MLDFR: A Multilevel Features Restoration Method Based on Damaged Images for Anomaly Detection and Localization

Yinghui Guo, Meng Jiang, Qianhong Huang, Yang Cheng, and Jun Gong

***Abstract*—For unsupervised anomaly detection and localization, a common approach is learning the distribution of normal samples and then use it as a criterion to identify abnormalities. This article proposes a multilevel features restoration method based on damaged images (MLDFR) for anomaly detection and localization. MLDFR seeks to restore the "normal feature" of the test sample. Specifically, we damage the training samples to generate the corresponding samples, and then design a concurrent feature extractor utilizing convolutional neural network and transformer pretrained on ImageNet to completely represent the multilevel features of samples. Additionally, we fully consider the dependencies among local features over long distances and design a feature restoration module. On the challenging, widely used anomaly detection datase (MVTec-AD), metal parts defect detection dataset (MPDD), and beantech anomaly detection dataset (BTAD) of real-world datasets, MLDFR achieves state-of-the-art anomaly localization performance, as well as image-level detection with virtually flawless score. We further report ablation studies, demonstrating MLDFR's effectiveness and generalizability.**

***Index Terms*—Anomaly detection (AD), anomaly localization (AL), features restoration.**

I. Introduction

ANOMALY detection (AD) and anomaly localization (AL) are the important section of industrial intelligent manufacturing [1] and widely used in many fields [2]. It is challenging because the types of anomalies are complex and diverse, and it is difficult to obtain abnormal samples in actual production. A common approach is unsupervised learning, which refers learning the distribution of normal samples and using that as a criterion to identify abnormalities.

In the recent development, deep neural networks have dramatically advanced the task with more powerful performances for representation learning. In the image space, an intuitive and effective method is to reconstruct the normal image of the test sample, and use the difference before and after restoration to detect and locate anomalies [3], [4], [5], [6]. Whereas, this pixel-level image reconstruction is vulnerable to noise interference, resulting in poor robustness of detection. Using pretrained networks to build feature representations in the feature space is an effective approach of detection. However, storing features and then retrieving "normal templates" incurs a significant storage overhead [10], [11], and modeling a probability distribution of features limits template diversity [12], [13], [14].

In order to comprehensively solve the abovementioned problems, we propose a new method based on multilevel damaged features restoration for AD and AL, termed multilevel features restoration method based on damaged images (MLDFR). We believe that images are characterized by low-density information, and restoration of all pixels is not required for AD and AL. Retaining most of the image structure is useful to improve the restoration effect. Restoring the most similar "normal templates" in the feature space capable of increasing the diversity of templates without storage.

We show the training framework of MLDFR in Fig. 4. Specifically, we obtain damaged samples by randomly corrupting normal samples before the start of each training round. In order to fully represent the multilevel features of the sample, we design a concurrent feature extractor utilizing the convolutional neural network (CNN) and transformer pretrained on ImageNet. To improve the feature restoration capability. We design the feature restoration module to capture the dependencies between local features over long distances. Then, we train the feature restoration module by using the features of normal samples as targets and the features of damaged samples as input. In inference, MLDFR first extracts the multilevel features of the test samples, then feeds them into the feature restoration model, and finally uses the difference before and after restoration to generate sufficient score maps for AD and AL. We describe the process of inference stage in Fig. 1. In summary, our contributions are as follows.

1) We propose a simple, yet effective method (MLDFR) for AD and AL. MLDFR is able to restore the normal

Manuscript received 7 March 2023; revised 21 May 2023; accepted 29 June 2023. Date of publication 18 July 2023; date of current version 19 January 2024. This work was supported by the National Key Research and Development Program of China under Grant 2021YFF0901300. Paper no. TII-23-0775. *(Corresponding author: Jun Gong.)*

Yinghui Guo, Meng Jiang, Qianhong Huang, and Yang Cheng are with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: 2100744@stu.neu.edu.cn; 2210312@stu.neu.edu.cn; 2100753@stu.neu.edu.cn; 20194110@stu.neu.edu.cn).

Jun Gong is with the College of Information Science and Engineering and the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China, and also with the Midea Key Laboratory Pattern Recognition and Artificial Intelligence, Foshan 528311, China (e-mail: gongjun@ise.neu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3292904.

Digital Object Identifier 10.1109/TII.2023.3292904

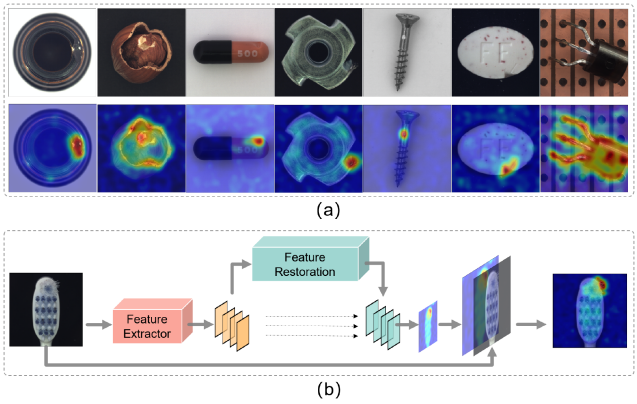

Fig. 1. (a) AD examples on MVTec. From top to bottom: abnormal samples and heatmap of pixel-level predictions. (b) Conceptual map for locating abnormal areas.

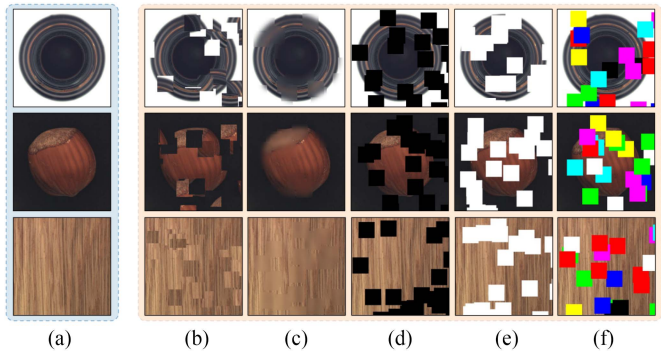

Fig. 2. Visualization of random damage. (a) Normal samples. (b)–(f) Damaged samples, bottle, hazelnut, and wood from MVtec, the damage methods include (b) cut paste, (c) Gaussian filter, (d) mask, (e) white, (f) RGB.

features that are most similar to the test sample, and use the difference before and after restoration to achieve AD and AL.

2) We introduce a concurrent feature extractor utilizing CNN and transformer pretrained on ImageNet, which can aggregate low-dimensional texture and high-dimensional semantic information of samples.
3) We propose a feature restoration module that can adequately capture long-distances dependencies and improve feature restoration capability.
4) Extensive experiments on several real-world datasets have shown that MLDFR is powerful for industrial detection, substantially outperforming previous methods.

