IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 February 2024, accepted 22 March 2024, date of publication 27 March 2024, date of current version 10 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382209

RESEARCH ARTICLE

A Double Self-Supervised Model for Pitting Detection on Ball Screws

XIAOMING WANG[1], **YONGXIONG WANG**[1], **ZHIQUN PAN**[1], **GUANGPENG WANG**[1], **AND JUNFAN CHEN**[2]

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]Intelligent Science and Technology, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Yongxiong Wang (wyxiong@usst.edu.cn)

This work was supported by the Natural Science Foundation of Shanghai under Grant 22ZR1443700.

ABSTRACT Automatic detection of pitting on Ball Screw Drive (BSD) is essential to ensure normal production activities. However, the scarcity of defective samples and precisely labeled data poses a significant challenge. To address this, we propose an efficient double self-supervised model that operates at both the image and pixel levels, aiming to construct a high-performance model trained with defect-free data for detecting unknown defects in BSD images. By incorporating global and local information and extracting features at multiple hierarchical levels, the model's generalization performance is enhanced. The image-level self-supervised representation is first learned by classifying normal images from the PasteNoise, a data augmentation approach by pasting noise patches at random locations in normal images. Meanwhile, the pixel-level self-supervised representation is learned by segmenting the noise patch to locate abnormal regions. Then, we introduce a novel feature masking strategy in a masking and prediction task for accurate defect localization. In addition, we use Histogram of Oriented Gradients (HOG) features with local contrast normalization as prediction targets to capture local shapes and appearances to improve the robustness of the model. The proposed method achieves competitive receiver operating characteristic curves of 97.42 (image-level) and 94.57 (pixel-level) on the BSD dataset. In experiments on the MVTec AD, the proposed model shows good performance, indicating the broad adaptability of our approach.

INDEX TERMS Convolutional neural network, defect detection, deep learning, histogram of oriented gradients, self-supervised learning.

I. INTRODUCTION

Ball screw drive is a commonly used mechanical device for rotational motion transmission [1], mainly consisting of a ball screw, guide rail, and drive mechanism. It is also one of the components most susceptible to wear. Over time, the rolling elements induce material fatigue on the spindle, ultimately leading to small wear on the spindle's surface, called pitting (Figure 1). These pittings can lead to high roughness of machined parts and unstable processing accuracy. Traditional methods rely mainly on manual inspection, which is inefficient and vulnerable to human interference. At the same time, mechanical wear is a gradual process. The conventional force signal, ultrasonic signal, vibration signal, current signal, and acoustic emission signal are indirect signals that are not sensitive to minor wear. Recently, BSD images taken by observing holes are used for defect detection. The new approach is convenient, low-cost,

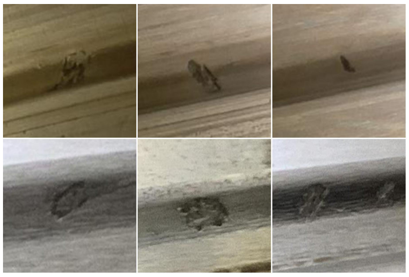

FIGURE 1. Examples of spindle pittings.

The associate editor coordinating the review of this manuscript and approving it for publication was Chaker Larabi.

and online. Subsequently, the automatic detection system based on BSD images gradually became a research focus.

In recent years, computer vision has achieved remarkable results in many automatic detections [2], [3], [4]. Deep learning methods, especially Convolutional Neural Networks (CNN), frequently appear in industrial production applications [5], [6]. However, in highly automated production scenarios, where the yield rate of products is particularly high, collecting defect samples is time-consuming. In addition, since defects are generated by uncontrolled factors in the production process, the forms of defects are various, and it is difficult to collect complete defective samples of various forms.

To reduce overdependence on data annotations, many self-supervised methods solving different proxy tasks have been widely studied [7], [8], [9], [10]. DeVries and Taylor [11] randomly cut areas of input images to reduce overfitting and improve the generalization ability of convolutional neural networks. Li et al. [9] propose a data augmentation strategy, which cuts image patches and then pastes them to random locations in the image to simulate real defects, but the trimmed patches are only a rough approximation of real defects. Pittings in the BSD [12] inspection are typically irregular and subtle(Figure 2), which is a challenge for the detection task. The differences in pitting on BSD between normal and anomaly patterns are often fine-grained as defective areas might be small and subtle in high-resolution images. However, most self-supervised methods have poor generalization due to a focus on object-centered natural images. Thus, it is essential for self-supervised representation learning to define an appropriate pretext task in pitting detection.

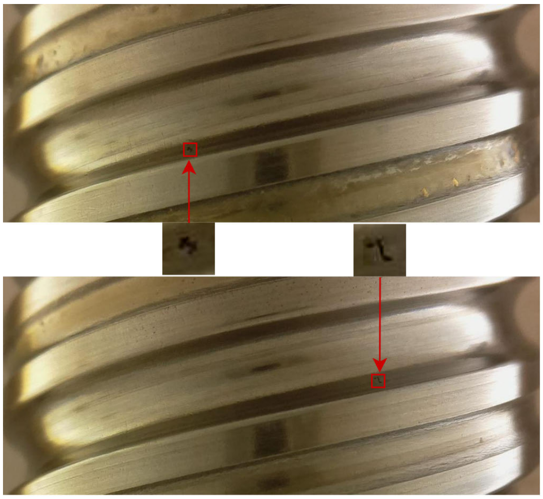

FIGURE 2. The initial pitting area is small, and compared to the high-resolution BSD image, the pitting is still less evident after magnification of 9 times.

In this work, we first construct a self-supervised framework that learns image representations by distinguishing anomaly data from normal data at image-level. We propose a novel proxy classification task that distinguishes between normal data and data augmented by PasteNoise. PasteNoise augmented samples are generated using a Perlin noise generator [13] along with an anomaly source image, which remains unrelated to the input image. Compared with common pseudo anomaly patches, such as rectangles or polygons, the texture of the noise patch is more natural, and the shape transformation is more diverse. Although we cannot obtain real defects, experiments show that representations learned by detecting noise patches generalize well to detecting real defects.

Moreover, masking the input image is one of the most popular augmentations used in transformer-based autoencoders [14], [15]. Self-supervised learning with masking shows more scalability when combined with ViT [15], [16]. However, self-supervised learning networks with naive masking do not work well in CNNs because masked inputs generate parasitic edges and distort the balance between global and local features [17]. Inspired by the focal mask [18], we introduce a new mask module, which avoids distorting the balance among features. High-level features are randomly masked to overcome the parasitic edge problem. We construct the second self-supervised framework by introducing the masking method into the convolutional network to improve the performance in detecting subtle defects. We specifically select HOG features with local contrast normalization and cheap computational overhead as the predictive target at pixel-level, further improving the robustness of the model. The self-supervised learning network with masking makes the model learn to find subtle differences between normal and defective images and generalizes better to real defects.

