EasyChair Preprint

№ 10825

Anomaly Detection of Industrial Products Considering Both Texture and Shape Information

Shaojiang Yuan, Li Li, Neng Yu, Tao Peng, Xinrong Hu and Xiong Pan

September 2, 2023

Anomaly Detection of Industrial Products Considering both Texture and Shape Information*

Shaojiang Yuan[1][0000−0002−8757−9646], Li Li[1,2,3][0000−0002−2027−1145], Neng Yu[1], Tao Peng[1,2,3], Xinrong Hu[1,2,3], and Xiong Pan[1,2,3]

[1] School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200, China
[2] Engineering Research Center of Hubei Province for Clothing Information, Wuhan, 430200, China
[3] Hubei Provincial Engineering Research Center for Intelligent Textile and Fashion, Wuhan, 430200, China

Abstract. Anomaly detection of industrial products is an important issue of the modern industrial production in the case of shortage of abnormal samples. In this work we design a novel framework for unsupervised anomaly detection and localization. Our method aims to learn global and compact distribution from image-level and feature-level processing of normal images. For image-level information, we present a self-supervised shape-biased module(SBM) aimed at fine-tuning the pre-trained model to recognize object shape information. As for feature-level information, our research proposes a pretrained feature attentive module (PFAM) to extract multi-level information from features. Moreover, given the limited and relatively small amount of texture-based class feature information in existing datasets, we prepare a multi-textured leather anomaly Detection(MTL AD) dataset with both the texture and shape information to shed a new light in this research field. Finally, by integrating our method with multiple state-of-the-art neural models for anomaly detection, we are able to achieve significant improvements in both the MVTec AD dataset and the MTL AD dataset.

Keywords: Anomaly detection · Self-supervised learning · Attention mechanism · Jigsaw puzzle.

1 Introduction

With the advent of industry 4.0 [14], production intelligence has become more and more important. Due to the difficulty of obtaining defects, the traditional defect detection system based on a large amount of data also needs to be dynamically changed. It is urgent to apply unsupervised or few-shot reliable methods to solve the problem of industrial product defect detection.

* Supported by National Natural Science Foundation of China with No.61901308.

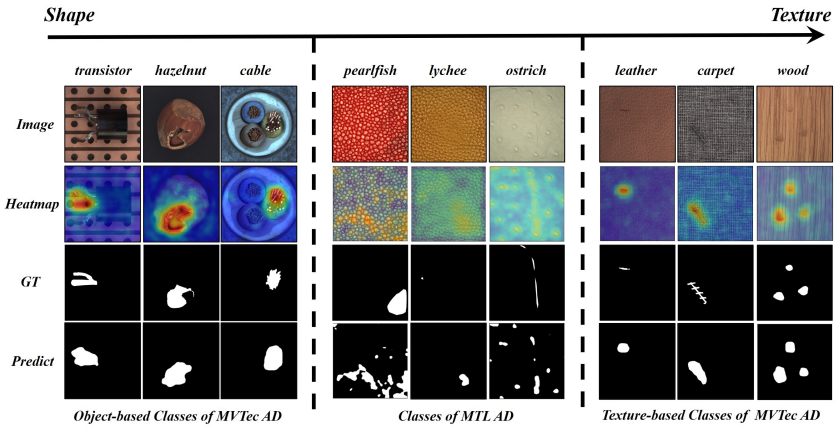

Fig. 1. The visualization results of the PaDiM [8] method on the two datasets(Image,Heatmap,GT,Predict anomaly mask). The top and bottom of the image are texture-based and object-based classes of MVTec AD respectively, and the middle of the image is our MTL AD dataset, which incorporates texture and shape features. The predict results show that the model cannot handle our MTL AD dataset effectively.

With the in-depth study of pretrained models [10,11], the convolutional neural network tend to have a preference for local texture information, and make judgments only by these. As shown in Fig. 1, high-precision tasks often require the features extracted by the model to be more comprehensive and in line with the characteristics of specific task. In the unsupervised anomaly detection task, it is necessary to obtain high-level abstract features as much as possible. This information extraction capability is a great challenge to the existing ImageNet-based convolutional neural network framework.