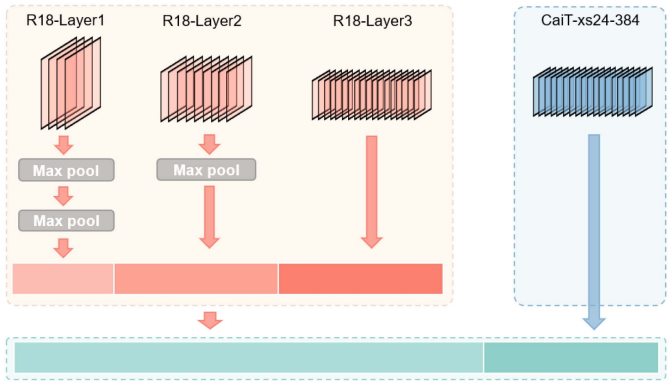

Fig. 3. Multilevel features fusion (MLFF). MLFF retains local features and global representation to the maximum extent.

II. Related Work

We group previous work on unsupervised AD into the following two main categories: image similarity and feature similarity. We will highlight the differences between our proposed MLDFR approach and previous work.

A. Image Similarity

These approaches usually compare at the image pixel level, and the core idea is to reconstruct the normal image that most closely resembles the input sample. They usually use autoencoder (AE), variational auto-encoders (VAE), or generative adversarial networks (GANs) to learn the distribution of normal data and then make judgments based on the differences between before and after reconstruction of test samples. The quality of the reconstructed image has a great impact on the detection performance, and to improve the reconstruction clarity, deep feature reconstruction (DFR) [3] carry out related designs for multiscale feature information, which provides different fine-grained context information for image reconstruction. Skip-GANomaly [4] refers to the structure of U-Net, which introduces cross layer connection to provide more spatial information for the decoder. However, recent studies have shown that the excellent generalization ability of deep models can also reconstruct anomalous regions [5]. For this reason, memory mechanisms [6] and image masking strategies [5] have been incorporated into reconstruction-based approaches. Nonetheless, these pixel-level image reconstruction methods still lack strong discriminative power for real-world AD [8].

B. Feature Similarity

These approaches transform from image space to feature space with the feature extraction capability of deep neural networks. The core idea is to find distinguishable feature embeddings and reduce the interference of irrelevant features. One of the ideas is to match "normal templates" by storing features. Cohen N [10] adopts a CNN pretrained network with a multiscale pyramid pool, and then conducts k-means clustering. However, this approach comes with additional space overhead. To address this issue, PatchCore [11] introduce local neighborhood aggregation to explore the feature extraction part. Another idea is to model the probability distribution of normal sample features. Rippel et al. [12] extract features using a pretrained network and model each feature map as a multivariate Gaussian distribution, and in the test stage, use the Mahalanobis distance between the feature vector to be tested and the "normal" distribution to measure the presence of defects in the sample. Dedard et al. [13] further model the multivariate Gaussian distribution at the

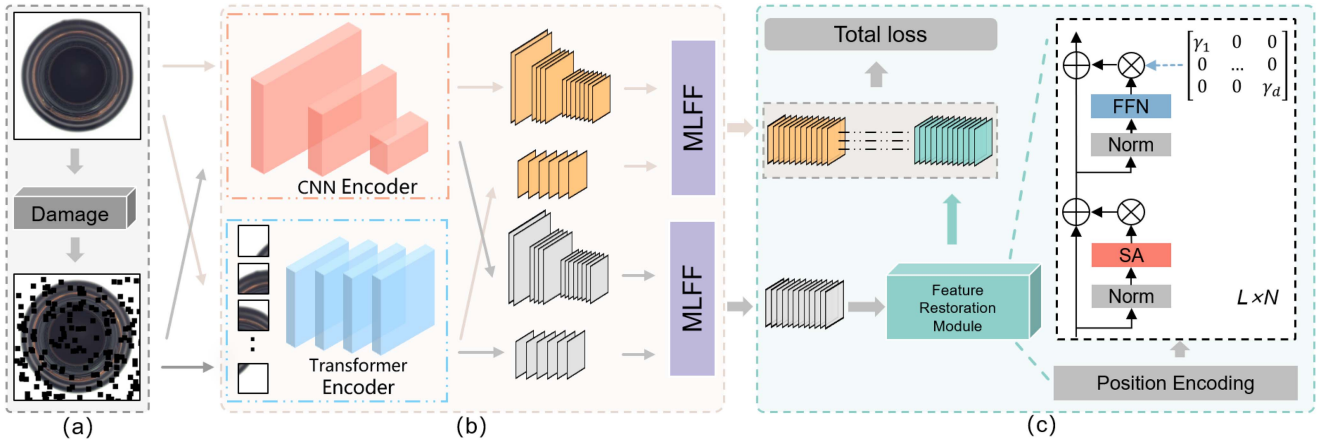

Fig. 4. Overview of MLDFR training framework. (a) To obtain the corresponding damaged samples during the training stage, the normal samples are randomly damaged. (b) Concurrent feature extractor composed of CNN and Transformer pretrained networks is used to extract features. Then, multilevel features of normal samples and abnormal samples are obtained by MLFF module. (c) Feature restoration module is trained by using the features of normal samples as targets and the features of damaged samples as input.

fine-granularity of the image blocks, thus achieving pixel-level segmentation, but this approach requires a high degree of strict pixel alignment. Rudolph et al. [14] introduce the normalizing flow (NF) model to improve the flexibility. Nonetheless, the NF is difficult to handle out-of-distribution problem due to its focus on low-dimensional features [9].

Unlike the abovementioned approaches, we introduce the concept of image restoration into the feature space and design a concurrent feature extractor using a pretrained network of CNN and transformer to reconstruct the best "normal feature" from the multilevel features of the damaged image. In the inference stage, AD and AL are realized by using the differences before and after feature restoration of test samples.

III. Proposed MLDFR Framework

A. Destroy Normal Samples

Superimposed noise, attribute elimination, jigsaw puzzle restoration, and cut paste are popular methods for constructing abnormal samples [18]. Different from the abovementioned methods, MLDFR does not simulate actual defects and only damages images arbitrarily and irregularly. We conjecture that damaged samples can retain part of the image structure, which is conducive to feature restoration. Our aim is to design a simple and universal image damage method. We show five distinct damage modes in Fig. 2. The specific damage steps are as follows.

1) Set the side length d of an area, which can be any size smaller than the image size.
2) Randomly select N square areas with the same size of $d \times d$ on the training image.
3) Damage the pixel value of this area.

B. Features Extraction and Fusion

In order to fully describe the sample features. We explore a concurrent multilevel features extraction method based on different pretrained networks. The motivation behind this is that CNN can extract local descriptors for low-level information, while the ViT [28] has wider receptive fields and can extract global representation and structural information. The shallow layers extracted by the pretrained CNN network show good results in the AD task [10], [11], [26], so we select the output of the last layer in the first three blocks of the pretrained ResNet18 [19] to represent local descriptors for low-level information. We select the deep layer output of the pretrained CaiT [20] to represent the high-level features of normal samples, while select the shallow layer output to represent the high-level features of damaged samples. We found that shallow layer output of CaiT can conducive to feature restoration.

