Enhanced Anomaly Detection Using Spatial-Alignment and Multi-scale Fusion

Keming Jiao[1], Xincheng Yao[1], Lu Wang[3], Baozhu Zhang[4], Zhenyu Liu[4], and Chongyang Zhang[1,2(✉)]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
{15834075118,i-Dover,sunny_zhang}@sjtu.edu.cn
[2] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
[3] School of Intelligent Manufacturing, Wuxi Vocational College of Science and Technology, Wuxi, China
[4] Ningbo Haitang Information Technology Co., Ltd., Ningbo, China

Abstract. Despite the great progress of anomaly detection technology, the mainstream anomaly detection (AD) methods still face the challenge of accurate detection of semantic anomalies. In this work, we propose a novel AD framework, to mitigate this problem. We introduce two specially-designed modules: Input-Reference Alignment (I-RA) and Adaptive Multi-scale Ensembled Scoring (A-MES). In I-RA, one ORB (Oriented Fast and Rotated Brief)-based spatial alignment block is introduced to constrain the patch matching from feature-only to a consistent measure in both feature and location, which can make the position-related semantical anomaly, such as wrong-printed letters in garment printings, be detected more easily. In order to against large variance of anomaly scale, A-MES is also developed to generate patches of different scale and multi-scale-fused abnormal scores, so that the detect performance for both the small-scale defects and semantic anomalies can be increased further. On the widely used MVTec AD dataset and our specially constructed Garment Printing Defects (GPD) dataset, our method achieves performance comparable to or even better than SOTA, especially in semantic anomalies and small-scale anomaly detection.

Keywords: Anomaly Detection · Garment Printing Defects · Semantic Anomalies · Small-scale Defects

1 Introduction

The ability to discern anomalies within an image is inherent to humans, while it still poses challenges for machines. Nevertheless, anomaly detection has widespread demands in reality, such as in industrial product defect detection [4], road anomaly detection [27], and video surveillance [26]. Nowadays, thanks to the powerful representation capability of deep neural networks for complex

Z. Lin et al. (Eds.): PRCV 2024, LNCS 15043, pp. 294–308, 2025.
https://doi.org/10.1007/978-981-97-8493-6_21

data, deep learning-based anomaly detection methods [19] have achieved significant advances. Due to the scarcity and uncertainty properties of anomalies, deep anomaly detection methods are mostly based on unsupervised learning paradigm, i.e., only using normal samples during training to learn normal patterns. During testing, anomalies can usually be identified by assuming that they do not obey the learned normal patterns (or distribution).

Currently, the mainstream unsupervised anomaly detection methods mainly consist of two branches. The first branch is based on reconstruction, which utilizes auto-encoders [12] or generative adversarial networks [9] to generate reconstructed images. Because only normal samples participate in training, the model would fail in abnormal regions. The difference between the reconstructed image and the input original image usually serves as the criterion for anomaly detection and localization. For example, Schlegl et al. propose AnoGAN [25] based on deep convolutional generative adversarial networks for learning images, with a novel anomaly scoring scheme based on mapping from the image space to the feature space. Bergmann et al. introduce a perceptual loss function based on structural similarity [3], which examines the interdependencies between local image regions by simultaneously considering brightness, contrast, and structural information. The second branch is based on feature embedding. Typically, a network is trained to map normal samples close to each other in the feature space, making it easier to distinguish anomalies. For instance, Reiss et al. propose to employ pre-trained features to adapt to the target domain distribution and applied transfer learning to maintain performance when dealing with one-class classification problems [20]. Constructing a memory bank is another common type in feature embedding methods. During the training phase, features are stored. During the testing phase, anomalies are determined through direct nearest patch features matching. This greatly simplifies the process of anomaly detection and improve the performance, represented by PatchCore [22], which achieves previous SOTA anomaly detection and localization performance.