The image-level self-supervised task and the pixel-level self-supervised task are both important methods in unsupervised learning. The image-level task contributes to learning image-level representation and data augmentation, while the pixel-level task focuses on image segmentation and pixel-level annotation, allowing models to achieve more fine-grained image understanding. Two self-supervised learning strategies play a crucial role in addressing data annotation challenges, improving model generalization, and enhancing robustness. We integrate both into a unified network model, enabling the model to learn multi-level feature representations and comprehensively capture the semantic and detailed information in images. The model combines global and local information for a thorough and accurate understanding of images, effectively adapting to data from various domains or tasks.

The main contributions of our study are as follows:

1) We propose a double self-supervised learning detection model for BSD defect detection. At image-level, we utilize data augmentation to construct a proxy task, which facilitates learning self-supervised representations by distinguishing data with anomaly patterns from normal data. At pixel-level, we design a new masking feature pattern for the self-supervised training strategy to improve the model generalization ability. After masking the high-level features,

the network is guided to learn the detail representations of the image by predicting HOG features.

2) We propose a novel data augmentation method called PasteNoise that creates visually coherent anomaly patches with varying degrees of detail and smoothness. Experiments show the effectiveness of PasteNoise in detecting unknown defects.

3) Our model achieves 97.42 image-level detection AUC and 94.57 pixel-level localization AUC on BSD data, outperforming current state-of-the-art models and meeting the requirements of industrial scenarios. Experiments on MVTec AD show that our approach achieves SOTA performance, confirming the proposed method's generalizability.

The structure of the remaining sections in this paper is as follows. The latest research on defect detection and data augmentation is stated in Section II. The proposed network model is introduced in Section III. Extensive experiments and analysis are conducted in Section IV. Finally, Section V provides the conclusion for the article.

II. RELATED WORKS

A. DEFECT DETECTION METHODS

In the field of industrial production, machine learning has been applied to analyze industrial images for defect detection. Background subtraction [19] is a popular defect detection method. The background is subtracted from the image while preserving defects and random noise [20]. However, these methods are susceptible to the position and noise of the image. In recent years, deep learning has developed rapidly in computer visual fields. Since the first CNN LeNet [21] was proposed, several excellent networks have been designed, such as Alex [22], VGG [23], Inception [24], ResNet [25], etc. The remarkable achievements of CNNs in various computer vision tasks have led to their gradual application in the field of defect detection. In [26], bilinear class activation maps are utilized to optimize the detect defection model. Cross-domain and domain adaptation methods are adopted to detect ball screw faults [27].

Although computer vision has been successfully applied in defect detection, most of the work is limited by training datasets. Insufficient diverse annotated data and an imbalance between normal and defective samples can cause the model's performance to decline when facing unknown scenarios or variations. To mitigate reliance on the dataset, anomaly detection, trained solely on normal images, is widely used in defect detection [9], [10]. Reconstruction is the most common method for anomaly detection. However, generative adversarial networks sometimes construct anomaly samples that are very similar to normal samples, which affects the network's ability to distinguish anomaly samples. By reconstructing the masked filtering, Ristea et al. [7] use pretext tasks to learn object representations in self-supervised learning. Li et al. [9] obtain deep representations by reconstructing randomly pasted parts. Cao et al. [28] optimizes the feature distributions of normal and abnormal data separately, effectively alleviating the problem of overfitting abnormal features. Due to imprecise boundary descriptions, anomaly scores may unexpectedly exhibit low values [28]. To alleviate excessive generalization of anomalies, synthetic anomalies are introduced in Draem [10], CutPaste [9], and DAF [29]. Unlike these methods, the proposed dual self-supervised model incorporates two self-supervised tasks, extracting features from different scales and perspectives. Multi-level feature representations are enabled to be learned by the model and anomalies are distinguished more accurately from normal samples. Additionally, we overlay noise patches as an anomaly on the normal image. Various defect features are extracted in training by randomly masking defective patches on images, and normal image representations are learned simultaneously. Our patch-based model performs better, as shown in Section IV-D.

B. DATA AUGMENTATIONS

Defect detection methods synergize with data augmentations to improve model robustness by generating diverse abnormal samples. Data augmentation is a critical way to avoid model overfitting. When the data set has some apparent characteristics, such as the images being basically taken in the same scene, employing technologies like Cutout [11] and its variants [9], [30], [31], [32] can aid the model in avoiding the learning of irrelevant information unrelated to the target task. Cutout [11] introduces irregularities by randomly masking out some rectangular portions of an input image with zero or other uniform values. In Random Erasing [31], the length and width of the erased area and the replacement value of the pixel value in the area are random. In CutMix [30], a new training sample is created by combining two or more input images in such a way that parts of each image are visible in the combined image. Mixup [32] works by taking a weighted average of the two images according to a certain ratio. CutPaste [9] is a simple data augmentation method that randomly cuts a patch and pastes it at a random position of the original image.

In anomaly detection, self-supervised networks learn the irregularities introduced by data augmentations to generalize the representation of defects. In this work, we learn representations by classifying normal data from PasteNoise during training, which is a straightforward and practical data augmentation strategy. PasteNoise is used to generate more natural and diverse abnormal samples for training. The model's generalization capability is enhanced, resulting in precise recognition of unseen anomalies in detection.

III. METHOD

In this section, we introduce a double self-supervised learning model combined with PasteNoise data augmentation and a mask-and-predict task, as shown in Figure 3. The backbone network is modified based on the lightweight ResNet-18 [25] with 18 weight layers, including 17 convolutional layers and a fully connected layer. Feature maps are extracted by 17 convolutional layers, as shown in Table 1, and

