

## Article

# Unsupervised Anomaly Detection on Metal Surfaces Based on Frequency Domain Information Fusion

Wenfei Wu <sup>1</sup>, Tao Tao <sup>1</sup>, Jinsheng Xiao <sup>1</sup>, Yichu Yao <sup>2</sup> and Jianfeng Yang <sup>1,\*</sup>

<sup>1</sup> School of Electronic Information, Wuhan University, Wuhan 430072, China; wuwenfei@whu.edu.cn (W.W.); tt1295@whu.edu.cn (T.T.); xiaojs@whu.edu.cn (J.X.)

<sup>2</sup> School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China; yichu\_yao@mail.nwpu.edu.cn

\* Correspondence: yjf@whu.edu.cn

**Abstract:** Metal products are widely used in industrial manufacturing, and the quality of metal products is becoming more and more demanding. At present, although there are many methods for detecting defects on metal surfaces, there are still various limitations. The limited number of defect samples, unpredictable defect characteristics, and the interference of metal grain bring great challenges to metal surface defect detection. For this reason, this paper proposes an unsupervised algorithm, FFnet, based on the fusion of frequency domain information, which introduces the frequency domain features into the unsupervised detection. A method of the adaptive enhancement of features in the frequency domain is proposed to make the features on the frequency domain more concerned with anomalies rather than textures. A scale-adaptive feature reconstruction module is used to effectively fuse the spatial and frequency domain features to fully utilize the information from different domains. In addition, a feature selection module is designed to improve the anomaly detection capability and reduce the computational redundancy by selecting the most representative subset of features. The proposed method outperforms other state-of-the-art methods on the connecting rod surface image dataset. In addition, in the generalization experiments of Kolektor Surface-Defect Dataset 2, our method also achieves optimal results and demonstrates strong generalization ability.

**Keywords:** defect detection; unsupervised learning; frequency domain; feature fusion



Received: 22 January 2025

Revised: 28 March 2025

Accepted: 31 March 2025

Published: 2 April 2025

**Citation:** Wu, W.; Tao, T.; Xiao, J.; Yao, Y.; Yang, J. Unsupervised Anomaly Detection on Metal Surfaces Based on Frequency Domain Information Fusion. *Sensors* **2025**, *25*, 2250. <https://doi.org/10.3390/s25072250>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Defect detection has long been a closely observed topic in both industrial production and scientific research. As widely used industrial products, metallic materials play a crucial role in various fields such as manufacturing, aerospace, and the automotive industry. The surface of metallic materials often contains various types of defects, such as scratches, cracks, corrosion, and holes. These defects can significantly affect the strength, durability, and safety of the products. Therefore, the timely and accurate detection of surface defects in metals is of great practical significance for improving product quality, reducing production costs, and ensuring safety.

In traditional production settings, defect detection has often been manually performed. This method is not only inefficient but also highly subjective, making it difficult to standardize. With the development of computer technology and image processing, defect detection has gradually shifted from traditional manual inspection to automated methods based on computer vision and deep learning, especially with the application of deep

learning techniques, which have significantly improved the accuracy and efficiency of defect detection.

However, metal surface defect detection technology still faces several challenges. Traditional image processing methods rely on hand-crafted features, which are susceptible to variations in lighting, surface contamination, and changes in viewing angles, leading to unstable detection results. Additionally, supervised learning-based methods [1,2] often require large annotated datasets. In the case of metal surface defect detection, the types and manifestations of defects are complex and varied. Defects often exhibit randomness and scarcity, and they may not appear in all samples. As a result, traditional supervised learning methods require a vast amount of annotated data to cover all possible defect types and scenarios, while acquiring defect samples in actual production environments is extremely difficult, leading to significant challenges when constructing datasets.

In recent years, with the advancement of deep learning technologies, an increasing number of unsupervised learning methods, such as autoencoders [3–6] and generative adversarial networks (GANs) [7–9], have been widely applied in anomaly detection. Unsupervised learning methods identify defects by learning the characteristics of normal samples and then detecting discrepancies between the sample to be inspected and the normal sample. These methods not only reduce the reliance on large annotated datasets but also enhance the robustness and generalization ability of the detection. However, the surface of metallic products differs from general industrial products in that it often contains complex and highly random metallic textures, and some defects may be concealed by these textures, causing significant interference in detection. Furthermore, metal surface defects typically appear in very subtle ways, making it difficult for existing detection models to effectively distinguish normal features from anomalous ones.

This study aims to analyze the challenges and difficulties associated with unsupervised anomaly detection algorithms in practical applications. Given the characteristics of metallic products, which feature random textures and subtle defects, we propose FFnet, an unsupervised anomaly detection method designed to address these challenges. FFnet is a novel framework that integrates both spatial and frequency domain features. The input image is passed through two feature extraction branches: The spatial branch utilizes the feature extraction capability of a large-scale pre-trained CNN to capture spatial information, while the frequency branch uses a patch-based discrete cosine transform (DCT) to extract information from the frequency domain. We have designed a frequency feature enhancement module (FFE) to fully leverage frequency information. Additionally, to fuse features from different domains, we have also designed two modules: the scale-adaptive feature reconstruction (SAFR) and reconstruction feature election (RFS) modules. SAFR aligns and reconstructs features from different scales, while RFS selects the most representative subset of features from the reconstructed ones. They effectively integrate features from different domains, enhancing the network performance while reducing redundant features, thus ensuring the network remains lightweight. The main contributions of this paper are as follows:

1. In this paper, we propose FFnet, a new framework for unsupervised anomaly detection that combines features in the spatial and frequency domains.
2. A frequency domain feature enhancement (FFE) method is designed to maximize the utilization of frequency domain information, thereby improving the detection performance of metal surface defects.
3. We also designed two additional modules: scale-adaptive feature reconstruction (SAFR) and reconstruction feature selection (RFS). SAFR and RFS effectively integrate features from different domains, enhancing network performance while reducing redundant features, thereby ensuring the network remains lightweight.

## 2. Related Works

### 2.1. Unsupervised Anomaly Detection

Currently, the problem of anomaly detection is mainly addressed using unsupervised and semi-supervised methods [10]. In the field of image anomaly detection, the prevailing approaches are still based on unsupervised algorithms, which can be broadly categorized into the following three types: image reconstruction-based methods, deep feature embedding-based methods and anomaly simulation-based methods.

**Reconstruction-based approaches** focus on anomaly detection and localization by training a model that uses only normal images and exploiting the inability of the module to effectively reconstruct anomalous regions. Common reconstruction techniques include the autoencoder (AE) [3–6], generative adversarial network (GAN) [7–9], transformer [11,12], and diffusion model [13], etc. Matsubara et al. [4] firstly introduced the variational autoencoder (VAE) into the field of industrial anomaly detection. Kozamernik et al. [14] proposed a model for the visual quality control of KTL coatings based on VAE. This method successfully detects anomalous regions containing surface defects by calculating the negative log-likelihood of the return distribution of the decoder, and Schlegl et al. [7] were the first to apply generative adversarial networks (GANs) to anomaly localization. Balzategui et al. [15] employed a GAN-based anomaly location (AL) method to conduct quality inspection of monocrystalline solar cells. Hou et al. [16] utilized the method to construct a segmentation and assembly framework for anomaly localization. Recent studies [17–19] have improved the performance of anomaly detection by pre-training CNN models and reconstructing image features at multiple scales. However, in the case of complex image texture or structure, the anomalous regions may share the same features with normal regions, which leads to the anomalies being incorrectly reproduced in the image reconstruction process, thus affecting the detection results.

**Embedding-based approaches** obtain the deeply embedded features of an image through feature extraction and then generate an anomaly score map by comparing the differences between normal and test features. Typical methods [20–22] utilize pre-trained networks on ImageNet [23] for feature extraction. Defard et al. [20] represent the embedding of extracted anomalous patch features through multivariate Gaussian distribution. Roth et al. [21] adopt a core subset by means of a greedy strategy to form a memory bank for anomaly detection. Bae [22] et al., on the other hand, considered patch location and neighborhood relationship and used location information by constructing histograms of representative features at each location. In the anomaly detection stage, the input features are usually scored using either the Mahalanobis distance or the maximum feature distance. Embedding-based methods have achieved remarkable results in the field of anomaly detection; however, since these methods rely on pre-trained models of ImageNet for feature extraction, the embedded features are inevitably affected by the bias of the ImageNet dataset, which may affect the effectiveness of anomaly detection in specific tasks.

