phenotypic feature matrix. However, as mentioned earlier, the PatchCore method employs a pixel-by-pixel search approach, conducting nearest neighbor searches on each pixel and disregarding pixel relationships. We recognize that 2D average pooling can increase the receptive field, which is crucial for anomaly detection tasks. Hence, we employ the pooling approach illustrated in Figure 4.

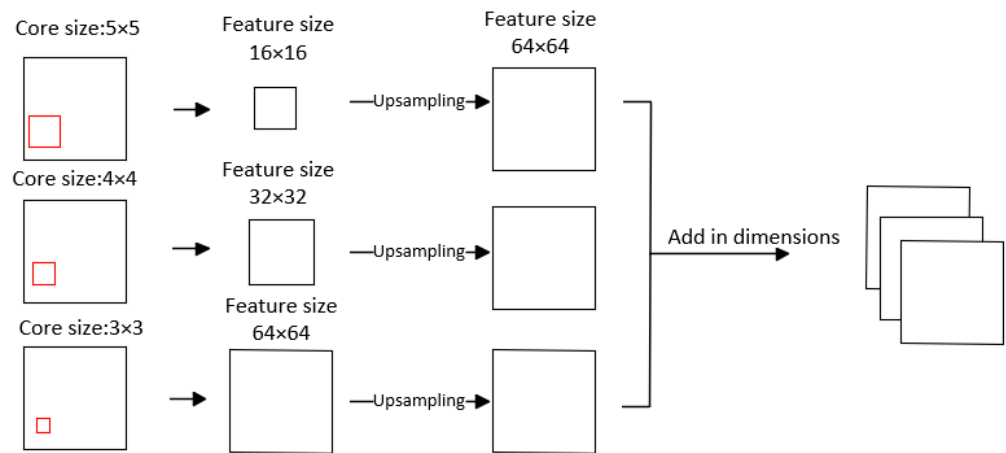


Figure 4. The structure of SPPM.

Assuming an input feature size of 64×64 , we perform three pooling operations with pool core sizes of 3×3 , 4×4 , and 5×5 . Subsequently, tensors of sizes 64×64 , 32×32 , and 16×16 are obtained. Next, the three pooled tensors are unsampled to the same size of 64×64 , and their dimensions are concatenated. The earlier pooling regions are included within subsequent pooling regions using this approach to enlarge the receptive field and give more attention to edge information due to an overlap between pooling regions, thus establishing closer relationships between the pixels.

After the pooling step, the eigenvector of the domain $\varphi(h, w)$, as represented by Formula (9), is enacted.

$$N_p^{(h,w)} = \left\{ (a, b) \mid a \in \left[h - \frac{p}{2}, \dots, h + \frac{p}{2} \right], b \in \left[w - \frac{p}{2}, \dots, w + \frac{p}{2} \right] \right\} \quad (9)$$

The feature aggregation operation is utilized to obtain the locally aware patch feature set p of the feature mapping tensor $\varphi_{i,j}$, enabling the successful realization of the clustering $P_{s,p}(\varphi(h, w))$ of the feature tensor, as shown in Formula (10).

$$P_{s,p}(\varphi(h, w)) = \left\{ \varphi(h, w) \left(N_p^{(h,w)} \right) \mid h, w \bmod s = 0, h < h^*, w < w^*, h, w \in N \right\} \quad (10)$$

In this case, the feature repository M_1 can be described as Formula (11):

$$M_1 = \bigcup_{xi \in N} P_{s,p}(\varphi(xi)) \quad (11)$$

3.2.3. Anomaly Detection

The samples used in self-supervised learning are exclusively normal. The training process aims to identify representative features of the “normal category” and utilize them

as a reference for evaluating positivity and negativity, ultimately achieving anomaly detection. During testing, we index the memory bank that stores characteristic information of positive samples and calculate the Euclidean distance between sample patches to obtain an anomaly score. In n -dimensional space, if there are two points $a(x_{11}, x_{12}, \dots, x_{1n})$ and $b(x_{21}, x_{22}, \dots, x_{2n})$, then the Euclidean distance d is defined as Formula (12).

$$d = \sqrt{\sum_{k=1}^n (x_{1k} - x_{2k})^2} \quad (12)$$

After completing the feature aggregation and pooling module in Step 2, we proceed to the feature clustering operation. The utilization of negative cosine similarity loss during training greatly facilitates feature clustering by ensuring that similar vectors are consistently mapped to proximate locations with each iteration, while dissimilar vectors are assigned to distant points. This approach effectively reduces the clustering time and enhances the operational efficiency of the model. To optimize the memory library size and improve the testing efficiency, we implement the PatchCore method and employ greedy subsampling for reducing and optimizing the memory bank based on Formula (13).

$$M_2^* = \frac{\operatorname{argmin}}{M_2 \subset M_1} \frac{\max}{m \in M_1} \frac{\min}{n \in M_2} \|m - n\|_2 \quad (13)$$

The objective of our purpose is to streamline the testing process by solely conducting nearest neighbor searches for the test sample's feature in M_2 , identifying its closest neighboring feature and subsequently calculating the maximum Euclidean distance from this feature to its clustering center. This approach allows us to obtain an anomaly score, facilitating effective anomaly detection.

In the memory bank, for each patch-level feature $m \in M_2$ of the training data, we select the m^* nearest neighbors from the patch-level features of the test data. The patch-level anomaly score of the test image X_{test} is estimated based on the distance between the patch-level feature m_{test} and m^* , as shown in Formulas (14) and (15).

$$m^* = \operatorname{argmin}_{m \in M_2} \|m^{test} - m\|_2 \quad (14)$$

$$s^* = \|m^{test} - m^*\|_2 \quad (15)$$

3.2.4. Post-Processing Method

In anomaly detection, the ground truth serves as a baseline measurement obtained from a reliable method. It is used to calibrate and improve the accuracy of new measurement methods. Many current anomaly detection methods rely on the calculated AUROC curve to determine the threshold of the entire dataset. This involves setting the threshold based on the data distribution and other characteristics. The regions above the threshold are considered abnormal, while those below are considered normal. However, if the overall threshold of an abnormal image is lower than the dataset threshold, the anomaly cannot be detected. Otherwise, false detections may occur. To address this, we propose a threshold determination method based on the image itself, which identifies the critical point between normal and anomalous regions, enabling the accurate labelling and localization of anomalies.

For each test sample, as shown in Figure 5a, we artificially construct five rectangular sampling frames in its upper left, lower left, upper right, lower right, and right center regions, as shown in Figure 5b.

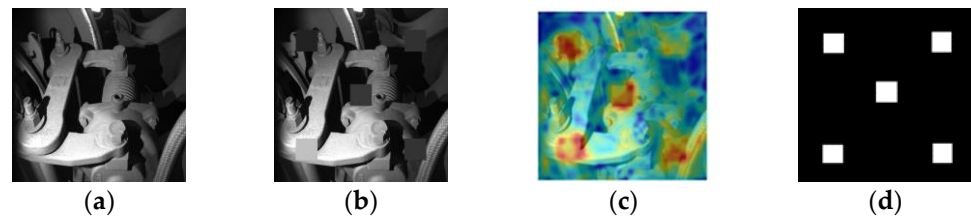


Figure 5. (a) source image; (b) artificial construction; (c) heat map; (d) ground truth of the constructed image.