To balance the shape and texture information, as shown in Fig. 2, we propose a novel framework to learn global and compact distribution from image-level and feature-level processing of normal images. For shape-based class images, we introduce a shape-biased module (SBM) composed of a defect synthesis block and a defect jigsaw block to fine-tune the pre-trained network for global information recognition. We use the pre-trained feature attentive module (PFAM) to apply different processing to multi-level features extracted by the model. The MFCSAM attention mechanism is introduced to aggregate multi-channel and long-distance information, allowing the model to focus on global shape information. In addition, since the texture-based class feature information in the existing anomaly detection datasets is relatively small and simple, we especially collect a large leather texture dataset MTL AD dataset with both texture information and shape information. We combine our method with various state-of-the-art anomaly detection methods and conduct extensive experiments on MTL AD

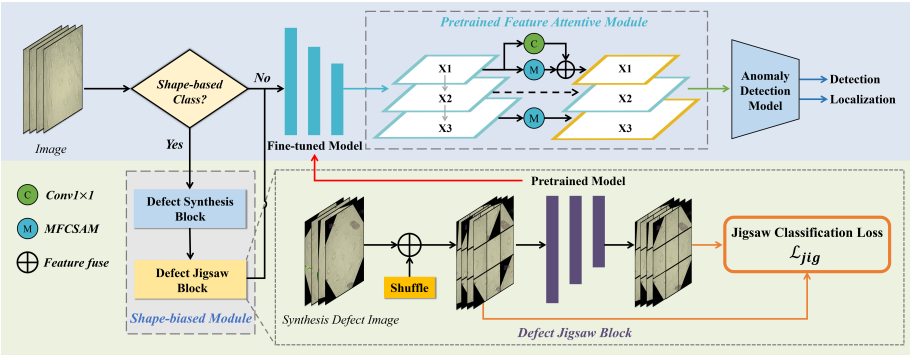

Fig. 2. The overview of our anomaly detection framework. The shape-biased module(SBM) fine-tunes the pretrained model at the image level, enhancing the model's ability to perceive the global shape information. The pretrained feature attentive module (PFAM) processes the features at the feature level, acquiring richer contextual semantic information.Incorporating these two modules can mitigate the issue of convolutional locality in existing anomaly detection models and improve the effect of anomaly detection.

and MVTec AD datasets [5]. Our experimental results show that the method can bring significant improvements in both anomaly detection and localization.

In summary, the main innovations of our work are listed as follows:

1) We propose a unique jigsaw block to adapt the pretrained model to shape information, helping the pretrained features overcome the local shape bias of the pretrained network in anomaly detection.

2) We integrate our method into several state-of-the-art models [8, 15, 21] for anomaly detection, showing significant performance improvements across multiple models and baselines.

3) We prepare a Multi-Textured Leather Anomaly Detection(MTL AD) dataset with both the texture and shape information to effectively expand the authenticity and diversity of anomaly detection datasets.

2 Related work

Anomaly detection models learn feature representation from normal data and apply it to both normal and abnormal data during testing due to the large amount of unstructured and unlabeled data of real-world abnormal samples. Depending on the representation learning models used, anomaly detection models can be divided into discriminative models based on pretrained models and generative models based on AE/GAN.

2.1 Discrimination Models Based on Pretrained Feature

The anomaly detection model learns discriminant features in nominal data through their own unique methods, and then compares the distribution of the test data

and the extracted feature in the inference stage to obtain an anomaly score. Depending on the method of establishing the distribution, it can be subdivided into probability-based [1, 8, 9, 23] and distance-based [4, 19] methods.

Probability-based algorithm calculates the probability density distribution for each point of the feature map to form a distribution map, and obtain anomaly score by computing the distribution difference between the test feature point and the K closest points in the distribution map. Shi et al. [23] use normalization flow instead of Gaussian distribution to compute a richer probability distribution for each location. Distance-based method finds the most representative feature information for each feature point, and then uses the feature map to calculate the distance of the K nearest feature points. Reiss et al. [19] used the KNN clustering method to collect the core features to establish a memory bank. Bergman et al. [4] used the nearest neighbor algorithm to calculate the distance of test features after reducing the feature dimension, and Roth et al. [21] introduced the coreset selection method on the basis of [19] to optimize the steps of establishing a memory bank.

