*applied sciences*

MDPI

*Article*

# A Feature-Oriented Reconstruction Method for Surface-Defect Detection on Aluminum Profiles

**Shancheng Tang, Ying Zhang *, Zicheng Jin, Jianhui Lu, Heng Li and Jiqing Yang**

College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; tangshancheng@xust.edu.cn (S.T.); 21207035005@stu.xust.edu.cn (Z.J.); 21207223046@stu.xust.edu.cn (J.L.); 22207223073@stu.xust.edu.cn (H.L.); 22207223130@stu.xust.edu.cn (J.Y.)
* Correspondence: 19207040027@stu.xust.edu.cn

**Abstract:** The number of defect samples on the surface of aluminum profiles is small, and the distribution of abnormal visual features is dispersed, such that the existing supervised detection methods cannot effectively detect undefined defects. At the same time, the normal texture of the aluminum profile surface presents non-uniform and non-periodic features, and this irregular distribution makes it difficult for classical reconstruction networks to accurately reconstruct the normal features, resulting in low performance of related unsupervised detection methods. Aiming at such problems, a feature-oriented reconstruction method of unsupervised surface-defect detection method for aluminum profiles is proposed. The aluminum profile image preprocessing stage uses techniques such as boundary extraction, background removal, and data normalization to process the original image and extract the image of the main part of the aluminum profile, which reduces the influence of irrelevant data features on the algorithm. The essential features learning stage precedes the feature-optimization module to eliminate the texture interference of the irregular distribution of the aluminum profile surface, and image blocks of the area images are reconstructed one by one to extract the features through the mask. The defect-detection stage compares the structural similarity of the feature images before and after the reconstruction, and comprehensively determines the detection results. The experimental results improve detection precision by 1.4% and the *F*1 value by 1.2% over the existing unsupervised methods, proving the effectiveness and superiority of the proposed method.

**Keywords:** defect detection; aluminum profile; unsupervised learning; complex texture processing

check for updates

**Citation:** Tang, S.; Zhang, Y.; Jin, Z.; Lu, J.; Li, H.; Yang, J. A Feature-Oriented Reconstruction Method for Surface-Defect Detection on Aluminum Profiles. *Appl. Sci.* **2024**, *14*, 386. https://doi.org/10.3390/app14010386

Academic Editors: Chen Jiang, Zhenzhong Chen, Xiaoke Li, Xiwen Cai and Zan Yang

Received: 28 November 2023
Revised: 23 December 2023
Accepted: 26 December 2023
Published: 31 December 2023

## 1. Introduction

In recent years, with the intensification of global economic competition, many countries have been actively promoting the "Industry 4.0" strategy, which aims to accelerate the development of digitalization, intelligence, and automation in the manufacturing industry in order to improve the competitiveness of national manufacturing industries. As an industrial base material, aluminum profiles are widely used in many fields such as power equipment, mechanical equipment, aerospace, infrastructure construction, and transportation, and are favored for their design flexibility, hardness, light weight, and wear resistance. However, constrained by production equipment and environmental conditions, aluminum profiles can hardly avoid surface defects, such as coating cracking, orange peel, bumping, convex powder, bubbles, and dirty spots, during the manufacturing process. These defects directly affect the quality, appearance, and safety of aluminum profiles. Especially in high-risk industries (e.g., aerospace and transportation), surface defects of aluminum profiles may lead to equipment failures and accidents, posing a serious threat to the safety of life and property. Therefore, determining how to efficiently and accurately detect defects on the surface of aluminum profiles is of great academic and practical significance. Currently, manual inspection is still the main method for the detection of aluminum profile surface defects. Although the method has a certain accuracy, it takes a lot of time and manpower and is easily affected by the subjective judgment and fatigue of staff, such that efficiency and reliability are limited. In addition, as the production speed of aluminum profiles increases, manual inspection can no longer meet the production line's demand for inspection speed. Therefore, there is an urgent need to replace inefficient manual work by intelligent means.

With the development of computer image-processing technology, the use of machine vision detection methods [1,2] to determine the surface-defect information in images has emerged. However, surface-defect detection methods based on traditional machine learning models need to extract features manually; have relatively harsh imaging environment requirements; are sensitive to changes in light and viewing angle; find it difficult to detect undefined defects; and are not suitable for complex defect detection. For aluminum profiles with complex surface textures and large differences in defect scales, it is difficult for current models to be effective in practical engineering applications and they cannot meet industrial demands.

In recent years, the rise of deep learning techniques has had a wide impact in the field of image processing [3–6], especially for the tasks of natural image classification, target detection, and localization, which have made significant progress. Compared to traditional machine vision methods, deep learning methods achieve automatic feature extraction through neural networks, which map the original image by layer-by-layer feature transformations, transforming it from the original feature space to a new one, and this method makes target detection easier. However, the convolutional neural network (CNN) method [7,8] is inefficient in capturing global contextual information and also fails to provide a holistic perception and macroscopic understanding of the image due to the limitation of the sensory field size, which results in limited local information being extracted by the shallow network. Moreover, supervised deep learning methods rely on labeled data in the training phase, which usually needs to cover suitable diversity and complexity to accommodate various defect situations in real application scenarios. However, different defect types may vary depending on factors such as shape, size, color, and lighting conditions, etc., which makes it difficult for supervised methods to effectively generalize to unseen defect samples. In addition, in real industrial environments, factors such as the captured background, image resolution, and light reflections may negatively affect the image quality, resulting in poor detection accuracy.

In view of these problems, this paper proposes a feature-oriented reconstruction method for unsupervised surface-defect detection on aluminum profiles. In the preprocessing stage, the original image is processed through techniques such as boundary extraction, background removal, and data normalization to extract the main part of the image of the aluminum profile and reduce the influence of irrelevant data features. In the model-training stage, the image of the main part of the aluminum profile is cropped into an area image with uniform specifications as input, and the interference of complex texture is removed by the feature-optimization module on the premise of retaining its surface features, and the body features of the aluminum profile are extracted by mask reconstruction. In the detection stage, a specific mask is used to reconstruct the area-feature image, the average similarity of the reconstructed area-feature images before and after the reconstruction is determined, and a comprehensive evaluation of the detection results determines whether there are defects. The main contributions of this paper are the following two points:

(a) A new unsupervised surface-defect detection method for aluminum profiles is proposed. It solves the problems of the existing supervised learning surface-defect detection methods for aluminum profiles, which require a large number of manually labeled defect features in advance, and a small number of aluminum profile samples and incomplete defect types lead to the insufficient detection capability of undefined defect categories.

(b) Incorporating the feature-optimization module into the Masked Auto-Encoders (MAE) model eliminates the complex texture randomly distributed on the surface of aluminum profiles and retains its surface feature information, which excludes the interference of irregular texture on the generated model and improves the performance of the model.

This article is organized structurally as follows.

Section 2 describes deep learning detection methods as well as transformer models. Section 3 presents a feature-oriented reconstruction method for unsupervised detection of surface defects on aluminum profiles. Section 4 describes the datasets, training details, and evaluation indicators used and discusses the results of each experiment. Section 5 summarizes the experimental results and looks at future research directions.

## 2. Related Works

### *2.1. Deep Learning Detection Methods*

In surface-defect detection, the earliest manual visual method consumes a lot of manpower and material resources, and it has the problems of low efficiency and poor accuracy. In recent years, machine vision has attracted much attention in surface-defect detection.

Some scholars have proposed different methods for object-surface-defect detection using supervised deep learning methods. For example, Dahai Liao et al. [9] proposed a non-destructive identification and classification method for surface defects based on the improved YOLOv5 algorithm, and they applied a new mobile network attention mechanism, coordinate attention, to the backbone of the YOLOv5 algorithm, giving better detection results. Kechen Song et al. [10] proposed a cross-layer semantic guidance network (CS-GNet) based on the YOLOv6 algorithm, which introduces a cross-layer semantic guidance module (CSGM) that uses deeper semantic information to guide the shallower feature layer and improves performance for detecting tiny defects. Shenqi Guan et al. [11] proposed a new method to detect fabric surface defects by target-driven features, Compared with the classical defect algorithm, the algorithm is able to realize accurate segmentation of surface defects, has better noise resistance, higher detection accuracy, and has strong applicability to fabric defect detection. Chenglong Wang et al. [12] proposed a defect-detection model, MeDERT, for aluminum profiles based on the improvement of the classical detector YOLOv4, which is suitable for dealing with the image features of aluminum profiles, and the experimental results show that MeDERT is superior to the models such as YOLOv5 and DERT, which effectively improves the defect detection performance. The above methods can more efficiently use the model to automatically extract good features and defective features, and the detection results are more accurate, but the model-training process requires the labeling of a large amount of data and defective samples in advance, and it is very difficult to collect complete defective samples.

Several scholars have proposed different methods for object-surface-defect detection using unsupervised methods. For example, Sizhe Xiao et al. [13] proposed a gradient-based unsupervised model, Grad MobileNet, based on MobileNetV3, in which the model the model can be trained using only a few normal images, extracting the feature gradient of the input image, classifying welding defects through the gradient distribution, and achieving 99% accuracy on the welding defect dataset RIAM, which was constructed by the authors. Qunying Zhou et al. [14] proposed a knowledge-distillation model based on attention mechanism and feature fusion, which enhances the ability of the model to extract features through attention, improves the pixel-level localization of the model, and provides better detection results in the MVTecAD dataset. Jin Rui et al. [15] proposed a fabric-defect-detection method based on an improved generative adversarial network, introducing a center loss constraint to improve the recognition performance of the method, which was evaluated on the publicly available Tianchi dataset with good results. Yijing Guo et al. [16] proposed a new unsupervised small-sample-defect-detection model based on the DAGM2007 dataset that performs well with a small number of training samples. Although the above methods are more effective on specific detection targets, their reconstruction networks are difficult to accurately reconstruct irregularly distributed images, and their detection results are susceptible to various factors such as the color, size, and illumination of the detection target.

Differing from the above method, this paper integrates the feature-optimization module into the MAE model, which on the one hand reduces the redundant data, reduces the model reconstruction time, and makes the model reconstruction faster and more accurate, and, on the other hand, eliminates the negative impact of the complex texture of the aluminum profile's surface that is irregularly distributed for the reconstruction process. Through the above method, we can improve the Structural Similarity Index Measure (SSIM) of the aluminum profile image before and after reconstruction, and then effectively determine whether the input image has defects.

### 2.2. Transformer

Transformer [17] is an attention-based structure originally proposed as a sequence-to-sequence model for machine translation tasks. In recent years, by virtue of its outstanding results in the field of Natural Language Processing (NLP) [18–21], it has attracted a wide range of attention from researchers in the field of computer vision [22], and more and more researchers are migrating its application to computer vision tasks such as target detection, video processing, image processing.

Image restoration algorithms based on the transformer structure perform well in terms of image global structure understanding, generalization ability of generalized datasets, etc., with results comparable to or even surpassing contemporaneous convolutional neural network-based algorithmic models. For example, Nicolas Carion et al. [23] innovatively applied the transformer to the field of target detection, and they proposed a new framework, DEtection TRansformer (DETR), based on the transformer and the dichotomous matching loss of direct set prediction. Haitao Yu et al. [24] proposed a dynamic transformer network for surface-defect detection, which utilizes the transformer's ability to extract global contextual features and achieves accurate and fast defect detection on steel surface images by fusing it with a dynamic network. Junpu Wang et al. [25] proposed an efficient hybrid transformer architecture for surface-defect detection with better detection results on the SD-saliency-900 dataset, the Fabric defect dataset, and NRSD-MN dataset. Hongbing Shang et al. [26] proposed a method for intelligent visual surface-defect detection using the Defect-aware transformer network (DATN) for industrial inspection, which works better on the publicly available dataset MVTec. Alexey Dosovitskiy et al. [27] proposed the Vision Transformer (ViT), using transformer instead of standard convolution, and applied it to image classification tasks, achieving state-of-the-art classification results at that time.

Unlike the local perception property of convolutional neural networks, the learning process of transformer is based on the interaction of global information, the multi-head attention mechanism can better focus on global information compared to convolution, which is more advantageous for detecting occluded targets and can keep the number of parameters relatively low, in addition to the fact that these methods change the traditional way of thinking for indirectly solving the problem using classification and regression, which is usually pre-trained using large-scale unlabeled datasets, and then the learned features are fine-tuned on the downstream task in an unsupervised manner, which improves performance and reduces the cost of manual labeling.

## 3. Proposed Methods

The framework is divided into three parts, which are aluminum profile image adaptive preprocessing, essential feature learning, and surface-defect detection, as shown in Figure 1.

(a) Adaptive preprocessing of aluminum profile images: Images of aluminum profiles with different background colors, lighting, and placement angles are extracted by adaptive boundary extraction, removal of background colors and data normalization to obtain images with no background, only the main part of aluminum profiles, and uniform specifications.

(b) Essential Feature Learning: Firstly, the boundary extraction, background removal, and data normalization operations are performed on the non-defective images aluminum profile dataset. Then these are cropped to 224 × 224 specifications and input into the model one by one. The model performs feature extraction on them, removes the masked image blocks based on the randomly generated mask image (mask rate of 75%), and inputs 25% of the image blocks that are not removed into the encoder and decoder for prediction of the removed image blocks. Finally, the loss constraint is utilized to make the reconstructed image as consistent as possible with the input image.