The constructed image is also fed into the network for feature extraction, and then we search for distances in the feature database, compute scores, and obtain the score matrix. Its heat map visualization is shown in Figure 5c. Obviously, as artificially constructed anomalies, these five areas will have significantly higher anomaly scores than the rest of the picture. We set the pixel values of these five areas to 0 and the other parts to 1, thus obtaining its ground truth, as depicted in Figure 5d. Then, we traverse the five regions, find the maximum and minimum fractional values ($thres_{max}$ and $thres_{min}$) and define the *interval* as outlined in Formula (16), setting *iter_times* to 10 to ensure the code runs efficiently:

$$interval = (thres_{max} - thres_{min}) / iter_times \quad (16)$$

For each iteration, we set the threshold as outlined in Formula (17):

$$th = thres_{max} - iter_times \cdot thres_{min} \quad (17)$$

The iteration is carried out in the range of minimum and maximum thresholds, and each *th* can output an anomaly mask accordingly, where the white area is abnormal and the black area is normal, as shown in Figure 6a–e, which are the masks corresponding to the thresholds 9.0, 8.3, 7.6, 6.4 and 6.1, respectively. As can be seen from the five pictures shown in Figure 6, with the continuous increase in the threshold, the area of the abnormal region in the mask image corresponding to each threshold value increases first and then decreases. Finally, we calculate the intersection over union (IOU) between the prediction box and the ground truth box. The threshold is updated according to the highest IOU, and the best threshold (*best_th*) is obtained after 10 iterations. The mask graphs corresponding to different responsiveness are presented in Figure 6a–e, serving the purpose of facilitating the description of this method's principle. It is important to note that these graphs do not represent the final segmentation results but rather serve as temporary variables within the code and are subsequently deleted after calculation to optimize memory usage.

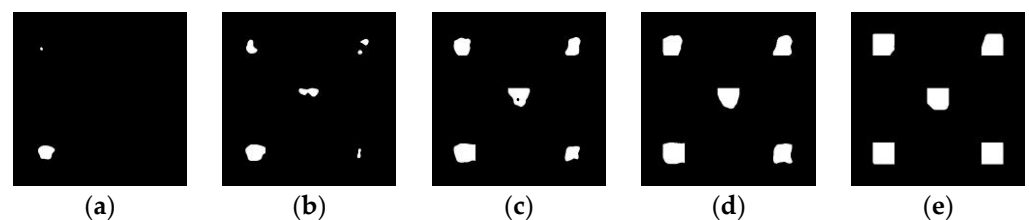


Figure 6. The corresponding anomaly mask under a certain *th*. (a) $th = 9$; (b) $th = 8.3$; (c) $th = 7.6$; (d) $th = 6.4$; (e) $th = 6.1$.

After determining the optimal threshold value (*best_th*), based on industrial detection expertise, we establish the region of abnormal pixels ($S = 25$) and devise an algorithm. For the neighboring pixel region exceeding *best_th*, if it is smaller than *S*, the pixel value in this region will be set to 0 (normal); otherwise, it will be set to 1 (abnormal). The implementation of this setting will effectively eliminate the occurrence of false positives in small areas, resulting in a reduction in the overall false alarm rate observed in the test results.

4. Experiment

4.1. Datasets and Metrics

We first evaluate the performance of our method on two public datasets, MVTec and MPDD, which cover both classical industrial anomaly detection and defect detection in manufacturing processes.

- MVTec dataset: The MVTec dataset is a well-known benchmark for industrial anomaly detection. It consists of 15 categories with a total of 3629 training and validation images and 1725 test images. The training set contains only normal images, while the test set includes images with various defects and normal images. The images have resolutions ranging from 700×700 to 1024×1024 pixels, and pixel-level ground truth labels are provided for each defect.
- MPDD dataset: The MPDD dataset is a newly proposed dataset focused on defect detection in the manufacturing process of painted metal parts. It comprises six types of metal parts captured under different spatial orientations, positions, distances, light intensities, and backgrounds. We utilize this dataset to evaluate the registration effect of our method and compare it with several other methods to highlight the superiority of our approach.
- Industrial scenario dataset: To demonstrate the robustness and generalization of our approach, we have selected an additional three datasets from the field of high-speed rail (HSR) component inspection for experimentation. Each of these industrial datasets consists of 460 normal samples and 115 abnormal samples, all of which exhibit visual bias in terms of their positioning. Figure 7 showcases the appearance of these normal samples.

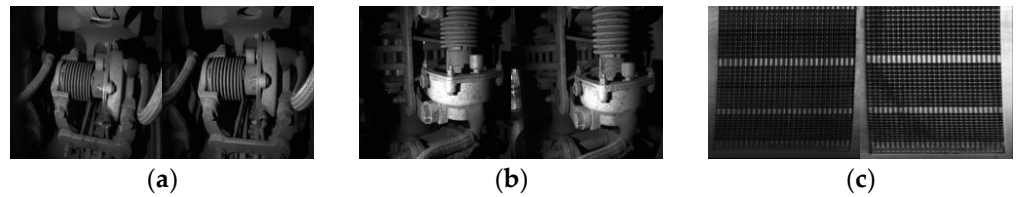


Figure 7. (a) HSR dataset 1; (b) HSR dataset 2; (c) HSR dataset 3.

We first adopt the traditional model evaluation method, randomly sampling normal data and abnormal data, and divide the dataset according to the ratio of training set to test set of 8:2. In order to avoid overfitting and other effects caused by unreasonable dataset division, we adopt the cross-validation method to evaluate the model in Section 4.4.

For the evaluation, we employ two threshold-independent metrics. The first is the area under the receiver operating characteristic curve (AUROC), which measures the true positive rate (percentage of correctly classified anomalous pixels). The image-level AUROC scores assess the model's detection accuracy, denoted as detection (Det.), while the pixel-level AUROC scores evaluate the model's positioning and segmentation accuracy, indicated as segmentation (Seg.). The second metric is the per-region-overlap score (PRO-Score) [28], which considers the mean rate of correctly classified pixels as a function of the false positive rate between 0 and 0.3 for each connected component, as shown in Formula (18). Higher scores indicate better localization of both large and small anomalies [29].

$$PRO = \frac{1}{N} \sum_{n=1}^N \frac{P \cap G_n}{G_n} = \frac{1}{N} \sum_{n=1}^N \frac{TP_n}{TP_n + FN_n} \quad (18)$$

where P represents the prediction result, G_n represents the true value, then the intersection of the two is the true positive sample TP_n , and FN_n represents the false negative sample.

The above two indicators are commonly employed in anomaly detection tasks and are widely acknowledged and utilized as benchmarks within the detection industry. However, due to the intricacy of industrial detection scenarios, there exists variation in data quality as well as resolution. To demonstrate the robustness of our method, we also employ three

additional indicators, namely the recall rate, accuracy rate, and false alarm rate (*FAR*), for testing and cross-validation using the high-speed railway dataset. The calculation formulas for these indicators are presented in Formulas (19)–(21), respectively.

$$Recall = \frac{TP}{TP + FN} \quad (19)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

$$FAR = \frac{FP}{FP + TN} \quad (21)$$

The term *FP* denotes the number of false positives, while *TN* represents the count of true negatives.