**Anomaly Simulation-Based approaches** simulate anomalous data by artificially synthesizing anomalies on normal images. Vitjan Zavrtanik et al. [24] proposed an end-to-end network, DRAEM, which learns and detects anomalies just synthesized from distribution patterns through discriminative training. CutPaste [25] proposed a simple strategy to generate synthetic anomalies for anomaly detection by cutting image chunks and pasting them to random locations in a large image. Simplenet [26] proposed a method of directly adding noise to the feature dimensions to simulate anomalies. In [27], a new end-to-end memory-assisted segmentation network, Memseg, is proposed to simulate anomaly samples by generating anomaly masks combined with DTD texture data [28], while RealNet [29] simulates anomalies using a diffusion model, adding interference during the diffusion process to make the synthesized anomaly images closer to actual anomalies. Although the

anomaly simulation methods are effective to a certain extent, they rely too much on the quality of anomaly simulation, and therefore, there are still some limitations and challenges for unpredictable or rare types of anomalies.

## 2.2. Deep Learning in Frequency Domain

Frequency analysis has long been a powerful tool in the field of signal processing, and in recent years, its application in deep learning has gradually gained attention. The study in [30] enhanced deep convolutional neural networks for ultrasonic concrete inspection by utilizing continuous wavelet transform and transfer learning. In the field of image processing, the frequency domain information of images is also widely integrated into convolutional neural networks. In [31], frequency domain information from images was introduced into convolutional neural networks (CNNs) through JPEG encoding. The research in [32] proposed a model transformation algorithm that converts CNN models from the spatial domain to the frequency domain. In [33], discrete cosine transform (DCT) was used to integrate frequency domain information into CNNs in the form of residuals, avoiding the complex model transformation process, and squeeze-and-excitation block (SE-Block) was used to select frequency channels. Other studies [34] designed a frequency channel attention network to further optimize the use of frequency information. Additionally, [35] introduced a learnable frequency enhancement module and aligned the RGB domain with the frequency domain to more effectively utilize frequency information.

In the studies presented in [33–35], frequency domain information is extracted using discrete cosine transform and integrated into the CNN as residuals. These works have demonstrated the feasibility and effectiveness of fusing frequency domain features with CNN. As illustrated in the Figure 1, we present an example of a metallic surface defect, where the blue box indicates the normal region, and the red box highlights the defect region. Due to the presence of random metallic textures on the surface, there is a certain degree of interference with the detection of defects. We perform frequency analysis on these two regions and plot the statistical results of the frequency signal. The curve graph reveals a clear difference in the frequency signals between the two regions. Therefore, frequency information can be used as a supplement to enhance the CNN model's ability to perceive and detect metal anomalies.

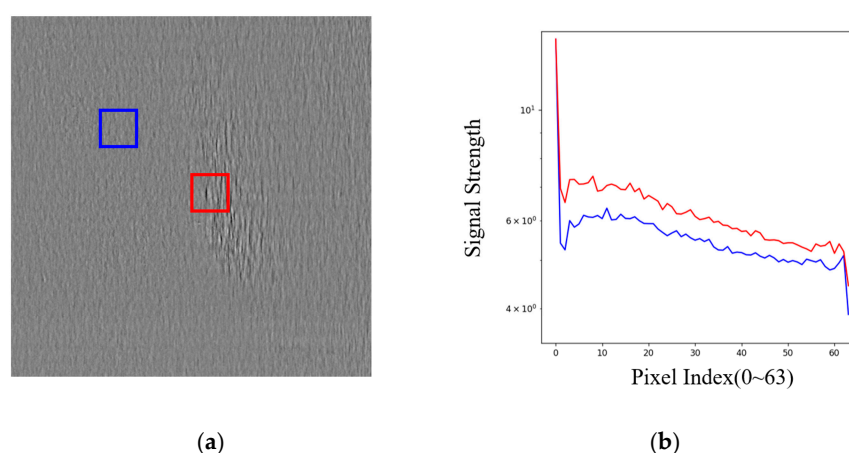


**Figure 1.** (a) Display of metal surface defect sample, with the abnormal region highlighted in the red box and the normal region highlighted in the blue box; (b) display of the frequency spectra for the two regions. The blue curve represents the frequency signal of the normal region within the blue box; the red curve represents the frequency signal of the abnormal region within the red box.

Although unsupervised anomaly detection methods have made significant progress in recent years, research on combining frequency domain and spatial domain features for

anomaly detection remains scarce. Unlike existing works, we innovatively integrated both spatial and frequency domain features during feature extraction, effectively combining these two domains. Furthermore, we utilized a reconstruction module to further learn and exploit the rich information from both types of features. This cross-domain fusion approach enables the model to more effectively capture anomaly patterns in images, significantly improving detection accuracy and robustness, thus providing a more comprehensive and precise solution for anomaly detection tasks.

### 3. Proposed Method

The FFnet algorithm proposed in this paper is a novel anomaly detection framework that integrates spatial and frequency domains features. The network architecture, as shown in Figure 1, mainly consists of two feature extraction branches, a feature reconstruction module (SAFR), a feature selection module (RFS), and a discriminator.

As illustrated in Figure 2, our proposed method is a dual-branch feature extraction network framework that combines spatial and frequency features. The spatial feature extraction branch is similar to that of traditional methods [20–22], utilizing a pre-trained convolutional neural network (CNN) for feature extraction. In the frequency domain branch, patch-based discrete cosine transform is used to extract frequency domain information. To improve the applicability of the frequency domain information, we have designed a learnable frequency feature enhancement module named FFE, which adaptively enhances features across different frequency bands, allowing the network to focus more on the useful frequency bands. To effectively integrate the features from different branches, we also designed the SAFR and RFS modules. SAFR is responsible for fusing spatial and frequency features from different scales, while RFS selects the most useful channels from the reconstructed features and discards redundant ones. Finally, the anomaly discriminator, similar to previous method [26], detects anomalies by evaluating anomaly scores.

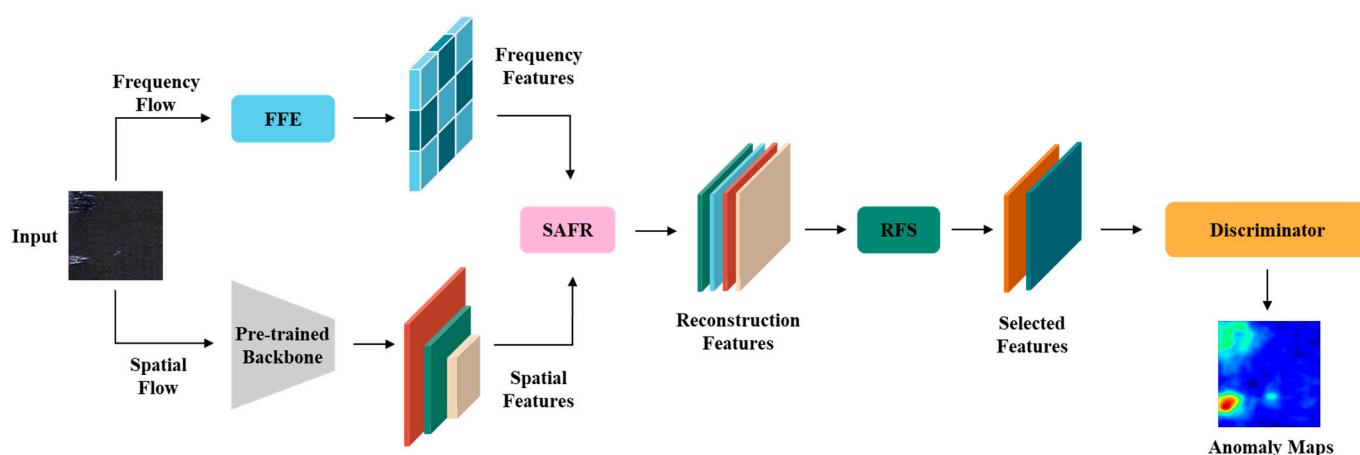


**Figure 2.** Our method employs a dual-branch feature extraction architecture that integrates both spatial and frequency information for anomaly detection.

#### 3.1. Frequency Domain Feature Enhancement

As mentioned earlier, frequency domain information can serve as a supplement to CNN, enhancing the representation of features. To better perceive and distinguish abnormal patterns, we have developed a new feature extraction framework. This framework adds an additional frequency domain feature extraction branch on top of the traditional pre-trained network feature extractor. This dual-branch feature extraction architecture enhances the representation of traditional CNN features by incorporating frequency domain information. We design a frequency feature enhancement (FFE) module to extract and enhance frequency

domain features. The implementation of the FFE module primarily involves two key steps. One is a patch-based DCT transformation for extracting frequency domain features, and the other is a learnable frequency enhancement module. Its structure is shown in Figure 3.