To achieve representation alignment in feature concatenation, we set input size of Resnet18 as same as CaiT-xs24-384, followed down-sampling the shallow layers outputs of ResNet18 through one or more maximum pooling layers with stride of 2, to aligning it with the high-level features extracted by CaiT. Then, concatenate these features along the channel. We describe the process of multilevel features fusion in Fig. 3. Then, the extracted multilevel features are recorded as

$$\begin{cases} FX_i = MLFF\left[f_{\text{cnn}}\left(x\right), f_{\text{vit}}\left(x\right)\right] \\ FX_i' = MLFF\left[f_{\text{cnn}}\left(x'\right), f_{\text{vit}}\left(x'\right)\right] \end{cases} \tag{1}$$

where x represents normal sample, x' represents a corresponding damaged sample, $f(\cdot)$ is a feature extractor, and MLFF$[\cdot]$ represents multilevel features fusion. And i represents the index of channels in the features. FX_i represents features of normal sample and FX_i' represents features of damaged sample.

C. Feature Restoration Module

Damaged areas are randomly distributed in the position of the image and dispersed in feature maps through the pretrained network. Creating reasonable textures and structures and for damaged regions require context understanding, and it is difficult to describe the semantic correspondence among remote regions

TABLE I
COMPARISON OF PREVIOUS IMAGE-SIMILARITY-BASED METHODS AND MLDFR ON THE MVTEC DATASET USING IMAGE-LEVEL AUROC AND PIXEL-LEVEL AUROC METRIC

Method	Skip-GANomaly [4]	MemAE [6]	DAGAN [7]	TrustMAE [29]	RIAD [13]	InTra [23]	**MLDFR**
Texture	82.6/-	89.0/85.2	93.1/-	90.7/93.8	95.1/93.9	98.9/96.1	**99.5/98.8**
Objects	79.5/-	78.3/86.0	84.0/-	86.7/94.0	89.9/94.3	93.0/96.9	**99.3/98.5**
Total Average	80.5/-	81.9/85.7	87.3/-	88.0/93.9	91.7/94.2	95.0/96.6	**99.4/98.6**

TABLE II
DETAILED COMPARISON OF PREVIOUS FEATURE-SIMILARITY-BASED METHODS AND MLDFR ON THE MVTEC DATASET FOR EVERY CLASS USING IMAGE-LEVEL AUROC, PIXEL-LEVEL AUROC, AND PRO METRIC

	Method	DifferNet [14]	CSFLOW [24]	DPAEM+SSPCB [25]	PFM [22]	Reverse Distillation [26]	PatchCore [11]	**MLDFR**
textures	Carpet	84.0/-/-	100.0/-/-	98.2/95.0/59.4	**100.0**/99.2/96.9	98.9/98.9/97.0	98.7/98.9/-	98.9/**99.3**/**98.5**
	Grid	97.1/-/-	99.0/-/-	**100.0**/**99.5**/61.1	98.0/98.8/96.0	**100.0**/99.3/97.6	98.2/98.7/-	**100.0**/**99.3**/**97.9**
	Leather	99.4/-/-	100.0/-/-	**100.0**/99.5/76.0	100.0/99.4/98.8	**100.0**/99.4/99.1	**100.0**/99.3/-	**100.0**/**99.6**/**99.4**
	Tile	92.9/-/-	100.0/-/-	**100.0**/**99.3**/95.0	99.6/96.2/88.7	99.3/95.6/90.6	98.7/95.6/-	99.2/98.3/**96.1**
	Wood	99.8/-/-	100.0/-/-	**99.5**/96.8/77.1	99.5/95.6/92.6	99.2/95.3/90.9	99.2/95.0/-	99.4/**97.3**/**94.6**
	Average	94.6/-/-	**99.8**/-/-	**99.5**/98.0/73.7	99.4/97.8/94.6	99.5/97.7/95.0	99.0/97.5/-	**99.5**/**98.8**/**97.3**
objects	Bottle	99.0/-/-	99.8/-/-	98.4/98.8/87.9	100.0/98.4/95.4	100.0/98.7/96.6	100.0/98.6/-	**100.0**/**98.9**/**96.8**
	Cable	86.9/-/-	99.1/-/-	96.9/96.0/57.2	98.8/96.7/94.2	95.0/97.4/91.0	**99.5**/**98.4**/-	97.8/**98.4**/**93.5**
	Capsule	88.8/-/-	97.1/-/-	99.3/93.1/50.2	94.5/98.3/91.7	96.3/**98.7**/**95.8**	98.1/**98.8**/-	**99.2**/98.3/96.3
	Hazelnut	99.1/-/-	99.6/-/-	**100.0**/**99.8**/92.6	**100.0**/**99.1**/**96.7**	99.9/98.9/95.5	100.0/98.7/-	**100.0**/98.9/96.1
	Metal nut	95.1/-/-	99.1/-/-	**100.0**/**98.9**/98.1	100.0/97.2/94.6	100.0/97.3/92.3	100.0/98.4/-	**100.0**/**98.0**/**92.5**
	Pill	95.9/-/-	98.6/-/-	**99.8**/97.5/52.4	96.5/97.2/96.1	96.6/98.2/96.4	96.6/97.1/-	99.3/**98.3**/**96.9**
	Screw	99.3/-/-	97.6/-/-	97.9/99.8/72.0	91.8/98.7/93.4	97.0/99.6/98.2	98.1/99.4/-	**99.3**/**99.7**/**98.8**
	Toothbrush	96.1/-/-	91.9/-/-	**100.0**/98.1/51.0	88.6/98.6/90.7	99.5/**99.1**/**94.5**	100.0/98.7/-	98.9/98.8/92.3
	Transistor	96.3/-/-	99.3/-/-	92.9/87.0/48.0	97.8/87.8/74.9	96.7/92.5/78.0	100.0/96.3/-	**99.7**/**96.7**/**87.6**
	Zipper	98.6/-/-	99.7/-/-	**100.0**/**99.0**/77.1	97.4/98.2/94.8	98.5/98.2/95.4	98.8/98.8/-	99.1/**99.2**/**97.5**
	Average	95.5/-/-	98.2/-/-	98.5/96.8/68.6	96.5/97.0/92.3	98.0/97.9/93.4	99.1/98.3/-	**99.3**/**98.5**/**94.8**
Total Average		95.2/-/-	98.7/-/-	98.9/97.2/69.9	97.5/97.3/93.0	98.5/97.8/93.9	99.1/98.0/-	**99.4**/**98.6**/**95.7**

TABLE III
COMPARISON OF PREVIOUS METHODS USING THE SAME BACKBONE ON THE MVTEC DATASET USING IMAGE-LEVEL AUROC AND PIXEL-LEVEL AUROC METRIC

Backbone	ResNet18			WideResNet-50			
Method	PaDiM [13]	CutPaste [18]	**MLDFR**	SPADE [10]	PaDiM [13]	DRAEM [15]	**MLDFR**
Texture	95.6/91.3	97.5/**96.3**	**97.9**/96.2	92.9/88.4	96.9/93.2	99.1/**97.9**	**99.5**/96.2
Objects	97.3/89.4	95.5/95.8	**98.7**/**98.0**	97.6/93.4	97.8/91.6	97.4/97.0	**98.6**/**97.7**
Total Average	96.7/90.1	96.1/96.0	**98.4**/**97.4**	96.5/91.7	97.5/92.1	98.0/**97.3**	**98.9**/97.2