(c) Surface-defect detection: Firstly, the image to be detected is adaptively preprocessed to get the aluminum profile image with uniform specifications. Then it is cropped to obtain all of the area images, and the area images are input into the model one by one. The feature-reconstructed image is then compared with the area-feature image by mean structural similarity index measure (MSSIM) comparison to determine whether the input image is a defective image. Finally, the detection results of all area images of the image to be detected are used to determine the final detection results.

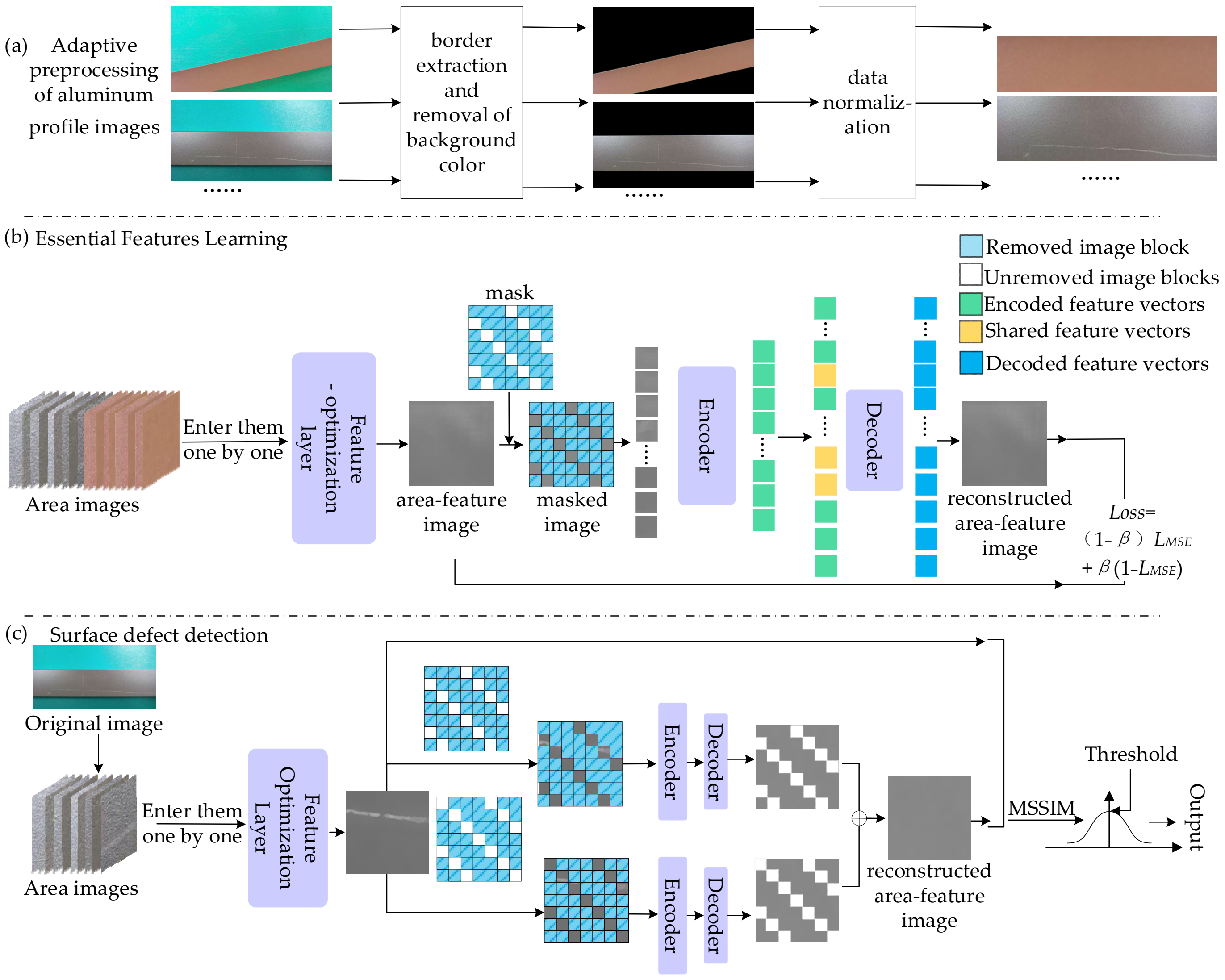

(a) Adaptive preprocessing of aluminum profile images
border extraction and removal of background color
data normaliz-ation
......
(b) Essential Features Learning
Removed image block
Unremoved image blocks
Encoded feature vectors
Shared feature vectors
Decoded feature vectors
mask
Area images
Enter them one by one
Feature - optimization layer
area-feature image
masked image
Encoder
Decoder
reconstructed area-feature image
Loss= （1- β ） LMSE + β (1-LMSE)
(c) Surface defect detection
Original image
Area images
Enter them one by one
Feature Optimization Layer
Encoder
Decoder
Encoder
Decoder
reconstructed area-feature image
MSSIM
Threshold
Output

**Figure 1.** Architecture of surface-defect detection methods for aluminum profiles.

### 3.1. *Aluminum Profile Image Preprocessing*

Aluminum profile image preprocessing is mainly divided into two parts: image boundary extraction [28] and data normalization. Image boundary extraction is performed on the original image of the aluminum profile to obtain the upper- and lower-boundary lines of the main part of the aluminum profile from the aluminum profile image. Data normalization is performed on the image with the background removed, which is rotated and cropped to reduce redundant information. The aluminum profile image preprocessing algorithm is shown in Figure 2.

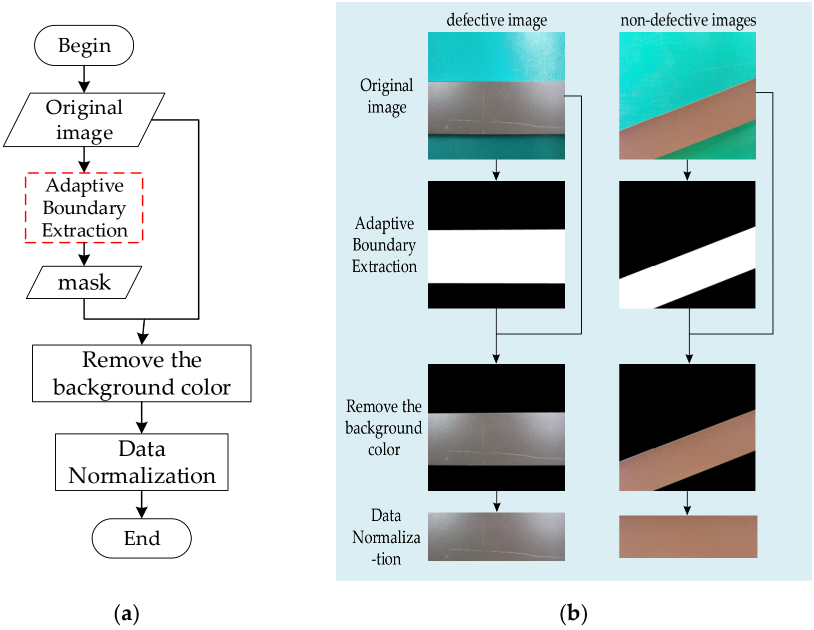


**Figure 2.** Aluminum profile image preprocessing algorithm. (**a**) Aluminum profile image preprocessing algorithm flowchart; (**b**) schematic diagram of the aluminum profile image preprocessing algorithm.

### 3.1.1. Adaptive Boundary Extraction

Image boundary extraction extracts the upper- and lower-boundary lines of the main part of the aluminum profile on the aluminum profile image by color channel selection [29], binarization processing, and extraction of the maximum connectivity domain [30]. The boundary extraction algorithm is shown in Figure 3.

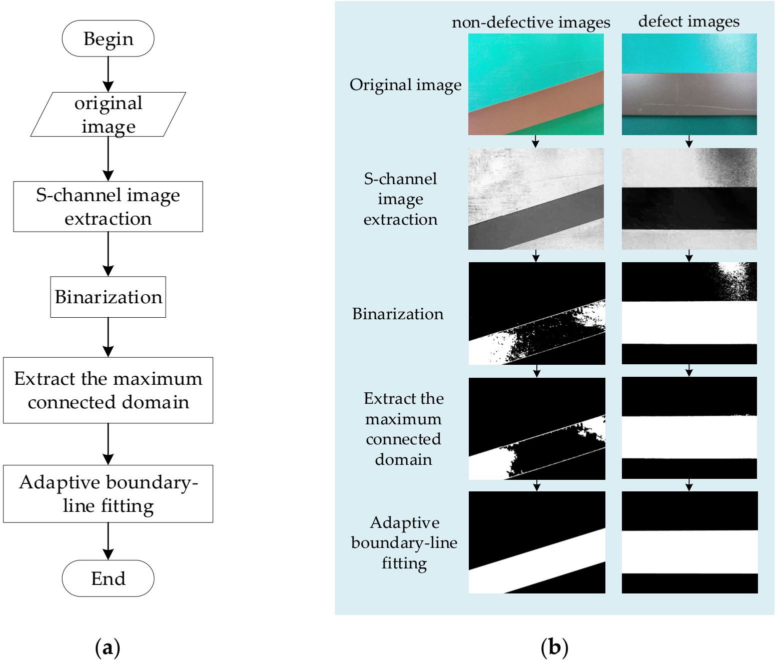

Begin
original image
S-channel image extraction
Binarization
Extract the maximum connected domain
Adaptive boundary-line fitting
End
non-defective images
defect images
Original image
S-channel image extraction
Binarization
Extract the maximum connected domain
Adaptive boundary-line fitting
(a)
(b)

**Figure 3.** Boundary extraction algorithm. (**a**) Boundary extraction algorithm flowchart; (**b**) schematic diagram of the boundary extraction algorithm.

The algorithm is specifically described below:

(a) S-channel image extraction.

The original image of the aluminum profile was segmented on the RGB [31,32] and HSV [33,34] color spaces to obtain the S-channel image with the most effective color information of the overall data, as shown in Figure 4.

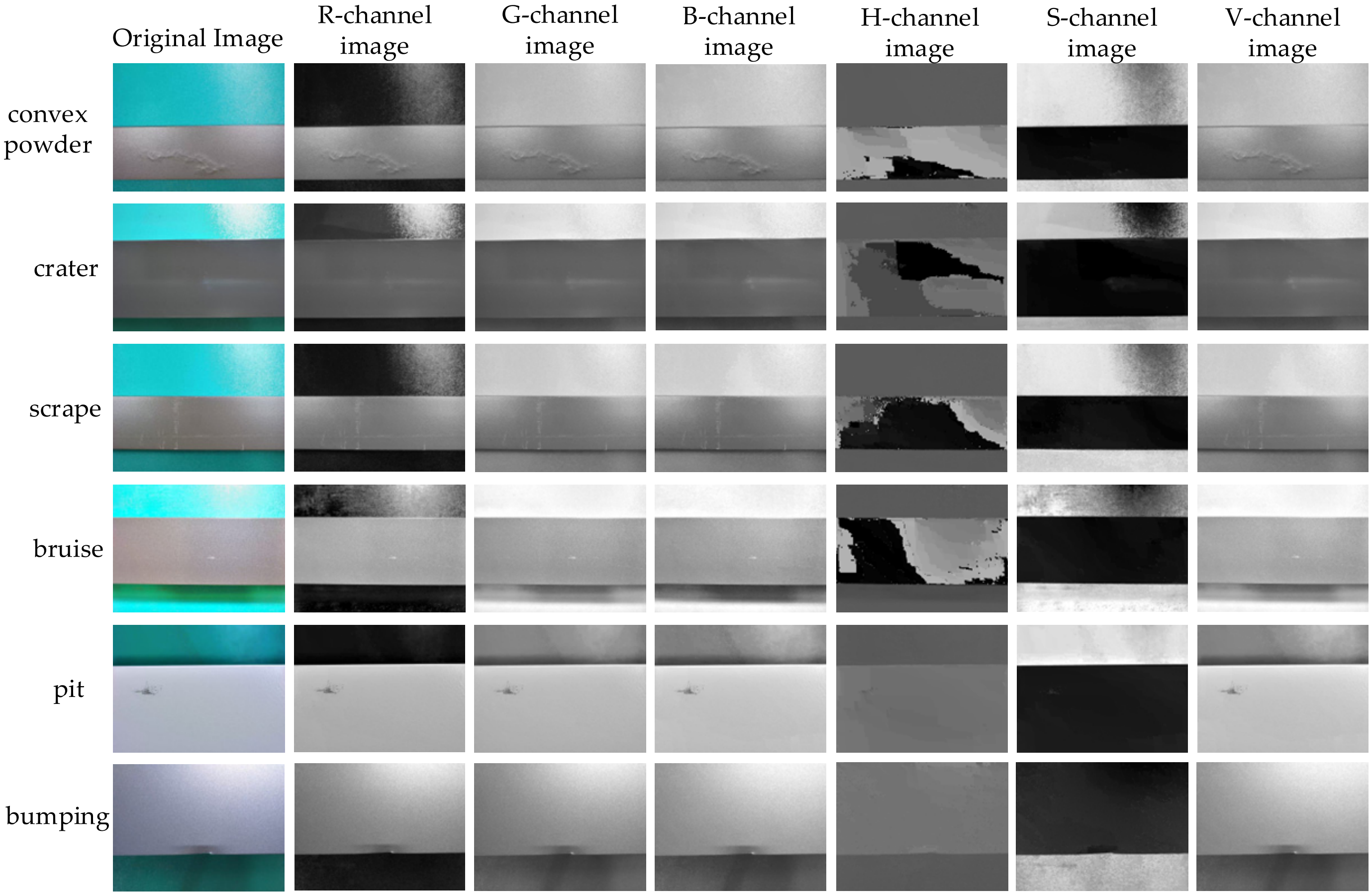


**Figure 4.** Examples of images on different color spaces.

(b) Binarization processing.