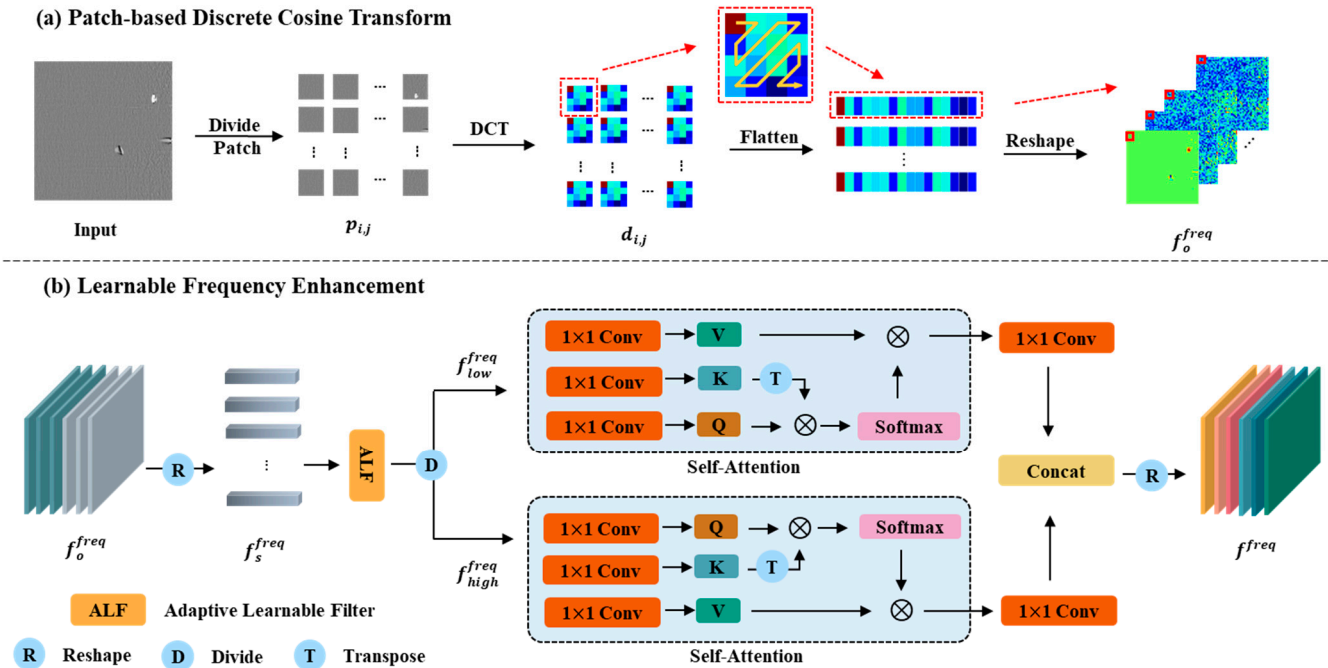


**Figure 3.** The structure of FFE consists of two steps: (a) patch-based DCT transformation to extract partitioned frequency domain features; (b) a learnable frequency feature enhancement module.

**Patch-based Discrete Cosine Transform:** In this section, the input image needs to be transformed by DCT to obtain the frequency domain information. As shown in Figure 3a, the image is first divided into patches, with each patch having a size of  $k \times k$ , resulting in a set of patches  $\{p_{i,j} \mid 1 \leq i \leq \frac{H}{k}, 1 \leq j \leq \frac{W}{k}\}$ . Then, a DCT transformation is applied to each patch, converting them into frequency spectra  $d_{i,j} \in \mathbb{R}^{k \times k}$ . After the DCT transformation, the frequency signals of the image are represented in the frequency spectrum, with low-frequency signals concentrated in the top-left corner and high-frequency signals in the bottom-right corner. Following the sequence shown in Figure 3a, we expand the frequency signals from high frequency to low frequency. At this point, the frequency signals of each patch are expanded into a sequence of length  $k^2$ . Next, based on the position of each patch in the original image, the frequency signals of the same frequency band are concatenated together. In this way, we transform the original image into the frequency domain, obtaining the frequency domain feature  $f_o^{freq} \in \mathbb{R}^{\frac{H}{k} \times \frac{W}{k} \times k^2}$ .

The frequency domain features  $f_o^{freq}$  obtained through the patch-based segmentation method are completely independent of each other between different patches. To enhance the correlation between neighboring patches and increase the receptive field, we apply a local feature aggregation strategy to  $f_o^{freq}$ . We define the feature vector at position  $(h, w)$  in  $f_o^{freq}$  as  $f_o^{freq}(h, w)$ , and the neighborhood of the feature vector at position  $(h, w)$  as  $N_p^{(h,w)}$ . The formula is as follows:

$$N_p^{(h,w)} = \{(a, b) \mid a \in [h - \lfloor p/2 \rfloor, \dots, h + \lfloor p/2 \rfloor], b \in [w - \lfloor p/2 \rfloor, \dots, w + \lfloor p/2 \rfloor]\}. \quad (1)$$

In this case,  $p$  represents the range of the neighborhood. We choose  $p = 3$ , meaning that the neighborhood consists of the surrounding  $3 \times 3$  patches. For each position  $(h, w)$ ,

we calculate the aggregated feature  $f_a^{freq}(h, w)$  for its neighborhood  $N_p^{(h,w)}$ . The formula is as follows:

$$f_a^{freq}(h, w) = \text{agg}\left(\left\{f_o^{freq}(a, b) \mid (a, b) \in N_p^{(h,w)}\right\}\right). \quad (2)$$

In Formula (2),  $\text{agg}$  is the function that performs the aggregation of the vectors in the neighborhood  $N_p^{(h,w)}$ . We use the adaptive mean pooling method for aggregation, and the formula is as follows:

$$\text{agg}(h, w) = \frac{1}{p \times p} \sum_{(a,b) \in N_p^{(h,w)}} f_o^{freq}(a, b). \quad (3)$$

**Learnable Frequency Enhancement:** In practical industrial scenarios, we have found that the textures and defects on metal surfaces are highly complex and variable. Features derived solely from fixed DCT transforms may not meet the requirements for anomaly detection. Therefore, we designed a learnable adaptive frequency enhancement module to improve the applicability of frequency information. As shown in Figure 3b, this module consists of two main components: the adaptive learnable filter (ALF) and the self-attention mechanism (SA) [36,37]. Due to the learnable nature of ALF, it can adaptively enhance features across different frequencies. The SA mechanism can dynamically consider the mutual interactions between features across different frequencies.

First, the frequency domain features undergo a reshaping operation, where  $f_a^{freq} \in \mathbb{R}^{\frac{H}{k} \times \frac{W}{k} \times k^2}$  is reshaped into  $f_s^{freq} \in \mathbb{R}^{\frac{H \times W}{k^2} \times k^2}$ . Define  $f_{Rj}^{freq}$  as the  $j$ -th column vector of  $f_s^{freq}$ . ALF is a parameterized learnable adaptive filter with a total of  $H \times W$  parameters. Let its parameters be defined as follows:

$$W_{ALF} = \left\{w_j \mid w_j \in \mathbb{R}^{\frac{H \times W}{k^2}}, 1 \leq j \leq k^2\right\}. \quad (4)$$

Then, the  $f_s^{freq}$  is fed into the ALF for adaptive filtering, resulting in  $f_f^{freq} \in \mathbb{R}^{\frac{H \times W}{k^2} \times k^2}$ . The formula is as follows:

$$f_f^{freq} = F_{ALF}(f_s^{freq}), \quad (5)$$

$$F_{ALF}(f_s^{freq}) = \left\{w_j \odot f_{Rj}^{freq} \mid 1 \leq j \leq \frac{H \times W}{k^2}\right\}, \quad (6)$$

$\odot$  is the element-wise multiply operation. Next, we divide the filtered features into two parts, low-frequency and high-frequency, as  $f_{low}^{freq}, f_{high}^{freq} \in \mathbb{R}^{\frac{H \times W}{k^2} \times \frac{k^2}{2}}$ . These two parts are then sent into two separate SA modules for further processing. The SA module allows features from different frequencies to interact with each other. By enhancing the low-frequency and high-frequency signals separately, we maximize the interaction between features across different frequencies while ensuring the distinction between low-frequency and high-frequency information. Finally, the outputs of the two SA modules are concatenated and reshaped back to the original dimensions, resulting in the enhanced feature  $f^{freq} \in \mathbb{R}^{\frac{H}{k} \times \frac{W}{k} \times k^2}$ .