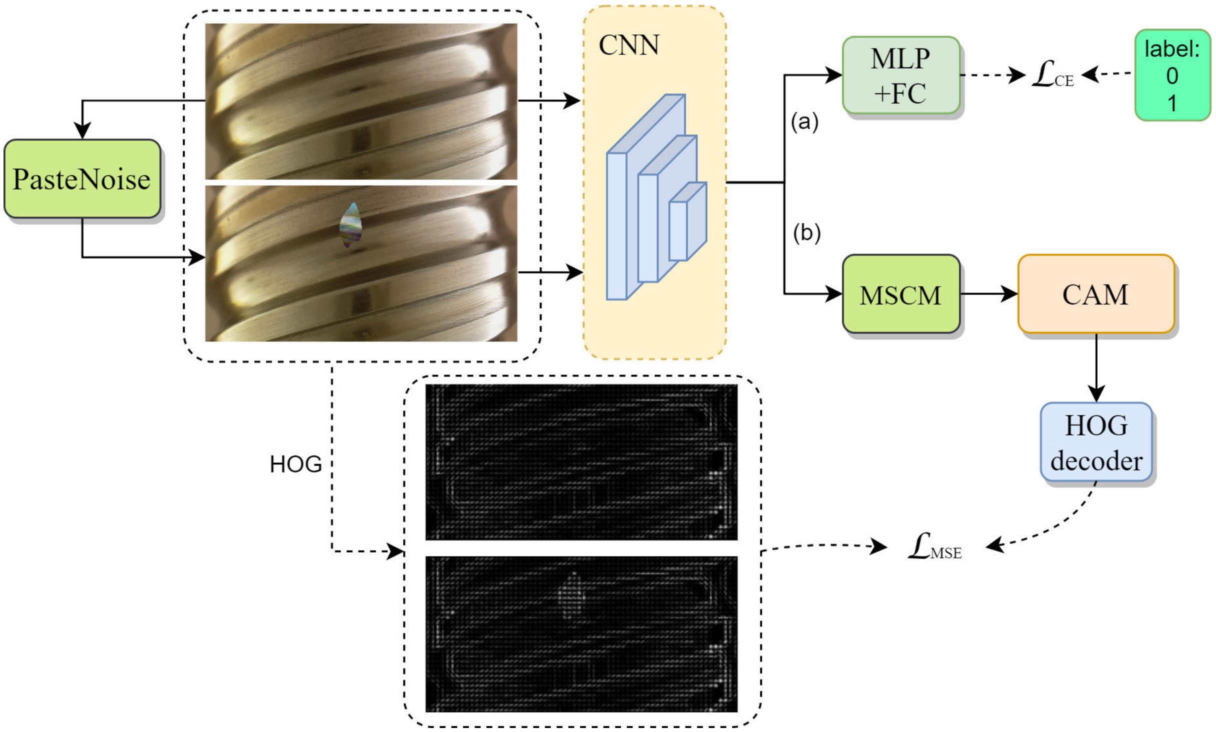

FIGURE 3. The overview of the double self-supervised model architecture. Anomaly images generated by PasteNoise, along with input images, are fed into a Convolutional Neural Network (CNN). The extracted feature maps are directed into two branches. (a)Through a Multi-Layer Perceptron (MLP) and a Fully Connected layer (FC), image-level classification predictions are generated. Our model acquires high-level semantic information within the images by minimizing the prediction loss L_{CE}. (b) Following the masking process in the Masked Split Convolution Module (MSCM), a Channel Attention Module (CAM) is utilized to scale the channel weights. Subsequently, an HOG decoder maps the feature maps back to a one-dimensional vector representing Histogram of Oriented Gradients (HOG). By minimizing the similarity loss L_{MSE} between the extracted HOG features from the input image.

TABLE 1. Architecture of backbone network.

Layername	*Outputsize*	*Layerstructure*
Conv1	128 × 128	7 × 7, 64, stride 2
Conv2_x	64 × 64	3 × 3 max pool, stride 2 $\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 2$
Conv3_x	32 × 32	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 2$
Conv4_x	16 × 16	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 2$
Conv5_x	8 × 8	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 2$

then they enter two streams of the network. One stream is a 3-layer multilayer perceptron (MLP) projection head and a fully connected layer with the cross-entropy loss (L_{CE}), which is used to measure the classification loss in self-supervised learning with PasteNoise. The other stream sequentially passes through the masked split convolution module (MSCM), channel attention module (CAM), and HOG decoder. At last, our network is equipped with a mean-squared error loss (L_{MSE}) that minimizes the reconstruction error between the original HOG and the predicted feature.

Next, the network structure is explained in detail from the perspective of two self-supervised tasks. Self-supervised learning with PasteNoise is described in section III-A. Self-supervised learning with HOG and masked features is introduced in section III-B. Additionally, we give a brief description of the anomaly score in section III-C.

A. SELF-SUPERVISED LEARNING WITH PASTENOISE

In this section, we present a novel pretext task for learning self-supervised representations by predicting PasteNoise augmentation and constructing a one-class classifier.

We insert an MLP projector between the backbone network and the classifier for better transferability [33]. Therefore, the image-level self-supervised learning sub-network includes a feature extractor $f(\cdot)$, an MLP projector $g(\cdot)$, and a classifier W. Given the input image x, the outputs $F_x = f(x)$ of CNN is a 512-dimensional 8×8 feature map. F_x is mapped into a projection vector $\mathbf{g} = g(F_x)$ by an MLP. The MLP consists of two fully connected layers. In the MLP projector, the hidden feature dimension is set to 512. Input images consist of the original image x and the image augmented with PasteNoise. The objective function for the self-supervised task is defined as:

$$F_x = f\left(x, PN\left(x\right)\right) \tag{1}$$

$$L_{CE} = \mathbb{BCE}\left(W \cdot g\left(F_x\right), (0, 1)\right) \tag{2}$$

where x is the normal image. PN denotes covering a noise patch on the normal image. $\mathbb{BCE}$ is a binary cross-entropy function.

PasteNoise: Since the shape of true anomalies is unpredictable, we expect to generate anomalies outside the normal distribution of characteristics. Following [10], a noise image with various anomaly shapes is first generated by the Perlin noise generator [13]. And then a random anomaly map (Figure 4 M_p) is obtained after binarizing the noise image by a threshold. The mask map (Figure 4 M) preserves the largest masked region from M_p. The anomaly texture source image (Figure 4 I_e) is selected from an image dataset that is different from the input image distribution. RandomAugment [34] is applied by three random augmentation methods from the set (Autocontrast, Posterize, Solarize, Sharpness, ChangeBrightness, ChangeColor). Finally, an anomalous image (Figure 4 I_s) is generated, where the mask region is filled with a linear combination of the augmented image I_e and an anomaly-free image (Figure 4 I) with a ratio factor α randomly chosen between [0.5,1]. The above operation can be formulated as follows:

$$I_s = (1 - M) \odot I + \alpha(M \odot I) + (1 - \alpha)(M \odot I_e) \tag{3}$$

Here, $\odot$ presents the element-wise multiplication operation. This algorithm for generating anomalous images has the following benefits. Random Perlin noise is more irregular and closer to real anomalous. The diversity of anomaly images is increased by introducing the hyperparameter α.

B. SELF-SUPERVISED LEARNING WITH HOG AND MASKED FEATURES

In the second self-supervised learning task, we construct a mask-and-predict task. This self-supervised sub-network includes a feature extractor $f(\cdot)$ shared with the classification task in Section III-A, a masked split convolutional module $\varepsilon(\cdot)$, a channel attention module $\varphi(\cdot)$, and an HOG decoder $h(\cdot)$. Here we utilize Mean Squared Error (MSE) loss to minimize the distance between the original HOG features and the predicted HOG as follows:

$$H_x = h\left(\varphi\left(\varepsilon\left(F_x\right)\right)\right) \tag{4}$$

$$L_{MSE} = \mathbb{MSE}\left(H_x, \mathbb{H}\left(x, PN\left(x\right)\right)\right) \tag{5}$$

where x is a normal data. F_x is expressed by Equation 1. H_x presents the predicted HOG. $\mathbb{H}$ means extracting HOG features from images. $\mathbb{MSE}$ is an MSE loss function.

Next, we will show MSCM, CAM, and HOG decoder in detail.