However, these methods are mainly designed for structural anomalies (e.g., broken, crack, etc) and have not effectively considered semantic anomalies (e.g., wrong-printed letters), causing them still unproficient in semantic anomaly detection. Semantic anomalies are often more ambiguous and challenging than structural anomalies, as they may not appear to abnormal locally, but rather forming overall semantic errors in the image [19], i.e., the wrong-printed letters themselves are usually normal. Therefore, position information is essential for semantic anomaly detection. However, in PatchCore, all patch features are aggregated into a unified memory bank equivalently, it will cause the loss of positional information. As a result, PatchCore has the difficulty to effectively detect semantic anomalies such as wrong-printed letters in clothes, because the features extracted from letters are already present in the memory bank. To address this issue, one intuitive but effective way is to align test images and reference samples without any anomaly first. After that, we can construct novel memory banks with both multi-scales and specific positions. During inference, for each position and scale, only the features from the corresponding memory bank will

be used for feature matching. Additionally, this method observes each position of the images separately at a smaller scale, making small-scale defects more prominent and easier to be detected. In real-world applications, semantic anomalies and small-scale defects will also occur frequently, especially in garment printings. However, the popular MVTec AD dataset [2] mainly contains structural anomalies and is also not focused on a specific industrial scenario. Therefore, to better evaluate the models' ability to detect semantic anomalies and also promote the progress in garment printings, we further establish the Garment Printing Defects dataset, which will be released available to the community.

The primary contributions of this paper are as follows:

- We propose a novel AD framework: AMFS (Aligned Matching and Fused Scoring). We introduce two modules, Input-Reference Alignment (I-RA) and Adaptive Multi-scale Ensembled Scoring (A-MES), to integrate multi-scale patch features and positional information. Thus, the ability to detect semantic anomalies is significantly improved.
- To promote research in defects detection of garment printings, we specially establish the Garment Printing Defects (GPD) dataset for anomaly detection on printing defects. Compared to MVTec AD, our dataset has two charercteristics: more focused and diverse. The GPD dataset comprises 36 types of printings, with 1510 images for training and 2531 images for testing. There are 8 different types of defects. Part of them are collected from apparel factories, while the others are carefully crafted. For each defect image, pixel-level ground truth regions are provided to facilitate performance measurement.
- Our method achieves competitive performance with PatchCore and significantly surpasses SOTA methods on our GPD dataset, up to a image-level AUROC of 96.1 and a pixel-level AUROC of 95.8. This indicates its ability to effectively detect semantic anomalies and small-scale defects.

2 Related Work

2.1 Memory Bank-Based Anomaly Detection

Most deep learning-based anomaly detection focus on making the representation of the abnormal image different from the normal ones, thereby detecting abnormal images and localizing the abnormal regions. In this process, the encoder, which extracts the representation, plays a crucial role. However, this implies that the encoder needs specialized training for the task, making it challenging to adapt to the diverse range of industrial products in actual production. Therefore, industrial product anomaly detection algorithms have shifted towards using fixed pre-trained encoders, such as ResNet [10] trained on ImageNet [7], without fine-tuning them for specific tasks. Features extracted from normal samples form a prototype of normal images. During detection, anomaly scores are calculated based on the similarity between the test image patch features and the prototype, followed by anomaly detection and localization. Cohen, N. et al. propose the SPADE [5], which stores features from different levels of the encoder in a

memory bank and performs anomaly detection and localization based on kNN [8]. Defard, T. et al. introduce the PaDiM [6], which uses a pre-trained CNN to encode images and employs a multivariate Gaussian distribution to obtain the probability representation of normal samples. Additionally, to address the domain shift issue encountered when using pre-trained models, student-teacher knowledge distillation [11] can be introduced for domain adaptation.

Building on the ideas of the aforementioned anomaly detection methods, Roth, K. et al. propose the PatchCore [22]. PatchCore constructs a memory bank using patch features from middle layers of the pre-trained encoder and subsamples them to obtain a coreset for anomaly detection and localization. While ensuring detection speed, the PatchCore method achieved the previous SOTA results on MVTec AD. Inspired by PatchCore, several following works emerge. Lee et al. propose CFA [16], which utilizes transfer learning to obtain the centers and hyperspheres of the memory bank, addressing the issue of normal parts being overestimated in anomaly images. Xie, G. et al. introduce Graph-core [28], which applies PatchCore to few-shot industrial detection using graph representation. Kim, D. et al. propose FAPM [14], designing an adaptive coreset sampling algorithm to improve matching speed during detection. Our work is also based on PatchCore with improvements focusing on addressing the problems of printing defects detection, especially semantic anomalies and small-scale defects. Unlike other improved methods, our enhancement is plug and play. It can be easily incorporated into other memory bank-based anomaly detection methods to enhance their detection capabilities for semantic anomalies and small-scale defects.