### 3.2. Scale-Adaptive Feature Reconstruction

We introduce frequency domain features to assist the network in better detecting anomalies on metal surfaces. To effectively fuse frequency domain features with spatial features, we design a scale-adaptive feature reconstruction (SAFR) module. Due to the strong randomness of defects on metal surfaces, their size and shape can vary significantly. Features at different scales have varying receptive fields. Deeper features focus more on structural information, while shallower features are more sensitive to textures. Fusing features at different scales can enhance the network's stability in dealing with scale varia-

tions. SAFR primarily consists of four scale transformation modules (STM) and a feature reconstruction module based on MLP (multi-layer perceptron), as shown in Figure 4.

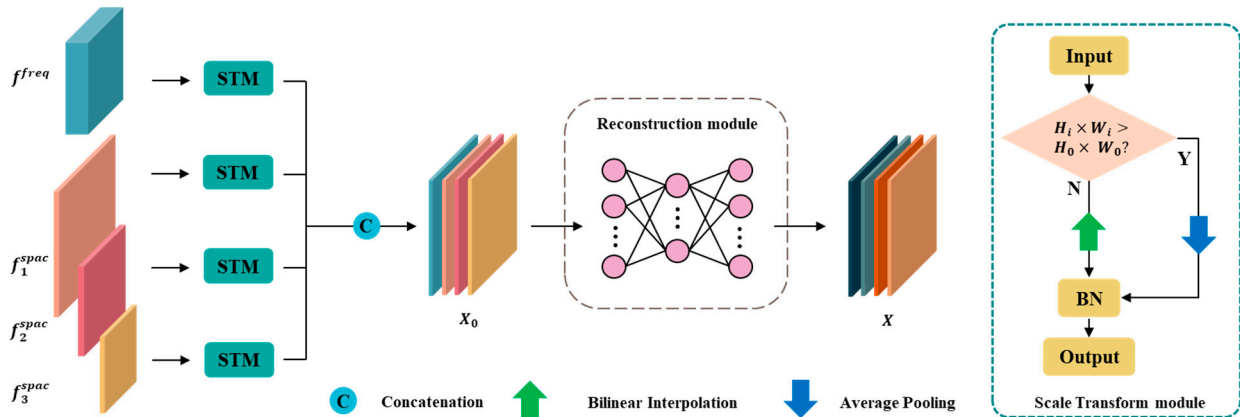


**Figure 4.** SAFR mainly consists of the STM and the reconstruction module, which are capable of fusing and reconstructing features of different sizes.

In the spatial feature extraction section, we use the pre-trained WideResNet50 [38] as the backbone network. Unlike PatchCore, to retain more detailed information, we use the outputs from three layers of the backbone network as feature extractors. The extracted spatial domain features can be represented as  $\{f_1^{spac}, f_2^{spac}, f_3^{spac}\}$ . Since spatial domain features and frequency domain features have different sizes, they need to be aligned before fusion. Here, we use the scale transformation module (STM) to perform scale normalization, ensuring that all feature blocks are adjusted to the size  $(H_0, W_0)$ . In the STM, we use average pooling layer and bilinear interpolation to achieve the downscaling and upscaling of feature sizes. We denote the STM as  $T_s$ , and the transformation process is as follows:

$$X_i^{spac} = T_s(f_i^{spac}), \quad (7)$$

$$X^{freq} = T_s(f^{freq}). \quad (8)$$

The feature sizes after transformation by the STM module are listed in Table 1. Next, along the channel dimension, we concatenate the spatial features with the frequency domain features according to their corresponding positions, as shown below:

$$X_0 = \text{Concat}(X_i^{spac}, X^{freq}). \quad (9)$$

**Table 1.** The sizes of the input and output features in the STM module.

| Input Feature | Size                                           | Output Feature | Size                         |
|---------------|------------------------------------------------|----------------|------------------------------|
| $f_1^{spac}$  | $\frac{H}{4} \times \frac{W}{4} \times 256$    | $X_1^{spac}$   | $H_0 \times W_0 \times 256$  |
| $f_2^{spac}$  | $\frac{H}{8} \times \frac{W}{8} \times 512$    | $X_2^{spac}$   | $H_0 \times W_0 \times 512$  |
| $f_3^{spac}$  | $\frac{H}{16} \times \frac{W}{16} \times 1024$ | $X_3^{spac}$   | $H_0 \times W_0 \times 1024$ |
| $f^{freq}$    | $\frac{H}{k} \times \frac{W}{k} \times k^2$    | $X^{freq}$     | $H_0 \times W_0 \times k^2$  |

At this point, the spatial and frequency domain features are in an independent state. Then, we use an MLP structure as the feature reconstruction module  $G_R$ . The MLP structure is capable of capturing complex nonlinear relationships in the data, enabling the more effective integration of features from different domains. The concatenated features  $X_0$  are

fed into the reconstruction module, producing the fused features  $X$ , which combine both spatial domain and frequency domain information, as shown below:

$$X = G_R(X_o). \quad (10)$$

### 3.3. Reconstruction Feature Selection

As mentioned in Section 3.2, to preserve more useful features, we selected the outputs of three intermediate layers in the spatial module and integrated them with frequency domain features. Not all of these features directly contribute to anomaly perception and detection. To enhance the feature representation of the model and reduce feature redundancy, we designed a reconstruction feature selection (RFS) module. Inspired by the channel attention mechanism [39], this approach directs the network to focus more on the feature information from important channels. Different channels carry varying amounts of information. Some channels contain more useful features, while others may be redundant or irrelevant. During training, a weight coefficient is assigned to each channel of the input features, where the magnitude of the coefficient indicates the importance of the corresponding channel. The channels are ranked according to their weight coefficients, retaining the most important channels while discarding the redundant ones. The RFS module selects the most representative subset of features from the reconstructed feature map in this manner, while also reducing feature redundancy, making the network more lightweight. The structure of the RFS is shown in Figure 5.

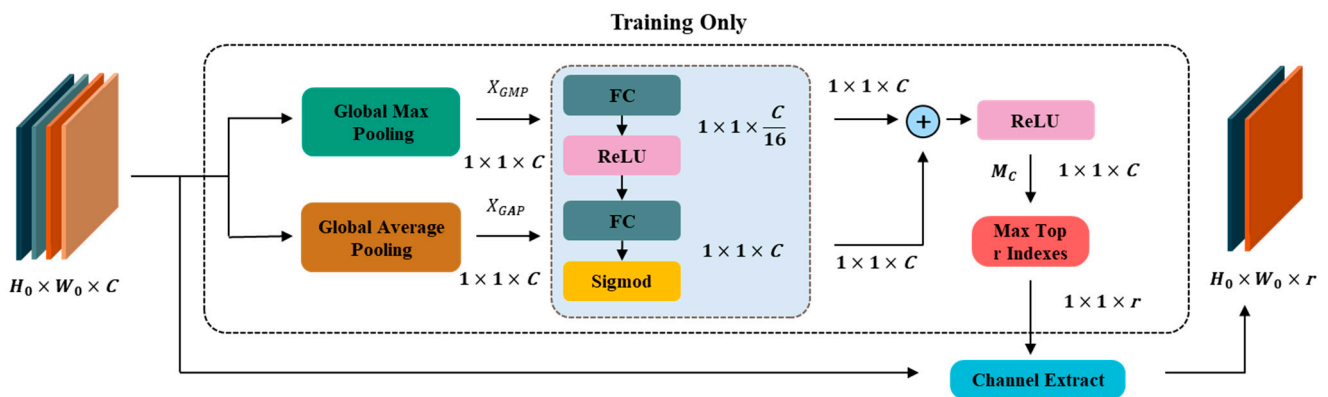


**Figure 5.** By applying global max pooling and global average pooling to the reconstructed features, we selectively extract the parts of the features that have local and global representativeness, respectively.

First, global maximum pooling (GMP) and global average pooling (GAP) are applied to  $X$ , yielding  $X_{GMP}$  and  $X_{GAP}$ , respectively. Then, both are passed through a shared MLP, and the output results are summed and activated by a ReLU function to obtain a weight matrix  $M_C$ . We select the top  $r$  values from  $M_C$  and obtain the position indexes of these  $r$  values. The channels at these  $r$  positions represent the most representative reconstructed feature channels after GMP and GAP. To ensure model consistency, the step of obtaining the  $r$  indexes through global pooling is performed only during the training process.