TABLE IV
COMPARISON OF PREVIOUS METHODS ON THE MPDD DATASET USING IMAGE-LEVEL AUROC AND PIXEL-LEVEL AUROC METRIC

Method	Skip-GANomaly [4]	DAGAN [7]	SPADE [10]	PaDiM [13]	CFLOW-AD [27]	PatchCore [11]	**MLDFR**
Bracket Black	61.3/88.9/-	68.6/89.7/-	44.7/94.3/-	75.6/94.2/-	72.7/96.9/-	81.9/**98.4**/-	**97.9**/97.6/**93.8**
Bracket Brown	62.1/78.1/-	77.1/81.5/-	91.0/97.2/-	85.4/92.4/-	88.8/97.8/-	78.4/91.5/-	**100.0**/**96.0**/**84.0**
Bracket White	73.3/78.8/-	72.1/70.6/-	79.9/96.8/-	82.2/98.1/-	87.8/98.6/-	76.0/97.4/-	**99.9**/**99.2**/**98.1**
Connector	72.6/80.2/-	99.8/85.7/-	95.2/98.4/-	91.7/97.9/-	94.7/98.4/-	96.7/95.0/-	**99.8**/**99.4**/**96.3**
Metal Plate	73.2/89.7/-	85.4/89.9/-	95.6/93.0/-	56.3/92.9/-	99.5/98.2/-	100.0/96.6/-	**100.0**/**99.1**/**96.4**
Tubes	46.4/77.3/-	31.9/82.3/-	56.1/95.9/-	57.5/93.9/-	73.1/96.4/-	59.7/95.1/-	**99.4**/**99.6**/**99.2**
Average	64.8/82.2/-	72.5/83.3/-	77.1/95.9/-	74.8/96.7/-	86.1/97.7/-	82.1/95.7/-	**99.5**/**98.5**/**94.6**

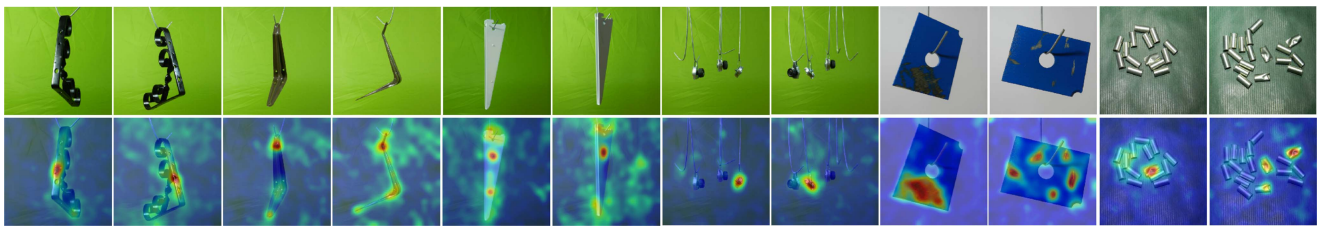

Fig. 5. AL results where MLDFR can precisely segment the anomalous regions. From top to bottom: abnormal samples and heatmap of pixel-level predictions.

TABLE V
COMPARISON OF PREVIOUS METHODS AND MLDFR ON THE BTAD DATASET USING IMAGE-LEVEL AUROC, PIXEL-LEVEL AUROC METRIC

Method	AE (MSE)	AE (MSE+SSIM)	VT-IDL	**MLDFR**
01	-/49	-/53	**-/99**	**100.0**/97.8
02	-/92	-/96	-/94	**91.2/96.9**
03	-/95	-/89	-/77	**99.7/99.7**
Average	-/78.7	-/79.3	-/90	**97.0/98.1**

TABLE VI
COMPARISON OF PREVIOUS METHODS IN TERMS OF INFERENCE TIME (SECOND), MEMORY USAGE (MB), AND PERFORMANCE (AD-AUROC/AL-AUROC) ON MVTEC

Method	Infer. time	Memory	Performance
SPADE [10]	1.31	1400	85.5/96.5
PaDiM [13]	0.90	3800	95.5/97.5
Reverse Distillation [26]	**0.07**	352	98.5/97.8
Pacthcore [11]	0.86	262	99.1/98.0
MLDFR	0.13	**198**	**99.4/98.6**

using fully CNNs. However, transformer is a natural architecture for dealing with nonlocal modeling. In nonlocal modeling, attention is a basic component in each block. We customize the transformer block for feature embedding based on ViT [28], because it has excellent context representation capability. Furthermore, we refer to CaiT to introduce "layerscale" [20]. Specifically, we add a learnable diagonal matrix to the output of each residual block, which helps to improve the dynamics of training and can train deeper, high-capacity transformers. Then, we add a position coding layer with the same size as the multilevel features to form the feature restoration module. In addition, we remove the class token because we found it unnecessary for feature restoration. We describe the feature restoration module in Fig. 4(c). First, features are position-encoded and then input to layer normalization. Where SA represents self-attention, then the input processing of features is as follows:

$$x_l' = x_l + \mathrm{SA}\left(\mathrm{norm}\left(x_l\right)\right) \tag{2}$$

where l represents the number of blocks and x_l represents the features after position encoding. Then, input x_l' into layer normalization and feed-forward networks. γ is also a learnable parameter and they form a diagonal matrix. Then, the processed features are as follows:

$$x_l'' = x_l' + \mathrm{diag}\left(\gamma_{l,1}, \ldots, \gamma_{l,d}\right) \times \mathrm{FFN}\left(\mathrm{norm}\left(x_l'\right)\right). \tag{3}$$

D. Loss Function

The main target loss is the average mean square error (MSE) of each channel before and after feature restoration as follows:

$$\mathcal{L}_{\mathrm{val}} = \frac{1}{N}\sum_{i}^{N}\left(\mathrm{FR}\left[\mathrm{FX}_i'\right] - \mathrm{FX}_i\right)^2 \tag{4}$$

where $\mathrm{FR}[\cdot]$ represents the feature restoration. And N represents the number of channels in the multilevel features. When using the concurrent multilevel features extractor of ResNet18 and CaiT, we introduce cosine similarity. Specifically, we calculate the cosine similarity of the feature vectors before and after restoration along the channel axis as follows:

$$\mathcal{L}_{\mathrm{dir}} = \frac{1}{N}\sum_{i}^{N}\left(1 - \frac{\mathrm{vec}\left(\mathrm{FR}\left[\mathrm{FX}_i'\right]\right)^T \cdot \mathrm{vec}\left(\mathrm{FX}_i\right)}{\left\|\mathrm{vec}\left(\mathrm{FR}\left[\mathrm{FX}_i'\right]\right)\right\| \left\|\mathrm{vec}\left(\mathrm{FX}_i\right)\right\|}\right) \tag{5}$$

where $\mathrm{vec}(\cdot)$ is the vectorization function that converts an arbitrary dimensional matrix into a one-dimensional vector. The total training loss of the feature restoration module is as follows:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{val}} + \lambda \cdot \mathcal{L}_{\mathrm{dir}} \tag{6}$$

where λ is set to make the scale of both constituent terms the same approximately.