The S-channel image of the aluminum profile image is processed using binarization to reduce a large amount of redundant information in the image, thus highlighting the contour of the main part of the aluminum profile.

(c) Extract the maximum connected domain.

Using the following (a) formula on the binarized image to find out the maximum contour of the parts with pixel value of 255, the regions with consecutive pixel value of 255 are concatenated over a large area, and the scattered white areas are also filtered out to reduce their adulterated noise.

$$\oint_L P(x,y)dx + Q(x,y)dy = \iint_D [\frac{\partial Q(x,y)}{\partial x} - \frac{\partial P(x,y)}{\partial y}]dxdy \tag{1}$$

In Equation (1), $L$ is a segmented smooth closed curve and takes the positive direction, $D$ is a bounded closed region in the plane enclosed by $L$, $P(x,y)$ and $Q(x,y)$ has a first-order continuous partial derivative on $D$.

(d) Adaptive boundary-line fitting.

On the extracted image of the maximum connectivity domain, the parts with a pixel value of 255 are searched from the upper and lower ends sequentially towards the middle, and the upper and lower two boundary lines of the main part of the aluminum profile on the image are fitted sequentially; after that the inner part of the outline is filled with white pixels sufficiently to extract its mask.

### 3.1.2. Data Normalization

The data normalization process mainly involves adaptive rotation of the input image without background color and the mask at the same angle, then finding the maximum internal rectangle on the rotated mask image, which is later cropped with the rotated background-removed image. The relevant algorithm is shown in Figure 5.

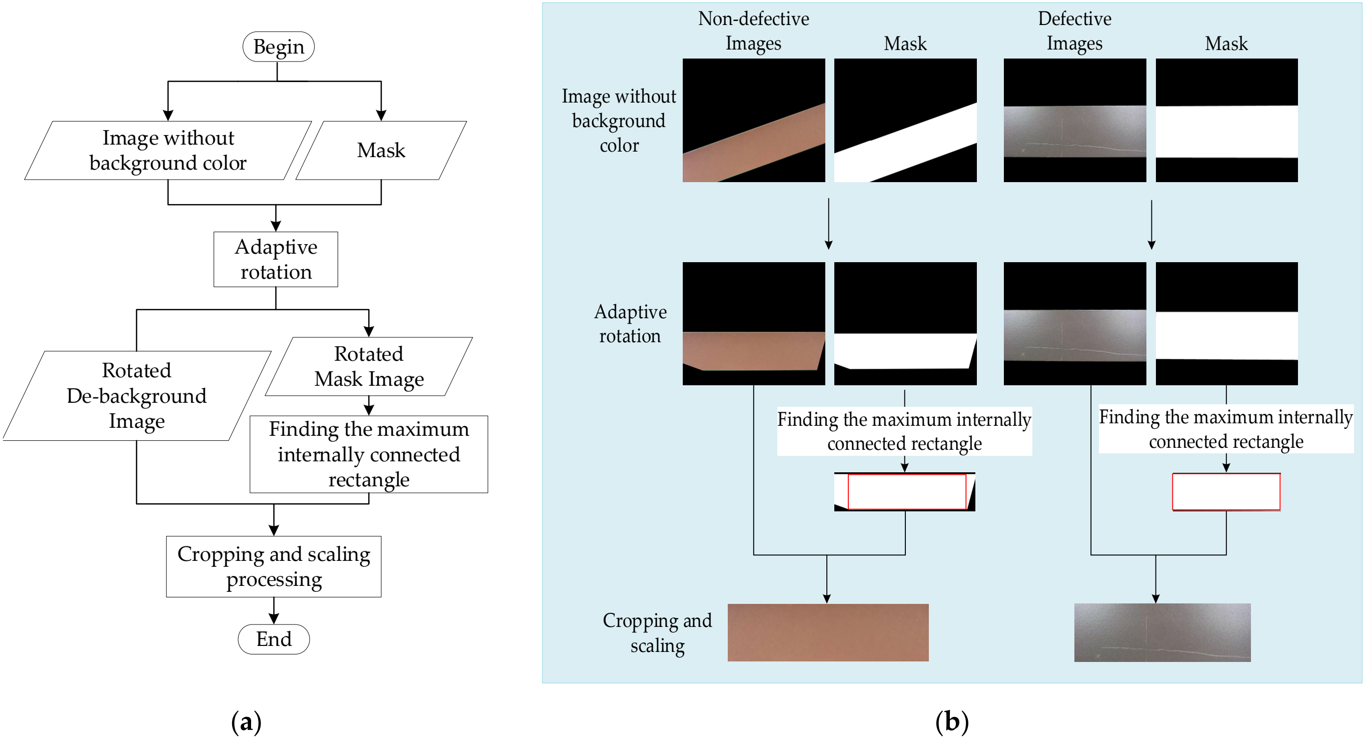

Begin
Image without background color
Mask
Adaptive rotation
Rotated De-background Image
Rotated Mask Image
Finding the maximum internally connected rectangle
Cropping and scaling processing
End
Non-defective Images
Mask
Defective Images
Mask
Image without background color
Adaptive rotation
Finding the maximum internally connected rectangle
Finding the maximum internally connected rectangle
Cropping and scaling
(a)
(b)

**Figure 5.** Data normalization algorithm. (**a**) Flowchart of the data normalization algorithm; (**b**) schematic of the data normalization algorithm.

The algorithm is specifically described below:

(a) Adaptive Rotation.

Using the different slopes of the upper- and lower-boundary lines of the main part of the aluminum profile fitted in the mask, the main part of the aluminum profile is rotated adaptively, and the angle of placement of the main part of the aluminum profile is adjusted so that it can be placed parallel to the upper and lower boundaries of the image and become more normalized.

$$\tan\alpha = \frac{\Delta y}{\Delta x} \tag{2}$$

(b) Finding the maximum internally connected rectangle.

The mask image after adaptive rotation is utilized to explore its inner rectangular boundaries.

(c) Cropping and scaling processing.

The rotated background-removed image is cropped for background and redundant information using (b) to normalize the dataset.

### *3.2. Image Essential Feature Learning Model*

#### 3.2.1. Transformer Model

The transformer encoder [35] consists of a stack of N transformer layers; each transformer layer consists of multi-head attention (MHA) layer [36] and feed-forward neural network layer. The data output from each layer is then fused with the input data using residual connections, and the normalization is performed before input to the next layer. The output dimension of each layer is d-dimensional, as shown in Figure 6.

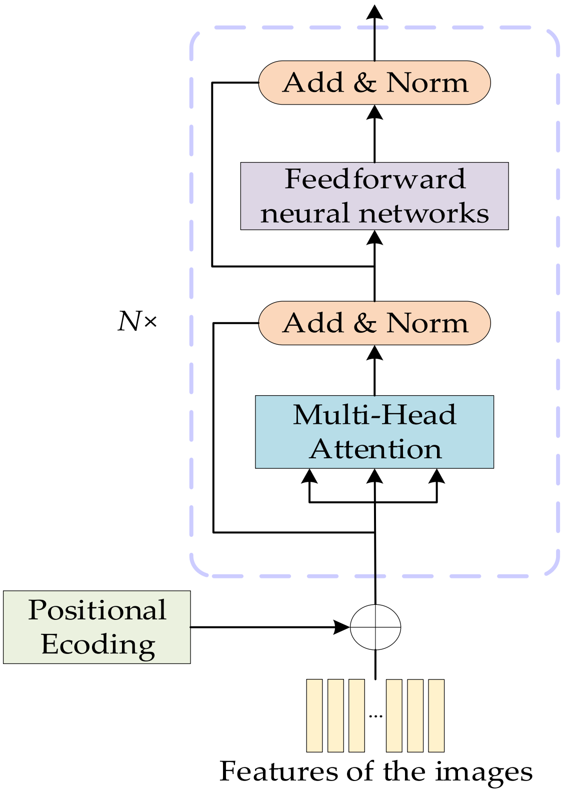


**Figure 6.** Transformer model structure.

The working principle of the multi-head attention mechanism is shown in Figure 7 for a 2-head attention model.

For a given input character $\{x^1, x^2, x^3\}$, vectorization (word embedding) yields $a^1, a^2$, $a^3 \in \mathbb{R}^{d_l \times 1}$, then for vectors $a^i, i \in \{1,2,3\}$ yields query vector $q^i \in \mathbb{R}^{d_k \times 1}$, key vector $k^i \in \mathbb{R}^{d_k \times 1}$, and value vector $v^i \in \mathbb{R}^{d_l \times 1}$ by the first linear transformation through $W^q \in \mathbb{R}^{d_k \times d_l}$, $W^k \in \mathbb{R}^{d_k \times d_l}$, and $W^v \in \mathbb{R}^{d_l \times d_l}$ matrices. Then, the obtained query vector $q^i$ is second linearly transformed through matrices $W^{q1} \in \mathbb{R}^{d_m \times d_k}$, $W^{q2} \in \mathbb{R}^{d_m \times d_k}$ to get $q^{i1} \in \mathbb{R}^{d_m \times 1}$, $q^{i2} \in \mathbb{R}^{d_m \times 1}$. Similarly, the obtained key vector $k^i$ is linearly transformed twice through the matrices $W^{k1} \in \mathbb{R}^{d_m \times d_k}$, $W^{k2} \in \mathbb{R}^{d_m \times d_k}$ to get $k^{i1} \in \mathbb{R}^{d_m \times 1}$, $k^{i2} \in \mathbb{R}^{d_m \times 1}$, the value vector $v^i$ is linearly transformed twice through the matrices $W^{v1} \in \mathbb{R}^{d_l/2 \times d_k}$, $W^{v2} \in \mathbb{R}^{d_l/2 \times d_k}$ to get $v^{i1} \in \mathbb{R}^{d_l/2 \times 1}$, $v^{i2} \in \mathbb{R}^{d_l/2 \times 1}$, and the specific computational process can be expressed as:

$$\begin{aligned} q^{ih} &= W^{qh} \cdot W^q \cdot a^i \\ k^{ih} &= W^{kh} \cdot W^k \cdot a^i \; i = \{1,2,3\},\ h = \{1,2\} \\ v^{ih} &= W^{vh} \cdot W^v \cdot a^i \end{aligned} \tag{3}$$

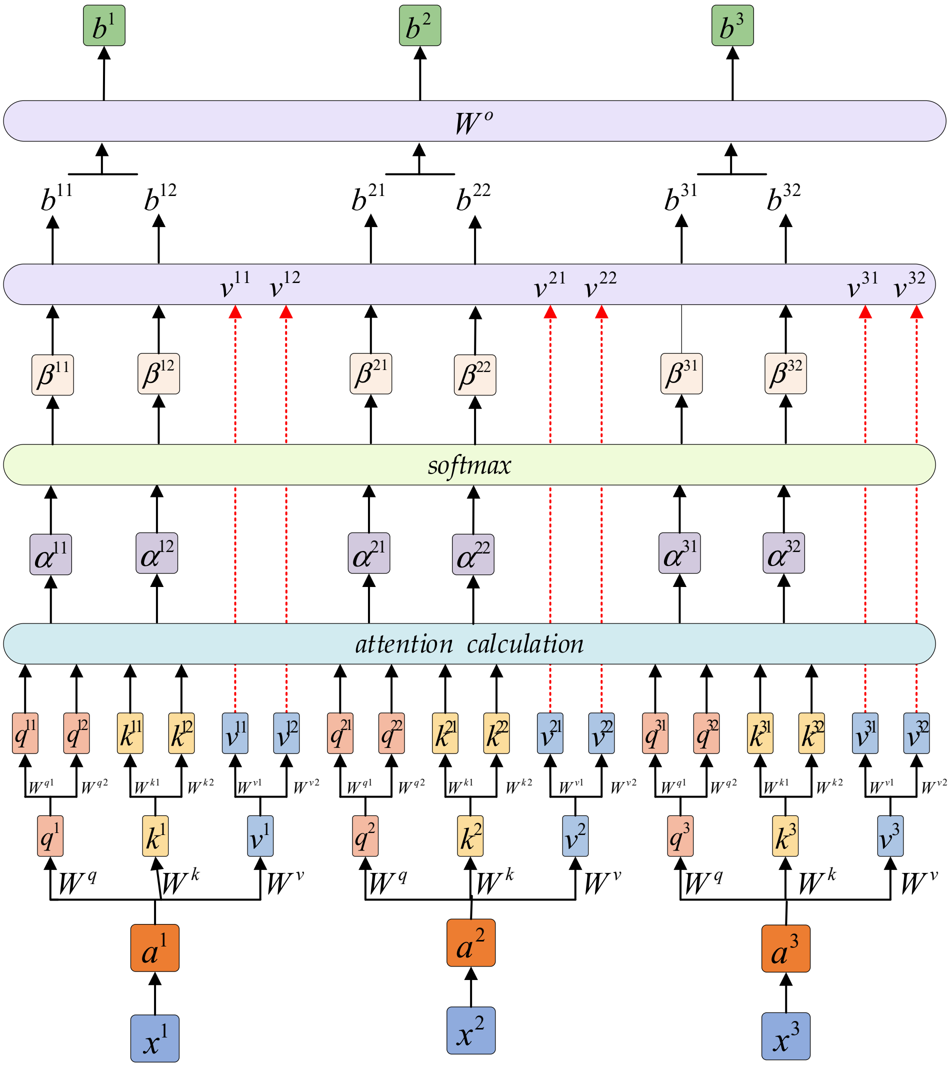


**Figure 7.** Two-head attention model.

Let the matrix

$$\begin{aligned}
Q^1 &= \left(q^{11}, q^{21}, q^{31}\right) \in \mathbb{R}^{d_m \times 3} \\
K^1 &= \left(k^{11}, k^{21}, k^{31}\right) \in \mathbb{R}^{d_m \times 3} \\
V^1 &= \left(v^{11}, v^{21}, v^{31}\right) \in \mathbb{R}^{d_l/2 \times 3} \\
Q^2 &= \left(q^{12}, q^{22}, q^{32}\right) \in \mathbb{R}^{d_m \times 3} \\
K^2 &= \left(k^{12}, k^{22}, k^{32}\right) \in \mathbb{R}^{d_m \times 3} \\
V^2 &= \left(v^{12}, v^{22}, v^{32}\right) \in \mathbb{R}^{d_l/2 \times 3}
\end{aligned} \tag{4}$$