2.2 Generative Models Based on Autoencoder and GAN

Generative models such as autoencoder [13] and GAN [12], directly encode the original information of the image to obtain latent space features, and learn the feature distribution of nominal data, then finally compare the test information and the generated features at the image level or pixel level in the inference stage to obtain abnormal results. Zhou et al. [26] used a deep autoencoder to introduce deep learning and some nonlinear activation functions to learn image feature information more robustly. After the GAN was proposed, the field of anomaly detection gradually began to use that network with a stronger generative effect instead of the autoencoder [22]. However, the disadvantage of simply using GAN is that it is irreversible, i.e. it cannot use the generated image to infer the latent space input that generated this image. Liang et al. [17] reconstructed images from multiple scales using multiple frequency components, making the image reconstruction more effective.

3 Methodology

First of all, we emphasize that our framework is an embeddable model enhancement method. For different data categories, we have designed different methods. As shown in Fig. 2, for texture-based classes, we have designed PFAM to help the model obtain richer feature information;For shape-based classes, on the basis of PFAM, we also designed shape-biased module(SBM) to fine-tune its pretrained model to help it learn shape information better. Next, we will introduce the component of the proposed method in detail. It needs to be explained in advance that our method only preprocesses pretrained models and features, and does not involve specific anomaly detection methods.

3.1 Shape-biased Module

Inspired by [3], we design a shape-biased module(SBM) for the shape bias of pretrained model, which focuses on the global shape information by making the pretrained model solve the jigsaw puzzle while reducing the local texture preference. In terms of input data, compared with the general classification problem using jigsaw [6, 7], the amount of nominal data available for training of a single category in industrial data is less, and the feature information is often relatively fixed, which makes the model easy to overfit. Hence, we refer to the method in [25] and use deformation and texture noise to act on nominal data to obtain synthetic anomaly data with complex feature information in the defect synthesis block.Then we combine this data with nominal data as input data for defect jigsaw block.

Let us assume to observe the t class from MVTec AD datasets, with the class containing N_{nom}^t images. After the augmenting of these images, we can get N_{syn}^t synthetic anomaly images, and we merge them into N^t images. Then, we use a regular $n \times n$ grid of patches to crop the source images and shuffled them into one of the n^2 grid positions. In the $n^2!$ possible permutations, we randomly select a set of P elements and assign an index to each patch. Then we define a jigsaw classification task on N^t labeled instances $\{(z_i^t, p_i^t)\}_{i=1}^{N_t}$, where z_i^t indicates the recomposed samples and $p_i^t \in \{1, \ldots, P\}$ is the related permutation index. The objective of defect jigsaw block is to minimize the jigsaw loss $\mathcal{L}_c\left(h\left(z \mid \theta_f, \theta_p\right), p\right)$ that measures the errors between the true permutation index and the index predicted by pretrained model function h, parametrized by θ_f and θ_p. These parameters define the feature embedding space and the final classifier, respectively for the convolutional network and fully connected layer dedicated to permutation recognition. We trained the defect jigsaw block to obtain the shape-adapted model, where $\mathcal{L}_{jig}$ is a standard cross-entropy loss:

$$\underset{\theta_f, \theta_p}{\operatorname{argmin}} \sum_{i=1}^{N^t} \mathcal{L}_{jig}\left(h\left(z_i^t \mid \theta_f, \theta_p\right), p_i^t\right) \tag{1}$$

3.2 Pretrained Feature Attentive Module

This study introduces pretrained feature attentive module(PFAM) to learn the feature. The module integrates low-level and high-level pretrained features through Multi-scale Frequency Channel Self-Attention Module(MFCSAM) to enhance features, mitigate shape bias in the pretrained features, and facilitate the model in obtaining richer information from the global context.