1) MASKED SPLIT CONVOLUTION MODULE (MSCM)

CNN is widely utilized in various computer vision tasks. In CNN training, image features are continually abstracted from low to high levels. However, as Sabour et al. [35] stated, CNN lacks the global arrangement ability for local features. To address this, we employ MSCM, aiming to leverage contextual information to predict masked feature information. This compels the model to learn the global structure of local features for high-precision reconstruction results. Specifically, MSCM conducts expansion, split, convolution, and fusion operations on the feature maps extracted by the CNN network. During training, the model learns to reconstruct the masked information, while providing useful features to locate and understand anomalies.

We show the MSCM module in Figure 5. The input feature $Y \in \mathbb{R}^{w \times h \times c}$ is expanded, and the dilation factor is d, where c is the number of input channels, and w and h are the width and height of the input feature map, respectively.

We aim to predict the spatial location of input feature Y. Therefore, we add d zero pixels around the input feature. The feature map is expanded to $(w + 2d) \times (h + 2d)$. Four sub-feature maps of size $w \times h$ are taken from four vertices, $K_i \in \mathbb{R}^{w \times h \times c}, \forall i \in \{1, 2, 3, 4\}$. Convolution operations are performed on these sub-feature maps to learn the input spatial location information by predicting the masked portion. Because of the same size, the spatial dimension of the module output feature is the same as the input feature. In the MSCM, the configurable super-parameter is only the expansion rate d, which is set to 2. Subsequently, we express MSCM with the following formulas:

$$K_i = S_i\left[F_{\text{pad}}(Y)\right] \tag{6}$$

$$Y_{\text{out}} = \sigma \sum_{i=1}^{4} F\left(K_i\right) \tag{7}$$

where F_{pad} and S_i represent padding and splitting operations, respectively. $F(\cdot)$ is convolution, and σ denotes the sigmoid activation function.

2) CHANNEL ATTENTION MODULE (CAM)

The output features of the Mask Splitting Convolution Module are partially masked, which may impact the activation ratio between channels. To enhance the modeling ability of MSCM and find the importance of different channels, we introduce the SE module [36] into the network. SE is a channel attention module (CAM), and the structure is shown in Figure 6.

CAM recalibrates the previously obtained features through three operations, Squeeze, Excitation, and Reweight. The Squeeze operation is to perform feature compression along the spatial dimension, turning each two-dimensional feature channel into a real number, using a global average pooling completed. The second is the Excitation operation, a mechanism similar to gates in recurrent neural networks. Weights are generated for each feature channel via a parameter w, which is learned to explicitly model the correlation between feature channels. The scale factor $s \in \mathbb{R}^{1 \times 1 \times c}$ is the core of

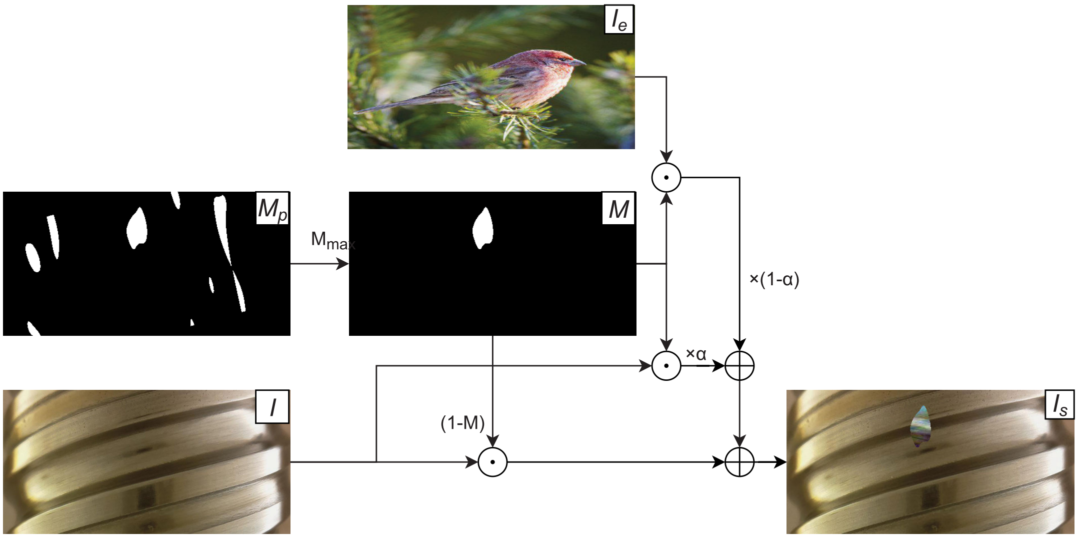

FIGURE 4. Simulated PasteNoise anomaly generation process. The binary anomaly mask M_p is generated from binarized Perlin noise. Mask M preserves the largest masked area from M_p. The anomalous region is sampled from I_e and placed on the anomaly-free image I to generate the anomalous image I_s.

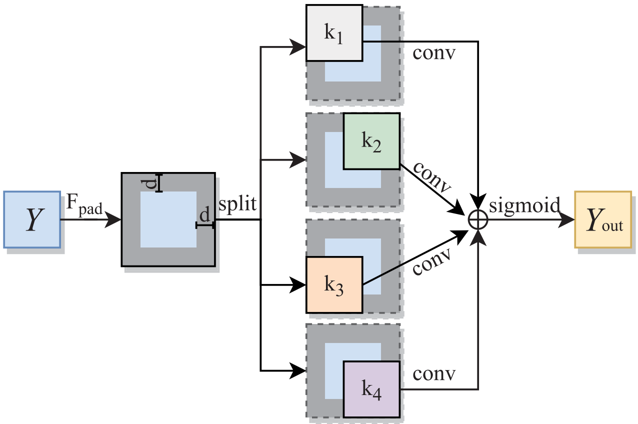

FIGURE 5. Masked split convolution module (MSCM).

CAM and can be expressed by Equation 8.

$$s = \sigma\left(W_2 \cdot \delta\left(W_1 \cdot Z\right)\right) \tag{8}$$

where δ represents the ReLU function, σ refers to the sigmoid function, and $Z \in \mathbb{R}^{h\times w\times c}$ is the input descriptor. $W_1 \in \mathbb{R}^{(c/r)\times c}$ and $W_2 \in \mathbb{R}^{c\times(c/r)}$ denote the weights of two fully connected layers, a dimensionality reduction layer with a reduction ratio r and a dimensionality increasing layer. The last is a Reweight operation, which weights s to the previous features by multiplying channel by channel, rescaling the original features in the channel dimension.

3) HOG DECODER

We introduce the HOG decoder to perform the inverse operation of encoding, extracting HOG features from the feature map, aligning with our prediction target. The backbone network employed in this paper is ResNet18 [25], featuring a small number of layers and relatively few parameters. Consequently, we construct a simple regression unit composed of two convolution layers and regress the feature image to a one-dimensional vector representing HOG in Table 2. These parameters of the regression unit are continuously corrected by the HOG loss function as Equation 5.

TABLE 2. Architecture of HOG decoder.