Then the following equation is available at this point:

$$\begin{aligned}
Q^1 &= W^{q1} \cdot W^q \cdot A \\
K^1 &= W^{k1} \cdot W^k \cdot A \\
V^1 &= W^{v1} \cdot W^v \cdot A \\
Q^2 &= W^{q2} \cdot W^q \cdot A \\
K^2 &= W^{k2} \cdot W^k \cdot A \\
V^2 &= W^{v2} \cdot W^v \cdot A
\end{aligned} \tag{5}$$

The corresponding attention scores are computed from the obtained query vector and key vector, where the $\alpha^{ih}$-th component $l$ of the attention vector can be expressed as:

$$\alpha_l^i = \left(q^{ih}\right)^{\mathrm{T}} \cdot k^{lh}, \; i,l \in \{1,2,3\}, \; h \in \{1,2\} \tag{6}$$

The attention vector is normalized by the Softmax layer to obtain the attention distribution, which can be expressed as equation:

$$\beta_j^{ih} = \frac{e^{\alpha_j^{ih}}}{\sum\limits_{n=1}^{3} e^{\alpha_n^{ih}}}, \; i,j \in \{1,2,3\}, \; h \in \{1,2\} \tag{7}$$

The final output $b^{ih} \in \mathbb{R}^{d_l/2\times 1}$ is obtained by dot-multiplying the attention distribution vector $\beta^{ih}$ obtained from each head with the value matrix $V^h$, which can be expressed as equation:

$$b^{ih} = \sum_{n=1}^{3} \beta_l^{ih} \cdot v^{lh}, \; i \in \{1,2,3\}, \; h \in \{1,2\} \tag{8}$$

The $b^{ih}$ obtained from the two heads are spliced together to obtain $B$, which can be expressed as equation:

$$B = \begin{pmatrix} b^{11}, b^{21}, b^{31} \\ b^{12}, b^{22}, b^{32} \end{pmatrix} \in \mathbb{R}^{d_l \times 3} \tag{9}$$

Given the parameter matrix $W^o \in \mathbb{R}^{d_l \times d_l}$, the final output can be expressed as:

$$O = W^o \cdot B \in \mathbb{R}^{d_l \times 3} \tag{10}$$

In summary, then there are the following:

$$\begin{aligned} O &= \mathrm{MultiHead}(Q,K,V) \\ &= W^o \cdot \mathrm{Concat}\left( \begin{array}{c} V^1 \cdot softmax\left(\frac{\left(K^1\right)^T \cdot Q^1}{\sqrt{d_m}}\right) \\ V^2 \cdot softmax\left(\frac{\left(K^2\right)^T \cdot Q^2}{\sqrt{d_m}}\right) \end{array} \right) \end{aligned} \tag{11}$$

### 3.2.2. Feature-Optimization Module

The image of the aluminum profile area is manipulated one by one using feature optimization [37] so that it attenuates the presence of complex textures on the non-defective parts of the aluminum profile surface and maintains the defective-feature information well.The feature-optimization process is shown in Figure 8.

The specific algorithm is as follows:

(a) Convert the image format, then convert the area image to a grayscale image one by one, using conversion equations as follows:

$$I = \frac{299}{1000}R + \frac{587}{1000}G + \frac{144}{1000}B \tag{12}$$

where each pixel is represented by 8 bits, $I$ represents the grayscale to be converted, and $R$, $G$, and $B$ represent red (R), green (G), and blue (B) in the RGB color space, respectively.

(b) The feature discrimination of the area grayscale image, eliminating the complex texture features randomly distributed on the surface of the main image of the aluminum profile and highlighting its essential features, is processed as follows:

$$\min_{u \in BV(\Omega)} \left\{ E(u) = \int \Omega |\nabla_u| dxdy + \frac{\lambda}{2} \int \Omega |u(x,y) - u_0(x,y)|^2 dxdy \right\} \tag{13}$$

$$u_0(x,y) = u(x,y) + \eta(x,y) \tag{14}$$

where $\Omega \subset R^2$ is a bounded region, $BV(\Omega)$ is the space consisting of all bounded variance functions in $\Omega$, $\lambda$ is a scale parameter that depends on the noise level, $u_0(x,y)$ is the input grayscale image to be manipulated, $u(x,y)$ is the feature-resolved image, $\eta(x,y)$ is the noise function of the image, which is an additive Gauss noise with a mean of 0 and a variance of $\sigma$, $\int \Omega|\nabla_u|dxdy$ is the smoothing term, and $\int \Omega|u(x,y) - u_0(x,y)|^2 dxdy$ is the approximation term.

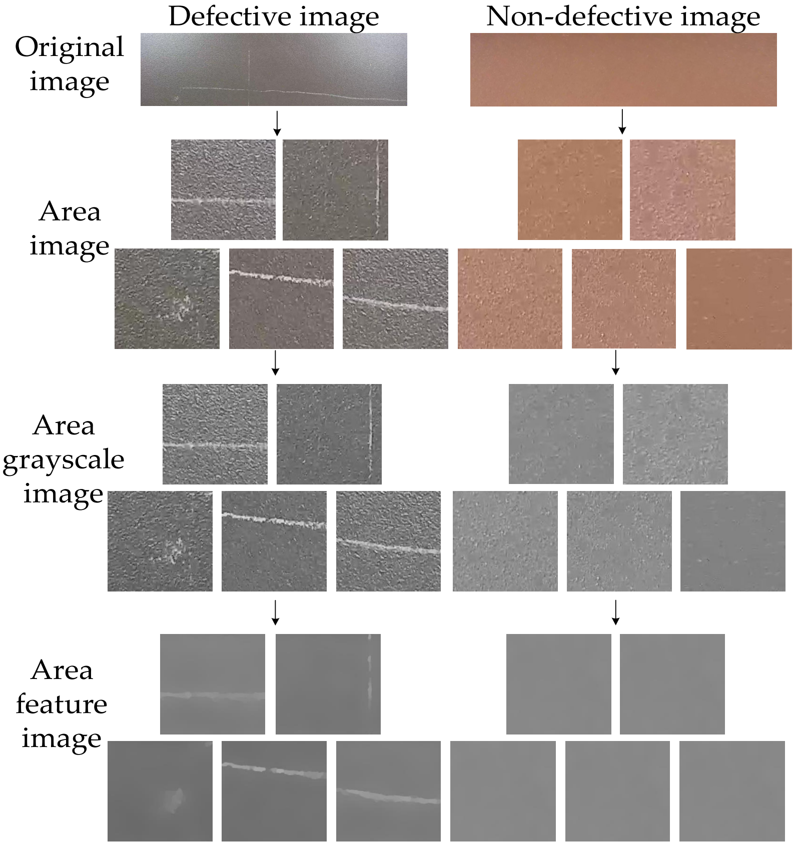


**Figure 8.** Feature-optimization module flowchart.

The Euler equations corresponding to the feature-discrimination model can be obtained using the gradient-descent method as follows:

$$-\nabla \cdot \left(\frac{1}{|\nabla u|}\nabla u\right) + \lambda(u - u_0) = 0 \tag{15}$$

The above equation is transformed and integrated over the entire image area $\Omega$ to obtain the following equation:

$$\lambda = \frac{\int \Omega\nabla \cdot (|\nabla u|/\nabla u) \cdot (u - u_0)dxdy}{\int \Omega(u - u_0)^2 dxdy} = \frac{1}{\sigma^2}\int \Omega\nabla \cdot (|\nabla u|/\nabla u) \cdot (u - u_0)dxdy \tag{16}$$

### 3.2.3. Essential Features Extraction Network

The structure of the essential features extraction model network, which is asymmetric, is shown in Figure 9. The encoder part firstly divides the input 224 × 224 3D-area images into 16 × 16 blocks after the feature-optimization layer and transforms it into a one-dimensional sequence. Next, it removes 75% of the random blocks by masking and introduces the Position and Class token encoding, and, finally, it outputs the encoder after the Transformer Block, which is made up of 24 Transformers stacked on top of each other, is used for the feature learning, and is normalized by the Layer Norm. The decoder part takes the output of the encoder as input, introduces mask tokens and decoder-position-embedding encoding, which is repaired by a Transformer Block of 8 Transformer stacks, and finally outputs the same normalized by Layer Norm.

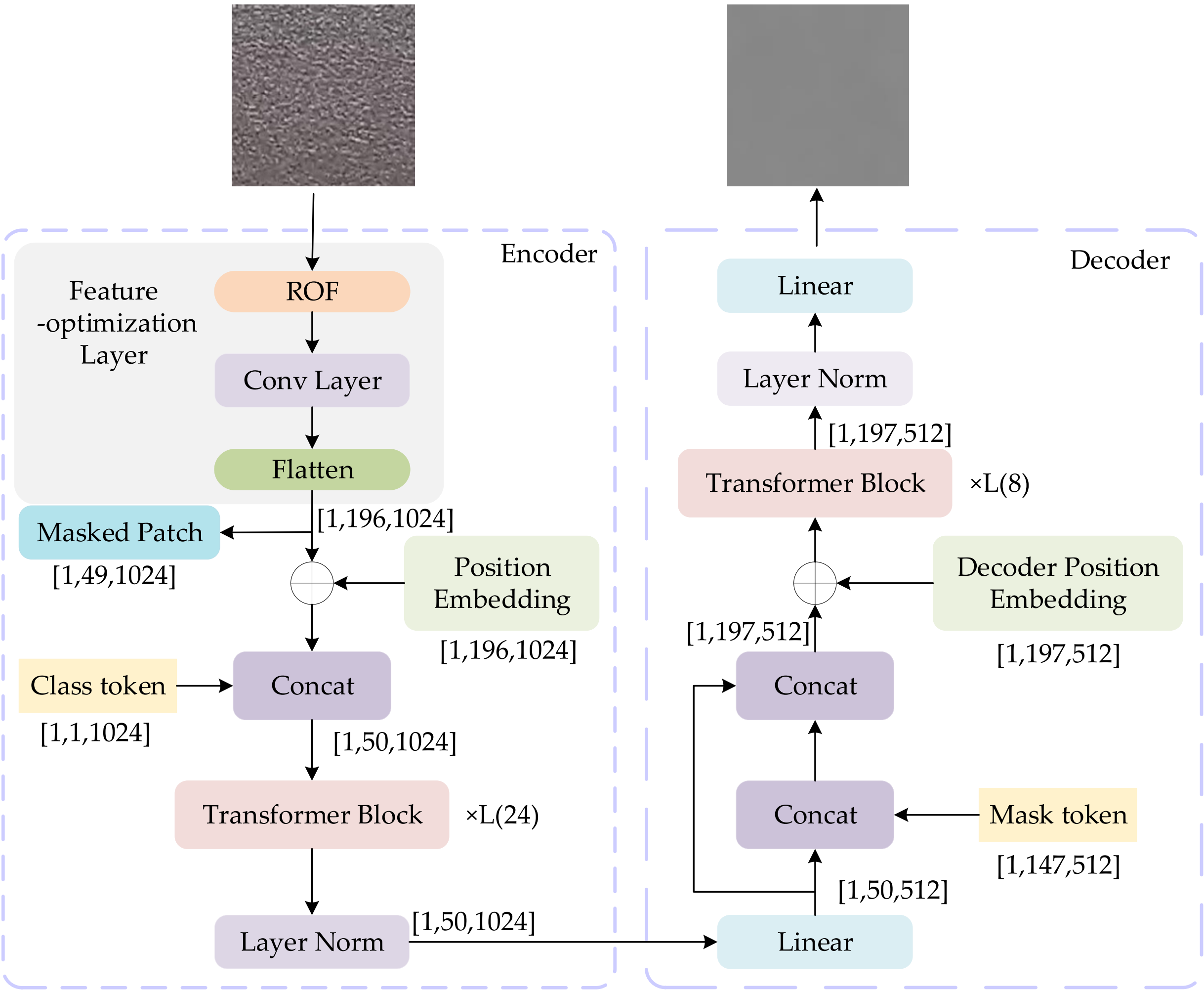


**Figure 9.** Essential features extraction network.

(a) Feature-Optimization Layer: the image input to the encoder first passes through the feature-optimization layer, which consists of three parts: the feature-optimization module, the convolutional layer, and the flatten function. Firstly, the essential features of the area image are extracted by the full variational image restoration algorithm, and then the essential features image has its features extracted by the convolutional layer, which divides the area image X into N blocks as in Equation (17). Next, the N blocks of the image are transformed into a one-dimensional sequence using the Flatten function, and the linear transformation of the sequence $\left\{x_p^i\right\}_{i=1}^N$ is performed as in Equation (18):

$$N = \frac{H \times W}{p^2} \tag{17}$$

$$z_0 = \left[x_p^1 E; x_p^2 E; \ldots; x_p^n E\right] \tag{18}$$

where $X \in H \times W \times C$ and $H$ and $W$ represent the width and height of $X$, respectively; $C$ represents the number of channels; $P^2$ represents the pixel size of the sequence; and $P^2 \cdot C$ represents the dimension of each sequence.