2.2 Existing Industrial Defects Dataset

Due to the strong demands for detecting product defects in industrial production and the need to improve the model detection capabilities, various datasets for industrial product defects have emerged. In 2015, Mery, D. et al. introduce the GDXray dataset [18], which collects X-ray images of various materials and products, including castings, welds, etc. In 2019, Bergmann, P. et al. propose the most commonly used benchmark for industrial product defect detection today, the MVTec AD dataset [2]. It comprises images of various industrial products with significant differences, such as toothbrushes, transistors, capsules, carpets, etc. Moreover, each product category includes different types of annotated defects, such as blemishes, cracks, etc. The emergence of this dataset also signifies a shift from traditional methods to deep learning methods in research on industrial product defect detection. Additionally, there are many defect datasets tailored for specific production domains. Huang, Y. et al. utilize the Magnetic Tile Defects (MTD) dataset in [13], which includes both defect-free and defective images of magnetic tiles with varying illumination levels and image sizes. Zhang, C. et al. construct the Zju-leaper dataset [30], containing a large number of normal and abnormal fabric images.

However, for the printing defects detection problem, due to the scarcity of samples for printing defects, it is challenging to measure model performance,

and there are virtually no public benchmark datasets specifically for printing defects. Liu, B. et al. collected garment printing images in [17]. However, they end up with only 135 defect images alongside 4035 defect-free images. The severe lack of garment printing defects images makes it difficult for us to conduct a comprehensive evaluation of the models. To address this issue, we mimic the primary categories of defects that occur in actual production and artificially create defects approximating real ones on defect-free images to compensate for the shortage of samples.

3 Method

The overview of our method is illustrated in Fig. 1. Our method is mainly based on PatchCore to achieve anomaly detection. While we further propose Input-Reference Alignment (I-RA) (3.1) and Adaptive Multi-scale Ensembled Scoring (A-MES) (3.2) to enhance the semantic anomalies and small-scale defects detection ability.

3.1 Input-Reference Alignment

To ensure that the positional information in the memory banks of different regions can be effectively utilized and remains unaffected by features outside the corresponding region during detection, it is necessary to ensure that the main subject's positions are as similar as possible in all training and testing images. To achieve this, for each type of printing, we select one image from the training set as a reference image. This reference image is not used for training but serves as a reference for aligning all other images. To ensure the alignment process is sufficiently efficient, we employ the fast ORB [23] feature detector to first extract feature points, and then the feature points from the input image and reference image are matched among each other. The matched feature point pairs are used to compute the homography matrix. With the matrix, we can apply affine transformations to the input images to align them with the reference image. An example of the alignment process is illustrated in Fig. 1.

3.2 Adaptive Multi-scale Ensembled Scoring

During training, for the aligned image with size (H, W), we construct an image pyramid [1] consisting of N layers. In layer-i, the image size is $(H/2^{i-1}, W/2^{i-1})$. In each level, the aligned image is evenly divided into $(2^2)^{i-1}$ blocks of the same size as the layer-N image. In total, there are $\sum_{i=1}^{N}(2^2)^{i-1} = (4^N - 1)/3$ blocks for one image. Subsequently, memory banks, $M_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$, are established for each block at corresponding positions across all images. Then they are subsampled to coresets [22], $C_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$. The details of training procedure is shown in Algorithm 1.

For testing, the detected images undergo the same construction of image pyramid and division. Then anomaly maps, $m_{ij},, i \in \{1, \ldots, N\}, j \in$