Layer	Kernel Shape	Kernel Number	Stride	Output
Conv	3×3	64	1	
MaxPool	2×2	32	2	32×32
Conv	3×3	32	1	
MaxPool	2×2	16	2	16×16
Flatten				4096
Linear				3780

C. ANOMALY SCORE

The Mahalanobis distance [37] can be used for anomaly detection, which needs to calculate the boundary threshold from the data center with normal data and then determine the point is an anomaly if it is a point from the center of the data set that exceeds the threshold. Similar to [38], we compute the Mahalanobis distance as the anomaly score. Mahalanobis distance is an effective method for calculating the similarity between two unknown sample sets. Unlike Euclidean distance, it considers the relationship between various properties and is scale-independent. For a given test datum $\mathbf{x}$, the Mahalanobis distance is defined as follows:

$$M(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T S^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{9}$$

where $\boldsymbol{\mu}$ and S^{-1} are the vector mean and covariance matrix, respectively, learned from the training data.

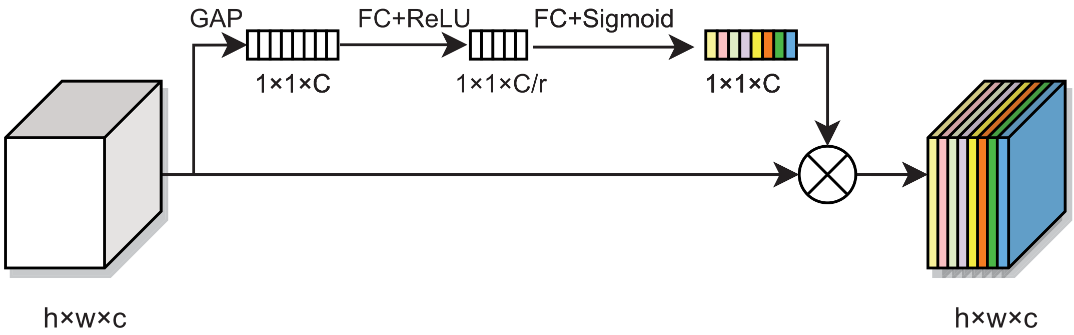

FIGURE 6. Architecture of the channel attention module (CAM).

For each sample, the anomaly score can be obtained by Equation 9. If the anomaly score exceeds the threshold θ, the pixel is classified as an anomaly point. In the experimental section, the Receiver Operating Characteristic (ROC) [39] metric is computed by varying the threshold θ.

IV. EXPERIMENTS

The experimental setup is first described in this section, including the datasets and experimental details. Then, we demonstrate the effectiveness of our approach on the BSD dataset and the MVTec AD dataset separately. We compare our approach with several state-of-the-art models. Finally, various aspects of the proposed method are explored through ablation experiments.

A. EXPERIMENTAL SETUP

BSD dataset [12]: This dataset comprises 1,104 images, including 710 normal images and 394 images with defects. These images are recorded by a camera system, which is a standard Raspberry Pi V2 microcontroller camera. The original photos are 1130 × 460 and a small number of about 2000×1000. Finally, 631 normal images are randomly divided into the training set. The rest 79 normal images and 394 defective images form the test set.

MVTec AD dataset [39]: The MVTec AD dataset is aimed at anomaly detection, which provides multi-object and multi-defect real-world images. It contains 5,354 color images divided into five texture categories and ten object categories. Each category includes a set of training images without defects and a set of images with various defects as well as defect-free images. For the MVTec AD dataset, the training set comprises 3,629 images, all of which are normal images. In the test set, there are 1,725 images, including 467 normal images and 1,258 abnormal images.

Evaluation Metrics: Following prior works, we in-depth study model performance from both detection and localization perspectives. For anomaly detection, the area under the receiver operation characteristic (AUC) is taken as the evaluation metric. For anomaly localization, AUC is selected to evaluate the pixel-level localization result.

Experimental Details: These experiments are conducted on the workstation with two INTEL XEON E5-2678 CPUs and two GeForce RTX 2080S GPUs (8GB). We implement our models in Python 3.7 and PyTorch 1.6. All images are resized to 256×256. We adopt ResNet-18 [25] as a backbone in our model. In experiments, we use the SGD optimizer with a base learning rate of 0.03, the momentum of 0.9, and weight decay of 0.00003. We train 800 epochs with a batch size of 32.

TABLE 3. Detection and localization performances on BSD data. For comparison, we report image-level detection and pixel-level localization AUCs of CutPaste [9], and SSPCAB [7].

Method	Image-level AUC	Pixel-level AUC
CutPaste [9]	95.63	93.89
SSPCAB [7]	95.82	94.01
Ours	97.42	94.57

B. DEFECT DETECTION ON BSD DATASET

We conduct anomaly detection and localization experiments leveraging our representations trained with a double self-supervised model. We use visual heatmaps to highlight the region-predicted anomaly for accurate localization of defects.

We compare with two recent works, including CutPaste [9] and SSPCAB [7]. Using the same backbone network (ResNet18) helps eliminate the impact of differences in network structures, ensuring a fair comparison. Identical basic configurations, including hyperparameters, learning rates, optimizers, etc., contribute to the reliability and repeatability of the comparison. The CutPaste [9] method shows high performance in image anomaly detection. It utilizes a technique of cutting images and pasting them into different positions to construct negative samples. However, such operations may lead to overfitting, particularly when the model is trained on smaller datasets. In SSPCAB [7], a self-supervised predictive module is integrated with masked convolution kernels and channel attention mechanisms to enhance features. We utilize two self-supervised learning tasks simultaneously to acquire comprehensive feature representations across various levels, encompassing both global and local characteristics. This strategy enables us to effectively capture both semantic understanding and intricate details within images.

In Table 3, we report the detection and localization performance on the BSD dataset. Our method achieves an image-level detection AUC of 97.42 and a pixel-level localization AUC of 94.57, outperforming CutPaste [9] and

TABLE 4. Comparison of image-level detection AUC with state-of-the-art methods on MVTec AD. The best result is bold.

Category		UStu-Tea [40]	patch-SVDD [41]	CutPaste [9]	SSPCAB [7]	CDO [28]	ours
object	bottle	93.10	98.60	98.30	98.60	**100.00**	**100.00**
	cable	81.80	90.30	80.60	82.90	**97.43**	95.60
	capsule	96.80	76.70	96.20	**98.10**	91.78	85.84
	hazelnut	96.50	92.00	97.30	98.30	98.21	**99.64**
	metal nut	94.20	94.00	99.30	**99.60**	98.53	96.48
	pill	**96.10**	86.10	92.40	95.30	94.35	95.23
	screw	**94.20**	81.30	86.30	90.80	82.97	81.88
	toothbrush	93.30	**100.00**	98.30	98.80	89.17	99.72
	transistor	66.60	91.50	95.50	96.50	94.54	**96.71**
	zipper	95.10	97.90	**99.40**	99.10	97.74	98.90
	average	90.77	90.84	94.36	**95.80**	94.47	95.00
texture	carpet	87.90	92.90	93.10	90.70	**97.75**	97.19
	grid	95.20	94.60	**99.90**	**99.90**	96.49	98.07
	leather	94.50	90.90	**100.00**	**100.00**	**100.00**	**100.00**
	tile	94.60	97.80	93.40	94.00	98.77	**99.86**
	wood	91.10	96.50	98.60	**99.20**	99.04	99.03
	average	92.66	94.54	97.00	96.76	98.41	**98.83**
all average		91.40	92.07	95.24	96.12	95.78	**96.28**

TABLE 5. Comparison of pixel-level localization AUC with state-of-the-art methods on MVTec AD. The best result is bold.