(b) Positional Embedding: The positional encoding $E_{pos}$ is introduced in order to prevent the loss of sequentiality of the positional information of the accession sequence, as shown in Equation (19).

$$z_1 = \left[x_p^1 E; x_p^2 E; \ldots ; x_p^n E\right] + E_{pos} \tag{19}$$

(c) Class token: The Concat function is utilized to add a learnable category encoding $x_{class}$ that is used to represent the global features of the image after encoding:

$$z = \left[x_{class}; x_p^1 E; x_p^2 E; \ldots ; x_p^n E\right] + E_{pos} \tag{20}$$

(d) Transformer Block: The data $Z$ is encoded through this process as in Equations (21) and (22).

$$z'_n = MHA(LN(Z_{n-1})) + Z_{n-1} \; (n = 0, 1, 2 \cdots , N) \tag{21}$$

$$z_n = MLP(LN(Z'_n)) + Z'_n \; (n = 0, 1, 2 \cdots , N) \tag{22}$$

where $MHA$ stands for Multiple Headed Attention, $MLP$ stands for Multi-Layer Perceptual Machine, and $n$ represents after n layers of transformer.

(e) Layer Norm: The data output from (4) is normalized by this process as in Equation (23):

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} * \gamma + \beta \tag{23}$$

where $E(x)$ denotes the mean, $Var(x)$ denotes the variance, $\varepsilon$ the auxiliary variable, and the initial values of $\gamma$ and $\beta$ are 1 and 0, respectively.

(f) Decoder: The output vector of the encoder is used as the input for the decoder, and after the data enters the decoder, it first passes through the Linear layer for dimensional conversion, and in order to ensure that it can distinguish between the different positions of mask tokens in the image, it will be added to the data as a whole with decoder-positional embedding.

(g) Loss: The mean square error ($MSE$) and $SSIM$ [38] are used as loss functions to calculate the loss of the original and restored images of the feature-optimization map as shown in (24), where the $MSE$ expression is shown in (25):

$$L_{PMRM} = (1 - \alpha) MSE_{Y_i, \hat{Y}_i} + \alpha(1 - SSIM_{Y_i, \hat{Y}_i}) \tag{24}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{25}$$

where $Y_i$ denotes feature-optimized image original data, $\hat{Y}_i$ denotes feature-optimized image reconstruction data, and $\alpha$ denotes a weight factor that balances the relative importance between pixel and $SSIM$, and in this paper, we set $\alpha = 0.5$.

## *3.3. Aluminum Profiles Surface-Defect Detection*

### 3.3.1. Defect-Detection Process

The flow of the unsupervised surface-defect inspection method for aluminum profiles is shown in Figure 10.

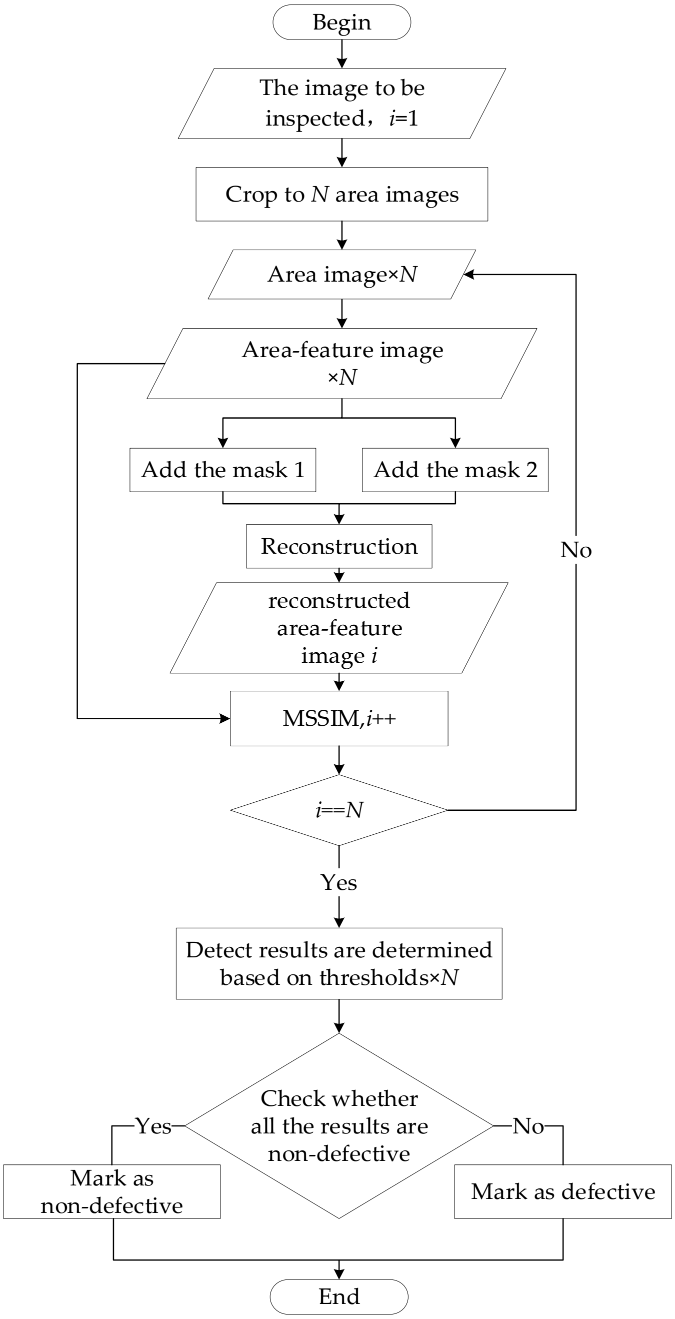
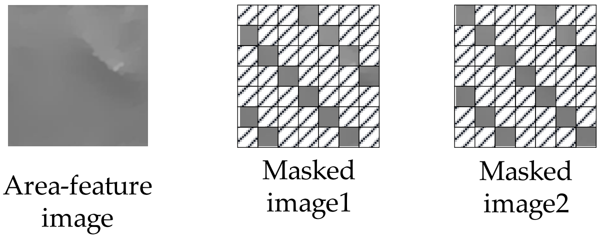


**Figure 10.** Flowchart of the surface-defect detection method.

The aluminum profile surface-defect detection method process is as follows:

(a) Area-feature image acquisition. The main image of the aluminum profile obtained by the aluminum profile image preprocessing method is cropped to an image of uniform specification of 224 × 224, i.e., an area image, and then image restoration operations are performed one by one on the area image to remove its randomly distributed complex texture, and obtain an area-feature image that effectively maintains the essential feature information of the aluminum profile surface.

(b) Mask image production. The entire area image obtained from one aluminum profile image is sequentially masked with two fixed masks each with a removal rate of 75%, respectively, to obtain two mask images, as shown in Figure 11.



Area-feature image

Masked image1

Masked image2

**Figure 11.** Mask images.

For mask image 1, the mask region is denoted by $\left\{\{M_{4i+j}\}_{i=0}^{48}\right\}_{j=0}^{2}$. For mask image 2, the mask region is denoted by $\left\{\{M_{4i+j}\}_{i=0}^{48}\right\}_{j=1}^{3}$.

(c) Feature Reconstruction: The mask image is fed into the detection model, and the feature-reconstructed image of the corresponding area-feature image is reconstructed sequentially.

(d) Defective or non-defective judgment: The MSSIM comparison is performed on the feature-reconstructed image and the regional feature image, and the regional image whose obtained MSSIM value is less than the judgment threshold is judged to be a defective image, and the regional image whose obtained MSSIM value is greater than the judgment threshold is judged to be a non-defective image.

(e) Aluminum profile surface-defect detection results are classified: If all the regional images cropped out of an aluminum profile image are non-defective images, the aluminum profile is judged to be a non-defective product, otherwise it is judged to be a defective product.

### 3.3.2. Definition of Judgment Thresholds

The MSSIM comparison between the eliminated complex-texture image mentioned in Section 3.3.1 and its corresponding restored image is performed using the following principle:

$$MSSIM = \frac{1}{N}\sum_{k=1}^{N} SSIM(x_k, y_k) \tag{26}$$

$N$ in the formula is the number of image division blocks, $SSIM$ is a number from 0 to 1, the larger indicates that the gap between the output image and the distortion-free image is smaller, i.e., the better the quality of the image, and the principle of $SSIM$ is shown in Equations (27)–(30):

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{27}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2\mu_y^2 + c_1} \tag{28}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2\sigma_y^2 + c_2} \tag{29}$$

$$s(x,y) = \frac{\sigma_{x,y} + c_3}{\sigma_x\sigma_y + c_3} \tag{30}$$

where $l(x,y)$ is the luminance comparison, $c(x,y)$ is the contrast comparison, $s(x,y)$ is the structural comparison, $\mu_x$ and $\mu_y$ denote the mean of $x$ and $y$, respectively, $\sigma_x$ and $\sigma_y$ denote the standard deviation of $x$ and $y$, respectively, $\sigma_{xy}$ denotes the covariance of $x$ and $y$, and $c_1, c_2, c_3$ denotes the constants, $\alpha > 0$, $\beta > 0$, and $\gamma > 0$, respectively, and in this paper we take $\alpha = \beta = \gamma = 1$.

The “3$\sigma$ criterion” [39] is used to select the judgment threshold, and the selection of the judgment threshold in this paper can be described as:

$$T = \mu + \eta * \sigma \tag{31}$$

In the formula, $\mu$ denotes the mean of the image, $\sigma$ denotes the standard deviation of the image, $\eta$ is the coefficient of the standard deviation $\sigma$, and $T$ is the threshold. The mean and standard deviation used in this paper are defined as:

$$\mu = \frac{\sum \Delta x_{(i,j,k)}}{n} \tag{32}$$

$$\sigma = \sqrt{\frac{\sum \Delta x_{(i,j,k)} - \mu}{n}} \tag{33}$$

where $n$ denotes the number of all pixel points involved in the computation of the image. After experimentation $\eta$ finally takes the value of 3.

## 4. Experimental Results and Discussion

### *4.1. Experimental Environment*

The experimental environment is a 64-bit Win10, Intel(R) Core(TM) i9-12 900H@2.50 GHz processor, and an NVIDIA GeForce RTX 3 07 Laptop’s GPU (NVIDIA Corporation, San Jose, CA, USA). The test platform comprises Python 3.7, CUDA Toolkit 10.0, and OpenCV 4.6.0. The hardware and software environment is shown in Table 1.

**Table 1.** Experimental environment.

| Environment | Name | Model |
|---|---|---|
| Hardware environment | Processor | Intel(R) Core(TM) i9-12900H@2.50 GHz |
| | Internal memory | 32GB DDR5 |
| | GPU | NVIDIA GeForce RTX 3070Ti Laptop GPU |
| Software environment | CUDA Toolkit | 10.0 |
| | Pytorch | 1.7.1 |
| | Data | Data |

### *4.2. Data Description*

The raw dataset of aluminum profiles used for the experiments was provided by the Aliyun Tianchi Competition organized by Alibaba (https://tianchi.aliyun.com/competition/entrance/231682/information, accessed on 12 October 2023) [40], in which the part of the data with flat surfaces of aluminum profiles is selected for the experiment. Figure 12 shows an illustration of the contrast between defective and non-defective aluminum profile images with and without flat surfaces.

A total of 330 sheets of non-defective aluminum profile data were selected as the training set. A total of 217 sheets with ten types of defective aluminum profile data such as scrape, bruise, crater, coating cracking, orange peel, bumping, pit, convex powder, bubbles, and dirty spots defects, and the remaining 123 sheets of non-defective aluminum profiles were selected as the test set. Figure 13 shows an example of some of the types of defects in an image of an aluminum profile with a flat surface.