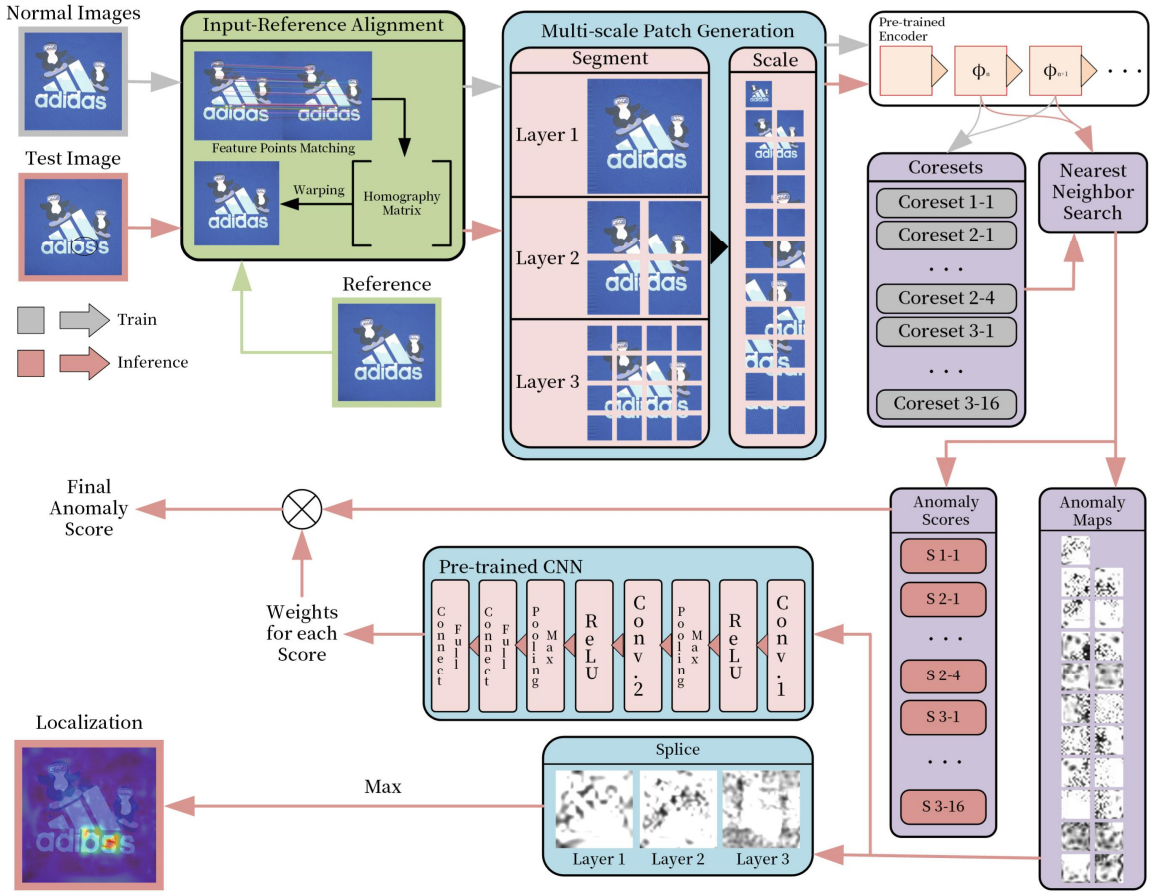


Fig. 1. The overview of AMFS. All training and testing images are aligned based on a reference image. Then blocks of multi-scale and different positions are generated. During training, corresponding coresets are established for each block. During inference, feature matching is performed on the corresponding coreset to obtain anomaly maps and anomaly scores. The maximum value is taken to obtain the final anomaly map, and the pre-trained convolutional network is used to obtain the final anomaly score.

Algorithm 1 The training procedure of AMFS

Require: training dataset I_{tr}, number of layers N, image input size (H, W), batch size b
Ensure: $\frac{4^N-1}{3}$ coresets $C_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$
  Initialize $\frac{4^N-1}{3}$ memory banks $M_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\} \rightarrow \{\}$
  Set the size for images to be divided and resized $\rightarrow (\frac{H}{2^{N-1}}, \frac{W}{2^{N-1}})$
  for b images in I_{tr} **do**
    for serial number of layer i in $[1, N]$ **do**
      Divide the images to $(2^{i-1})^2$ parts by dividing the width and height equally
      Obtain patch features of each part $p_{ij}, j \in \{1, \ldots, 4^{n-i}\}$
      Save patch features to corresponding memory bank
    end for
  end for
  for each memory bank **do**
    Subsample to $\frac{4^N-1}{3}$ coresets $C_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$
  end for