GMP and GAP represent local and global features in space, respectively.  $X_{GMP}$  focuses more on local information, while  $X_{GAP}$  emphasizes global information. By combining both, we enhance the model's ability to perceive anomalies at different scales. At this point, we obtain the final feature representation  $\chi \in \mathbb{R}^{H_0 \times W_0 \times r}$ , as shown in the following formula:

$$\chi = RFS(X, r). \quad (11)$$

### 3.4. Discriminator and Loss Function

In the anomaly detection section, we adopt a two-layer MLP architecture as the discriminator  $D$ , which is used to assess the normality of each patch. During training, we use only normal sample images as the training set. We simulate anomalous features by adding Gaussian random noise to the normal features. We represent the normal features as  $\chi_{h,w}$ , where  $(h, w)$  denotes the position of the patch. The simulated anomalous features are represented as  $\chi_{h,w}^-$ .  $\chi_{h,w}$  represents positive samples, and  $\chi_{h,w}^-$  represents negative samples. This anomaly simulation-based self-supervised method is used to learn the features of normal samples. The anomalous features  $\chi_{h,w}^-$  are obtained by adding Gaussian random noise  $\epsilon \in \mathbb{R}^r$  to the normal features  $\chi_{h,w}$ . The formula is as follows:

$$\chi_{h,w}^- = \chi_{h,w} + \epsilon. \quad (12)$$

We use a truncated L1 loss as the loss function for training [26].  $\mathcal{L}$  represents the training objective, where  $th^+$  and  $th^-$  are truncation terms to prevent overfitting. The formula is as follows:

$$l_{h,w}^i = \max\left(0, th^+ - D\left(q_{h,w}^i\right)\right) + \max\left(0, -th^- + D\left(q_{h,w}^{i-}\right)\right), \quad (13)$$

$$\mathcal{L} = \min \sum_{q^i \in \chi_{train}} \sum_{h,w} \frac{l_{h,w}^i}{H_0 * W_0}. \quad (14)$$

The discriminator  $D$  performs discrimination on the features at each position  $(h, w)$  and outputs an anomaly score  $s_{h,w}$  for the patch at that position. By performing this evaluation over all patches in the feature map, we obtain an anomaly score map of the same size as the feature map. We use the maximum anomaly score from all patches as the anomaly score  $S_A$  for the entire image.  $S_A$  is used to determine whether an image is anomalous, while  $s_{h,w}$  is used to assess the anomaly of each individual patch. The formula is as follows:

$$s_{h,w} = -D(\chi_{h,w}), \quad (15)$$

$$S_A = \max_{(h,w) \in H_0 \times W_0} s_{h,w}. \quad (16)$$

We upsample the anomaly score map of size  $H_0 \times W_0$  to  $H \times W$ , obtaining an anomaly score output  $S_{AL}$  that matches the size of the input image. The score at each position in  $S_{AL}$  represents the anomaly score of the corresponding pixel. The formula is as follows:

$$S_{AL} = \{s_{h,w} | (h, w) \in H \times W\}. \quad (17)$$

## 4. Experiments and Results

### 4.1. Datasets

The dataset used in this study consists of surface images of automotive connecting rods, collected from an industrial production line. Since the automotive connecting rods have cylindrical surfaces, their surface images cannot be captured using conventional area array cameras. Therefore, we designed a specialized imaging system for the connecting rods. The system rotates the connecting rod on a roller and uses a Keyence line scan camera (CA-HL08MX) to scan the surface, thereby obtaining an unfolded map of the surface. To ensure consistent image brightness, a parallel light source is employed for illumination. The specific configuration of the system and the placement of the equipment are shown in Figure 6.

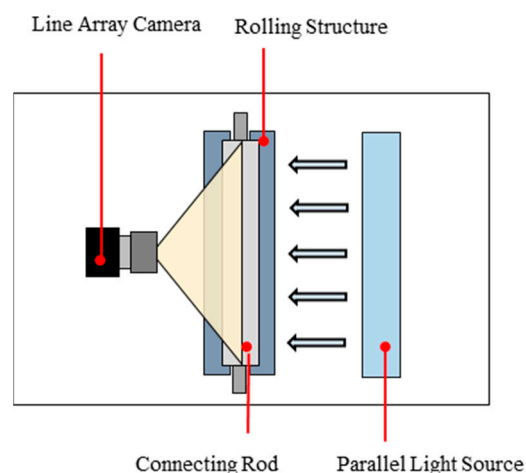


**Figure 6.** We designed a surface image acquisition system for the connecting rod, which primarily includes a line array camera, a rolling structure, and a parallel light source. The surface image of the connecting rod is captured by rotating it through one full rotation using the rolling structure.

In this study, data were collected for 100 automotive connecting rod samples, resulting in a total of 400 surface images with a resolution of  $6000 \times 8000$  pixels (four images per connecting rod). To facilitate network training, all full-sized images were cropped to a resolution of  $1024 \times 1024$  pixels. The dataset construction is based on the cropped images. As shown in Figure 7, Figure 7a shows the complete connecting rods, and Figure 7b–e represent cropped patches of the images. Due to the extreme rarity of surface defect samples, only 150 defect-containing images were collected. To balance the dataset, 150 normal surface images were randomly selected to form the test set. Considering the computational resource demands of unsupervised learning, we aimed to minimize the data size for training. As a result, 200 normal images were used as the training set. This dataset is named CRS (connecting rod surface image dataset), consisting of a total of 500 images, where the training set includes 200 normal images, and the test set contains 150 normal images and 150 anomalous images. The dataset is annotated using Labelme. The anomaly regions are annotated using a binary classification segmentation method.

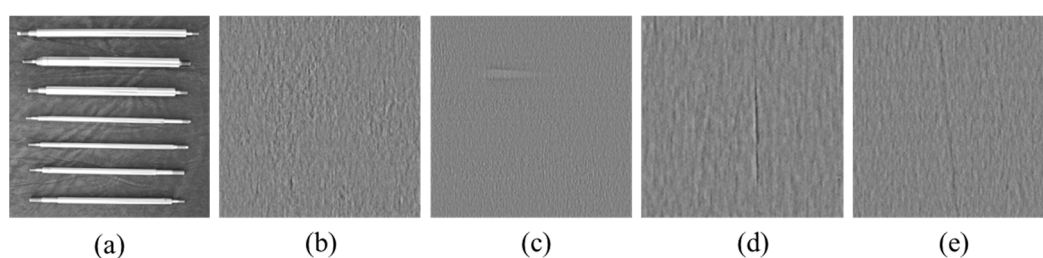


**Figure 7.** Examples of connecting rod images: (a) shows the full image; (b–e) display cropped local patches.

Additionally, we conducted generalization experiments on the Kolektor Surface-Defect Dataset 2 (KSDD2) [40]. Since KSDD2 is a supervised learning dataset, to meet the experimental requirements, we randomly selected 200 normal samples from the original training set as the training set, and 100 normal samples and 100 defect samples from the test set to form a new test set.

#### 4.2. Experimental Configuration and Experimental Details

The experiments in this paper were conducted on an Ubuntu 20.04 system, with an Intel i9–13900K CPU and an Nvidia GeForce RTX 4090 GPU, as detailed in Table 2. In the experiments, all input images are uniformly resized to  $256 \times 256$  pixels. The patch size

$k$  for the patch-based DCT transformation is set to 4. A pre-trained WideResNet50 was used as the backbone network, with the first, second, and third layers serving as spatial feature extractors. The reconstruction feature scale was unified to  $64 \times 64$ , and the feature selection dimension  $r$  was set to 1536. The Adam optimizer was used, with the learning rate set to 0.0001 and weight decay set to 0.00001. The number of training epochs for each dataset was set to 100, and the batch size was 4.

**Table 2.** Experiment environment parameters.

| Hardware or Software | Version      |
|----------------------|--------------|
| CPU                  | Intel 13900K |
| GPU                  | RTX 4090     |
| RAM                  | 64G          |
| Operating system     | Ubuntu 20.04 |
| CUDA version         | 12.0         |

#### 4.3. Results of the CRS Dataset

To validate the effectiveness and superiority of the proposed method, we compared it with several representative methods in the anomaly detection field, including PaDiM [20], PatchCore [21], MemSeg [27], and SimpleNet [26]. To assess the performance of these methods, we used the AUROC (area under the receiver operating characteristic curve) and F1 score for both classification and segmentation tasks as evaluation metrics. The calculation formulas are as follows:

$$\text{TPR(Recall)} = \frac{TP}{TP + FN}, \quad (18)$$

$$\text{FPR} = \frac{FP}{TN + FP}, \quad (19)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (20)$$

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x)) dx, \quad (21)$$

$$\text{F1} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (22)$$

Here,  $TP$  represents the anomalous samples correctly classified as anomalies, and  $FP$  represents the normal samples incorrectly classified as anomalies. For image-level, the statistics are calculated on a per-image basis, while for pixel-level, the results are computed on a per-pixel basis. The experimental results for the connecting rod are shown in Table 3. We compare the proposed method, FFNet, with several mainstream methods in terms of image classification AUROC, F1 score, as well as pixel segmentation AUROC and F1 score. Our method achieved the best results in both image AUROC and pixel AUROC, with scores of 99.4% and 95.4%, respectively. Additionally, the F1 scores for both image classification and pixel segmentation were the highest among all algorithms, reaching 98.4% and 31.8%, respectively.