For a given input image I, we denote the output features of the last three stages of pretrained backbone network as:

$$X^I = \{X_1, X_2, X_3\} \tag{2}$$

As shown in Fig. 2, we employ specialized operations for different features. For the feature X_1, since it contains a large number of low-level information

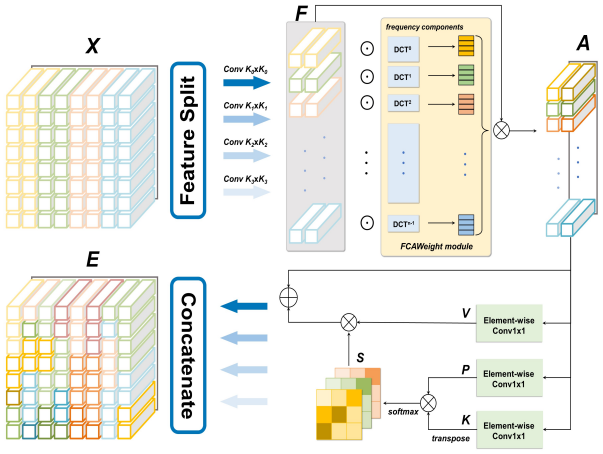

Fig. 3. The schematic diagram of the MFCSAM module is shown above. The module takes as input the pretrained features and first splits them into different scales using dedicated convolutional kernels. Next, frequency-domain attention and self-attention are applied to each scale to enable global feature learning, followed by the fusion of the multi-scale features to produce the final output.

that can guide small-size defects detection and shape features, we use a 1×1 convolution kernel to obtain its deeper information, and use MFCSAM to obtain its global information from multiple channels and different dimensions, Then, the two sets are combined as far as possible to obtain low-level information without the interference of useless information. For features X_2 and X_3, under the consideration of balancing parameters and effects, only use MFCSAM for the last layer of feature X_3 to obtain its highest level and richest information.

Multi-scale Frequency Channel Self-Attention. For feature information, our motivation is to build a more global and effective attention mechanism. Therefore, a novel Multi-scale Frequency Channel Self-Attention module is proposed. As illustrated in Fig. 3, the MFCSAM is mainly implemented in three steps.

In the multi-scale implementation, we use convolution kernels of different sizes, so the features of a single scale can be expressed as follows:

$$F_i = \mathrm{Conv}\left(k_i \times k_i\right)\left(X_i\right), \quad i = 0, 1, 2 \ldots S-1 \tag{3}$$

where the i-th kernel size $k_i = 2 \times (i+1) + 1$, and $F_i \in \mathbb{R}^{C' \times H \times W}$ denotes the feature map with different scales. By introducing frequency components to extract channel information from feature maps of each scale, the attention weight vectors of frequency channels at different scales can be obtained. Mathematically, the vector of channel attention vector can be represented as:

$$A_i = \ \mathrm{FCAWeight}\ \left(F_i\right) \tag{4}$$

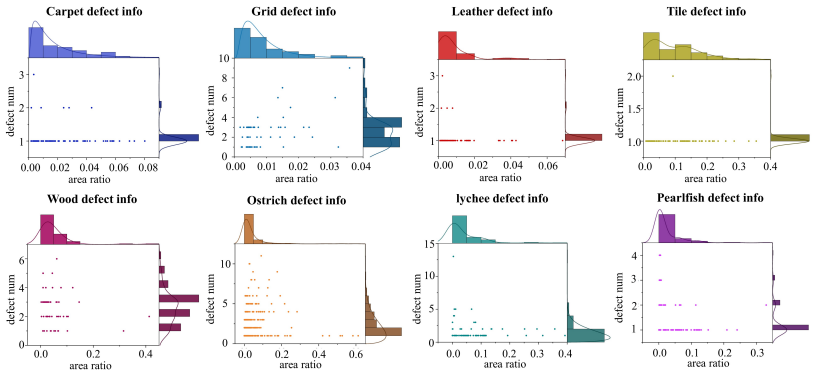

Fig. 4. Individual information of annotated bounding box for each of the 8 classes, including the proportion of defect area of this category and the number of defects in each image.

Where $A_i \in \mathbb{R}^{C' \times H \times W}$ is the split frequency channel attention vector, the FCAWeight module is used to generate frequency channel attention weights. After gaining frequency channel attention, we feed A_i into three convolution layers to generate three new feature maps P, K and V, respectively, where $\{P, K, V\} \in \mathbb{R}^{C' \times H \times W}$. Then we transpose and reshape them to $\mathbb{R}^{C' \times N}$, where $N = H \times W$ is the number of pixels. After that we perform a matrix multiplication between the transpose of K and P, and apply a softmax layer to calculate the position attention map $S \in \mathbb{R}^{N \times N}$:

$$s_{ji} = \frac{\exp\left(P_i \cdot K_j\right)}{\sum_{i=1}^{N} \exp\left(P_i \cdot K_j\right)} \tag{5}$$