4.2. Implementation Details

We utilize wide_resnet_50_2 and the *STN* as the underlying architecture to construct a convolution-based encoder and predictor. These models are trained on images with a resolution of 224×224 using an Asus NVIDIA GTX 3090 GPU sourced from Chengdu, China. For parameter updates, we employ momentum SGD for 55 epochs with a learning rate of 0.0001 and a batch size of 32. The duration of each training round is approximately 8 min, with an average training epoch lasting for about 8.7 s.

4.3. Anomaly Detection

To assess the efficacy of FR-PatchCore, we conduct a comparative analysis with other cutting-edge anomaly detection techniques, such as SPADE, PaDiM, Patch-SVDD [30], Cutpaste [31], RegAD, and PatchCore. We meticulously replicate their reported outcomes using each method's official source code to ensure an equitable comparison. The last row displays the macro-averaged scores for all categories, highlighting the highest scores in bold.

The MVTec dataset is the most widely used benchmark for anomaly detection, and FR-PatchCore achieves a 98.81% image-level AUROC score and a 97.86% pixel-wise AUROC score when compared with the other six methods in Table 1. In the comparison of detection results of “similar categories” such as bottle, leather, cable, etc., our method is slightly inferior to PatchCore, but it still has high detection scores and accuracy, and the average detection scores of all categories are better than the other six methods. For the three “spatial transformation classes” such as hazelnut, screw, and metal nut in the dataset, we present partial representative results of their visualizations in Section 4.6.

To evaluate the robustness of our method against the “rotation class” data that we defined, we conducted experiments on the MPDD dataset, which are displayed in Table 2, from ten runs. The final row presents the macro-averaged scores across all categories, with the best scores indicated in bold.

We present partial representative results of the MPDD dataset visualizations in Section 4.5. Unlike the MVTec dataset, all six categories in the MPDD dataset have spatial differences and position shifts. This is challenging for most methods because anomaly detection models are not trained enough to deal with morphologically diverse samples. We believe that increasing the training time is not enough to make the model learn more. On the contrary, it may cause the model to overfit. Most models cannot detect the “spatial transformation class” of the sample effectually. The FR-PatchCore proposed by us is innovative in model learning. As shown in Table 2, our method leads in comparison with the other six methods. This proves that the idea of using feature-level registration as a pretext task is worthwhile. Only when the model learns the spatial information that has been ignored before can it better deal with such problems, improve the generalization and be applied to the field of high-standard industrial anomaly detection.

Table 1. The results on the MVTec dataset (AUROC—image%, AUROC—pixel%).

Category	SPADE		PaDiM		SVDD		Cutpaste		RegAD		PatchCore		Ours	
	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
carpet	92.80	97.50	99.40	98.90	92.90	92.60	93.10	98.30	98.50	98.90	98.00	98.90	97.13	98.35
grid	47.30	93.70	95.70	94.90	94.60	96.20	99.90	97.50	91.50	88.70	97.50	97.50	98.65	95.14
leather	95.40	97.60	99.80	99.10	90.90	97.40	100.0	99.50	100.0	98.90	100.0	99.10	99.84	98.84
tile	96.50	87.40	97.40	91.20	97.80	91.40	93.40	90.50	97.40	95.20	99.40	94.90	98.46	97.67
wood	95.80	88.50	98.80	93.60	96.50	90.80	98.60	95.50	99.40	94.60	98.90	93.60	98.73	95.45
bottle	97.20	98.40	99.80	98.10	98.60	98.10	98.30	97.60	99.80	97.50	100.0	98.30	99.76	98.39
cable	84.80	97.20	92.20	95.80	90.30	96.80	80.60	90.00	80.60	94.90	99.30	98.90	99.78	98.17
capsule	89.70	99.00	91.50	98.30	76.70	95.80	96.20	97.40	76.30	98.20	97.60	98.50	98.83	98.82
hazelnut	88.10	99.10	93.30	97.70	92.00	97.50	97.30	97.30	96.50	98.50	100.0	98.40	99.59	97.96
metal nut	71.00	98.10	95.80	96.70	94.00	98.00	99.30	93.10	98.30	96.90	99.70	97.70	99.67	98.77
pill	80.10	96.50	94.40	94.70	86.10	95.10	92.40	95.70	80.60	97.80	95.90	97.70	98.82	98.21
screw	66.70	98.90	84.40	97.40	81.30	95.70	86.30	96.70	63.40	97.10	94.90	98.60	96.86	98.50
toothbrush	88.90	97.90	97.20	98.70	100.0	98.10	98.30	98.10	98.50	98.70	100.0	97.20	98.06	98.63
transistor	90.30	94.10	97.50	97.20	91.50	97.00	95.50	93.00	93.40	96.80	99.90	94.90	98.96	97.26
zipper	96.60	96.50	90.80	98.20	97.90	95.10	99.40	99.30	94.00	97.40	99.20	98.40	99.58	97.79
average	85.41	96.03	95.20	96.70	92.07	95.71	95.19	95.97	91.21	96.67	98.69	97.51	98.81	97.86

Table 2. The results on the MPDD dataset (AUROC—image%, AUROC—pixel%).

Category	SPADE		PaDiM		SVDD		Cutpaste		RegAD		PatchCore		Ours	
	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
bracket black	62.80	97.70	75.60	94.20	60.17	89.46	77.50	85.20	67.30	72.58	68.15	90.55	85.98	96.74
bracket brown	88.00	98.50	85.40	95.20	64.56	89.69	93.40	93.80	69.60	93.25	74.96	86.46	83.50	95.40
bracket white	83.60	97.00	82.20	98.10	50.89	89.48	74.60	82.70	61.40	95.68	72.56	98.31	88.25	99.50
connector	90.70	98.60	91.70	97.90	94.52	90.98	97.60	96.80	94.90	92.16	80.95	96.74	99.95	99.11
metal plate	91.70	97.40	56.30	93.40	92.80	92.70	88.00	92.50	90.20	89.26	89.98	96.30	98.48	98.22
tubes	51.00	97.50	57.50	92.10	22.15	85.81	98.20	97.40	67.90	92.27	69.07	94.11	81.25	98.85
average	77.97	97.78	74.78	95.15	64.18	89.69	88.22	91.40	71.90	89.20	75.95	93.75	89.57	97.97

To evaluate the applicability of our method in real-world scenarios, we conducted experiments using data from high-speed rail (HSR) components in industrial anomaly detection. The scores are displayed from ten runs in Table 3.

Table 3. The results on the HSR dataset (AUROC—image%, AUROC—pixel%).

Category	SPADE		PaDiM		SVDD		Cutpaste		RegAD(k = 8)		PatchCore		Ours	
	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
1	98.78	90.70	98.67	93.70	95.46	93.25	78.70	85.27	92.25	90.75	98.28	97.25	99.32	96.58
2	97.25	95.26	97.68	97.25	98.56	98.25	99.25	98.75	94.87	95.76	98.26	99.27	98.86	98.86
3	95.45	94.28	96.87	97.36	94.28	96.28	98.36	97.16	98.23	96.14	97.52	97.25	99.28	99.38
average	97.16	93.41	97.74	96.10	96.10	95.93	92.10	93.73	95.12	94.22	98.02	97.92	99.15	98.27

We present partial representative results of the HSR dataset visualizations in Section 4.5. As can be seen from Table 3, FR-PatchCore still achieves an excellent detection performance even when detecting real industrial datasets with a complex environment, noisy images, and many influencing factors, which proves that our method has good generalization and robustness.