For better training, the aluminum profile images are cropped into 224 × 224 uniformly sized images and the cropped training set is rotated by 90°, 180°, and 270°, and then flipped horizontally and left and right to get 53,904 aluminum profile images.

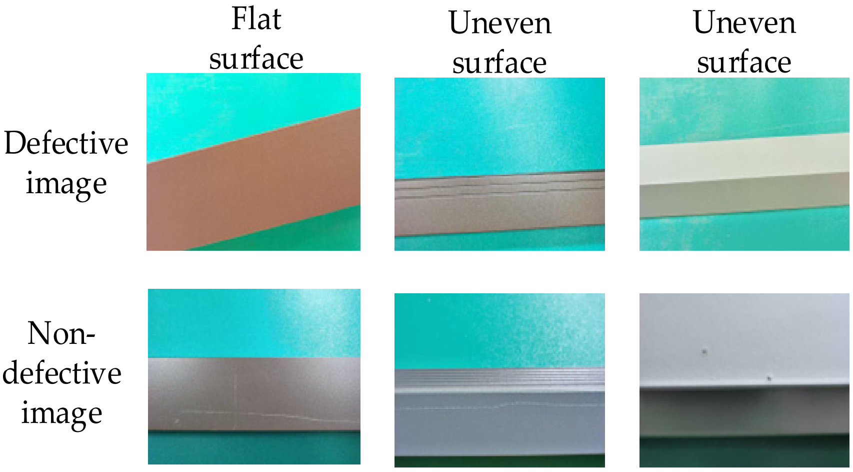


**Figure 12.** Comparison chart of aluminum profile surface flats with and without (Part of it).

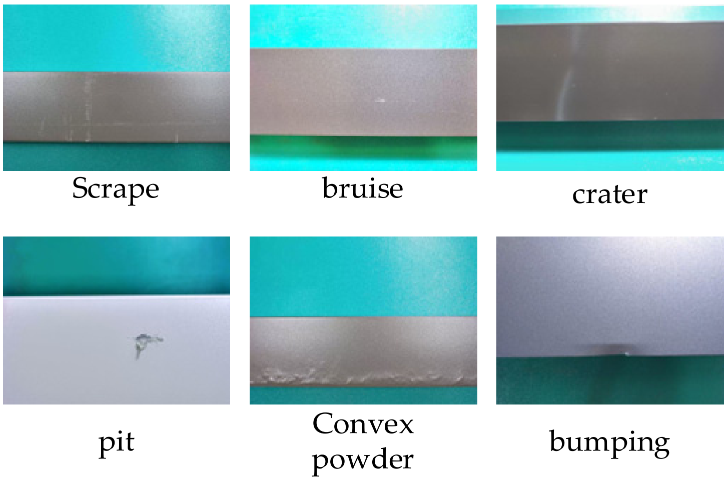


**Figure 13.** Select defect categories(part).

### *4.3. Experimental Result*

In this paper, experiments were carried out using the test set (tests were conducted using area images of size 224 × 224), and the detection results are shown in Figure 14. The vertical coordinate indicates the MSSIM value between the repaired image and the corresponding area image, and the horizontal black dashed line in the figure indicates the segmentation threshold, which actually takes the value of 0.99625, the vertical blue dashed line simply visually distinguishes the three datasets. From Figure 14, it can be seen that the MSSIM value between the good aluminum profile image and its repaired image is concentrated above the threshold segmentation line, which tends to be close to 1, while the MSSIM value between the defective aluminum profile image and its repaired image is mostly concentrated below the threshold segmentation line, from which it can be seen that the model is very sensitive to the defective image.

Results for the detection of aluminum profiles are shown in Figure 15. The images used in the first three rows are defective images and the images used in the last two rows are non-defective images. The corresponding part of the area image, feature-optimized image, feature-reconstructed image, and determination results during the detection process are shown in columns 2, 3, 4, and 5 of Figure 15.

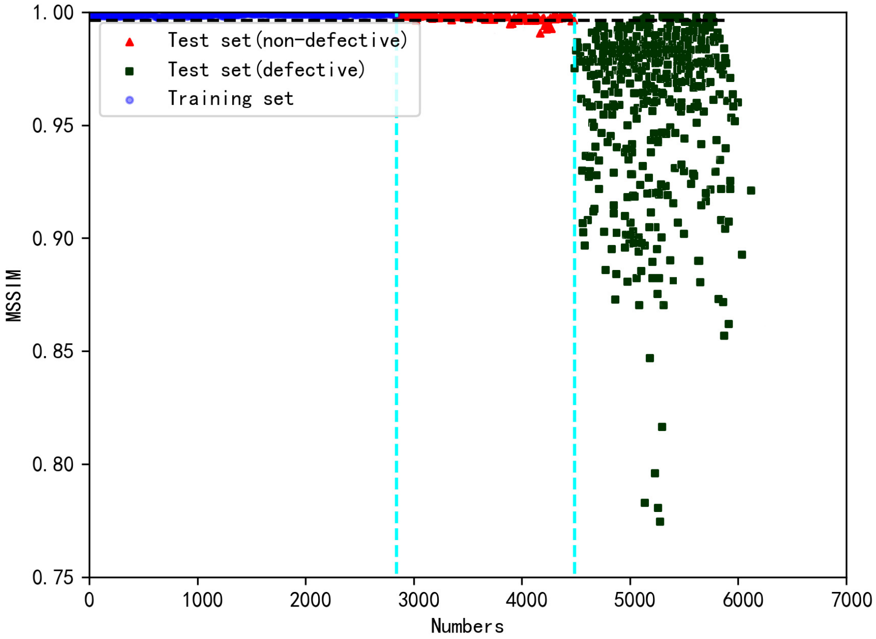


**Figure 14.** Detection effect graph.

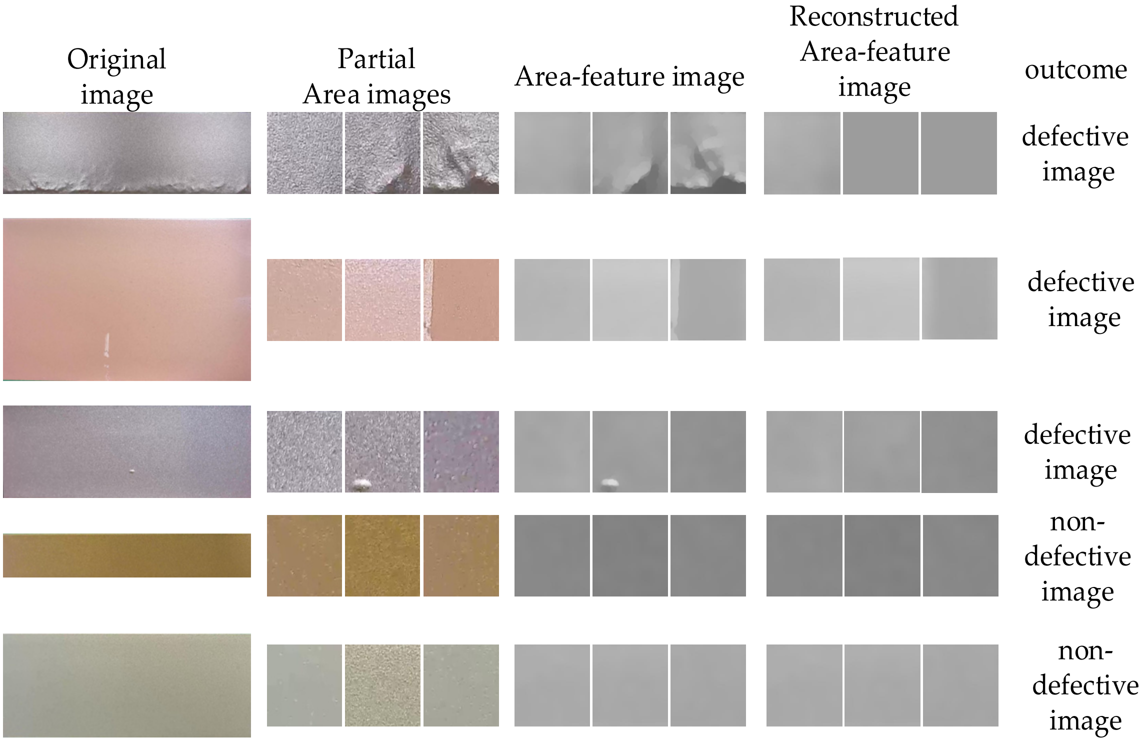


**Figure 15.** Diagram of test results.

### 4.4. *Comparative Experiments*

#### 4.4.1. Experiments Comparing Feature-Optimization Methods

Aluminum profile surfaces have a large number of complex, irregular texture features, and these complex texture features will greatly affect a model's feature extraction effect. In order to reduce data redundancy and enable the model to better extract the essential features of the aluminum profile surface, this paper adopts various feature-optimization methods to operate on the area image of the aluminum profile, and selects the best method for feature optimization through comparison. This process is shown in Figure 16, in which the left four columns are the effect diagrams after feature optimization through the methods of BIOR, DWTN, and HAR, and the right four columns are the effect diagrams after operating on the area image with defects through the same methods.

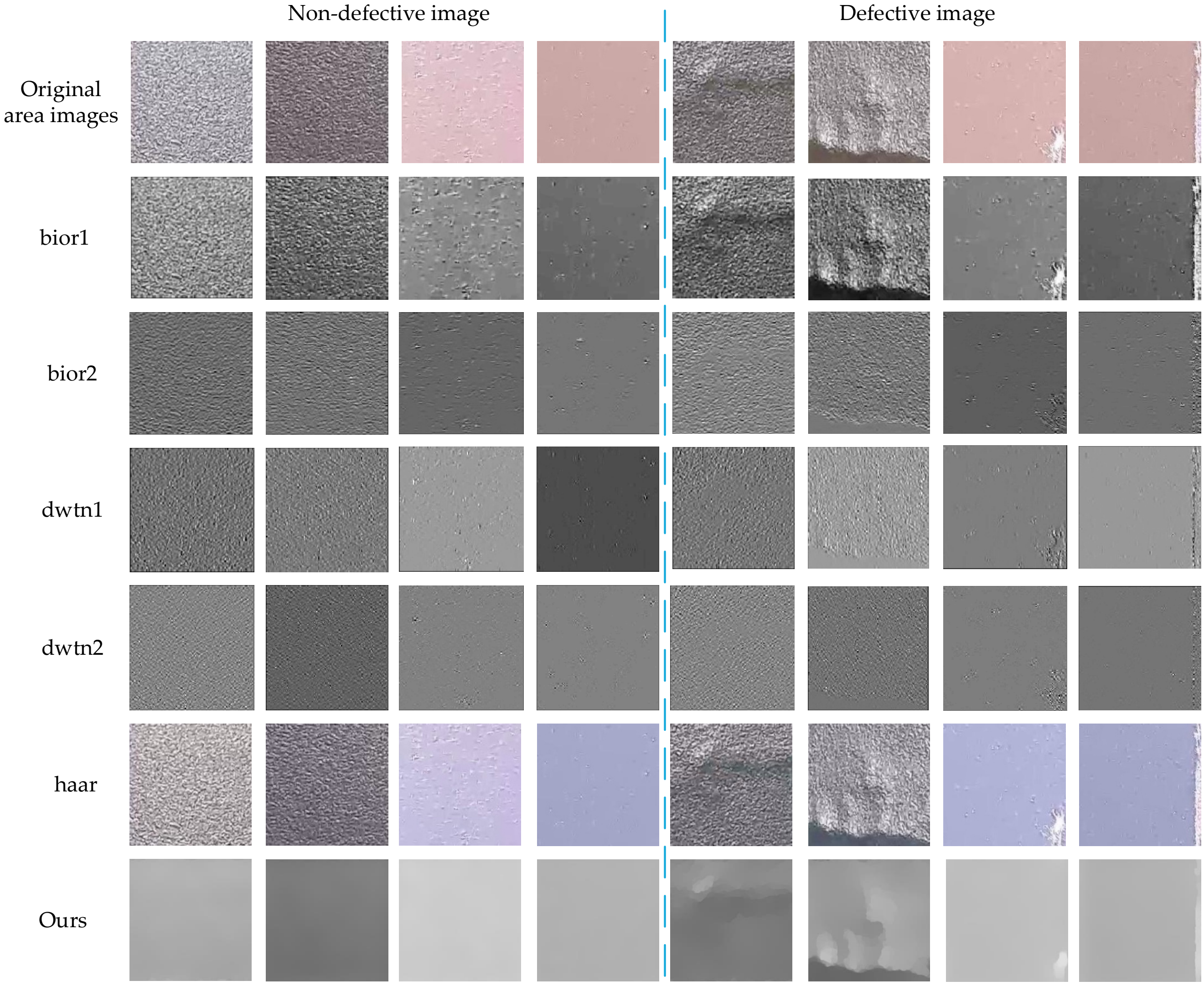


**Figure 16.** Comparison of the effectiveness of different feature-optimization methods.

Through comparison, it can be seen that the feature-optimization module used in this paper can effectively eliminate the interference of complex texture features on the surface of aluminum profiles. At the same time, the method effectively retains the features of defective parts in the image, while the remaining methods have difficulty eliminating the interference by complex textures, and can easily to create the model of the surface of the aluminum profile using good features mistakenly judged as defective features, thus affecting the later detection results.