where S_{ji} represents the i_{th} position's impact on j_{th} positions. Meanwhile, we perform a matrix multiplication between V and the transpose of S. After a leaky ReLU layer, we multiply it by a scale parameter and perform a element-wise sum operation with the feature A to obtain the output $E_j \in \mathbb{R}^{C' \times H \times W}$ as follows:

$$E_j = \alpha \sum_{i=1}^{N} \left(s_{ji} V_i\right) + A_j, \quad j = 0, 1, 2 \cdots S - 1 \tag{6}$$

where α is initialized as 0 and gradually learns to assign more weight. From this, we obtain features with channel and position attention, which come from fully convolutional layers without information loss, thus improving the consistency of anomaly detection classification and segmentation. Finally, we re-concatenate the split part as the output of MFCSAM, the final output can be represented by:

$$E = E_0 \oplus E_1 \oplus \ldots \oplus E_{s-1} \tag{7}$$

Table 1. Detection(I-AUROC) and localization(P-AUROC) (in %) of state-of-the-art methods on MVTec AD and MTL AD, before and after adding our method. The best result for each before-versus-after pair is highlighted in bold.

Dataset	Class	DRAEM +SSPCAB [20]	ViTLnet [24]	GLAD [2]	Pyramidflow [16]	PaDiM [8]		PatchCore [21]		CFA [15]	
							+Our method		+Our method		+Our method
Texture-based	Carpet	(98.2, 95.0)	(-, 98.9)	(99.0, 97.8)	(-, 97.4)	(99.5, **99.1**)	(**99.7**, 99)	(98.4, **98.8**)	(**99.2**, 98.7)	(**99.5**, 98.7)	(99.5, **99.1**)
	Grid	(100, 99.5)	(-, 97.8)	(98.7, 99.7)	(-, 95.7)	(94.2, 97)	(**95.6, 97.4**)	(95.9, 96.8)	(**98, 97.6**)	(99.2, 97.8)	(**99.6**, **98.6**)
	Leather	(100, 99.5)	(-, 99.7)	(100, 99.8)	(-, 98.7)	(100, **99.3**)	(100, 99.1)	(100, **99.1**)	(100, 99)	(100, 99.1)	(**100**, **99.5**)
	Tile	(100, 99.3)	(-, 97.5)	(99.6, 96.1)	(-, 97.1)	(97.4, **95.5**)	(**98.2**, 94.9)	(100, **96.1**)	(100, 95.4)	(99.4, 95.8)	(**100**, **97.1**)
	Wood	(99.5, 96.8)	(-, 97.4)	(99.1, 95.8)	(-, 97.0)	(**99.3**, **95.7**)	(99.2, 95)	(98.9, 93.4)	(**99.2, 95.1**)	(99.7, 94.8)	(**100**,**96.4**)
	Average	(99.54, 98.02)	(-, 98.3)	(99.1, 97.8)	(-, 97.18)	(98.08, **97.28**)	(**98.54**, 97.08)	(98.64, 96.84)	(**99.28**, **97.16**)	(99.56, 97.24)	(**99.82**, **98.14**)
object-based	Bottle	(98.4, 98.8)	(-, -)	(100, 96.9)	(-, 97.8)	(99.9,98.5)	(**100,98.7**)	(100.98.4)	(100,**98.8**)	(100,98.6)	(100,98.6)
	Cable	(96.9, 96.0)	(-, -)	(99.8, 98.6)	(-, 91.8)	(87.8, 97.0)	(**91.8,98.0**)	(99.0, **98.8**)	**99.2**,98.7	(99.8,**98.7**)	(**99.9**,98.6)
	Capsule	(99.3, 93.1)	(-, -)	(97.8, 98.7)	(-, 98.6)	(**92.7**,98.8)	(92.2,**99.0**)	(**98.2**,98.8)	(97.4,**99.2**)	(97.3,98.9)	(**98.2**,98.9)
	Hazelnut	(100, 99.8)	(-, -)	(99.8, 98.2)	(-, 98.1)	(**96.4**, 98.5)	(96.2,**98.6**)	(100,98.7)	(100,**98.9**)	(100,98.6)	(100,98.6)