Category		patch-SVDD [41]	CutPaste [9]	PaDiM [42]	CDO [28]	Ours
object	bottle	98.10	97.60	98.10	**98.99**	96.64
	cable	96.80	90.00	95.80	96.71	**96.83**
	capsule	95.80	97.40	98.30	**98.50**	97.72
	hazelnut	97.50	97.30	97.70	**98.94**	98.03
	metal nut	**98.00**	93.10	96.70	97.18	96.48
	pill	95.10	95.70	94.70	96.86	**97.30**
	screw	95.70	96.70	97.40	98.31	**98.32**
	toothbrush	98.10	98.10	**98.70**	98.70	97.45
	transistor	97.00	93.00	97.20	87.50	**97.28**
	zipper	95.10	**99.30**	98.20	98.42	97.73
	average	96.72	95.82	97.28	97.01	**97.38**
texture	carpet	92.60	98.30	**98.90**	98.78	98.79
	grid	96.20	97.50	94.90	**97.49**	96.54
	leather	97.40	**99.50**	99.10	99.05	99.10
	tile	91.40	90.50	91.20	95.52	**95.54**
	wood	90.80	95.50	93.60	**96.04**	95.38
	average	93.68	96.26	95.54	**97.38**	97.07
all average		95.71	95.97	96.70	97.13	**97.28**

SSPCAB [7]. As shown in Figure 7, the proposed method can locate the pittings more accurately than other methods. We note that BSD images have similar backgrounds, fixed local attributes, high resolution, and relatively stable textures in corresponding regions of the images, which can be classified as texture categories. Compared with the texture category on the MVTec AD dataset, which generally achieves 99+ AUC, the reduced performance on BSD data could be explained by contaminated images. The examples can be seen in Figure 8, where some contaminated regions are close to the anomaly image distribution. Our method detects anomalous features in these contaminated regions, which are difficult to distinguish from real anomaly regions.

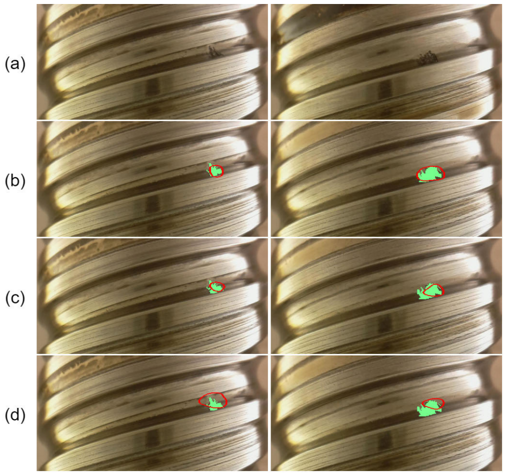

FIGURE 7. Visual comparisons with state-of-the-art methods on the BSD dataset. Rows (a)-(d) represent input images, and results predicted by the following methods: our proposed method, SSPCAB [7], and CutPaste [9]. The green masks are true anomalous pixels and the red regions are located by the corresponding method.

C. COMPARISON WITH STATE-OF-THE-ART MODELS ON MVTEC AD DATASET

Our method is evaluated on the challenging MVTec anomaly detection dataset. We use image-level AUC for evaluating

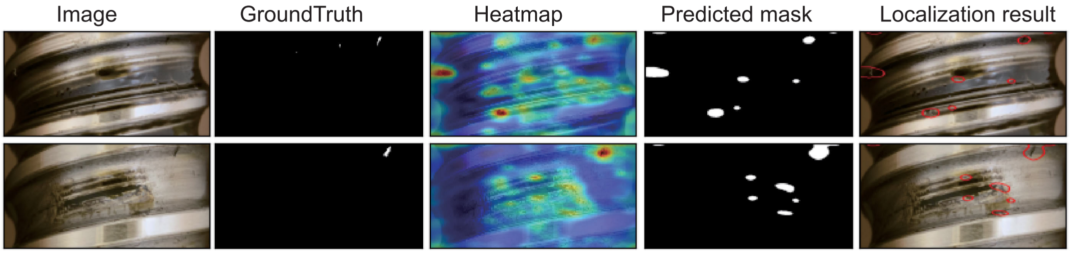

FIGURE 8. The original image contains contaminated areas that are difficult to mark in the ground-truth maps, which causes a discrepancy between the ground-truth and the predicted mask map.

anomaly detection. At the same time, for localization accuracy, the pixel-level AUC is used for anomaly localization.

Anomaly Detection: In Table 4, we report the detection performance compared with state-of-the-art models on MVTec AD, including patch-level SVDD [41], uninformed students [40], cut-paste self-supervised learning (CutPaste) [9], self-supervised predictive convolutional block (SSPCAB) [7], and CDO [28]. The proposed method outperforms recent anomaly detection methods, achieving the best AUC in 5 out of 15 classes and achieving the best average AUC.

Anomaly Localization: Table 5 compares our method to the recent state-of-the-art on the task of pixel-level anomaly detection. Our method achieves comparable results to these well-performing methods, patch-level SVDD [41], CutPaste [9], PaDiM [42] and CDO [28] with ResNet-18 [25]. The proposed method achieves a significant improvement in anomaly localization accuracy.

D. ABLATION STUDY

Prediction Target Type: We explore the impact of two different prediction targets in self-supervised learning with masked features in Table 6. RGB is an image representation method based on color channels, suitable for tasks that involve consideration of color. HOG is a feature descriptor method based on image gradients, primarily used to capture texture and shape information in images. It exhibits strong descriptive capabilities for the edges and contours of objects. Compared to HOG features, regressing RGB values produces a slight drop of about -0.5 for image classification. RGB as a target has a potential downside of over-fitting to high-frequency details and local statistics, which play an insignificant role in the interpretation of image content. HOG contributes to capturing features with strong representation from high-level semantic specialties and focuses on obtaining the texture-related representation of images. As a result, HOG with local-contrast normalization is more robust to overfit high-frequency details [43].

Double Self-Supervised Learning Tasks: We evaluate the effect of two self-supervised learning tasks on the BSD dataset in Table 7. We note that self-supervised learning of representations with noise patches achieves decent results alone. Self-supervised learning of spatial representation with masked features, using HOG as the prediction target, is also a distillation mode and can obtain a detection effect (95.16 detection AUC and 93.75 localization AUC). Interestingly, when these two self-supervised tasks are integrated into the same backbone network model, a better detection effect (97.42 detection AUC and 94.57 localization AUC) is achieved than when used alone. Integrating dual self-supervised tasks at both image and pixel levels enables the network to extract multi-level abstract features. This enhances the model's robustness to noise or complex environments, resulting in more reliable and versatile representations.