As shown in Figure 8, our method demonstrates a clear visual advantage over other methods, particularly excelling in handling samples with mild defects. Compared to SimpleNet, which also uses WideResNet as the backbone, our method performs better in detecting shallow defects that are easily obscured by surface metal textures. This advantage is attributed to the incorporation of frequency domain features, which enhance our method's ability to perceive and distinguish between textures and defects.

**Table 3.** Comparison results on CRS. Anomaly detection and localization performance are measured based on Image AUROC (%), Image F1 (%), Pixel AUROC (%) and Pixel F1 (%). Additionally, we calculated the execution efficiency of the algorithm, expressed in FPS.

|             | Image AUROC | Image F1    | Pixel AUROC | Pixel F1    | FPS       |
|-------------|-------------|-------------|-------------|-------------|-----------|
| PaDiM       | 94.2        | 93.6        | 91.5        | 20          | 1         |
| PatchCore   | 99.1        | 97.8        | 93.6        | 28          | 16        |
| MemSeg      | 95.8        | 94.2        | 92.1        | 22.6        | 25        |
| SimpleNet   | 98.5        | 97.5        | 87.4        | 21.5        | 51        |
| <b>Ours</b> | <b>99.4</b> | <b>98.4</b> | <b>95.4</b> | <b>31.8</b> | <b>43</b> |

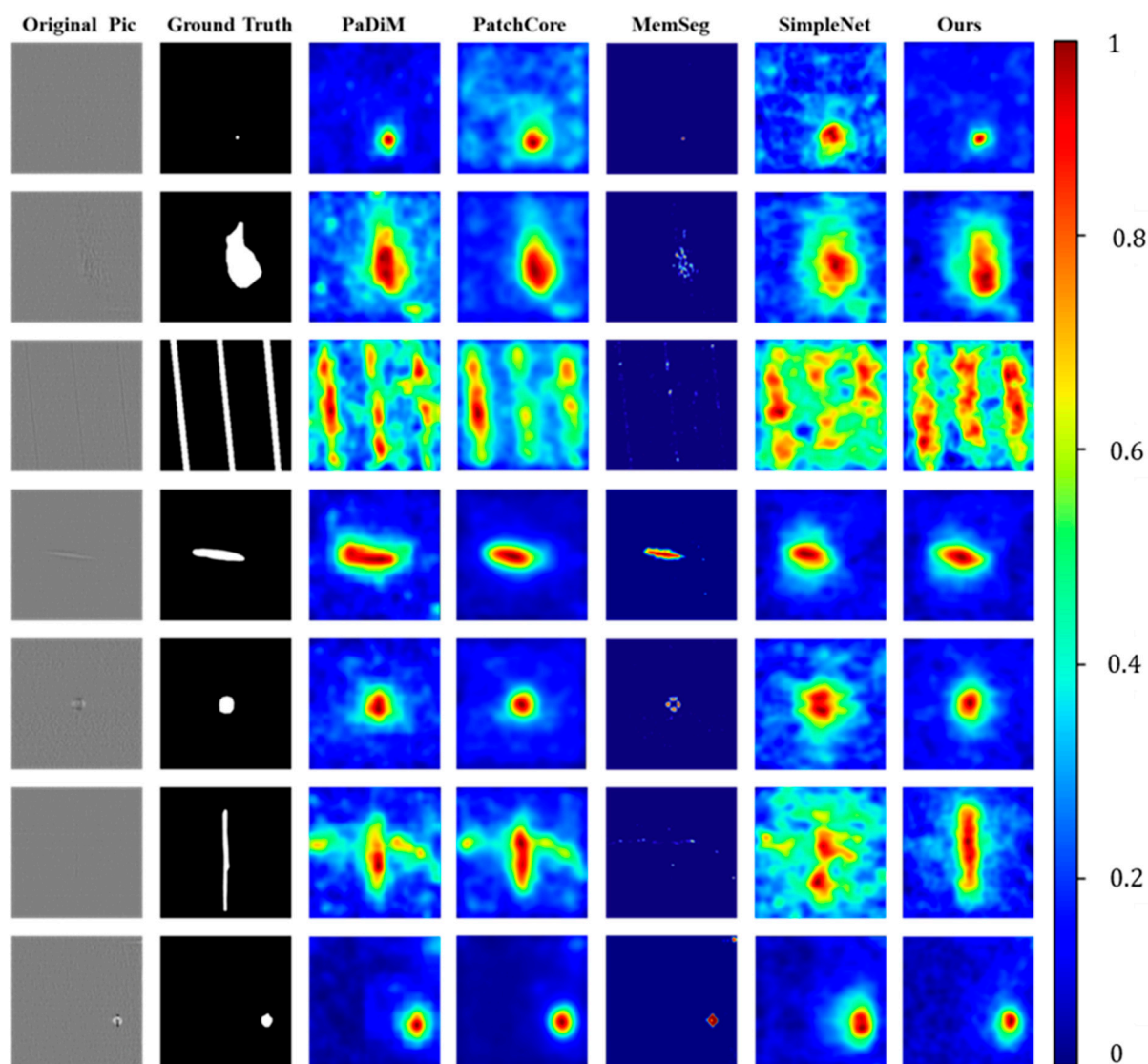


**Figure 8.** The experimental results on the CRS dataset are visualized as heatmaps. Anomalous outputs are represented in these heatmaps, where higher anomaly scores are more prominently displayed, indicating regions with more significant abnormalities.

In the segmentation task, although MemSeg excels in detecting obvious defects by adding a segmentation subnet at the end of the network; this design leads to noticeable false negatives in samples with less prominent defect features. As shown in Figure 8, the defect in the fourth sample is more prominent, and the segmentation performance of MemSeg is relatively good. However, the performance on the third and sixth samples

is noticeably worse. In industrial applications, the risk of false negatives often has low tolerance. Therefore, our method offers a stronger advantage over these classic anomaly detection methods, particularly in industrial scenarios where high precision is required, ensuring more stable and reliable performance.

Additionally, we conducted a comparative analysis of inference speeds across different algorithms. As shown in Figure 9, the  $y$ -axis represents I-AUROC while the  $x$ -axis denotes FPS. In this visualization, positions further to the right indicate faster inference speeds (higher FPS values), whereas positions higher up correspond to superior detection performance (elevated Image AUROC). Our method achieves the best detection performance (highest I-AUROC) while closely following SimpleNet in inference speed (second highest FPS). As evidenced by its positioning in the upper-right quadrant of Figure 9, the proposed approach demonstrates an optimal balance between detection efficacy and computational efficiency. Despite the incorporation of additional frequency domain features, our method maintains competitive inference speeds through the optimized architecture of the RFS module.

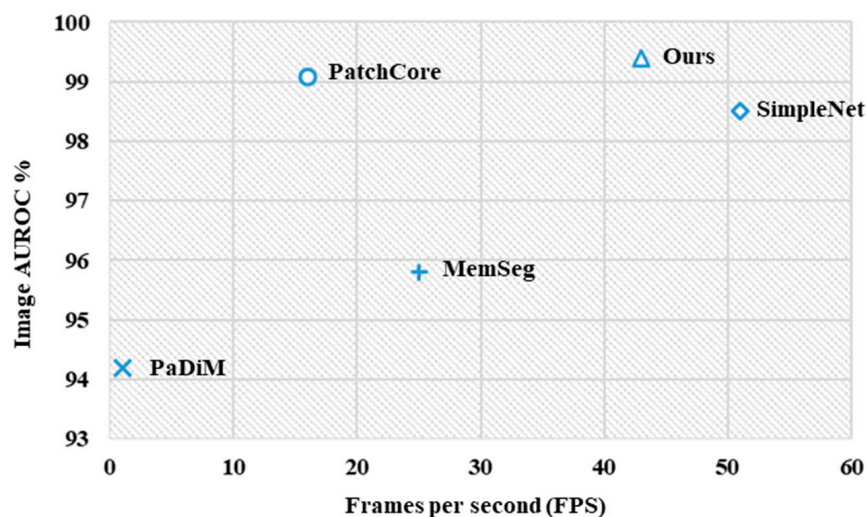


**Figure 9.** Inference speed versus Image AUROC on CRS.

#### 4.4. Results of the KSDD2 Dataset

The results are shown in Table 4. On the KSDD2 dataset, our method still achieved the best performance. In terms of image-level AUROC and F1 scores, we reached 96.9% and 93.6%, the highest values among all methods. At the pixel level, our method achieved AUROC and F1 scores of 97.6% and 52.8%, respectively. Although our Pixel F1 score is slightly lower than that of MemSeg, we rank first in all other metrics, especially in Image AUROC, where we outperform the second-best PatchCore by 1.7%. As shown in Figure 10, our method still demonstrates significant visual advantages, particularly excelling in detecting smaller defects. Although MemSeg slightly outperforms our method in Pixel F1 score, its introduction of a segmentation subnet at the end of the network leads to more severe false negatives. Considering all the metrics and false negative situations, our method shows stronger generalization capability compared to MemSeg. This is particularly valuable in practical industrial applications, where it holds great potential for handling various defect detection tasks.