	Metal_nut	(100, 98.9)	(-, -)	(99.4, 96.2)	(-, 97.2)	(98.9. 98.2)	(**99.2,98.6**)	(**99.4**,98.9)	(98.6,**99.3**)	(**100,98.8**)	(99.6,98.7)
	Pill	(99.8, 97.5)	(-, -)	(96.3, 96.2)	(-, 96.1)	(93.9, 96.6)	**94.7,96.8**	(92.4,**98**)	**92.5**,97.0	(97.9,**98.6**)	(**98.7**, 98.2)
	Screw	(97.9, 99.8)	(-, -)	(97.9, 99.9)	(-, 94.6)	(84.5, 98.8)	**87.2,98.9**	(96.0,98.9)	(**96.2, 99.5**)	(97.3,**99.0**)	(97.3,98.9)
	Toothbrush	(100, 98.1)	(-, -)	(100, 98.9)	(-, 98.5)	(94.2, 99.1)	(**99.7,99.2**)	(93.3,98.8)	(**100,99.0**)	(**100**,98.8)	(99.7,**98.9**)
	Transistor	(92.9, 87.0)	(-, -)	(99.6, 96.5)	(-, 96.9)	(**97.6**, 97.6)	(94.3,**98.7**)	(**100,98.1**)	(97.3,97.8)	(100,98.3)	(100,**98.4**)
	Zipper	(100, 99.0)	(-, -)	(99.9, 99.1)	(-, 96.6)	(88.2, 98.6)	(**91.0,98.8**)	(**98.2**,98.3)	(**96.4**,99.0)	(99.6,98.6)	(**99.7**,**98.8**)
	Average	(98.52, 96.8)	(-, -)	(99.0, 97.9)	(-, 96.62)	(93.41, 98.17)	(**94.55,98.29**)	(97.65, 98.57)	(97.76, 98.72)	(99.19,**98.69**)	(**99.31**,98.66)
Overall		(98.86, 97.20)	(-, -)	(99.1, 97.9)	(-, 96.80)	(95.,97.9)	(**95.93,98.04**)	(98, 98)	(98.27, 98.20)	(99.3,98.2)	(**99.48, 98.48**)
MTL AD Dataset	Ostrich	(-, -)	(-, -)	(-, -)	(-, -)	(72.7, 74.8)	(**73.7, 75.6**)	(84.8, **82.3**)	(**86.9**, 80.8)	(91.3, **86.7**)	(**93.2**, 86.4)
	Lychee	(-, -)	(-, -)	(-, -)	(-, -)	(**87.9**, 91.6)	(87.5, **92.7**)	(**75.2**, 89.4)	(73.9, **89.9**)	(87.7, 93.9)	(**93.6**, **94.8**)
	Pearlfish	(-, -)	(-, -)	(-, -)	(-, -)	(71.4, 82.9)	(**73.4, 84.9**)	(75.3, 87.5)	(**81.2, 90.9**)	(79.5, 91.3)	(**82.8**, **91.8**)
	Average	(-, -)	(-, -)	(-, -)	(-, -)	(77.33, 83.10)	(**78.20, 84.40**)	(78.43 86.40)	(**80.67, 87.20**)	(86.17, 90.63)	(**89.86**, **91**)

4 Results and discussion

4.1 DataSets

In this paper, we use MVTec AD dataset and our own MTL AD dataset to conduct experiments of our proposed method. MVTec AD dataset is a commonly used anomaly detection dataset, which contains images from 10 object categories and 5 texture categories. The number and size distribution of MTL AD dataset is shown in Fig. 4. From the size comparison, we can see that the size of defects in our dataset is more diverse and in line with the actual situation, while the defects in MVTec AD data set are generally larger and easier to identify.

4.2 Experimental setup

Since the input image size of this experiment is generally larger than 900x900, in order to reduce the amount of calculation, we resize the image and center cropped it to 224x224. The GPU is Nvidia RTX 3080Ti, and the CPU is 12th Gen Intel(R) Core(TM) i7-12700 to measure the throughput of the proposed method. Anomaly detection is generally divided into defect detection and localization. Referring to previous methods [8,21], we adopt Area Under the Receiver Operator Curve (AUROC), and then evaluated the performance of the proposed method in terms of anomaly detection and localization.