Most anomaly detection researchers evaluate the performance of their methods using both image-level and pixel-wise AUROC curves. AUROC is a widely used metric for assessing the ability of models to discriminate between abnormal and normal samples.

However, in anomaly detection, it is not only important to identify anomalies but also to accurately localize them. The pixel-wise AUROC evaluation provides a broad assessment, where the score is significantly improved if a large region is correctly localized but has a minimal impact if a small region is incorrectly positioned.

To address this limitation, we also employed the PRO score [24] criteria, which considers the rate of correctly classified pixels as a function of the false positive rate (fpr). We calculated the average PRO score of the three datasets using an fpr of 30%, following the methodology described in [25]. In Table 4, we compare our method with SPADE, PaDiM, and PatchCore, which are all based on feature embedding. Our approach achieves the highest PRO-Score, indicating a superior localization performance.

Table 4. Average PRO scores on three datasets.

Datasets	SPADE	PADIM	PatchCore	Ours
MVTec	91.70	90.10	90.17	92.90
MPDD	91.20	84.70	93.20	94.90
HSR 1–3	90.70	94.81	94.72	95.81
Mean score	91.20	89.87	92.69	94.54

In the comparison of the PRO score, we only compare four methods based on feature embedding: SPADE, PaDiM, PatchCore and our proposed method. Since these four methods have the same basic principle, they are easier to compare, and it is more convenient to use the principle of control variables. For example, the same backbone wide_resnet_50_2 is used at the same time. As shown in Figure 8, we report the results of the PRO score comparison on three datasets of different difficulty, which indicates that FR-PatchCore has an excellent detection performance and still obtains good indicators under this more stringent score limitation.

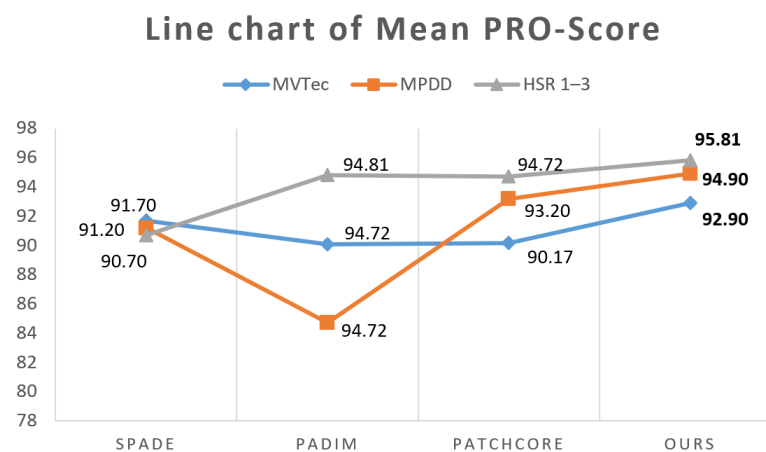


Figure 8. Average PRO score line chart.

In addition, as presented in Tables 5–7, we have documented the scores for the accuracy rate, recall rate, and false positive rate of the three high-speed rail datasets. Among the four comparative methods employed, our approach stands out as the most advanced, with the lowest occurrence of false alarms. Furthermore, apart from these metrics, industrial tests also assess efficient reasoning performance. By employing identical parameter settings across all four methods, we conducted comprehensive evaluations. The inference time considered here encompasses both inputting samples to be detected into the network for forward propagation and the processing time. The original image resolution was adjusted from 1280×1280 to a 224×224 input network size. Our method's inference speed remains comparable to that of well-known PatchCore networks while maintaining superior anomaly detection and segmentation performance at the image-level.

Table 5. Industrial inspection score on HSR dataset 1.

Metrics	SPADE	PaDiM	PatchCore	Ours
Recall%	88.00	96.00	94.00	96.00
Accuracy%	87.80	97.60	97.92	97.39
FAR%	12.00	4.00	3.52	4.00
TIMES(s)	0.37	0.35	0.20	0.22

Table 6. Industrial inspection score on HSR dataset 2.

Metrics	SPADE	PaDiM	PatchCore	Ours
Recall%	84.00	94.00	96.00	98.00
Accuracy%	82.60	93.04	92.17	93.91
FAR%	16.00	6.00	4.00	2.00
TIMES(s)	0.42	0.34	0.33	0.30

Table 7. Industrial inspection score on HSR dataset 3.

Metrics	SPADE	PaDiM	PatchCore	Ours
Recall%	86.00	92.00	96.41	98.00
Accuracy%	86.09	92.17	97.92	98.45
FAR%	14.00	0.08	4.00	2.00
TIMES(s)	0.31	0.26	0.21	0.21

4.4. Cross-Validation Experiment

To assess the performance of our model on limited data samples, we conducted five-fold cross-validation using the high-speed rail component dataset. The training set was divided into five folds, with four folds used for training and the remaining one for testing. Additionally, within the four-fold training data, an additional 5% of the data was extracted as a verification set. Each training and test iteration involved 442 training data, 18 validation data, and 115 test data. This approach ensures that all samples in the training set are utilized for both training and testing purposes.

Since this study is based on self-supervised learning and utilizes unlabeled positive samples for training, there is no actual labeled data available. However, it should be noted that anomaly detection inherently involves a classification task where the “normal” label is carried by the data itself during training. Therefore, in each of the five cross-validations performed, classification accuracy serves as our evaluation metric. Subsequently, we select the model with highest classification accuracy on the verification set to evaluate its performance on separate test sets through five experiments. The resulting test errors from each experiment can be obtained using Formula (22).

$$E_T(\omega) = \sum_{x \in D}^n (\hat{f}(x|\omega) - f(x))^2 \quad (22)$$

Set D consists of N samples, where each sample $(x_i, y_i = f(x_i))$ represents a subset of the total set χ .

The generalization error in this case can be approximated as the average of the five test errors, as demonstrated in Formula (23).

$$E_G(\omega) = \sum_{x \in \chi} p(x) (\hat{f}(x|\omega) - f(x))^2 \approx E_T(\omega) \quad (23)$$

where $p(x)$ represents the probability that x occurs in the total set χ .

The results of the five cross-validation iterations are presented in Tables 8–10. Taking high-speed rail parts dataset 1 as an example, the recall rate is utilized as an evaluation metric to compare the four methods in the classification task of positive and negative

samples, as depicted in Table 10. The model achieving the highest score during iteration will be selected for testing purposes.

Table 8. Five train iterations on HSR dataset 1.

Iteration	SPADE	PADIM	PatchCore	Ours
1	83.33	88.89	94.44	94.44
2	94.44	94.44	88.89	100.00
3	83.33	83.33	94.44	88.89
4	100.00	94.44	100.00	94.44
5	88.89	100.00	94.44	94.44
Mean Recall	89.99	92.22	94.44	94.44

Table 9. Five cross-validation iterations on HSR datasets 1–3.

Iteration	SPADE	PADIM	PatchCore	Ours
1	94.78	98.26	95.65	98.26
2	95.65	93.04	96.52	95.65
3	92.17	93.91	94.78	93.91
4	97.39	95.65	94.78	97.39
5	90.43	94.78	99.13	96.52
Mean Accuracy	94.08	95.13	96.17	96.35

Table 10. Mean error calculation.