**Table 4.** Comparison results on KSDD2. Anomaly detection and localization performance are measured based on Image AUROC (%), Image F1 (%), Pixel AUROC (%) and Pixel F1 (%).

|             | Image AUROC | Image F1    | Pixel AUROC | Pixel F1    |
|-------------|-------------|-------------|-------------|-------------|
| PaDiM       | 92.5        | 89.3        | 97.1        | 49.2        |
| PatchCore   | 95.2        | 91.3        | 97.5        | 50.1        |
| MemSeg      | 93.3        | 88.7        | 93.2        | <b>53.8</b> |
| SimpleNet   | 94.9        | 92.9        | 96.8        | 46.1        |
| <b>Ours</b> | <b>96.9</b> | <b>93.6</b> | <b>97.6</b> | <b>52.8</b> |

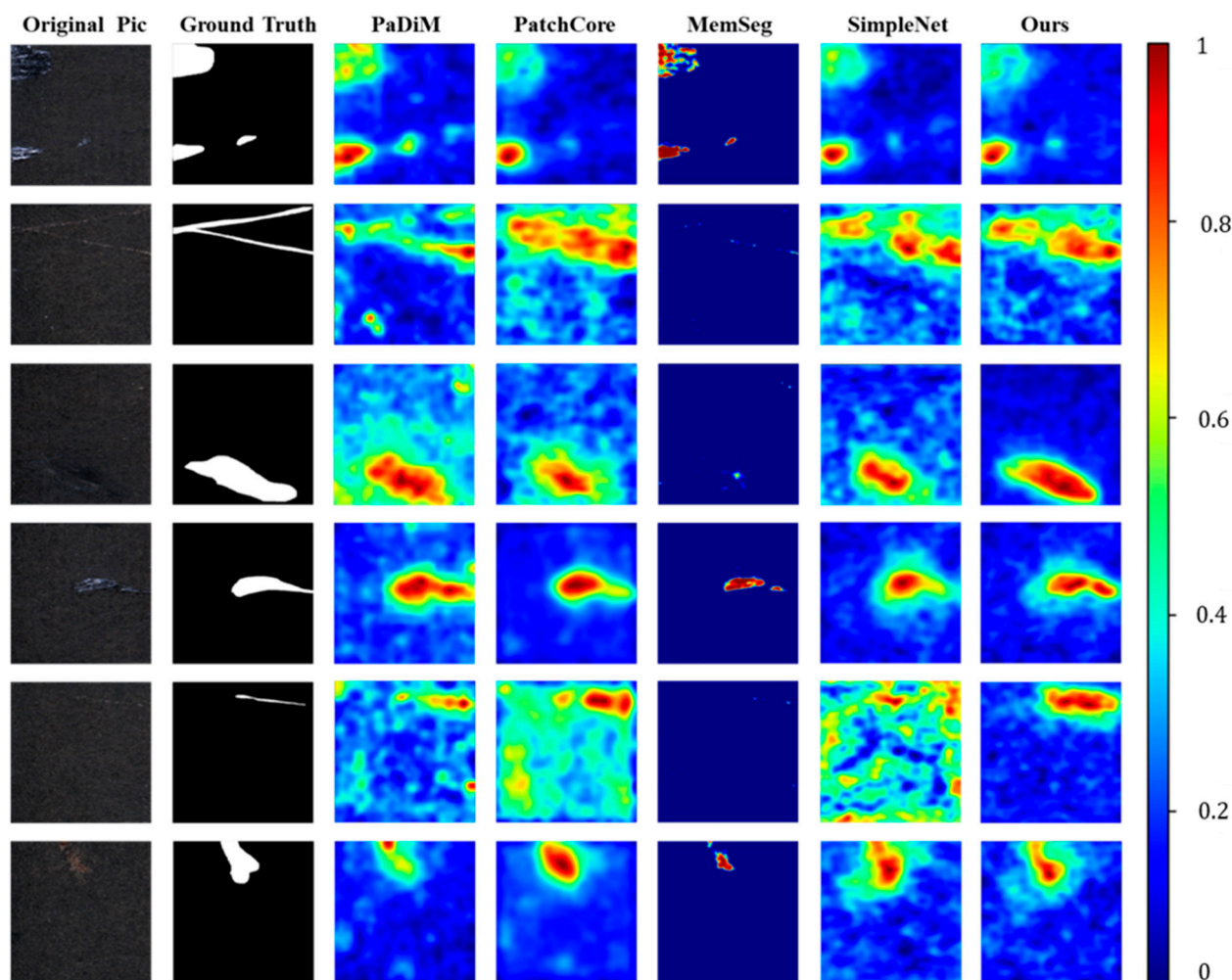


**Figure 10.** KSDD2 dataset visualization results.

#### 4.5. Ablation Experiments

##### 4.5.1. Patch Size for the DCT Transform on the Effectiveness of Detection

As shown in Table 5, we conducted experiments with different patch division sizes. The results indicate that the choice of patch size significantly affects the model's detection performance. The Patch-based DCT transform demonstrates a clear optimal solution between local and global contexts, with the best model performance achieved when  $k = 4$ .

**Table 5.** The impact of different patch division sizes  $k$  on detection performance. Anomaly detection and localization performance are measured based on Image AUROC [%], Image F1 [%], Pixel AUROC [%], and Pixel F1 [%].

|          | Image AUROC | Image F1    | Pixel AUROC | Pixel F1    |
|----------|-------------|-------------|-------------|-------------|
| $k = 2$  | 98.5        | 96.1        | 93.2        | <b>32.2</b> |
| $k = 4$  | <b>99.4</b> | <b>98.4</b> | <b>93.6</b> | 31.8        |
| $k = 8$  | 99.2        | 98.3        | 92.4        | 28.1        |
| $k = 16$ | 97.8        | 96.9        | 91.5        | 25.6        |

#### 4.5.2. Ablation Study on Modules of FFNet

To validate the effectiveness of each component in the algorithm, we conducted a series of ablation experiments on CRS. As shown in Table 6, to ensure consistent intermediate feature dimensions across all positions in the network, we replaced SAFR with concatenation and used the random dimensionality reduction (RDR) [20] instead of RFS.

**Table 6.** Ablation study of different modules on CRS. Anomaly detection and localization performance are measured based on Image AUROC (%), Image F1 (%), Pixel AUROC (%), and Pixel F1 (%).

| Models            |               | I-AUROC     | Image-F1    | P-AUROC     | P-F1        |
|-------------------|---------------|-------------|-------------|-------------|-------------|
| Without frequency | -             | 98.1        | 97.6        | 91.9        | 25.3        |
|                   | +RFS          | 98.6        | 98.3        | 92.5        | 27.6        |
| With frequency    | -             | 98.4        | 97.4        | 92.0        | 24.5        |
|                   | +FFE          | 98.6        | 97.9        | 93.1        | 26.4        |
|                   | +FFE+SAFR     | 99.1        | 98.3        | 93.5        | 28.9        |
|                   | +FFE+SAFR+RFS | <b>99.4</b> | <b>98.4</b> | <b>93.6</b> | <b>31.8</b> |

The following conclusions can be drawn. Without adding any modules, simply incorporating frequency domain information leads to improvements in both Image AUROC and Pixel AUROC compared to the baseline. This indicates that frequency domain features complement spatial domain features, thereby enhancing anomaly perception. After the frequency domain features are enhanced through the FFE module, they are better at representing the differences between anomalous and normal features than the fixed DCT-derived frequency domain features. Furthermore, after using SAFR to reconstruct the frequency and spatial features, the detection performance improved even further. Regardless of whether frequency domain features were added, RFS significantly contributed to improving detection performance.

Overall, these modules have a substantial impact on the model's detection performance. With the combined effect of these modules, the model's performance in both image-level and pixel-level tasks improved significantly: Image AUROC and Image F1 increased by 1.3% and 0.8%, respectively, while Pixel AUROC and Pixel F1 improved by 1.7% and 6.5%.