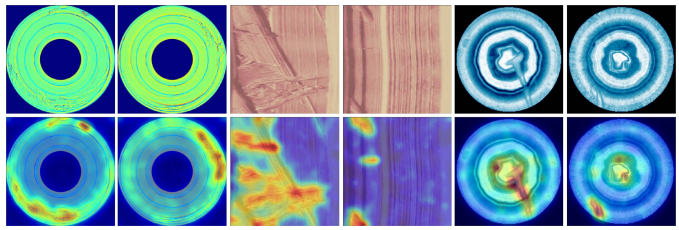

Fig. 6. AL results where MLDFR can precisely segment the anomalous regions. From top to bottom: abnormal samples and heatmap of pixel-level predictions.

E. Anomaly Localization and Detection

For pixel-level AL, we calculate the pixel-level anomaly score. We directly input the test sample without damaging it. Get its own multilevel features through the concurrent feature extractor as follows:

$$\mathrm{FS}_i = \mathrm{MLFF}\left[f_{\mathrm{cnn}}\left(s\right), f_{\mathrm{vit}}\left(s\right)\right]. \tag{7}$$

Then, the features are input into the restoration network and the most similar normal features are reconstructed. We calculate the MSE of the feature points under each channel, and then take the average value along the channel axis as the anomaly score. In order to locate the anomaly in the query image, the score map is up-sampled to the size of the input image. Let Ψ be the bilinear up-sampling operation used in this study. Then, an accurate score is as follows:

$$S = \sum_{i}^{N}\Psi[(\mathrm{FR}[\mathrm{FS}_i] - \mathrm{FS}_i)^2]. \tag{8}$$

Finally, we smooth the score map through a Gaussian filter to remove the noise in the score map refer to [10], [11], [13]. For image-level AD, set the threshold value according to all values in the score map.

IV. EXPERIMENTS

A. Experimental Settings

We will evaluate our proposed MLDFR approach on the three datasets: MVTec (2019) [8], BTAD (2021) [21], and MPDD (2021) [16]. In addition, we report ablation studies on a widely used MVTec dataset to explore the effects of different modules

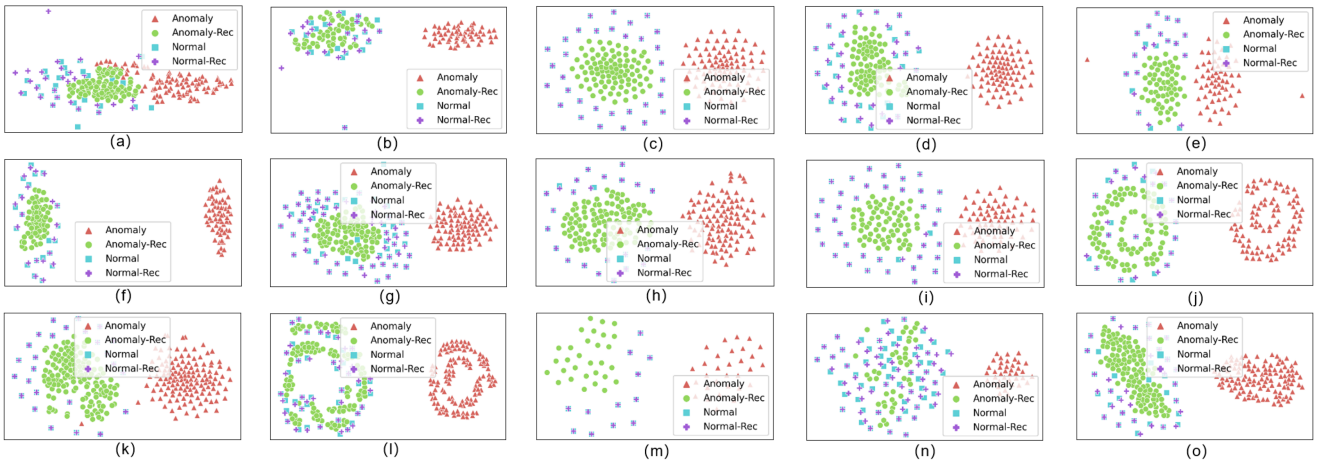

Fig. 7. t-SNE visualization of features that test samples were reconstructed before and after. We plot embeddings of anomaly (red), anomaly restoration (green), normal (blue), normal restoration (purple). (a) Carpet. (b) Grid. (c) Leather. (d) Tile. (e) Wood. (f) Bottle. (g) Cable. (h) Capsule. (i) Hazelnut. (j) Metal nut. (k) Pill. (l) Screw. (m) Toothbrush. (n) Transistor. (o) Zipper.

TABLE VII
DETECTION RESULTS USING DIFFERENT NUMBER OF LAYERS

Backbone		CaiT			ViT	R18				WR50	R18 + CaiT
Layer number		5−5	15−15	15−5	8−4	1	2	3	1+2+3	1+2+3	
Texture	Image AUROC	98.5	99.3	**99.4**	99.4	94.6	79.5	61.5	97.9	99.5	**99.5**
	Pixel AUROC	95.6	98.2	**98.6**	98.6	96.0	94.9	91.4	96.2	96.2	**98.8**
	Pixel PRO	92.5	96.0	**97.2**	96.8	91.9	88.4	80.6	92.7	92.4	**97.3**
Object	Image AUROC	99.3	94.1	96.3	93.2	91.8	96.5	90.2	**98.7**	98.6	**99.3**
	Pixel AUROC	98.2	96.9	98.2	98.0	97.3	97.2	97.3	**98.0**	97.7	**98.5**
	Pixel PRO	96.0	89.6	93.5	92.3	89.7	91.1	91.5	**93.0**	92.1	**94.9**
Average	Image AUROC	91.7	95.8	97.3	95.3	92.7	90.8	87.9	**98.4**	98.9	**99.4**
	Pixel AUROC	93.8	97.3	**98.3**	98.2	96.8	96.4	95.3	97.4	97.2	**98.6**
	Pixel PRO	85.0	91.7	**94.7**	93.8	90.4	90.1	80.6	92.9	92.2	**95.7**

TABLE VIII
DETAILED COMPARISON OF PREVIOUS FEATURE-SIMILARITY-BASED METHODS AND MLDFR ON THE MVTEC DATASET FOR EVERY CLASS USING IMAGE-LEVEL AUROC, PIXEL-LEVEL AUROC, AND PRO METRIC