Anomaly Patterns: We study the performance of representations trained by patching diverse patches to normal

TABLE 6. "HOG" and "RGB" present HOG features and pixels as prediction targets in the masked self-supervised task, respectively.

RGB	HOG	Image-level AUC	Pixel-level AUC
✓		94.63	93.55
	✓	95.16	93.75

TABLE 7. "HOG" denotes HOG features as prediction targets in the masked self-supervised task. "NP" means the self-supervised task with noise patches.

HOG	NP	Image-level AUC	Pixel-level AUC
✓		95.16	93.75
	✓	96.08	94.02
✓	✓	97.42	94.57

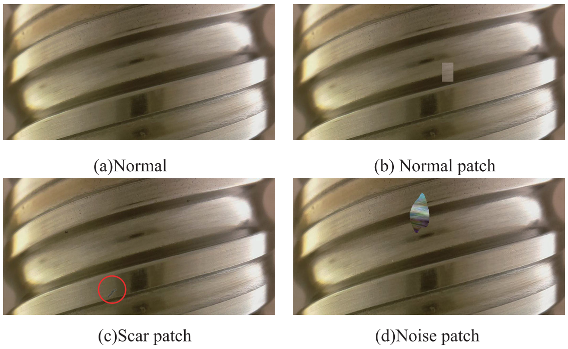
(a)Normal
(b) Normal patch
(c)Scar patch
(d)Noise patch

FIGURE 9. Visualization of normal and augmented normal samples.

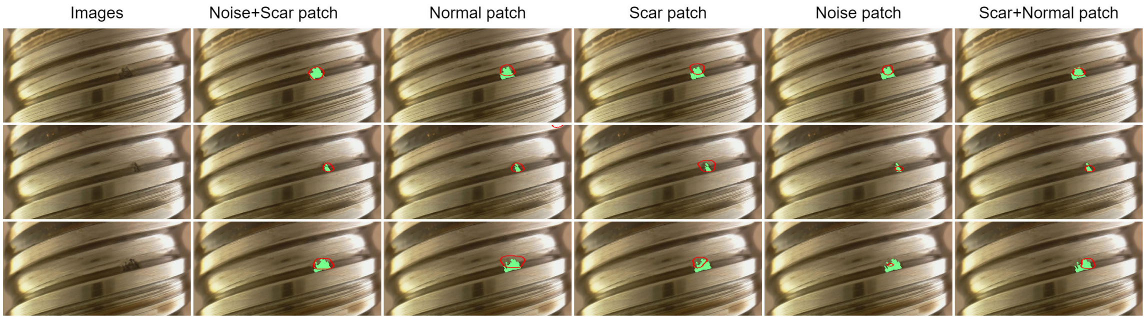

FIGURE 10. Visual comparisons of different anomaly patterns. The green masks are true anomalous pixels and the red regions are located by the corresponding method.

images, such as scar, normal, scar+normal, and noise patch. Scar patch, which is proposed in [9] for anomaly detection, is a scar-like thin rectangular box filled with an image patch. Normal patch denotes the patch that is cut from the input image. Noise patches generated by the Berlin noise generator in Section III-A have random and disorderly features while maintaining a uniform distribution at a specific scale. Combining with "Scar" enhances the diversity of synthesized anomalies, which help the model comprehend and learn to handle unknown anomaly patterns. With a finer-gained detection task to leverage scar and noise patch together augmentation, we achieve the best detection (97.42 AUC) and localization (94.57 AUC) performance. We show samples from considered augmentations patches in Figure 9 and report the detection and localization AUCs in Table 8. As shown in Figure 10, the accurate localization of defects is achieved by training with the anomaly pattern "Noise + Scar".

TABLE 8. Detect performance trained with various patch combinations.

Patch type	Image-level AUC	Pixel-level AUC
Scar patch	95.49	94.23
Normal patch	95.43	93.89
Scar + Normal patch	96.04	94.53
Noise patch	97.19	93.85
Noise + Scar patch	97.42	94.57

Different Mask Types: We explored the impact of other masking methods on our model, including "No", "Dropout", and "Mask". The detection results are shown in Table 9. We observe that "MSCM" brings about performance improvements with a 0.78 image-level AUC and a 1.33 pixel-level AUC. In contrast, "Dropout" is a technique that randomly zeros elements in the feature maps during the model training process, aiming to introduce randomness to prevent overfitting. However, this random zeroing might lead to incomplete information transmission with limited training data. In some cases, it performs worse than not using any feature map masking (i.e., "No"). On the other hand, "Mask" simulates the absence of partial information by applying a binary mask to the input feature map. However, the effectiveness of "Mask" may be influenced by the specific task and data distribution. As shown in Figure 11, MSCM demonstrates good localization accuracy, whereas "Dropout" shows more errors in predictions.

TABLE 9. Detection results of different mask types. "No" means not to mask the feature map. "Dropout" presents to randomly set the elements in the feature map to zero with a probability of 0.5, "Mask" denotes a rectangle to mask the feature map, and the masking rate is 50%. "MSCM" is our masking strategy described in Section III-B.

Mask type	Image-level AUC	Pixel-level AUC
No	96.75	93.05
Dropout	96.45	92.86
Mask	96.84	93.24
MSCM	97.42	94.57

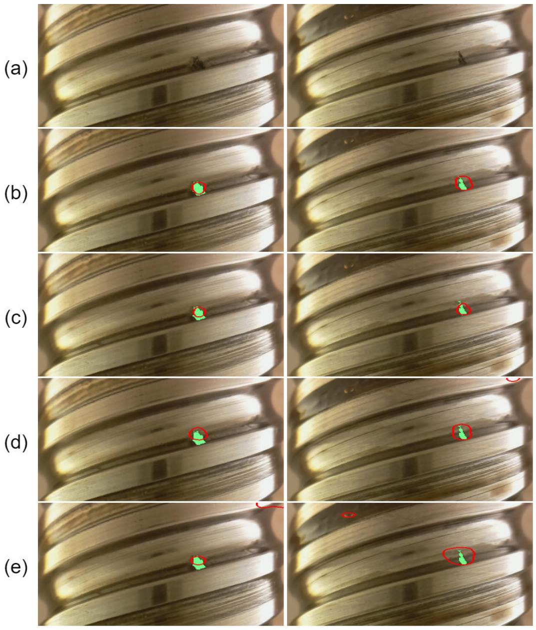

FIGURE 11. Visual comparisons of different mask types. Rows (a)-(e) represent input images, and results predicted by the following mask types: MSCM, Mask, No, and Dropout respectively. The green masks are true anomalous pixels and the red regions are located by the corresponding method.

These masking strategies play a certain role in improving detection effectiveness, although the effects are relatively modest. In future research, we aim to enhance the efficacy of masking methods by adjusting parameters, introducing new techniques, or combining them with other advanced approaches.