$\{1, \ldots, 4^{N-i}\}$, and scores, $s_{ij}, , i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$, for each block are obtained through the matching distance between the test patch and its nearest neighbor [22] in the corresponding coresets. For anomaly localization, we reconstruct N anomaly maps, $m_i, i \in \{1, \ldots, N\}$, by separately splicing blocks of each layer to form a complete anomaly map. Then the final anomaly map m will be obtained by Eq. 1,

$$m = \{Max(v_{i,h,w}) | i \in \{1, \ldots, N\}, h \in \{1, \ldots, H\}, w \in \{1, \ldots, W\}\} \quad (1)$$

where $v_{i,h,w}$ indicates the anomaly value in position (h, w) of m_i. To balance the contribution to final anomaly score of each block better, we input the anomaly maps into a pre-trained convolutional neural network, F_{fusion}, and obtain the weights $w_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$ corresponding to each anomaly score. Then the final anomaly score can be calculated by Eq. 2,

$$s = [w_{ij} | i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}] \cdot [s_{ij} | i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}] \quad (2)$$

where $w_{ij} = F_{fusion}(m_{ij})$. The details of inference procedure is shown in Algorithm 2.

To obtain the pre-trained network F_{fusion}, we need numerous anomaly scores and maps, s_{ij} and m_{ij}, of normal and abnormal images. Therefore, we utilize an anomaly generation method proposed by Zavrtanik, V. et al. [29] to generate anomalies in a randomly selected region of the images, while data augmentation, including translation, scaling, and rotation, is applied to the normal images to obtain a large number of normal images. Then we feed them into our algorithm and stop when getting anomaly scores s_{ij} and maps m_{ij}. Using this data, we train a convolutional neural network with a structure similar to AlexNet [15] to obtain the F_{fusion}. The input of this network is anomaly maps m_{ij}, and the output is weights of each corresponding anomaly score w_{ij}. The training objective is to maximize the final anomaly score s for anomaly images and minimize the final anomaly score s for normal images.

Algorithm 2 The inference procedure of AMFS

Require: test image I_{te}, number of layers N, image input size (H, W), $\frac{4^N-1}{3}$ coresets $C_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$
Ensure: anomaly map m, anomaly score s
  for serial number of layer i in $[1, N]$ **do**
    Divide the images to $(2^{i-1})^2$ parts by dividing the width and height equally
    Obtain $(2^{i-1})^2$ anomaly maps $m_{ij}, j \in \{1, \ldots, 4^{N-i}\}$ and anomaly scores $s_{ij}, j \in \{1, \ldots, 4^{N-i}\}$
    Splice a complete anomaly map m_i and resize to (H, W)
  end for
  Initialize an empty anomaly map m of size (H, W) whose all anomaly values $\rightarrow 0$
  for (h, w) ,h in $\{1, \ldots, W\}$, w in $\{1, \ldots, W\}$ **do**
    anomaly value in position (h, w) is set to $Max(v_{i,h,w}), i \in \{1, \ldots, N\}$
  end for
  Use $m_{ij}, , i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$, $s_{ij}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, 4^{N-i}\}$, and F_{fusion} to obtain anomaly score s

4 Dataset Description

The Garment Printing Defects (GPD) dataset consists of 36 categories of print images, including 1510 images for training and 2531 images for testing. The training set only contains defect-free images, with approximately 40 images per print category, while the test set includes both defective and defect-free images. Among them, 19 categories were collected on-site from apparel factories, while other 17 categories were collected from online sources, as shown in Fig. 2. The resolution of all images ranges from 600×600 to 2590×2590 pixels. We annotated the ground truth at the pixel level with labelme [24] for every defective image.

Fig. 2. Example images for all 36 printing categories of the GPD dataset.

To cover a wide range of potential defects that may occur on garment printings, we collect 8 types of defects, including color, dim, hole, wrong-printed letter, miss, overprint, scratch, and stain. However, due to the limited availability of real defects in actual production, it was challenging to collect a large number of authentic defects. To ensure that our dataset contains a sufficient number of defect samples, we meticulously fabricated a portion of anomalies following the logic of real anomaly generation. The details of each defect and their examples are showed in Table 1.

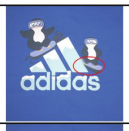
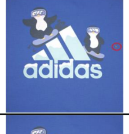
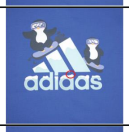
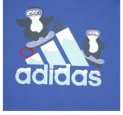
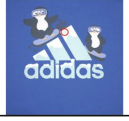
Compared to existing printing defects datasets [30,31], our dataset includes a richer variety of printing types, closely resembling real-world occurrences in factory production. Additionally, beyond common structural anomaly defect images, it contains numerous images with semantic anomalies. Therefore, our dataset can be used not only to improve printing defects detection but also to provide further data for semantic anomaly detection. Dataset at GPD_dataset.