Iteration	SPADE	PADIM	PatchCore	Ours
Mean error	5.92	4.87	3.83	3.65

After selecting the optimal model for each of the four comparison methods, we assessed its performance by conducting tests on five separate datasets and calculated the average accuracy based on the results obtained from these five tests, as presented in Table 9.

Finally, we computed the average test error after five iterations of the four models utilized in the testing process. It can be observed from Table 10 that our method demonstrates superior generalization capability for the HSR dataset, as evidenced by its significantly lower generalization error compared to other methods employed across the three industrial datasets.

4.5. Ablation Experiment

Experiments were conducted to assess the contribution of different components in our proposed approach. The MVTec dataset was used as the subject of these experiments, where we evaluated the impact of pooling, registration, and post-processing methods individually. The experimental results clearly demonstrated that our method outperformed the other methods in terms of performance.

4.5.1. Backbone Selection Process

The CNN and transformer have emerged as the predominant backbone networks for deep learning in recent years, owing to their robust feature extraction capabilities. Initially designed for natural language processing, the transformer has also been adapted by researchers for image processing through vision transformer (ViT) networks. In this section, we randomly select three subgraphs to visualize the initial three layers of features extracted from ViT and wide_resnet_50_2 networks, using the screw class as an example (Figures 9 and 10). Notably, the features extracted by ViT exhibit enhanced recognizability, with reduced noise compared to those obtained by wide_resnet_50_2.



Figure 9. Feature visualization extracted by wide_resnet_50_2 network.

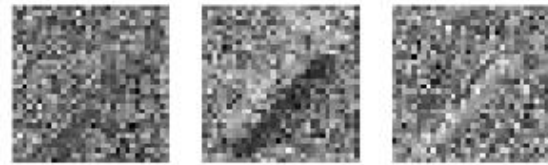


Figure 10. Feature visualization extracted by ViT network.

The disparity in the experimental outcomes arises from variations in the scale of the training data. In contrast to ViT's remarkable efficacy on extensive datasets, CNNs exhibit commendable feature extraction capabilities even with limited data, rendering them more amenable for deployment within data-scarce industrial domains.

4.5.2. SPPM

In the registration module depicted in Figure 1, we incorporate a 2D average pooling module alongside the CNN + STN for extracting image features. This addition serves to mitigate the over-sensitivity of the convolutional layers to positional information. We have opted not to use the SPPM in this module, as our experiments have shown that 2D average pooling yields better results. As shown in Table 11, the application of AvgPool2d in the registration module received the highest score because this module plays a crucial role in maximizing the absolute value of the negative cosine similarity loss, thereby significantly enhancing the performance of anomaly detection.

Table 11. Comparison of different pooling (AUROC—image%, AUROC—pixel%).

Metrics	No Pooling		SPPM		AvgPool2d	
	Det.	Seg.	Det.	Seg.	Det.	Seg.
scores	95.50	88.50	96.80	90.60	98.20	91.70

For evaluation purposes, we employ a pixel-wise anomaly detection AUROC score and PRO score as our chosen metrics. To illustrate the convergence of loss, we utilize the screw category as an example, which is depicted in Figure 11.

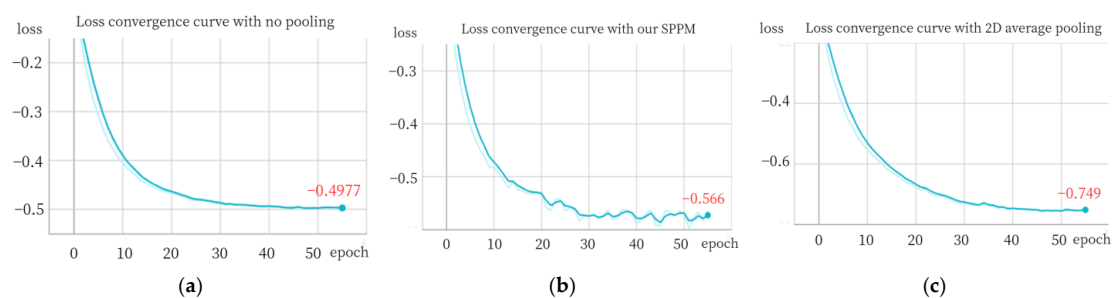


Figure 11. (a) Loss convergence curve with no pooling; (b) our SPPM; (c) 2D average pooling.

The convergence of the negative cosine similarity loss for the screw class is analyzed under different settings. Figure 11a demonstrates the absence of pooling results in a loss value of approximately -0.49 . When our SPPM is utilized, the loss slightly increases to

around -0.57 , as depicted in Figure 11b. However, by incorporating 2D average pooling, the loss significantly improves and reaches approximately -0.75 , as shown in Figure 11c. This observation suggests that integrating 2D average pooling enhances the model's capacity to learn features during the registration process.

4.5.3. Registration Module

For the SPPM and the registration module, we conducted ablation experiments to assess the impact of specific components in our model, namely the SPPM and the registration module. By removing and combining these components individually, we aim to understand their contributions to the overall performance. The results in Table 12 show that when neither the SPPM nor the registration module is used, the two metrics used for evaluation have the lowest score and the lowest anomaly detection result. When both are included in the model, we will get the best anomaly detection result.

Table 12. Module ablation (AUROC%).

Module	MVTec		MPDD	
	Det.	Seg.	Det.	Seg.
No SPPM	95.50	97.51	96.80	90.60
no Registration	95.50	97.64	96.80	90.60
neither	95.50	97.51	95.50	88.50
Both	95.50	97.92	95.50	88.50

4.5.4. Post-Processing Method

In Figure 12, we demonstrate the application of our proposed post-processing method to generate a visual representation of the mask. Specifically, we present an abnormal image from the high-speed rail dataset and show the original image, segmentation image, heat map, ground truth value, the mask generated without the post-processing method, and the mask generated after the post-processing method from left to right. It is evident from Figure 12e that there is a large false positive area; however, this issue is greatly reduced in Figure 12f due to our post-processing method.

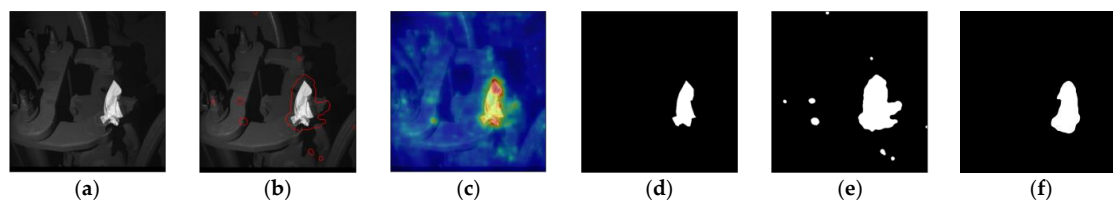


Figure 12. (a) Original image; (b) segmentation result, the red part represents the initial segmentation of abnormal areas; (c) predicted result; (d) ground truth; (e) the mask has been segmented without the use of any post-processing methods; (f) the mask has been segmented using a post-processing method.

The proposed post-processing method for segmentation exceptions employs intersection over union (IoU) as a metric to evaluate the test image, as depicted in Formula (24):

$$IoU = \frac{p_{ii}}{p_{ij} + p_{ji} - p_{ii}} \quad (24)$$

The true value is represented by i , the predicted value is represented by j , and p_{ij} represents the number of pixels predicting i as j . Therefore, Formula (25) is equivalent to Formula (11).

$$IoU = \frac{P \cap G}{P \cup G} \quad (25)$$

The letter P denotes the prediction, while G represents the ground truth.