4.3 Quantitative Results

Benchmark. We choose PaDiM [8], PatchCore [21] and CFA [15] models as benchmark models for anomaly detection. In addition, we introduce state-of-the-

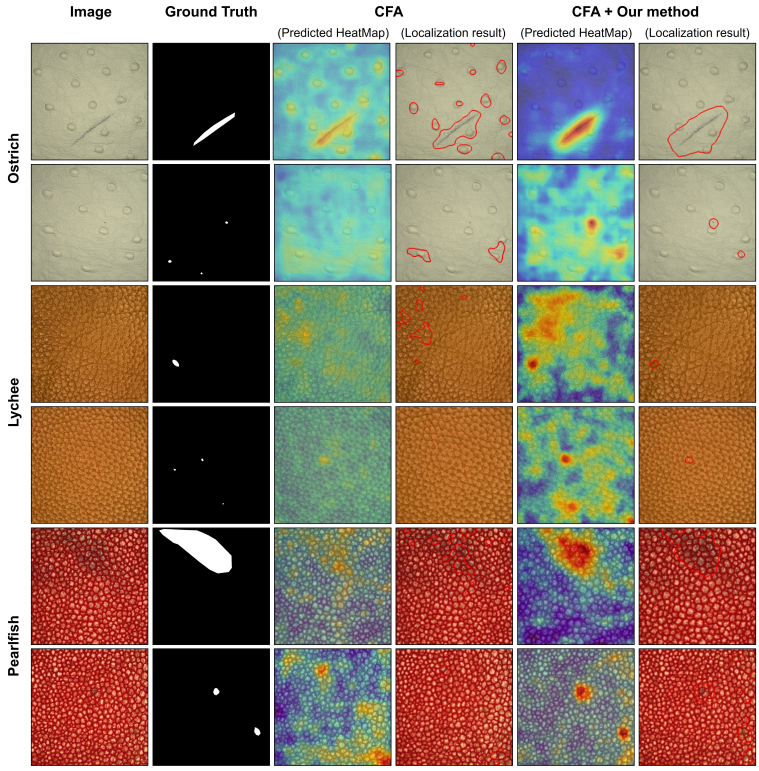

Fig. 5. Results on the MTL AD Dataset using the CFA model and adding our method. From left to right, each column is the predicted heatmap and localization results of the original image, GT, and the original CFA model (represented by the red line) and the predicted heatmap and localization results of the CFA model after adding our method(represented by the red line).

art algorithms for result comparison, including sspcab [20], PyramidFlow [16], GLAD [2], and ViTALnet [24].

Results. We present our results in table 1. From the results of texture-based classes of both MVTec AD and MTL AD datasets, it can be seen that except for the 0.2% loss in the localization effect of the PaDiM [8] model, the rest of the models all have a certain degree of improvement after adding our method, among PatchCore [21] model achieved a 0.64% improvement in the detection effect of MVTec AD data set. For the CFA [15] model, since the model utilizes the features of all layers and is adaptive to specific tasks, it is more suitable for our tasks for texture and shape, and the performance is also the best: the detection effect for MTL AD is improved by 3.69% and the detection effect for MVTec AD is improved by 0.18% overall. Moreover, compared to the state-of-the-art methods, the CFA algorithm with our proposed enhancements achieves

the optimal results in almost every category. Except for a 0.16 lower score in texture-based categories compared to ViTLnet, which is a specialized algorithm for texture-based categories.

Table 2. Average inference time (in milliseconds) for two frameworks [8], [15], before and after integrating our method,respectively.The running times are measured on an Nvidia GeForce GTX 3080Ti GPU with 12 GB of VRAM.

Method	Time(ms)	
	Baseline	+Our method
PaDiM [8]	503	507
CFA [15]	129	131

Ablation Study. To illustrate the effectiveness of the proposed method, we use the CFA [15] model for ablation experiments. We added the components of our method, i.e., SBM and PFAM, to the model to see the final effect. For SBM and PFAM, in table 1, we show the detection and localization effect of the CFA [15] model in texture-based and object-based classes according to the idea of whether to use our method. The results proves that our method is effective on most classes. It can also be seen from the visualization results in Fig. 5 that our module can better locate and detect industrial product defects.In summary, the experimental results prove that our method has a good ability to deal with the problem of shape-bias, and can more comprehensively utilize the pretrained model for anomaly detection on the surface of industrial products.