5 Experiments

5.1 Experimental Details

Dataset We mainly conduct experiments on the popular MVTec dataset and the proposed GPD dataset. On the GPD dataset, during both training and testing phases, all images are resized to dimensions of 512 × 512 pixels. We

Table 1. Descriptions and example images of defects

Type of defects	Morphological description	Example image
Color	A portion exhibits either excessively high or low color saturation, or it is incorrectly printed with another color.	
Dim	Blurring of the printings occurs due to the overflow of glue.	
Hole	Holes in the garments caused by machinery malfunctions.	
Letter	Letters in the garment printing are wrong-printed as other existing letters.	
Miss	A portion of the printing is omitted during production or embedded within the fabric, making it difficult to observe.	
Over	A portion of the printing is redundantly processed in different locations.	
Scratch	Linear scratches caused by machinery errors during production.	
Stain	Point-like defects caused by insufficient glue or the presence of impurities.	

also test using the MVTec AD dataset as a benchmark. This dataset comprises images of 15 industrial products, with 3629 images in the training set and 1725 images in the test set. The training set exclusively contains defect-free images. Pixel-level ground truth annotations are provided. The resolution of all images ranges from 700×700 to 1024×1024 pixels. The images are resized to 256×256 pixels for both training and testing.

Evaluation Metrics The ROC curve and the area under the ROC curve (AUC) are commonly used evaluation metrics for binary classification tasks. In this experiment, we adopted image-level AUROC as the evaluation metric for image-level anomaly detection and pixel-level AUROC for evaluating pixel-level anomaly localization.

Model Configuration To better simulate the rapid detection requirements in real-world production scenarios, we set the coreset sampling rate to 1% to reduce

detection time. When calculating anomaly scores, we used a neighbor size of 9 and employed a pyramid structure with 3 layers.

5.2 Anomaly Detection on MVTec AD

The experimental results on the MVTec AD dataset are shown in Table 2. Compared to PatchCore, anomaly localization performance improved by 0.3, with nearly all categories of objects achieving an increase in pixel-level AUROC, which was already at a high level. Even on the MVTec AD dataset, which primarily consists of structural anomalies and anomalies of normal size, our method maintains a comparable image-level AUROC to PatchCore while achieving an improvement in pixel-level AUROC. This indicates that our method is more conducive to anomaly localization.

Table 2. Experimental results on MVTec AD dataset

Category	Bot.	Cab.	Cap.	Car.	Gri.	Haz.	Lea.	Met.	Pil.	Scr.	Til.	Too.	Tra.	Woo.	Zip.	Average
image-level AUROC																
PatchCore	**100.0**	**99.3**	98.0	**98.0**	**98.6**	**100.0**	**100.0**	**99.7**	97.0	**96.4**	99.4	**100.0**	**99.9**	99.2	**99.2**	**99.0**
AMFS	**100.0**	97.8	**99.9**	96.8	97.7	**100.0**	98.9	97.2	**98.5**	96.3	**99.7**	**100.0**	**99.9**	**99.6**	**99.2**	98.8
pixel-level AUROC																
PatchCore	**98.5**	98.2	98.8	**98.9**	98.6	98.6	**99.3**	98.4	97.1	99.2	96.1	98.5	94.9	95.1	**98.8**	97.9
AMFS	**98.5**	**98.7**	**99.2**	98.4	**98.8**	**98.8**	99.1	**98.7**	**97.7**	**99.5**	**96.2**	**98.7**	**96.4**	**95.2**	98.7	**98.2**

Bold indicates the better performance

5.3 Anomaly Detection on the GPD Dataset

For printing defects detection, we conduct experiments on the entire GPD dataset. Furthermore, to more effectively validate the performance of our method on semantic anomalies and small-scale defects, we also conduct experiments separately on the letter subset and the small-scale defect subset. The experimental results obtained are shown in the Table 3. Compared with other memory bank-based anomaly detection methods, our detection and localization performance are the best, reaching an image-level AUROC of 96.1 and a pixel-level AUROC of 95.8. Compared to PatchCore, they have been improved by 1.3 and 2.5 respectively. When used for detecting small-scale defects, the performance of other methods will drop dramatically, while our method can achieve much better results than these methods. For instance, the results show that we can exceed PatchCore by 6.5 and 4.6 on detection and localization respectively. On the letter subset, our localization performance advantage is even more obvious, exceeding PatchCore by 8.4, while the detection performance still remains leading. This shows that our method has superiority on the detection and localization of these two types of defects.