The intersection over union (*IoU*) algorithm is commonly utilized in deep learning for object detection and semantic segmentation tasks to compute the overlap ratio between different images. In Table 13, we present the average *IoU* metrics for the three datasets used in our experiment, while Section 4.5 showcases visualizations of selected test results. The mask graph generated after applying our proposed post-processing method exhibits a higher degree of proximity to the ground truth value, thereby resulting in an elevated average *IoU* value. This substantiates the efficacy of our proposed approach.

Table 13. The mask generated by the non-post-processing method and the mask generated by our method.

Datasets	$\overline{IoU}$ (Before)	$\overline{IoU}$
HSR dataset 1	0.42	0.68
HSR dataset 2	0.33	0.75
HSR dataset 3	0.37	0.71

As can be seen from Figure 12e,f, the mask result obtained after segmentation via post-processing method is closer to the heat map. The responsiveness of the sample in Figure 12a is depicted in a histogram format, as illustrated in Figure 13b, while the responsiveness of the corresponding class's normal sample is presented in Figure 13a.

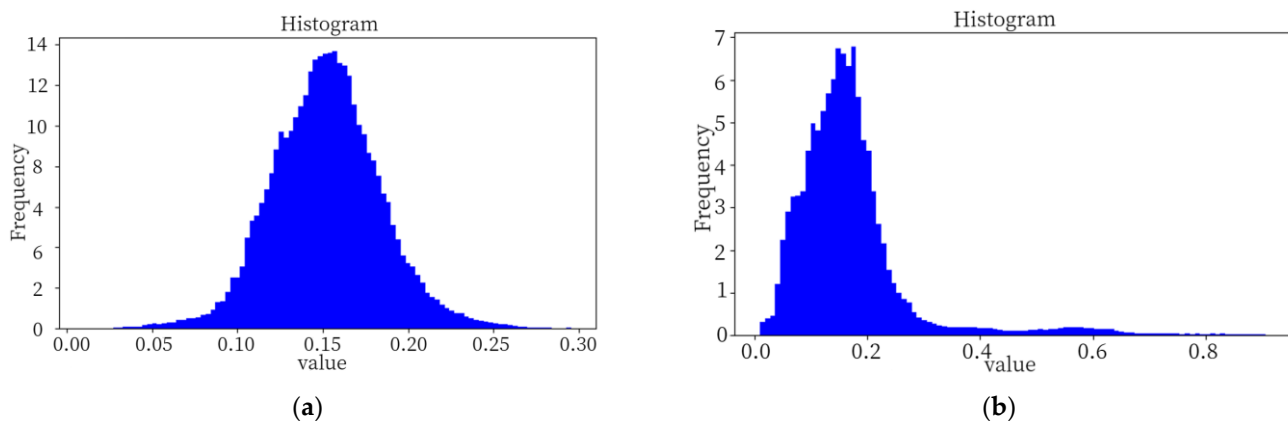


Figure 13. (a) Responsiveness histogram (normal); (b) responsiveness histogram (abnormal).

The test image predominantly consists of normal areas with a high frequency of response but a low level of responsiveness, whereas abnormal areas exhibit a low frequency of response but a high level of responsiveness. The responsivity distribution in the normal region typically conforms to an approximate normal distribution, while the responsivity in the abnormal region lies outside this distribution. The process of calculating responsiveness is analogous to binary classification. Initially, the positive and negative attributes of the image are evaluated. Subsequently, threshold-based truth rules may be employed to exclude certain normal regions exhibiting higher responsiveness and identify regions demonstrating elevated levels of responsiveness. The response value of the normal region is observed to be approximately distributed within the range of (0, 0.7), while our post-processing method calculates an optimal threshold of 0.75. The proposed approach ensures that the highly responsive normal region, depicted in Figure 13b, with a responsivity range of 0.4–0.6 is not erroneously classified as an exception, and the mask prediction yields superior results. By employing this post-processing method, the occurrence of false positives is effectively minimized.

4.6. Visualization

A visualization of selected categories with “spatial transformation” characteristics from various datasets used in the experiment was employed to provide a more intuitive analysis of the detection capability of FR-PatchCore, as depicted in Figure 14.

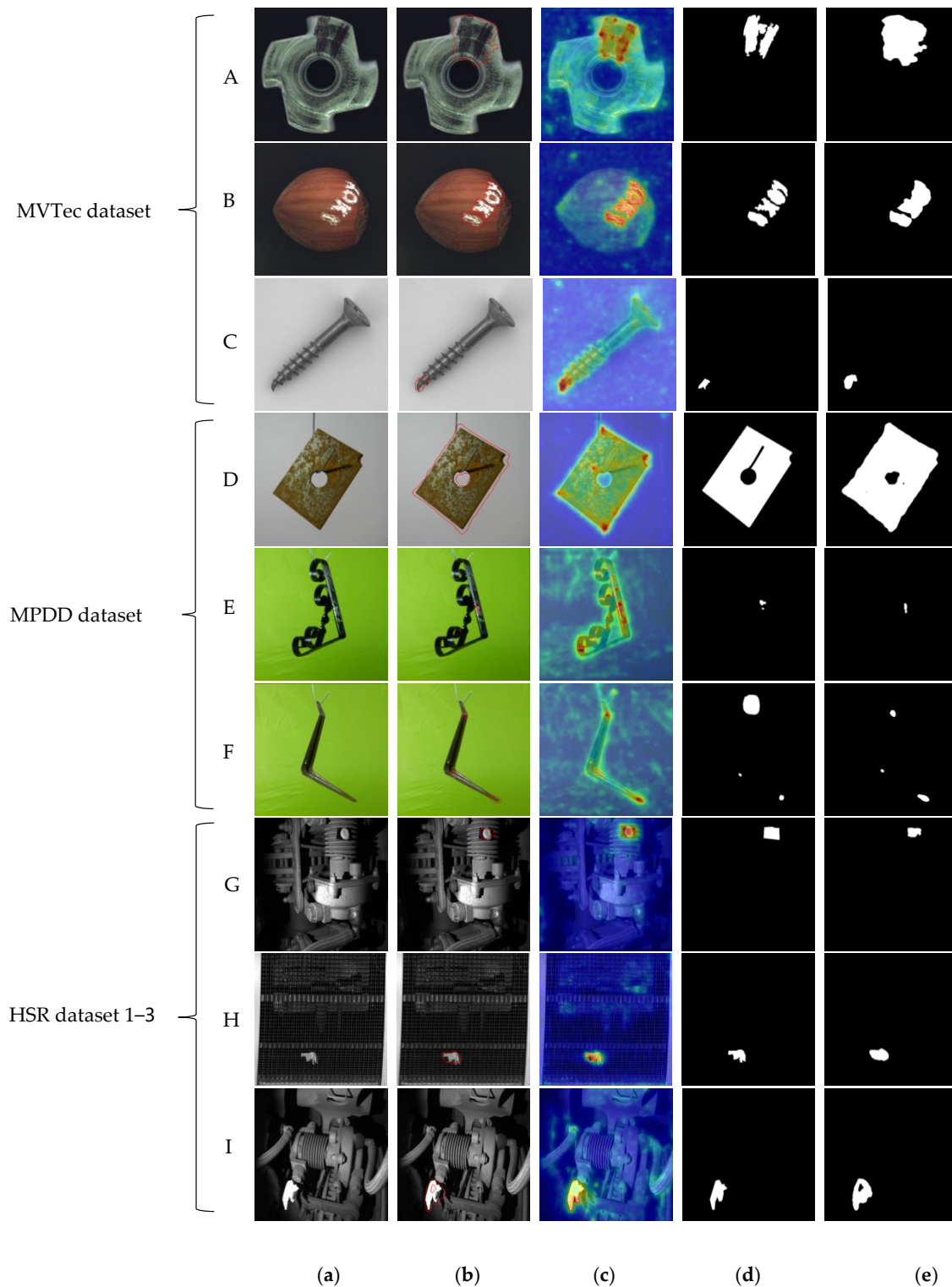


Figure 14. (a) original image; (b) segmentation result, the red part represents the initial segmentation of abnormal areas; (c) predicted result; (d) ground truth; (e) segmented mask.