	Method	Cut Paste	Gaussian Filter	Mask	White	RGB	Gaussian Filter or mask or white
textures	Carpet	98.6/**99.4**/**98.5**	98.6/99.3/**98.5**	98.8/99.3/**98.5**	**99.0**/99.3/**98.5**	98.4/98.2/**98.6**	98.9/99.3/98.5
	Grid	**100.0**/**99.4**/97.8	**100.0**/**99.3**/**97.9**	**100.0**/99.3/97.8	**100.0**/**99.3**/97.8	**100.0**/**99.3**/97.8	**100.0**/99.3/**97.9**
	Leather	**100.0**/**99.6**/**99.4**	**100.0**/**99.6**/**99.4**	**100.0**/**99.6**/**99.4**	**100.0**/**99.6**/**99.4**	**100.0**/**99.6**/**99.4**	**100.0**/**99.6**/**99.4**
	Tile	98.6/98.0/95.7	99.3/98.1/95.9	**99.8**/98.0/95.6	98.2/98.1/96.0	98.6/97.6/95.3	99.2/**98.3**/**96.1**
	Wood	98.9/96.8/93.8	**99.5**/**97.5**/94.5	99.4/97.2/**94.7**	99.3/96.9/94.4	**99.5**/96.8/94.5	99.4/97.3/94.6
	Average	99.2/98.6/97.0	99.5/**98.8**/97.2	**99.6**/98.7/97.2	99.3/98.6/97.2	99.3/98.3/97.1	99.5/**98.8**/**97.3**
objects	Bottle	99.9/98.8/96.6	**100.0**/**99.1**/**97.4**	99.8/99.0/97.0	**100.0**/99.0/97.0	99.8/98.9/96.8	**100.0**/**98.9**/**96.8**
	Cable	96.5/98.2/93.1	96.9/97.9/92.1	97.4/98.2/93.0	97.4/98.0/92.1	**97.5**/**98.4**/**93.1**	**97.8**/**98.4**/**93.5**
	Capsule	93.3/98.0/94.6	99.0/97.0/94.1	96.9/98.3/95.1	**99.2**/98.1/94.5	96.6/**98.3**/95.1	**99.2**/**98.3**/**96.3**
	Hazelnut	97.0/98.4/95.7	**100.0**/**98.9**/**96.1**	**100.0**/98.8/95.9	99.9/98.8/95.8	**100.0**/**98.9**/95.9	**100.0**/**98.9**/**96.1**
	Metal nut	**100.0**/97.8/91.9	**100.0**/97.8/93.0	**100.0**/98.0/91.8	**100.0**/98.0/**93.7**	**100.0**/**98.3**/**94.2**	**100.0**/**98.0**/**92.5**
	Pill	98.0/97.7/95.9	99.0/98.3/96.6	98.9/97.8/96.3	98.4/98.0/96.4	98.9/98.0/96.7	**99.3**/**98.3**/**96.9**
	Screw	92.8/99.0/95.5	97.0/99.5/97.9	97.4/99.3/97.2	98.0/99.6/98.2	97.9/99.6/98.2	**99.3**/**99.7**/**98.8**
	Toothbrush	96.7/98.9/92.8	96.4/98.7/92.2	98.6/98.9/93.0	96.4/**99.0**/**93.4**	96.4/98.9/92.4	**98.9**/98.8/92.3
	Transistor	98.5/96.7/86.2	98.9/94.9/82.3	**100.0**/96.5/86.3	**100.0**/**97.2**/**88.4**	**100.0**/96.9/87.6	**99.7**/**96.7**/**87.6**
	Zipper	91.2/98.6/96.0	**99.9**/**99.2**/**97.6**	**99.9**/99.1/97.3	97.8/97.1/97.2	95.2/98.9/96.8	99.1/**99.2**/97.5
	Average	96.4/98.2/93.8	98.7/98.1/93.9	99.0/98.4/94.3	98.7/98.3/94.7	98.2/98.5/94.7	**99.3**/**98.5**/**94.8**
Total Average		97.3/98.4/94.9	99.0/98.3/95.0	99.2/98.5/95.2	98.9/98.4/95.5	98.6/98.4/95.5	**99.4**/**98.6**/**95.7**

on the final results of MLDFR. Better results are highlighted in bold in the table.

All images of datasets are resized to a specific resolution of 384×384. The training and testing of AD and location are performed on one category at a time. In this experiment, we randomly choose a damage method from "Gaussian Filter (kernel = 9), Mask, White" for each sample of the same batch. The features of the normal samples are represented by fusing the output from the last layer in the first three blocks of ResNet18 and the 15th block of Cait-xs24-384. The features of damaged samples are represented by fusing the output from the last layer of the first three blocks of ResNet18 and the fifth block of Cait-xs24-384. The block number of feature restoration module is set to 8. The λ of loss function was set to 140. During training,

the optimization was set as Adam, the batch size 15, the number of epochs 500, the initial learning rate 0.001, which decays to 0.8 times every 30 epochs. Gaussian filtering was performed on the calculated anomaly score map, and the filter kernel was set to 8.

For AD, we take the area under the receiver operating characteristic (AUROC) as the evaluation metric. For AL, we use the AUROC curve where the true positive rate is the proportion of pixels correctly classified as anomalous. In addition, we report per-region-overlap (PRO) for AL [30]. It consists of plotting, for each connected component, a curve of mean values of the correctly classified pixel rates as a function of the false positive rate between 0 and 0.3. That means that it can evaluate more equally for abnormal areas of different sizes. A high PRO-score means that both large and small anomalies are well-localized [13].

B. Qualitative Results

MVTec-AD datasets contain real-world datasets of 5 texture classes and 10 object classes for AD and location. They are separated into two sets: a training set containing 3629 normal samples and a test set containing 1792 normal or abnormal sample. Each kind of test set contains multiple defects. In addition, pixel-level annotations are available in the test dataset for AL evaluation.

Table I shows the quantitative comparisons of MLDFR with image-similarity-based methods. Compared to the runner-up (InTra [23]), our method produces a significant response to the whole anomaly region. Table II shows the quantitative comparisons of MLDFR with feature-similarity-based methods and reports the detailed comparison results. As can be seen that our results remain the best. MLDFR achieves state-of-the-art image-level detection (AUROC 99.4%) and AL (AUROC 98.6%, PRO 95.7%) score. Table III presents the qualitative comparisons of different backbone networks as feature extractor. For fairness, we group together results with the same backbone architecture. The MLDFR has been significantly improved especially for image level AD.

MPDD datasets contain six categories of real-world industrial products with 1346 images. Unlike MVTec datasets which consists of monochromatic background from the laboratory environment. MPDD datasets can test the performance under conditions of variable spatial orientation, position, and distance of multiple objects concerning the camera at different light intensities and with a nonhomogeneous background [27]. The training set consists only of normal samples, while the test set contains both normal and abnormal samples, and provides pixel-level annotation for evaluating anomaly location result. Table IV displays the qualitative comparisons of MLDFR with the seven methods reported by MPDD. MLDFR achieves state-of-the-art image-level detection and AL performance. We beat the second competitor by 13.4% improvement (from 86.1% to 99.5%) in image-level AD results. We show AD examples of six categories on MPDD in Fig. 5.

BTAD datasets contain three types of real-world industrial products. A total of 2540 samples. As same as MVTec, the training set consists only of normal samples, while the test set contains both normal and abnormal samples, and provides pixel-level annotation for evaluating anomaly location result. Table V displays the qualitative comparisons of MLDFR with the three methods reported by a vision transformer network for image anomaly detection and localization (VT-ADL) [21]. MLDFR achieves state-of-the-art image-level detection and AL performance. We beat second competitor by a wide margin (from 90.0% to 98.1%) in pixel-level AL results. We display some AD examples on BTAD in Fig. 6.

Feature visualization: To better understand the discriminative abilities in the view of features before and after restoration, we utilize t-distributed stochastic neighbor embedding (t-SNE) to visualize the features before and after restoration of the sample of MVTec datasets and display them in Fig. 7. As shown, there is no significant change before and after the restoration of normal features, but the anomaly features are mapped to the normal feature area after restoration, which is significantly different from before. It is proved that the feature restoration module has a reliable representation capability for normal features, which ensure the excellent AD performance of MLDFR in various situations.