Inference time. Regardless of the underlying framework [15], [8], referring to Madan et al. [18] for the testing method of embeddable modules, we add the two modules of this paper, SBM and PFAM. To evaluate the additional amount of time for adding modules in this paper, we show the running time before and after integrating our method into two state-of-the-art frameworks [15], [8] in Table 4.3. For both baseline models, the time after adding our method is at most 0.5ms higher. Furthermore, the computation time of CFA differs by no more than 0.2 milliseconds relative to the original baseline.

5 Conclusion

Most anomaly detection methods use pretrained convolutional models to extract nominal data features. Due to the shape-bias, these features can have side effects when faced with shape or texture products. In this paper, We propose a novel framework to remove local bias without reducing the features of convolutional

layers. To show the superiority of our method, we combine it with current state-of-the-art models [8, 15, 21]. We demonstrate that each module in the method is necessary through extensive experiments on MVTec AD [5] and MTL AD dataset. Although a certain amount of computation is increased, our method can overcome shape-bias from pretrained model to a certain extent, and can achieve improvements at the image/pixel level of anomaly detection. In future work, we will continue to study the anomaly detection of industrial product defects, and explore how to improve the detection effect with as less computational cost as possible.

References

1. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE transactions on pattern analysis and machine intelligence **30**(3), 555–560 (2008)
2. Artola, A., Kolodziej, Y., Morel, J.M., Ehret, T.: Glad: A global-to-local anomaly detector. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5501–5510 (2023)
3. Asadi, N., Sarfi, A.M., Hosseinzadeh, M., Karimpour, Z., Eftekhari, M.: Towards shape biased unsupervised representation learning for domain generalization. arXiv preprint arXiv:1909.08245 (2019)
4. Bergman, L., Cohen, N., Hoshen, Y.: Deep nearest neighbor anomaly detection. arXiv preprint arXiv:2002.10445 (2020)
5. 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)
6. Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., Tommasi, T.: Domain generalization by solving jigsaw puzzles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2229–2238 (2019)
7. Chen, P., Liu, S., Jia, J.: Jigsaw clustering for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11526–11535 (2021)
8. 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)
9. Feng, Y., Yuan, Y., Lu, X.: Learning deep event models for crowd anomaly detection. Neurocomputing **219**, 548–556 (2017)
10. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3146–3154 (2019)
11. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313**(5786), 504–507 (2006)

14. Lasi, H., Fettke, P., Kemper, H.G., Feld, T., Hoffmann, M.: Industry 4.0. Business & information systems engineering **6**(4), 239–242 (2014)
15. Lee, S., Lee, S., Song, B.C.: Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. arXiv preprint arXiv:2206.04325 (2022)
16. Lei, J., Hu, X., Wang, Y., Liu, D.: Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14143–14152 (2023)
17. Liang, Y., Zhang, J., Zhao, S., Wu, R., Liu, Y., Pan, S.: Omni-frequency channel-selection representations for unsupervised anomaly detection. arXiv preprint arXiv:2203.00259 (2022)
18. Madan, N., Ristea, N.C., Ionescu, R.T., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M.: Self-supervised masked convolutional transformer block for anomaly detection. arXiv preprint arXiv:2209.12148 (2022)
19. 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)
20. Ristea, N.C., Madan, N., Ionescu, R.T., Nasrollahi, K., Khan, F.S., Moeslund, T.B., Shah, M.: Self-supervised predictive convolutional attentive block for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13576–13586 (2022)
21. 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)
22. 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)
23. Shi, H., Zhou, Y., Yang, K., Yin, X., Wang, K.: Csflow: Learning optical flow via cross strip correlation for autonomous driving. arXiv preprint arXiv:2202.00909 (2022)
24. Tao, X., Adak, C., Chun, P.J., Yan, S., Liu, H.: Vitalnet: Anomaly on industrial textured surfaces with hybrid transformer. IEEE Transactions on Instrumentation and Measurement **72**, 1–13 (2023)
25. Yang, M., Wu, P., Liu, J., Feng, H.: Memseg: A semi-supervised method for image surface defect detection using differences and commonalities. arXiv preprint arXiv:2205.00908 (2022)
26. 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. pp. 665–674 (2017)