The feature-optimized image is binarized and then classified directly according to the black and white pixel points with an accuracy of 0.83, and its partial binarization results are shown in Figure 17.

As can be seen in Figure 17, during the process of classifying the feature-optimized image directly according to the black and white pixel points after binarization, some of the binarized maps differ greatly from what is expected and affect the detection results so that their detection accuracy is low, making it necessary to extract the essential features by MAE so as to achieve accurate classification.

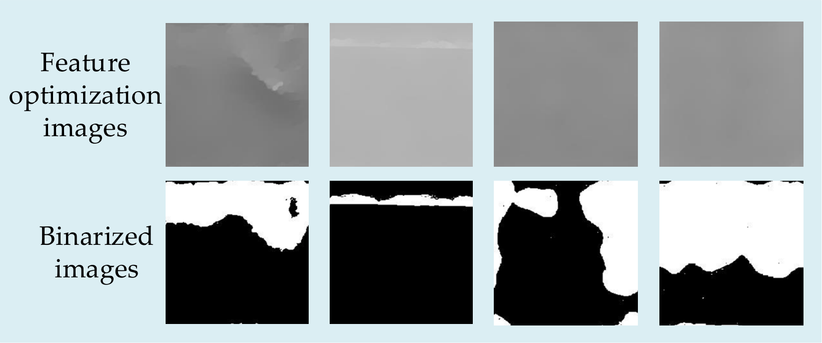


**Figure 17.** Partial feature-optimization image binarization results.

### 4.4.2. Model Comparison Experiment

In order to evaluate the performance of the surface-defect detection method for aluminum profiles described in this paper, in this section, the detection method is compared with three unsupervised detection methods, namely, Masked Auto-Encoders (MAE), Variational Auto-Encoders (VAE) incorporating feature-optimization methods (VAE + ROF), and the Generative Adversarial Network (GAN) incorporating feature-optimization methods (GAN + ROF).

In this paper, measures of recall (RC), precision (PR), accuracy (Acc), the $F1$ value, and the ROC curve are used as indicators to evaluate the performance of aluminum profile surface-defect detection. $RC$, $PR$, and $F1$ are defined as follows:

$$RC = \frac{TP}{TP + FN} \tag{34}$$

$$PR = \frac{TP}{TP + FP} \tag{35}$$

$$F1 = \frac{2PR \times RC}{PR + RC} \tag{36}$$

where $TP$ denotes the number of non-defective samples detected correctly, $FN$ denotes the number of defective samples detected incorrectly, and $FP$ denotes the number of non-defective samples detected incorrectly. $PR$ and $RC$ are used to evaluate the quality of the model's detection results, and the $F1$ value is an index used to reflect the overall detection results of the model.

The ROC curve is called the Subject Operating Characteristic curve, in which the horizontal axis is the false-positive rate ($FPR$) and the vertical axis is the true-positive rate ($TPR$). Scholars often use the ROC curve to evaluate the effectiveness of a model, and the closer the curve is to the upper left corner, the more accurate the work of the subject, that is, the better the detection effect. $FPR$, $TPR$ are defined as follows:

$$FPR = \frac{FP}{FP + TN} \tag{37}$$

$$TPR = \frac{TP}{TP + FN} \tag{38}$$

where $TN$ denotes the number of pixels that successfully detected a non-defective region.

The three algorithms were experimented with in terms of detection accuracy and ROC curves, and their reconstruction-comparison diagrams are shown in Figure 18, with the experimental results shown in Figure 19 (The black dashed line in the figure is the reference line.) and Tables 1 and 2.

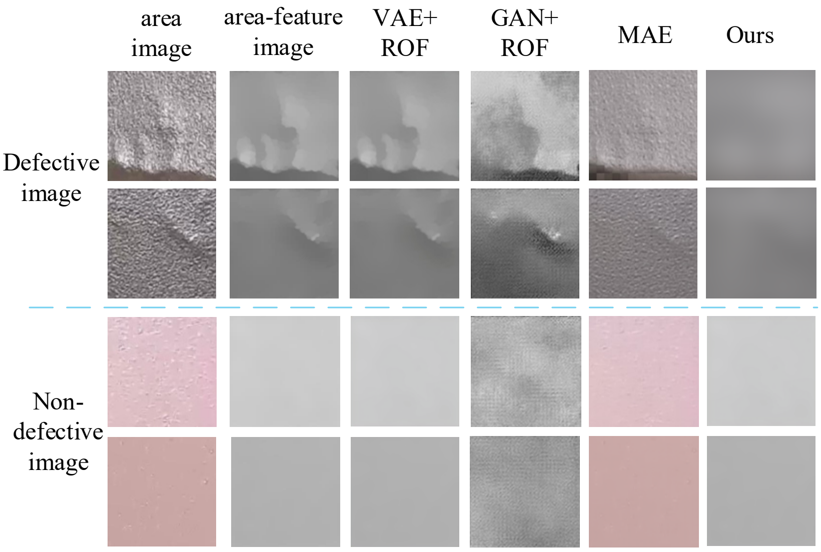


**Figure 18.** Comparison of the effects of different model reconstructions.

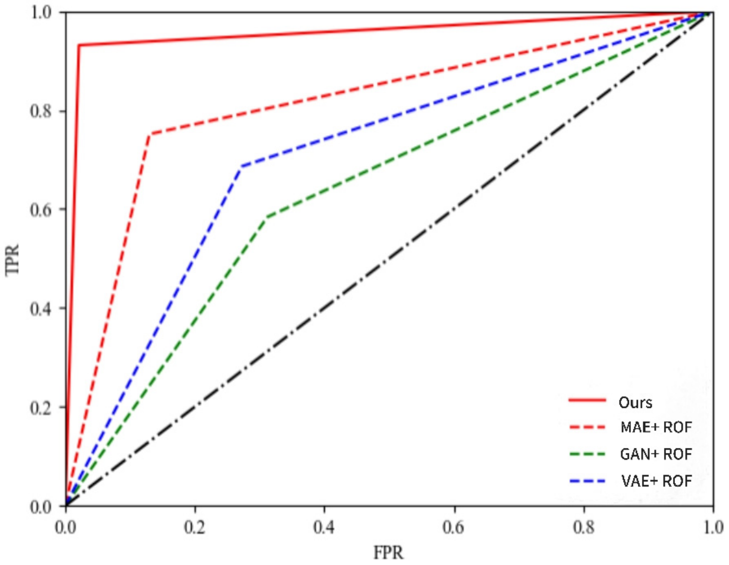


**Figure 19.** Comparison of ROC curves for the four assays.

**Table 2.** Comparison of model accuracy.

| Model | *PR* | *RC* | *F*1 |
|---|---|---|---|
| VAE + ROF | 0.686 | 0.603 | 0.642 |
| GAN + ROF | 0.624 | 0.584 | 0.598 |
| MAE | 0.850 | 0.751 | 0.798 |
| Ours | 0.974 | 0.931 | 0.952 |

As can be seen from Table 1, the precision of this aluminum profile dataset on MAE, VAE incorporating feature-optimization methods, and the Generative Adversarial Network are 0.850, 0.686 and 0.624, respectively. The recall is 0.751, 0.603 and 0.584, respectively, while the *F*1 values are 0.798, 0.642, and 0.598, respectively. Compared with other classical models, the precision rate, recall rate, and *F*1 value of this paper's model are improved by at least 12.4%, 18%, and 15.4%, respectively. The experiments show that the model in this paper can effectively eliminate the interference of complex textures on the surface of aluminum profiles and improve the detection accuracy.

The results of comparison with Models from the literature [41] are shown in Table 3:

**Table 3.** Model comparison results.

| Model | *PR* | *TPR* | *TNR* | *F*1 | *AUC* |
|---|---|---|---|---|---|
| AE(L2) | 0.78 | 0.74 | 0.45 | 0.76 | 0.68 |
| AE(SSIM) | 0.83 | 0.64 | 0.66 | 0.73 | 0.67 |
| VAE | 0.79 | 0.64 | 0.54 | 0.71 | 0.63 |
| AnoGAN | 0.69 | 0.68 | 0.58 | 0.63 | 0.64 |
| GANomaly | 0.79 | 0.74 | 0.49 | 0.76 | 0.72 |
| DPAE | 0.96 | 0.92 | 0.89 | 0.94 | 0.92 |
| Ours | 0.97 | 0.93 | 0.98 | 0.95 | 0.95 |

The DPAE method is the unsupervised detection method for aluminum profiles proposed in this literature. As can be seen from Table 2, the *PR*, *TPR*, *TNR*, *F*1, and *AUC* of the aluminum profile dataset of the Aliyun Tianchi Competition from this method are 98.2%, 97.6%, 97.9%, 97.9%, and 97.7%, respectively; all of these values are significantly higher than those detected using the methods of AE(L2), AE(SSIM), VAE, AnoGAN, GANomaly, and DPAE Results. Also, the detection results for this aluminum profile dataset using the method proposed in this paper improved *PR*, *TPR*, *TNR*, *F*1, and *AUC* by 2.2%, 5.6%, 8.9%, 3.9%, and 5.7%, respectively, when compared with the detection results on the DPAE method. Experiments have shown that the model proposed in this paper can effectively detect whether there are defects on the surface of aluminum profiles.

The results using our model are compared with the MA-YOLO method proposed in the literature [42], and some other mainstream supervised surface-defect detection methods mentioned in that literature, as shown in Table 4.

**Table 4.** Comparison results with supervised deep learning models.

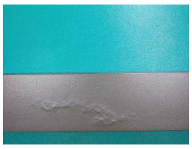
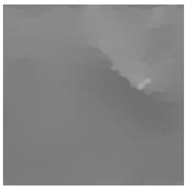
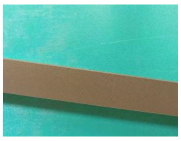
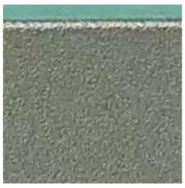
| Model | *PR* | *RC* | *F*1 |
|---|---|---|---|
| SSD300 | 0.958 | 0.511 | 0.672 |
| Faster R-CNN | 0.541 | 0.879 | 0.658 |
| YOLOv3 | 0.901 | 0.787 | 0.826 |
| YOLOv4 | 0.928 | 0.599 | 0.671 |
| YOLOv5s | 0.884 | 0.792 | 0.827 |
| YOLOX_s | 0.875 | 0.829 | 0.847 |
| YOLOv6s | 0.769 | 0.725 | 0.739 |
| YOLOv7 | 0.878 | 0.554 | 0.639 |
| DETR | 0.711 | 0.837 | 0.778 |
| MA-YOLO | 0.908 | 0.811 | 0.849 |
| Ours | 0.974 | 0.931 | 0.952 |

As can be seen from Table 4, the *PR*, *RC*, and *F*1 values of our proposed method on this dataset are significantly higher than those of SSD300, Faster R-CNN, YOLOv3, YOLOv4, YOLOv5s, YOLOX_s, YOLOv6s, YOLOv7, DETR, and MA-YOLO, and the *PR*, *RC*, and *F*1 values are at least improved by 1.6%, 5.2%, and 10.3%, respectively. Our experiments have shown that the model in this paper can effectively detect whether there are defects on the surface of aluminum profiles.

Table 5 shows an example of the identification process for some of the samples, where the first two rows are correctly identified samples and the last two rows are incorrectly identified samples, the penultimate column is the actual output and the last column is the desired output.

As can be seen from Table 5, due to lighting, shooting angle, and other issues resulting in large differences in the background of aluminum profiles in this dataset, the main part of the aluminum profiles is affected by varying degrees of exposure, uneven illumination, strong reflections, and background noise, etc. However, the model proposed in this paper can effectively remove the interference of the complex textures on the surfaces of the aluminum profiles while retaining their defective features, and can effectively identify defective samples. The main reason for the identification error of the third row of samples is that the lack of prior features of the sample is too small, and in the process of feature optimization, the defective features and texture are optimized together, which affects the detection results. For the last row of samples from the acquisition process using the shooting angle, the image has part of the aluminum profile side information; under the influence of the lighting effect, the side information color presentation and the rest of the main part of the difference, and in the process of data normalization is retained, the gloss, color difference affects the final results of the detection.

**Table 5.** Sample identification examples.

| Original Image | Normalization Data | Partial Area Image | Area-Feature Image | Reconstructed Area-Feature Image | Actual | Desired |
|---|---|---|---|---|---|---|
|  | | |  | | Defective | Defective |
|  | | | | | Non-defective | Non-defective |
| | | | | | Non-defective | Defective |
| | |  | | | Defective | Non-defective |

### 4.5. Ablation Experiment

In Section 3.2.3 we mentioned that when a 224 × 224 image is fed into the essential features extraction network, it is divided into 16 × 16 blocks at the feature-optimization layer, but we also tried dividing it into 32 × 32 blocks and 8 × 8 blocks and comparing the results obtained in these cases. Figure 20 shows an example of the reconstruction results from the model for the feature-optimized image in these three blocking scenarios.