The “spatial rotation” category of MVTec and MPDD in the two public datasets is selected for display in Figure 14, lines A–F. In Figure 14b, highly accurate anomaly segmentation results are showcased, while Figure 15c visualizes the anomaly score. Once the model successfully locates and evaluates anomalies, the threshold selection method proposed by us can be utilized to obtain mask prediction results, as depicted in Figure 14e.

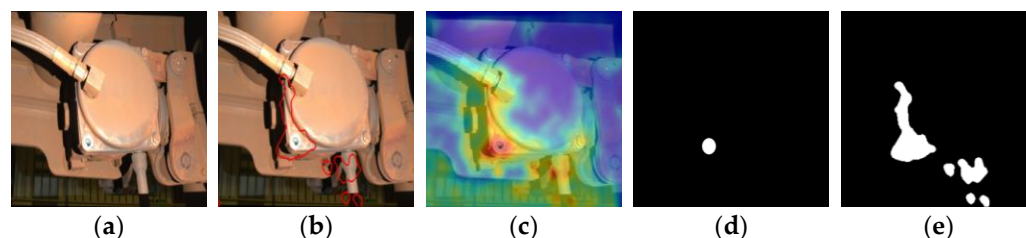


Figure 15. (a) original image; (b) segmentation result, the red part represents the initial segmentation of abnormal areas; (c) predicted result; (d) ground truth; (e) segmented mask.

The anomaly detection results for three different categories in HSR 1–3 datasets are shown in line G–I of Figure 14. In contrast to the single simple scenes of the public dataset, the data captured in industrial scenes exhibit higher levels of noise and complexity in the background, making anomaly detection susceptible to environmental factors. To capture more comprehensive visual information of the inspected objects and prevent data overexposure caused by lighting or other factors during raw data collection for industrial inspection, an unfixed shooting point is adopted instead of continuous shooting from a fixed position. This leads to a significant presence of “spatial transformation” category data in industrial inspections, which further validates the necessity of our research. Based on the segmentation and prediction results from these three columns, FR-PatchCore demonstrates accurate localization of anomalies amidst complex backgrounds while exhibiting high sensitivity toward abnormal areas and low response toward normal areas. Moreover, it maintains a low false positive rate when ensuring reliable anomaly detection, thereby showcasing its potential applications and strong generalization capabilities.

4.7. Discussion

The AUROC and PRO scores were utilized in Tables 1–4 to perform category-by-category anomaly detection on the three datasets employed. In comparison with other notable methods, FR-PatchCore demonstrated a superior average performance by not only achieving excellent anomaly detection for conventional “similar class” images but also effectively detecting the challenging “spatial transformation class” in the dataset.

An evaluation of three challenging datasets for high-speed rail parts, which are critical in industrial testing, was conducted using recall, accuracy, and false positive rates in Tables 5–7. Additionally, the test speed of the comparison method was assessed, and the results indicate that at an image resolution of 224×224 , the test speed is comparable to that of the PatchCore method, with a maximum speed of 0.21 s per frame. These findings demonstrate that our approach ensures a superior recall rate and accuracy while maintaining a minimal false positive rate without compromising efficiency.

Furthermore, to provide a more comprehensive discussion on FR-PatchCore’s capabilities, we designed a five-fold cross-validation experiment utilizing the three industrial datasets. The average error from five tests was adopted as the final evaluation index. Our proposed method exhibited the lowest average error among all tests conducted, further highlighting its competitiveness and strong generalization abilities.

In the ablation experiment conducted in Section 4.5, we performed comprehensive analyses and experiments on the utilized model modules and network selection process. Furthermore, IoU was employed as a metric to assess the effectiveness of the proposed post-processing approach. Additionally, a subset of experimental results was visually

presented in Section 4.6 to illustrate the recovery process of FR-PatchCore segmentation and anomaly positioning.

The effectiveness of self-supervised learning heavily relies on a well-designed auxiliary task. To enhance the model's understanding of local feature relationships during registration, we propose a feature-level registration task that captures and stores the maximum distance between normal features and their respective clustering centers in the feature database. In testing, precise image alignment is no longer necessary. Instead, we can simply query the memory database to calculate the maximum distance between target features and their most similar clustering centers. As a result, our approach achieves robust anomaly detection in two-dimensional images. However, it is important to acknowledge that our method also has certain limitations and disadvantages.

In the introduction, we define the “similarity category” and “spatial transformation category” based on the spatial position relations of objects in the picture, such as rotation and flip. However, this definition is limited to two-dimensional planes and does not account for variations in depth between images that may exist across different planes. Our method is therefore not suitable for capturing spatial position relationships with significant depth differences. As illustrated in Figure 15, we present detection results of data exhibiting substantial disparities in depth information within industrial detection scenarios, along with the corresponding registration loss curve for this class shown in Figure 16. The high occurrence of false positives can be attributed to our feature-grade registration method's inability to achieve accurate depth alignment. Figure 16 depicts the registration loss curve specifically for this category, revealing convergence when the loss reaches approximately 0.4. This indicates that our model struggles to effectively register features associated with this type of data, resulting in a scarcity of learned features compared to previous categories.

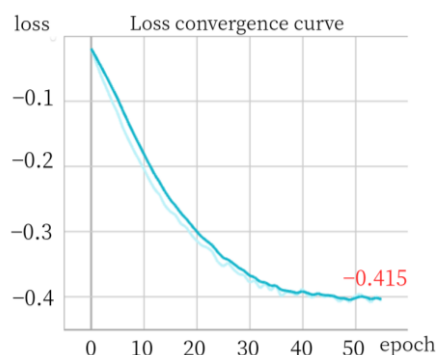


Figure 16. Registration loss function curve of Figure 15a.

The integration of two-dimensional images with depth maps in the above scenarios can potentially yield significant benefits, as it encompasses more comprehensive spatial information. By combining these two types of images, the model can acquire an enhanced feature representation, thereby enhancing its performance.

In addition, the proposed post-processing method is based on a single test image and determines the threshold by calculating the responsiveness of each part of the image, thereby mitigating the occurrence of false positives to some extent. However, when dealing with large-scale test data and high-resolution images, this approach may introduce significant computational overhead and pose challenges for practical implementation. Furthermore, our exploration into adaptive threshold calculations remains limited, without incorporating a comprehensive and universally applicable algorithm. The responsiveness values for each area of the image in Section 4.5.4 were visually represented, offering a novel perspective. If we consider the data distribution, assuming that responsiveness follows a normal distribution, then according to the three-sigma rule, an outlier is defined as a value in a set of result values that deviates from the mean value by more than three standard deviations. However, if the responsiveness does not follow a normal distribution and this distribution can be computed with minimal effort, then the outlier can be defined as k times the standard

deviation from the mean, where k serves as a threshold. Further investigation into this issue will be conducted in future research.