Table 3. Experimental results (image/pixel-level AUROC) on GPD dataset and subsets.

Methods	SPADE [5]	Mah.AD [21]	PaDiM [6]	PatchCore [22]	Ours
GPD dataset	62.8/90.7	72.5/-	72.0/93.0	94.8/93.3	**96.1/95.8**
Small-scale defects	62.7/50.6	70.5/-	71.0/77.2	91.3/88.3	**97.8/92.9**
Letter subset	41.6/57.7	48.8/-	66.0/85.8	90.8/86.7	**93.4/95.1**

5.4 Ablation Study

Spatial-Alignment When the pyramid layers are set to 2, 3, and 4, the experimental results with and without I-RA are presented in Table 5. It can be observed that the our I-RA module can significantly improve the image-level AUROC. Without spatial alignment during training, different images might have variations in angle, position, or size of their main subjects. If the main subject happens to be centrally located in the image, the contents of divided blocks will differ, especially in lower pyramid levels. This discrepancy causes the coresets to contain features beyond the predefined range. Similarly, the divided blocks of the test images might not entirely correspond to the intended range, resulting in abnormally high or low anomaly scores for that particular scale and position.

Layers In Table 4, we show ablation studies under different pyramid levels. When the number of pyramid layers was set to 3, the average pixel-level AUROC reached its maximum value. For a more intuitive illustration, we show the anomaly localization results of the "over" defect category in different pyramid levels in Fig. 3. Compared to using 1 layer (PatchCore) and 2 layers, the localization results from the 3-layer pyramid were evidently more accurate, with almost only the anomalous regions being delineated. The localization results from the 4-layer pyramid did not show significant improvement. The average image-level AUROC gradually decreased with an increase in the number of layers. Specifically, the 3-layer pyramid only decreased by 0.4, compared to 2 layers, but there was a drop of 3.6 when comparing 4 layers to 3 layers. Therefore, for input images of size 512×512 pixels, a 3-layer pyramid is the most suitable choice.

Fused Scoring methods The average image-level AUROC results obtained using various anomaly score calculation methods at different pyramid layers are presented in Table 5. Intuitively, when a portion of an image is anomalous, the entire image can be considered as anomalous. Therefore, it would seem appropriate to select the maximum value from the anomaly scores of different image parts as the final anomaly score. However, using the maximum value is susceptible to noise interference. Some normal samples may also produce excessively high anomaly scores at certain points. Experimental results also show that the maximum value performs the worst. Using the anomaly scores s_{ij} of different

Table 4. Experimental results on GPD dataset when selecting different numbers of layers.

Layers	image	pixel
1	94.8	93.3
2	96.5	94.2
3	96.1	95.8
4	92.5	94.6

Table 5. Experimental results (image/pixel-level AUROC) on GPD dataset when selecting different anomaly score computing methods and without I-RA.

Number of Layers	Max	Mean	w/o I-RA	AMFS
2	93.6/-94.5/-	92.4/93.2	96.5/94.2	
3	90.5/-92.9/-	83.6/94.8	96.1/95.8	
4	84.7/-87.3/-	76.6/89.6	92.5/94.6	

blocks mentioned in Sect. 3.2 for training, we directly fit whether an anomaly exists using a linear layer: 0 represents no defect, and 1 represents defects existing. The weights obtained for the scores of different parts at 2 and 3 pyramid layers are shown in Table 6. The weight distribution under the 2-layer pyramid is closer to taking the mean, while the 3-layer pyramid gives more weight to the lower-resolution part of the entire image. As a result, the average calculated image-level AUROC is higher. Furthermore, our method effectively balances the contribution of each region to the final anomaly score by training a weighting network. Experimental results also demonstrate the effectiveness of our method.