From this figure, we can see that when the feature-optimization image is divided into 8 × 8 blocks, its reconstructed image is more likely to have defective residues. When the feature-optimization image is divided into 16 × 16 blocks, its reconstruction effect is better compared to the other two cases. When the feature-optimization image is divided into 32 × 32 blocks, in addition to defective residues appearing in some of the images, non-defective parts of the image may also be disturbed by other factors interference, which leads to the problem of inconsistent texture of non-defective regions before and after reconstruction. The comparison of the detection precision for the three cases is shown in Table 6.

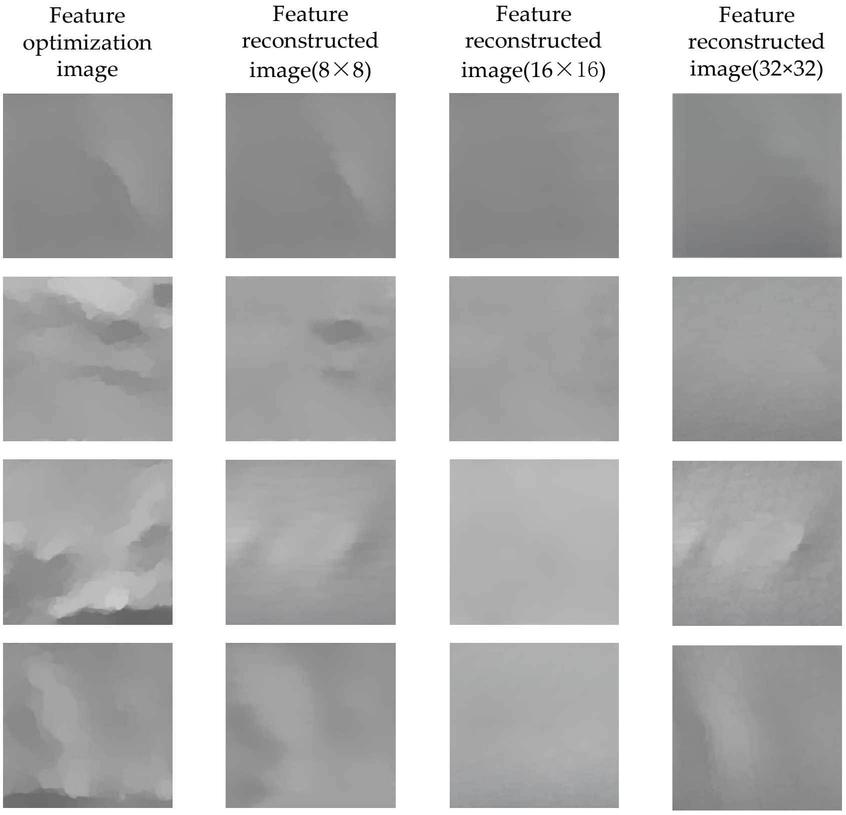

Feature optimization image
Feature reconstructed image(8×8)
Feature reconstructed image(16×16)
Feature reconstructed image(32×32)

**Figure 20.** Comparison of model outputs for different blocking scenarios (some examples).

**Table 6.** Comparison of detection results within different blocking scenarios.

| Blocks | 8 × 8 | 16 × 16 | 32 × 32 |
|---|---|---|---|
| *PR* | 0.581 | 0.974 | 0.738 |

From Table 6, it can be seen that the method dividing the image into 16 × 16 blocks has the best detection effect, that dividing it into 32 × 32 blocks has the second best detection effect, and that dividing it into 8 × 8 blocks has the weakest detection effect among the three.

## 5. Conclusions

The unsupervised surface-defect detection method for aluminum profiles focusing on feature reconstruction contains three parts: aluminum profile image preprocessing, essential feature learning and surface-defect detection. The adaptive boundary extraction method of the preprocessing process can accurately determine the main part of the aluminum profile under the background of different colors and complex lighting, can reduce redundant data, and can lay the foundation for improving the accuracy of the detection method. The feature-optimization module is integrated into the MAE model, which effectively eliminates the interference of the complex texture randomly distributed on the surface of aluminum profiles, improves the model's focus on the essential features of aluminum profiles, weakens the influence of irrelevant information on the model, and effectively extracts the surface features of the good aluminum profiles. The method also adds a fixed local masking strategy, which masks the defective information on the surface of the aluminum profile as much as possible during the detection process, and it is able to obtain its global features from the unmasked area, which ultimately achieves a better repair effect. The experimental results show that its detection accuracy reaches 97.7%, compared with the current common methods for detecting defects on the surface of aluminum profiles. The training process for this method does not require defect samples or a large amount of data labeled in advance, which reduces the labor cost. At the same time, the method is not subject to the limitations of the types of defect samples, giving higher accuracy, and the equipment required for the model training meets the cost requirements of industrial production. However, the proposed method still has challenges in distinguishing small and low-contrast defects, so future research will continue to explore more efficient feature-optimization methods.

**Author Contributions:** Conceptualization, Y.Z. and S.T.; Methodology, Y.Z. and Z.J.; software, Y.Z. and H.L.; validation, Z.J.; investigation, Z.J. and J.Y.; writing—original draft preparation, Y.Z. and J.L.; writing—review and editing, H.L. and J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key Research and Development Program of China (2018YFC0808300), Shaanxi Science and Technology Plan Key Industry Innovation Chain (Group)—Project in Industrial Field (2020ZDLGY15-07), Xi'an Science and Technology Program (23KGDW0032-2022).

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

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The dataset used in this study was downloaded at: https://tianchi.aliyun.com/competition/entrance/231682/information.

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

## References

1. Tian, H.; Wang, D.; Lin, J.; Chen, Q.; Liu, Z. Surface defects detection of stamping and grinding flat parts based on machine vision. *Sensors* **2020**, *20*, 4531. [CrossRef] [PubMed]
2. Xu, C.; Li, L.; Li, J.; Wen, C. Surface defects detection and identification of lithium battery pole piece based on multi-feature fusion and PSO-SVM. *IEEE Access* **2021**, *9*, 85232–85239. [CrossRef]
3. Le, N.; Rathour, V.S.; Yamazaki, K.; Luu, K.; Savvides, M. Deep reinforcement learning in computer vision: A comprehensive survey. *Artif. Intell. Rev.* **2022**, *55*, 2733–2819. [CrossRef]
4. Hesamian, M.H.; Jia, W.; He, X.; Kennedy, P. Deep learning techniques for medical image segmentation: Achievements and challenges. *J. Digit. Imaging* **2019**, *32*, 582–596. [CrossRef] [PubMed]
5. Martínez, S.S.; Vázquez, C.O.; García, J.G.; Ortega, J.G. Quality inspection of machined metal parts using an image fusion technique. *Measurement* **2017**, *111*, 374–383. [CrossRef]
6. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [CrossRef] [PubMed]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
8. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer 690 Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
9. Liao, D.; Cui, Z.; Zhu, Z.; Jiang, Z.; Zheng, Q.; Wu, N. A nondestructive recognition and classification method for detecting surface defects of $Si_3N_4$ bearing balls based on an optimized convolutional neural network. *Opt. Mater.* **2023**, *136*, 113401. [CrossRef]
10. Song, K.; Sun, X.; Ma, S.; Yan, Y. Surface Defect Detection of Aero-engine Blades Based on Cross-layer Semantic Guidance. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2514411. [CrossRef]
11. Guan, S.; Shi, H. Fabric defect detection based on the saliency map construction of target-driven feature. *J. Text. Inst.* **2018**, *109*, 1133–1142. [CrossRef]
12. Wang, C.; Xie, H. MeDERT: A Metal Surface Defect Detection Model. *IEEE Access* **2023**, *11*, 35469–35478. [CrossRef]
13. Xiao, S.; Liu, Z.; Yan, Z.; Wang, M. Grad-MobileNet: A Gradient-Based Unsupervised Learning Method for Laser Welding Surface Defect Classification. *Sensors* **2023**, *23*, 4563. [CrossRef] [PubMed]
14. Zhou, Q.; Wang, H.; Wang, Y. Defect detection method based on knowledge distillation. *IEEE Access* **2023**, *11*, 35866–35873. [CrossRef]
15. Rui, J.; Qiang, N. Research on textile defects detection based on improved generative adversarial network. *J. Eng. Fibers Fabr.* **2022**, *17*, 15. [CrossRef]
16. Guo, Y.; Zhong, L.; Qiu, Y.; Wang, H.; Gao, F.; Wen, Z.; Zhan, C. Using ISU-GAN for unsupervised small sample defect detection. *Sci. Rep.* **2022**, *12*, 11604. [CrossRef] [PubMed]
17. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]

18. Qi, L.; Zhang, Y.; Liu, T. Bidirectional Transformer with absolute-position aware relative position encoding for encoding sentences. *Front. Comput. Sci.* **2023**, *17*, 171301. [CrossRef]
19. Von der Mosel, J.; Trautsch, A.; Herbold, S. On the validity of pre-trained transformers for natural language processing in the software engineering domain. *IEEE Trans. Softw. Eng.* **2022**, *49*, 1487–1507. [CrossRef]
20. Mosin, V.; Samenko, I.; Kozlovskii, B.; Tikhonov, A.; Yamshchikov, I.P. Fine-tuning transformers: Vocabulary transfer. *Artif. Intell.* **2023**, *317*, 103860. [CrossRef]
21. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey. *ACM Comput. Surv.* **2022**, *55*, 109. [CrossRef]
22. Badaro, G.; Saeed, M.; Papotti, P. Transformers for Tabular Data Representation: A survey of models and applications. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 227–249. [CrossRef]
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
24. Yu, H.; Liu, D.; Zhang, Z.; Wang, J. A Dynamic Transformer Network with Early Exit Mechanism for Fast Detection of Multiscale Surface Defects. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5025710. [CrossRef]
25. 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. [CrossRef]
26. Shang, H.; Sun, C.; Liu, J.; Chen, X.; Yan, R. Defect-aware transformer network for intelligent visual surface defect detection. *Adv. Eng. Inform.* **2023**, *55*, 101882. [CrossRef]
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Zhou, W.; Hong, J. FHENet: Lightweight Feature Hierarchical Exploration Network for Real-Time Rail Surface Defect Inspection in RGB-D Images. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5005008. [CrossRef]
29. Chang, Q.; Li, X.; Zhao, Y. Reversible data hiding for color images based on adaptive three-dimensional histogram modification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5725–5735. [CrossRef]
30. Fan, Z.; Lin, H.; Li, C.; Su, J.; Bruno, S.; Loprencipe, G. Use of parallel ResNet for high-performance pavement crack detection and measurement. *Sustainability* **2022**, *14*, 1825. [CrossRef]
31. Huo, D.; Wang, J.; Qian, Y.; Yang, Y.H. Glass segmentation with RGB-thermal image pairs. *IEEE Trans. Image Process.* **2023**, *32*, 1911–1926. [CrossRef]
32. Song, K.; Bao, Y.; Wang, H.; Huang, L.; Yan, L. A Potential Vision-Based Measurements Technology: Information Flow Fusion Detection Method Using RGB-Thermal Infrared Images. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 5004813. [CrossRef]
33. Zhang, F.; Jiang, X.; Xia, Z.; Gabbouj, M.; Peng, J.; Feng, X. Non-Local Color Compensation Network for Intrinsic Image Decomposition. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 132–145. [CrossRef]
34. Zhang, Y.; Di, X.; Zhang, B.; Ji, R.; Wang, C. Better than reference in low-light image enhancement: Conditional re-enhancement network. *IEEE Trans. Image Process.* **2021**, *31*, 759–772. [CrossRef] [PubMed]
35. Mohsan, M.M.; Akram, M.U.; Rasool, G.; Alghamdi, N.S.; Baqai, M.A.A.; Abbas, M. Vision Transformer and Language Model Based Radiology Report Generation. *IEEE Access* **2022**, *11*, 1814–1824. [CrossRef]
36. Zhuang, X.; Liu, F.; Hou, J.; Hao, J.; Cai, X. Modality attention fusion model with hybrid multi-head self-attention for video understanding. *PLoS ONE* **2022**, *17*, e0275156. [CrossRef]
37. Lou, X.; Jia, Z.; Yang, J.; Kasabov, N. Change detection in SAR images based on the ROF model semi-implicit denoising method. *Sensors* **2019**, *19*, 1179. [CrossRef]
38. Bakurov, I.; Buzzelli, M.; Schettini, R.; Castelli, M.; Vanneschi, L. Full-Reference Image Quality Expression via Genetic Programming. *IEEE Trans. Image Process.* **2023**, *32*, 1458–1473. [CrossRef]
39. Pukelsheim, F. The Three Sigma Rule. *Am. Stat.* **1994**, *48*, 88–91.
40. Chen, R.; Cai, D.; Hu, X.; Zhan, Z.; Wang, S. Defect detection method of aluminum profile surface using deep self-attention mechanism under hybrid noise conditions. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3524509. [CrossRef]
41. Liu, J.; Song, K.; Feng, M.; Yan, Y.; Tu, Z.; Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Opt. Lasers Eng.* **2021**, *136*, 106324. [CrossRef]
42. Jiang, L.; Yuan, B.; Wang, Y.; Ma, Y.; Du, J.; Wang, F.; Guo, J. MA-YOLO: A Method for Detecting Surface Defects of Aluminum Profiles with Attention Guidance. *IEEE Access* **2023**, *11*, 71269–71286. [CrossRef]

**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.