Complexity analysis: Table VI lists the results of the comparison of MLDFR with previous work in terms of inference time (second), memory usage (MB). In contrast, we have a significant reduction in memory usage, mainly because MLDFR does not store features and only needs to train a restore module. However, inference time is slower than RD, mainly due to the longer computational time required for concurrent structures and self-attention-based restore module. In the overall performance, inference time is relatively excellent.

C. Ablation Analysis

Feature extractor: Table VII lists the results of the different layer outputs selected when using only the CaiT or ResNet18 model, and other backbones. It can be concluded: when only using CaiT as feature extractor, selecting the output of the deep layer for normal samples and the shallow layer for abnormal samples produces better results. We speculate that the output of deep layers can better describe the global representation of normal samples, and the output of shallow layers can retain more structural information of damage samples, which facilitates feature restoration. When only using ResNet18 as the feature extractor network, selecting the output of the different layers shows large differences in different types of samples. After multilevel feature fusion can achieve a better performance.

In addition, when only using CaiT as feature extractor, the performance on texture class is better than object class. When only CNN was used, the results were reversed. We speculate that CaiT can capture long-distance feature dependencies but unfortunately deteriorate local. On the contrary, the CNN is good at extracting local features but experiences difficulty to capture global representations. When using both ResNet18 and CaiT as concurrent feature extractor, significant improvements are achieved on both texture and object classes. It is fully verified

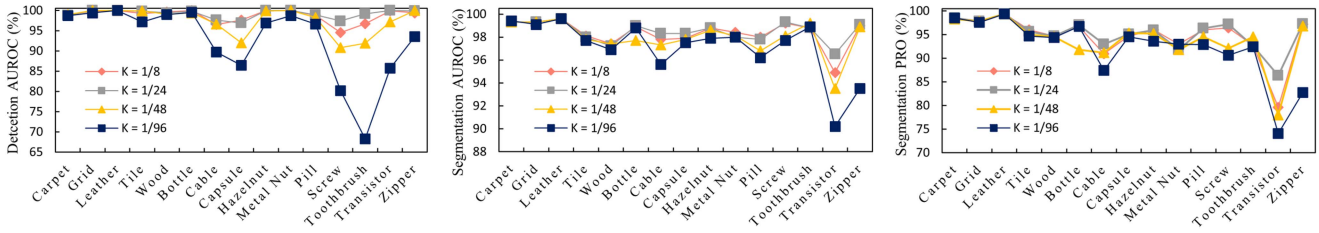

Fig. 8. Comparison of k (the side length of damage squares, which takes the size of the input as the unit).

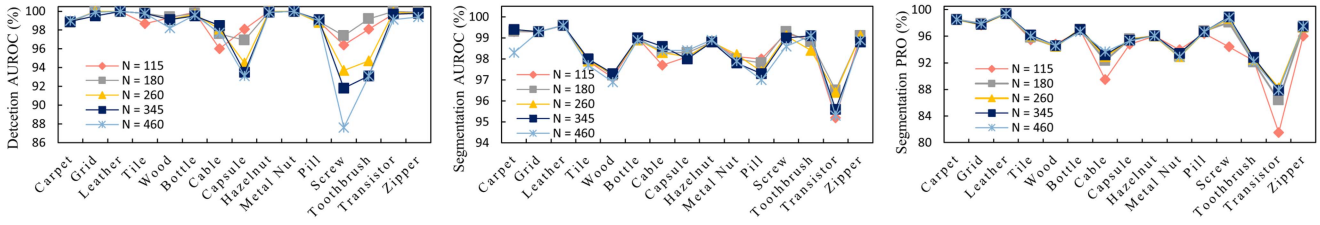

Fig. 9. Comparison of N (the number of damage squares).

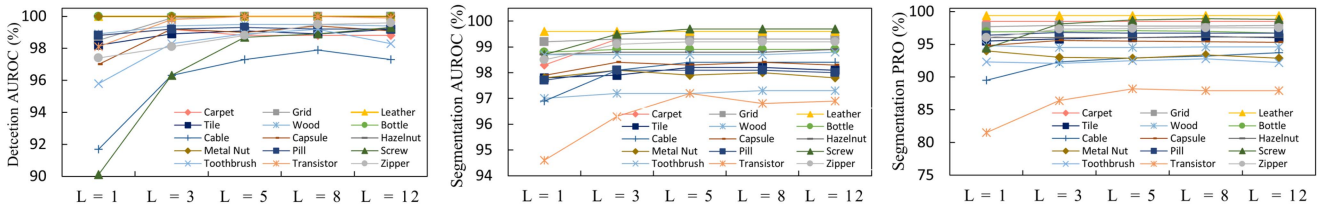

Fig. 10. Comparison with different number of blocks of the feature restoration models.

that this concurrent feature extractor effectively fuses the local features and global representation of samples.

Damage settings: Table VIII shows the qualitative comparisons of using different methods of damage images, including "CutPaste, Gaussian filter (kernel = 9), Mask, White, RGB." The result reveals that no definite winner works the best for all classes of detection. Nonetheless the average results with "Gaussian filter, mask, and white" are better than other methods. Hence, we randomly assign each sample a damage method from the "Gaussian filter, mask, white" and attempt to combine their advantages. Remarkably, the result is better than using only one of them. A plausible hypothesis on that combines three damage methods can balance the differences in the datasets. Fig. 8 displays qualitative comparison of k (the side the unit). Fig. 9 further illustrates qualitative comparison of N (the number of damage squares). As can be seen, that the side length and the number of damage squares have a low impact on texture class detection, but a remarkable impact on object class detection. We infer that indicates the appropriate degree of damage is particularly important. According to the result, the side length of damage squares is set to about 1/24 of the input size and the number of damage squares is set to about 180, with better test results.

Feature restoration module: Fig. 10 gives the quantitative comparisons of the feature restoration module with different number of blocks. As shown, that increasing the number of blocks improves the accuracy of AD and AL. Moreover, this phenomenon is more significant on classes with more complex structures. We speculate that increasing the number of blocks can more comprehensively capture the correlation between long-distance features and effectively improve the capability of feature restoration. Table IX shows the quantitative comparison of the feature restoration models with different structures. When the diagonal matrix is introduced into the multi-layer perception (MLP) layer, the performance is greatly improved, especially in the data types with complex structure. At the same time,

TABLE IX
QUANTITATIVE COMPARISON WITH DIFFERENT STRUCTURE OF THE FEATURE RESTORATION MODELS

Layer number		MLP	MLP + Diag(γ)	ViT	ViT + Diag(γ)
Texture	Image AUROC	98.9	99.5	99.5	99.5
	Pixel AUROC	98.4	98.7	98.7	98.8
	Pixel PRO	97.2	97.2	97.2	97.3
Object	Image AUROC	96.7	98.7	99.2	99.3
	Pixel AUROC	97.8	98.4	98.5	98.5
	Pixel PRO	93.2	94.5	94.8	94.9
Average	Image AUROC	97.6	99.0	99.3	99.4
	Pixel AUROC	98.0	98.5	98.6	98.6
	Pixel PRO	94.5	95.4	95.6	95.7
Average Convergence Epoch		200	100	320	120