V. CONCLUSION

In this paper, we propose a double self-supervised model for BSD defect detection. A noise generator is utilized to generate synthetic abnormal images with random and disordered characteristics. The diversity of training samples is enhanced. Then, self-supervised proxy tasks are constructed at both the image and pixel levels. These tasks have a positive effect on solving the problem of difficult or expensive data labeling and improving the performance of the model on vision tasks. Pitting defects on BSD surfaces can be effectively identified in complex production scenarios (97.42 image-level AUC and 94.57 pixel-level AUC). The proposed model outperforms state-of-the-art anomaly detection models on real BSD datasets by +2.0 image-level AUC and +0.56 pixel-level AUC. The proposed model is optimized from multiple perspectives (textured natural PasteNoise, multi-view feature representation optimization) for improving network performance. The self-supervised learning model for defect detection is an effective method in industrial production, improving detection accuracy and reducing reliance on manual labeling or defect data.

In practical applications, the limitations of the proposed method include the following several aspects. Firstly, the proposed model focuses on classification and localization tasks, and segmentation accuracy needs improvement. In future work, it is essential to further research and optimize defect pixel-level segmentation tasks. This involves adjusting the network architecture to extract features for inferring defect positions and utilizing defect features (size, shape, color) to better differentiate specific categories. Secondly, it is observed that the accuracy of locating small and inconspicuous defects is lower in complex backgrounds. Further research on the precise localization of subtle and inconspicuous anomalies should also be conducted in future work.

REFERENCES

[1] A. Kamalzadeh, D. J. Gordon, and K. Erkorkmaz, "Robust compensation of elastic deformations in ball screw drives," *Int. J. Mach. Tools Manuf.*, vol. 50, no. 6, pp. 559–574, Jun. 2010.

[2] W. Zhang, X. Li, and Q. Ding, "Deep residual learning-based fault diagnosis method for rotating machinery," *ISA Trans.*, vol. 95, pp. 295–305, Dec. 2019.

[3] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Rel. Eng. Syst. Saf.*, vol. 182, pp. 208–218, Feb. 2019.

[4] S. Saifullah and R. Dreżewski, "Non-destructive egg fertility detection in incubation using SVM classifier based on GLCM parameters," *Proc. Comput. Sci.*, vol. 207, pp. 3254–3263, Jan. 2022.

[5] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019.

[6] Q. Xuan, Z. Chen, Y. Liu, H. Huang, G. Bao, and D. Zhang, "Multiview generative adversarial network and its application in pearl classification," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 8244–8252, Oct. 2019.

[7] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-supervised predictive convolutional attentive block for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13566–13576.

[8] 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, 2022.

[9] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-supervised learning for anomaly detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9659–9669.

[10] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRÆM—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8330–8339.

[11] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[12] T. Schlagenhauf, M. Landwehr, and J. Fleischer, "Industrial machine tool component surface defect dataset," 2021, *arXiv:2103.13003*.

[13] K. Perlin, "An image synthesizer," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 287–296, Jul. 1985.

[14] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14648–14658.

[15] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15979–15988.

[16] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 162, 2022, pp. 1298–1312.

[17] L. Jing, J. Zhu, and Y. LeCun, "Masked Siamese ConvNets," 2022, *arXiv:2206.07700*.

[18] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked Siamese networks for label-efficient learning," in *Proc. 17th Eur. Conf. Comput. Vis.—(ECCV)*. Tel Aviv, Israel: Springer, Oct. 2022, pp. 456–473.

[19] A. H. Pratomo, W. Kaswidjanti, A. S. Nugroho, and S. Saifullah, "Parking detection system using background subtraction and HSV color segmentation," *Bull. Electr. Eng. Informat.*, vol. 10, no. 6, pp. 3211–3219, Dec. 2021.

[20] S. Sood, B. Blakeley, and V. Rebuffel, "Defect detection method in digital radiography for porosity in magnesium castings," in *Proc. 9th Eur. Conf. NDT*, Berlin, Germany, Sep. 2006. [Online]. Available: https://www.ndt.net/?id=3670

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

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

[26] C. Hu and Y. Wang, "An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10922–10930, Dec. 2020.

[27] M. Azamfar, X. Li, and J. Lee, "Intelligent ball screw fault diagnosis using a deep domain adaptation methodology," *Mechanism Mach. Theory*, vol. 151, Sep. 2020, Art. no. 103932.

[28] Y. Cao, X. Xu, Z. Liu, and W. Shen, "Collaborative discrepancy optimization for reliable image anomaly localization," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 10674–10683, Nov. 2023.

[29] Y. Cai, D. Liang, D. Luo, X. He, X. Yang, and X. Bai, "A discrepancy aware framework for robust anomaly detection," *IEEE Trans. Ind. Informat.*, vol. 20, no. 3, pp. 3986–3995, Mar. 2024.

[30] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.

[31] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 13001–13008.

[32] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk management," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.

[33] Y. Wang, S. Tang, F. Zhu, L. Bai, R. Zhao, D. Qi, and W. Ouyang, "Revisiting the transferability of supervised pretraining: An MLP perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9173–9183.

[34] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.

[35] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[37] P. C. Mahalanobis, "On the generalized distance in statistics," *Indian Journal Statistics*, vol. 80, pp. S1–S7, Dec. 2008.

[38] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6726–6733.

[39] 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. (CVPR)*, Jun. 2019, pp. 9584–9592.

[40] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4182–4191.

[41] J. Yi and S. Yoon, "Patch SVDD: Patch-level SVDD for anomaly detection and segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 1–16.

[42] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proc. Int. Conf. Pattern Recognit.* Cham, Switzerland: Springer, Jan. 2021, pp. 475–489.

[43] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

XIAOMING WANG received the B.S. degree from Shanghai University of Electric Power, Shanghai, China, and the M.S. degree from Shanghai Jiao Tong University, Shanghai. She is currently pursuing the Ph.D. degree in control engineering with the University of Shanghai for Science and Technology, Shanghai.

Her research interests include computer vision and applications of deep learning.

YONGXIONG WANG received the B.S. degree from Harbin Engineering University, Harbin, China, and the M.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China.

He is currently a Professor with the University of Shanghai for Science and Technology. His research interests include the computer vision and intelligent robot.

ZHIQUN PAN received the B.E. degree from Ludong University, Yantai, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China.

His current research interests include deep learning, computer vision, face analysis, weakly supervised learning, and domain adaptation.

GUANGPENG WANG received the B.S. degree from Liaoning University of Technology, Jinzhou, China, and the M.S. degree from Shandong University of Science and Technology, Qingdao, China. He is currently pursuing the Ph.D. degree in control science and engineering with the University of Shanghai for Science and Technology, Shanghai, China.

His research interests include the computer vision and applications of deep learning.

JUNFAN CHEN is currently pursuing the bachelor's degree in intelligent science and technology with the University of Shanghai for Science and Technology, Shanghai, China.

His research interests include the computer vision and applications of deep learning.