## 5. Conclusions

In this paper, we propose FFNet, a novel unsupervised anomaly detection method based on frequency domain feature fusion, capable of accurately identifying and segmenting surface defects in metals. Our method primarily consists of three core components: the FFE, responsible for extracting and enhancing frequency domain features; the SAFR, responsible for fusing spatial and frequency domain features; and the RFS, responsible for filtering the reconstructed features. These components work together to introduce frequency domain features and effectively integrate them with spatial domain features,

thereby enhancing the model's ability to perceive subtle anomalies while controlling redundant features.

FFNet achieved image-level AUROC and F1 of 99.4% and 93.6%, respectively, and pixel-level AUROC and F1 of 96.9% and 97.6% on the CRS dataset. Compared to other unsupervised anomaly detection algorithms, FFNet demonstrated the best detection performance. We also conducted generalization experiments on the KSDD2 dataset, achieving similarly advanced results, proving the strong generalization ability and robustness of our method. FFNet achieves 43 FPS on an RTX 4090 in terms of inference speed, which is second only to that of the fastest SimpleNet. This indicates that our method remains highly competitive and holds substantial promise for practical industrial applications.

Overall, FFNet has shown excellent detection capability in metal surface defect detection, providing important insights and experiences for future research in this field. It should be noted that our method currently exhibits certain limitations. While FFNet prioritizes distinguishing anomalous regions from normal areas, it achieves suboptimal precision in boundary delineation, particularly for defects with ambiguous edges. Additionally, the framework requires a substantial number of normal samples during training. In future work, we will address these limitations through architectural refinements to enhance boundary localization accuracy and explore few-shot learning paradigms to reduce dependency on large-scale datasets. Next, we also aim to further improve the detection of small defects and extend our algorithm to defect detection tasks for industrial products beyond metals.

**Author Contributions:** Software, W.W. and T.T.; formal analysis, J.X.; investigation, J.Y.; resources, Y.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by Pingyang, Zhejiang Province of China (No. 250071494).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data available on request from the authors.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Li, F.; Xi, Q.G. DefectNet: Toward fast and effective defect detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–9.
2. Xiao, J.; Guo, H.; Zhou, J.; Zhao, T.; Yu, Q.; Chen, Y.; Wang, Z. Tiny object detection with context enhancement and feature purification. *Expert Syst. Appl.* **2023**, *211*, 118665.
3. Youkachen, S.; Ruchanurucks, M.; Phatrapomnant, T.; Kaneko, H. Defect segmentation of hot-rolled steel strip surface by using convolutional auto-encoder and conventional image processing. In Proceedings of the 10th International Conference of Information and Communication Technology for Embedded Systems (ICICTES), Bangkok, Thailand, 25 March 2019; pp. 1–5.
4. Kang, G.; Gao, S.; Yu, L.; Zhang, D. Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning. *IEEE Trans. Instrum. Meas.* **2018**, *68*, 2679–2690. [[CrossRef](#)]
5. Akçay, S.; Atapour-Abarghouei, A.; Breckon, T.P. Skip-GANomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. *arXiv* **2019**, arXiv:1901.08954.
6. Matsubara, T.; Sato, K.; Hama, K.; Tachibana, R.; Uehara, K. Deep generative model using unregularized score for anomaly detection with heterogeneous complexity. *IEEE Trans. Cybern.* **2020**, *52*, 5161–5173. [[CrossRef](#)] [[PubMed](#)]
7. Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging, Proceedings of the International Conference on Information Processing in Medical Imaging, Boone, NC, USA, 25–30 June 2017*; Springer: Cham, Switzerland; pp. 146–157.
8. Lai, Y.T.K.; Hu, J.S. A texture generation approach for detection of novel surface defects. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Miyazaki, Japan, 7–10 October 2018; pp. 4357–4362.
9. Lai, Y.T.K.; Hu, J.S.; Tsai, Y.H.; Chiu, W.Y. Industrial anomaly detection and one-class classification using generative adversarial networks. In Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Auckland, New Zealand, 9–12 July 2018; pp. 1444–1449.

10. Mesgaran, M.; Hamza, A.B. Graph fairing convolutional networks for anomaly detection. *Pattern Recognit.* **2024**, *145*, 109960. [[CrossRef](#)]
11. Gao, L.; Zhang, J.; Yang, C.; Zhou, Y. Cas-VSwin transformer: A variant swin transformer for surface-defect detection. *Comput. Ind.* **2022**, *140*, 103689. [[CrossRef](#)]
12. Wang, J.; Xu, G.; Yan, F.; Wang, J.; Wang, Z. Defect transformer: An efficient hybrid transformer architecture for surface defect detection. *Measurement* **2023**, *211*, 112614.
13. Lu, F.; Yao, X.; Fu, C.-W.; Jia, J. Removing anomalies as noises for industrial defect localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 16166–16175.
14. Kozamernik, N.; Bračun, D. Visual Inspection System for Anomaly Detection on KTL Coatings Using Variational Autoencoders. *Procedia CIRP* **2020**, *93*, 1558–1563.
15. Balzategui, J.; Eciolaza, L.; Maestro-Watson, D. Anomaly detection and automatic labeling for solar cell quality inspection based on Generative Adversarial Network. *arXiv* **2021**, arXiv:2103.03518.
16. Hou, J.; Zhang, Y.; Zhong, Q.; Xie, D.; Pu, S.; Zhou, H. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8791–8800.
17. Mohammadi, B.; Fathy, M.; Sabokrou, M. Image/Video Deep Anomaly Detection: A Survey. *arXiv* **2021**, arXiv:2103.01739.
18. Yang, Y.; Xiang, S.; Zhang, R. Improving unsupervised anomaly localization by applying multi-scale memories to autoencoders. *arXiv* **2020**, arXiv:2012.11113.
19. Liu, Y.; Zhuang, C.; Lu, F. Unsupervised Two-Stage Anomaly Detection. *arXiv* **2021**, arXiv:2103.11671.
20. Defard, T.; Setkov, A.; Loesch, A.; Audigier, R. Padim: A patch distribution modeling framework for anomaly detection and localization. In Proceedings of the International Conference on Pattern Recognition, Montréal, QC, Canada, 10–11 January 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 475–489.
21. Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; Gehler, P. Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14318–14328.
22. Bae, J.; Lee, J.H.; Kim, S. Image anomaly detection and localization with position and neighborhood information. *arXiv* **2022**, arXiv:2211.12634.
23. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
24. Zavrtnik, V.; Kristan, M.; Draem, D.S. A discriminatively trained reconstruction embedding for surface anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8330–8339.
25. Li, C.-L.; Sohn, K.; Yoon, J.; Pfister, T. Cutpaste: Self-supervised learning for anomaly detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9664–9674.
26. Liu, Z.; Zhou, Y.; Xu, Y.; Wang, Z. SimpNet: A simple network for image anomaly detection and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 20402–20411.
27. Yang, M.; Wu, P.; Feng, H. A semi-supervised method for image surface defect detection using differences and commonalities. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105835.
28. Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; Vedaldi, A. Describing textures in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 3606–3613.
29. Zhang, X.; Xu, M.; Zhou, X. RealNet: A feature selection network with realistic synthetic anomaly for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 11–15 June 2024; pp. 16699–16708.
30. Wang, L.; Yi, S.; Yu, Y.; Gao, C.; Samali, B. Automated ultrasonic-based diagnosis of concrete compressive damage amidst temperature variations utilizing deep learning. *Mech. Syst. Signal Process.* **2024**, *221*, 111719.
31. Gueguen, L.; Sergeev, A.; Kadlec, B.; Liu, R.; Yosinski, J. Faster Neural Networks Straight from JPEG. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018; Volume 31.
32. Ehrlich, M.; Davis, L.S. Deep residual learning in the jpeg transform domain. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3484–3493.
33. Xu, K.; Qin, M.; Sun, F.; Wang, Y.; Chen, Y.K.; Ren, F. Learning in the frequency domain. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1740–1749.
34. Qin, Z.; Zhang, P.; Wu, F.; Li, X. Frequency channel attention networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 783–792.
35. Zhong, Y.; Li, B.; Tang, L.; Kuang, S.; Wu, S.; Ding, S. Detecting camouflaged object in frequency domain. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 4504–4513.

36. Vaswani, A. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
37. Xiao, J.; Wang, S.; Zhou, J.; Tian, Z.; Zhang, H.; Wang, Y.F. MIM: High-Definition Maps Incorporated Multi-View 3D Object Detection. *IEEE Trans. Intell. Transp. Syst.* **2024**, *26*, 3989–4001. [[CrossRef](#)]
38. Zagoruyko, S. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
39. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 11–15 June 2018; pp. 7132–7141.
40. Božič, J.; Tabernik, D.; Skočaj, D. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Comput. Ind.* **2021**, *129*, 103459.

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