TABLE X
QUANTITATIVE COMPARISON OF LOSS FUNCTION

Method	MSE	Cosine Distance	MSE + λ * Cosine Distance			
			λ = 50	λ = 140	λ = 500	λ = 1000
Texture	**99.6**/**98.8**/97.2	49.7/75.5/41.1	**99.6**/**98.8**/**97.3**	99.5/**98.8**/**97.3**	98.7/98.6/97.2	97.9/98.5/97.2
Object	98.9/98.4/**95.0**	38.6/59.3/62.2	99.0/**98.5**/**95.0**	**99.3**/**98.5**/94.8	99.0/**98.5**/94.8	98.9/98.4/94.8
Average	99.1/98.5/95.7	42.3/64.7/55.2	99.2/**98.6**/**95.8**	**99.4**/**98.6**/95.7	98.9/**98.6**/95.6	98.6/98.4/95.7

greatly reduce the training epochs. It can be fully verified that this diagonal matrix effectively aggregates the important feature information of different channels, which is helpful for feature restoration.

Loss function: Table X presents the quantitative comparison before and after adding cosine similarity to the loss function. In the inference stage, we calculate the MSE of the feature points as the score map. Therefore, only using Cosine distance as the loss function cannot locate the abnormal region accurately. There is a significant improvement in performance after adding the cosine similarity. But if the λ that makes the ratio of two components approximately the same is set too large, it is not conducive to AD and location.

V. CONCLUSION

In this work, we introduce the concept of image restoration into the feature space, and propose a novel paradigm for AD and AL. Extensive experimentation demonstrates that state-of-the-art performance is obtained via training the feature restoration module only using simple damage method under this paradigm. The virtually flawless image-level detection performance makes it possible for industrial AD applications. The limitations of this framework are reflected in two aspects. First, it will increase the calculation cost of retaining all the extracted features, so we can further study screen important features to reduce the inference time and storage space. Second, no definite method of the image damage works the best for all classes of detection. In future work, we can further study the adaptive damage method on the specified datasets to improve the performance on AD.

REFERENCES

[1] A. Castellani, S. Schmitt, and S. Squartini, "Real-world anomaly detection by using digital twin systems and weakly supervised learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4733–4742, Jul. 2021.

[2] Y. Zhang, Z. Y. Dong, W. Kong, and K. Meng, "A composite anomaly detection system for data-driven power plant condition monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4390–4402, Jul. 2020.

[3] Y. Shi et al., "DFR: Deep feature reconstruction for unsupervised anomaly segmentation," *Neurocomputing.*, vol. 424, pp. 9–22, Feb. 2021.

[4] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: Skip connected and adversarially trained encoderdecoder anomaly detection," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[5] V. Zavrtanik, M. Kristan, and D. Skcaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognit.*, vol. 112, 2021, Art. no. 107706.

[6] D. Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1705–1714.

[7] T.-W. Tang et al., "Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications," *Sensors*, vol. 20, no. 12, 2020, Art. no. 3336.

[8] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec Ad—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9592–9600.

[9] P. Kirichenko, P. Izmailov, and A. G. Wilson, "Why normalizing flows fail to detect out-of-distribution data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 20578–20589.

[10] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," *CoRR*, vol. abs/2005.02357, 2020.

[11] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14318–14328.

[12] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 6726–6733.

[13] T. Defard et al., "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 475–489.

[14] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but differnet: Semi-supervised defect detection with normalizing flows," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1907–1916.

[15] V. Zavrtanik, M. Kristan, and D. Skočaj, "Draem-a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8330–8339.

[16] S. Jezek et al., "Deep learning-based defect detection of metal parts: Evaluating current methods in complex conditions," in *Proc. 13th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops*, 2021, pp. 66–71.

[17] F. Ye, C. Huang, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," *IEEE Trans. Multimedia.*, vol. 24, pp. 116–127, Dec. 2022.

[18] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9664–9674.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[20] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 32–42.

[21] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A vision transformer network for image anomaly detection and localization," in *Proc. IEEE 30th Int. Symp. Ind. Electron.*, 2021, pp. 01–06.

[22] Q. Wan, L. Gao, X. Li, and L. Wen, "Unsupervised image anomaly detection and segmentation based on pre-trained feature mapping," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 2330–2339, Mar. 2023.

[23] J. Pirnay and K. Chai, "Inpainting transformer for anomaly detection," in *Proc. 21st Int. Conf. Image Anal. Process.*, Lecce, Italy, 2022, pp. 394–406.

[24] M. Rudolph, T. Wehrbein, B. Rosenhahn, and B. Wandt, "Fully convolutional cross-scale-flows for image-based defect detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1088–1097.

[25] N.-C. Ristea et al., "Self-supervised predictive convolutional attentive block for anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13576–13586.

[26] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9737–9746.

[27] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows," in *Proc. IEEE Winter. Conf. Appl. Comput. Vis.*, 2022, pp. 98–107.

[28] A. Dosovitskiy et al., "An image is worth 16x16 words: Trans-formers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–21.

[29] D. S. Tan, Y. C. Chen, T. P. C. Chen, and W. C. Chen, "TrustMAE: A noise-resilient defect classification framework using memory augmented auto-encoders with trust regions," in *Proc. IEEE Winter. Conf. Appl. Comput. Vis.*, 2021, pp. 276–285.

[30] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4183–4192.

Yinghui Guo received the B.S. degree in automation in 2021 from the School of Information Science and Engineering, Northeastern University, Shenyang, China, where he is currently working toward the M.S. degree in control science and engineering.

His current research interests are pattern recognition and anomaly detection.

Meng Jiang received the B.S. degree in information engineering from the Shenyang University of Chemical Technology, Shenyang, China, in 2021. She is currently working toward the Ph.D. degree in information science and engineering with Northeastern University, Shenyang, China.

Her current research interests are system engineering based on graph neural network and intelligent optimization.

Qianhong Huang received the B.S. degree in mechanical design, manufacturing and automation from the School of Mechanical and Electrical Engineering, Heilongjiang University, Harbin, China, in 2021. She is currently working toward the M.S. degree in control science and engineering with Northeastern University, Shenyang, China.

Her current research interests include pattern recognition and intelligent systems.

Yang Cheng received the B.S. degree in automation in 2023 from the School of Information Science and Engineering, Northeastern University, Shenyang, China, where he is currently working toward the M.S. degree in control science and engineering.

His current research interests are pattern recognition and large language model.

Jun Gong received the Ph.D. degree in system engineering from Northeastern University, Shenyang, China, in 2002.

He was a Visiting Scholar with the School of Industrial Engineering, Pennsylvania State University, State College, PA, USA. He was a Visiting Scholar with Akita Prefectural University, Akita, Japan, and also with The Hong Kong City University, Hong Kong. He is currently an Associate Professor with the School of Information Science and Engineering, Northeastern University. He has authored or coauthored more than 120 journal articles, conference papers, and book chapters. His current research interests include the areas of deep learning, intelligent control, medical image processing, and pattern recognition.

Dr. Gong has been a member of the technical committees of several scientific conferences. He is also a regular Reviewer for several journals.