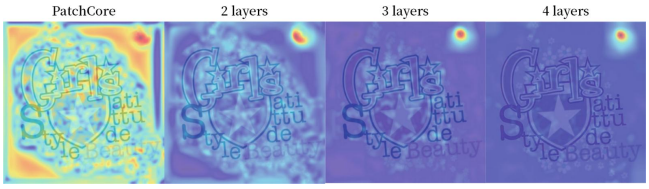


Fig. 3. Anomaly localization performance when selecting different numbers of layers.

Table 6. Weights of anomaly scores of each blocks. The order is from the highest level to the lowest level of the pyramid, with each layer arranged from left to right and top to bottom.

Number of Layers	Weights
2	0.281, 0.262, 0.334, 0.229, 0.313
3	0.411, 0.132, 0.028, 0.171, -0.068, 0.111, 0.147, 0.173, 0.202, -0.011, -0.028, -0.181, 0.065, 0.020, -0.147, -0.106, 0.126, 0.061, 0.119, 0.105, 0.324

6 Conclusion

In this paper, for the specific application of printing defects detection and to address the issue of semantic anomalies lacking in existing anomaly detection datasets, we propose the GPD dataset. Compared to other datasets, ours focuses on the industrial category of garment printings, featuring a wide variety and including numerous images with semantic anomalies. Furthermore, we improve upon the existing algorithm PatchCore, pointing out that for detecting semantic errors, it is essential to retain certain positional information. And we have small-scale defects occupy a relatively large area within the region being examined. Integrating these two points, we perform detection on aligned images at multiple scales and different positions, and then fuse the results.

Acknowledgements. This work was supported in part by the National Natural Science Fund of China (62371295), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Science and Technology Commission of Shanghai Municipality (22DZ2229005).

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Eng. **29**(6), 33–41 (1984)
2. 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, pp. 9592–9600 (2019)
3. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv:1807.02011 (2018)
4. Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z., Shao, L.: Surface defect detection methods for industrial products: a review. Appl. Sci. **11**(16), 7657 (2021)
5. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv:2005.02357 (2020)
6. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDiM: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition, pp. 475–489. Springer (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. IEEE (2009)
8. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Applications of Data Mining in Computer Security, pp. 77–101 (2002)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)
12. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
13. Huang, Y., Qiu, C., Yuan, K.: Surface defect saliency of magnetic tile. Vis. Comput. **36**(1), 85–96 (2020)
14. Kim, D., Park, C., Cho, S., Lee, S.: FAPM: fast adaptive patch memory for real-time industrial anomaly detection. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
16. Lee, S., Lee, S., Song, B.C.: CFA: coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access **10**, 78446–78454 (2022)
17. Liu, B., Chen, Y., Xie, J., Chen, B.: Industrial printing image defect detection using multi-edge feature fusion algorithm. Complexity **2021**, 1–10 (2021)
18. Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M.: GDXray: the database of X-ray images for nondestructive testing. J. Nondestr. Eval. **34**(4), 42 (2015)
19. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: a review. ACM Comput. Surveys (CSUR) **54**(2), 1–38 (2021)
20. Reiss, T., Cohen, N., Bergman, L., Hoshen, Y.: PANDA: adapting pretrained features for anomaly detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2806–2814 (2021)
21. Rippel, O., Mertens, P., Merhof, D.: Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 6726–6733. IEEE (2021)
22. 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, pp. 14318–14328 (2022)
23. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
24. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**, 157–173 (2008)
25. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging, pp. 146–157. Springer (2017)
26. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6479–6488 (2018)
27. Vojir, T., Šipka, T., Aljundi, R., Chumerin, N., Reino, D.O., Matas, J.: Road anomaly detection by partial image reconstruction with segmentation coupling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15651–15660 (2021)
28. Xie, G., Wang, J., Liu, J., Zheng, F., Jin, Y.: Pushing the limits of fewshot anomaly detection in industry vision: GraphCore. arXiv:2301.12082 (2023)

29. Zavrtanik, V., Kristan, M., Skočaj, D.: DRAEM—A discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8330–8339 (2021)
30. Zhang, C., Feng, S., Wang, X., Wang, Y.: ZJU-Leaper: a benchmark dataset for fabric defect detection and a comparative study. IEEE Trans. Artif. Intell. **1**(3), 219–232 (2020)
31. Zhang, E., Li, B., Li, P., Chen, Y.: A deep learning based printing defect classification method with imbalanced samples. Symmetry **11**(12), 1440 (2019)