5. Conclusions

In this study, we propose FR-PatchCore, a feature-level registration method that leverages PatchCore, one of the most advanced anomaly detection methods on the MVTec AD benchmark. Feature-level registration is employed as a pre-text task, and the features stored in memory are optimized through registration loss to approach an optimal feature representation. Additionally, we introduce an innovative ground-truth segmentation method to address false positives resulting from artificial threshold determination. The performance of FR-PatchCore excels in anomaly detection on MVTec datasets and achieves state-of-the-art functionality on MPDD datasets (datasets exhibiting spatial location differences). Furthermore, our method is validated using the high-speed rail dataset. The experimental results demonstrate that FR-PatchCore effectively handles two types of data, “similar category” and “spatial transformation category”, thereby enhancing the generalization capability of PatchCore. Our approach showcases its potential for industrial anomaly detection applications while maintaining high accuracy for effective generalization purposes. However, during our discussion, we also analyze certain limitations of this approach. When significant depth differences exist between samples within the same category, our approach does not sufficiently structure the registration task to cope with such variations, leaving room for future improvements.

Author Contributions: Conceptualization, Y.Z. and Z.J.; methodology, Y.W.; software, Z.J.; validation, Z.J. and J.L.; formal analysis, J.L.; investigation, Y.Z.; resources, X.G.; data curation, X.G.; writing—original draft preparation, Z.J.; writing—review and editing, X.G.; visualization, Z.J.; supervision, Y.Z.; project administration, J.L.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (Grant No. 61960206010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data underlying the result presented in this paper are available in Refs. [13,14]. Other data are not publicly available at this time but can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, Z.; Zhang, Y.; Luo, L.; Wang, N.; Lim, S. AnoDFDNet: A Deep Feature Difference Network for Anomaly Detection. *J. Sens.* **2022**, *2022*, 3538541. [[CrossRef](#)]
2. Nguyen, D.T.; Lou, Z.; Klar, M.; Brox, T. Anomaly detection with multiple-hypotheses predictions. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 31 May 2019; pp. 4800–4809.
3. Sakurada, M.; Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, Dunedin, New Zealand, 2 December 2013; pp. 4–11.
4. Akcay, S.; Atapour-Abarghouei, A.; Breckon, T.P. Ganomaly: Semi-supervised anomaly detection via adversarial training. In Proceedings of the Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Revised Selected Papers, Part III 14. pp. 622–637.
5. Arisoy, S.; Nasrabadi, N.M.; Kayabol, K. Unsupervised pixel-wise hyperspectral anomaly detection via autoencoding adversarial networks. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [[CrossRef](#)]
6. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.-K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In Proceedings of the International Conference on Artificial Neural Networks, Lugano, Switzerland, 15 January 2019; pp. 703–716.

7. Carrara, F.; Amato, G.; Brombin, L.; Falchi, F.; Gennaro, C. Combining gans and autoencoders for efficient anomaly detection. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3939–3946.
8. Lyu, Y.; Han, Z.; Zhong, J.; Li, C.; Liu, Z. A Generic Anomaly Detection of Catenary Support Components Based on Generative Adversarial Networks. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 2439–2448. [\[CrossRef\]](#)
9. Son, H.; Jang, Y.; Kim, S.-E.; Kim, D.; Park, J.-W. Deep Learning-Based Anomaly Detection to Classify Inaccurate Data and Damaged Condition of a Cable-Stayed Bridge. *IEEE Access* **2021**, *9*, 124549–124559. [\[CrossRef\]](#)
10. Mitra, J.; Qiu, J.; MacDonald, M.; Venugopal, P.; Wallace, K.; Abdou, H.; Richmond, M.; Elansary, N.; Edwards, J.; Patel, N.; et al. Automatic Hemorrhage Detection From Color Doppler Ultrasound Using a Generative Adversarial Network (GAN)-Based Anomaly Detection Method. *IEEE J. Transl. Eng. Health Med.* **2022**, *10*, 1800609. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Xiao, K.; Cao, J.; Zeng, Z.; Ling, W.-K. Graph-based Active Learning with Uncertainty and Representativeness for Industrial Anomaly Detection. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5016114. [\[CrossRef\]](#)
12. Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; Gehler, P. Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14318–14328.
13. Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9592–9600.
14. Yan, X.; Zhang, H.; Xu, X.; Hu, X.; Heng, P.-A. Learning semantic context from normal samples for unsupervised anomaly detection. *AAAI Conf. Artif. Intell.* **2019**, *35*, 3110–3118. [\[CrossRef\]](#)
15. Zhou, C.; Paffenroth, R.C. Anomaly detection with robust deep autoencoders. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 665–674.
16. Bergmann, P.; Löwe, S.; Fauser, M.; Sattlegger, D.; Steger, C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv* **2018**, arXiv:1807.02011. [\[CrossRef\]](#)
17. You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; Le, X. A unified model for multi-class anomaly detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4571–4584. [\[CrossRef\]](#)
18. Bergman, L.; Cohen, N.; Hoshen, Y. Deep nearest neighbor anomaly detection. *arXiv* **2020**, arXiv:2002.10445. [\[CrossRef\]](#)
19. Cohen, N.; Hoshen, Y. Sub-image anomaly detection with deep pyramid correspondences. *arXiv* **2020**, arXiv:2005.02357. [\[CrossRef\]](#)
20. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185. [\[CrossRef\]](#)
21. Defard, T.; Setkov, A.; Loesch, A.; Audigier, R. Padim: A patch distribution modeling framework for anomaly detection and localization. In Proceedings of the International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2020; pp. 475–489.
22. Zheng, Y.; Wang, X.; Deng, R.; Bao, T.; Zhao, R.; Wu, L. Focus your distribution: Coarse-to-fine non-contrastive learning for anomaly detection and localization. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
23. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28. [\[CrossRef\]](#)
24. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Huang, C.; Guan, H.; Jiang, A.; Zhang, Y.; Spratling, M.; Wang, Y.-F. Registration based few-shot anomaly detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 303–319.
26. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146. [\[CrossRef\]](#)
27. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
28. Cai, L.Z.; Lin, J.; Starr, M.R.; Obeid, A.; Ryan, E.H.; Ryan, C.; Forbes, N.J.; Arias, D.; Ammar, M.J.; Patel, L.G.; et al. PRO score: Predictive scoring system for visual outcomes after rhegmatogenous retinal detachment repair. *Br. J. Ophthalmol* **2023**, *107*, 555–559. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Yang, J.; Shi, Y.; Qi, Z. Dfr: Deep feature reconstruction for unsupervised anomaly segmentation. *arXiv* **2020**, arXiv:2012.07122. [\[CrossRef\]](#)
30. Yi, J.; Yoon, S. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In Proceedings of the Asian Conference on Computer Vision, Kyoto, Japan, 30 November–4 December 2020.
31. Li, C.-L.; Sohn, K.; Yoon, J.; Pfister, T. Cutpaste: Self-supervised learning for anomaly detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9664–9674.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.