electronics

MDPI

Article

Detection of Scratch Defects on Metal Surfaces Based on MSDD-UNet

Yan Liu †, Yunbai Qin *, Zhonglan Lin †, Haiying Xia * and Cong Wang

School of Electronic and Information Engineering, Guangxi Normal University, Guilin 541004, China; lyyy_ran@stu.gxnu.edu.cn (Y.L.); lin_zhonglan@stu.gxnu.edu.cn (Z.L.); cong@stu.gxnu.edu.cn (C.W.)

* Correspondence: qinyunbai@gxnu.edu.cn (Y.Q.); xhy22@mailbox.gxnu.edu.cn (H.X.)

† These authors contributed equally to this work.

Abstract: In this work, we enhanced the U-shaped network and proposed a method for detecting scratches on metal surfaces based on the Metal Surface Defect Detection U-Net (MSDD-UNet). Initially, we integrated a downsampling approach using a Space-To-Depth module and a lightweight channel attention module to address the loss of contextual information in feature maps that results from multiple convolution and pooling operations. Building on this, we developed an improved attention module that utilizes image frequency decomposition and cross-channel self-attention mechanisms, as well as the strengths of convolutional encoders and self-attention blocks. Additionally, this attention module was integrated into the skip connections between the encoder and decoder. The purpose was to capture dense contextual information, highlight small and fine target areas, and assist in localizing micro and fine scratch defects. In response to the severe foreground–background class imbalance in scratch images, a hybrid loss function combining focal loss and D_{ice} loss was put forward to train the model for precise scratch segmentation. Finally, experiments were conducted on two surface defect datasets. The results reveal that our proposed method is more advantageous than other state-of-the-art scratch segmentation methods.

Keywords: defect detection; attention mechanism; hybrid loss; U-Net; SPD module; semantic segmentation

Citation: Liu, Y.; Qin, Y.; Lin, Z.; Xia, H.; Wang, C. Detection of Scratch Defects on Metal Surfaces Based on MSDD-UNet. *Electronics* **2024**, *13*, 3241. https://doi.org/10.3390/electronics13163241

Academic Editor: Andrea Bonci

Received: 18 July 2024
Revised: 12 August 2024
Accepted: 14 August 2024
Published: 15 August 2024

1. Introduction

Metals and metal products are indispensable in modern industrial manufacturing. However, defects such as scratches are often unavoidable on their surfaces due to limitations in production processes, tool and equipment wear, careless operation, or collisions and compression during transportation. These scratch defects negatively affect the aesthetic appeal of such products, making them difficult to clean and significantly reducing important properties such as their dimensional stability, surface roughness, and fatigue limits. Therefore, the timely repair of scratches is crucial to ensure the performance of downstream products. Manufacturers can take a series of measures, such as recoating or applying surface treatment techniques, to enhance the aesthetics and durability of metals and metal products. However, before proceeding with a repair, a crucial and indispensable step is detecting the quality of the metal.

Accurate metal defect detection is crucial for modern industrial production. The detection results can be used for the precise quality assessment of products, raw materials, etc., and serve as a basis for making maintenance decisions. In recent years, deep learning methods have significantly improved the performance of the detection of metal surface scratch defects. To achieve high-precision defect detection, large and complex deep learning models are typically designed. However, these models require substantial computational resources and have lengthy training and inference times, making it challenging to achieve the high defect detection efficiency required in industrial production. Moreover, the accurate detection of scratch defects remains challenging due to the inherent class imbalance

between the foreground and background in metal surface scratch defect data, as well as complex factors such as small or elongated targets. To address these issues, this paper proposes a downsampling mode based on a Space-To-Depth (SPD) [1] module and a lightweight channel attention module (LCAM) and enhances the self-attention mechanism used. Based on this, an algorithm is proposed for localizing metal scratch defects at the pixel level for defect segmentation (refer to Figure 1). The main component of the proposed model is an enhanced U-shaped network (U-Net) designed to accurately segment scratches of different sizes from a metal surface image. The primary contributions of this paper are as follows:

(1) A downsampling convolutional block is proposed based on an SPD module and LCAM. It is embedded into the U-Net to acquire more information and mitigate the loss of contextual information in feature maps after multiple convolution and pooling operations in the encoder.

(2) Building upon the above contribution, a lightweight anti-redundant self-attention module is designed to address the geometric features of scratches, such as their direction, thickness, and inherent high-frequency components, by integrating refined edge information. This module is embedded into the skip connections between the encoder and decoder, serving as a precursor for encoding–decoding feature fusion.

(3) Furthermore, a hybrid loss function that combines the focal loss [2] and D_{ice} loss [3] functions is introduced to deal with the inherent foreground–background class imbalance in metal surface scratch defect data. By leveraging their respective unique advantages, the proposed function is utilized to train a model for precise scratch segmentation.

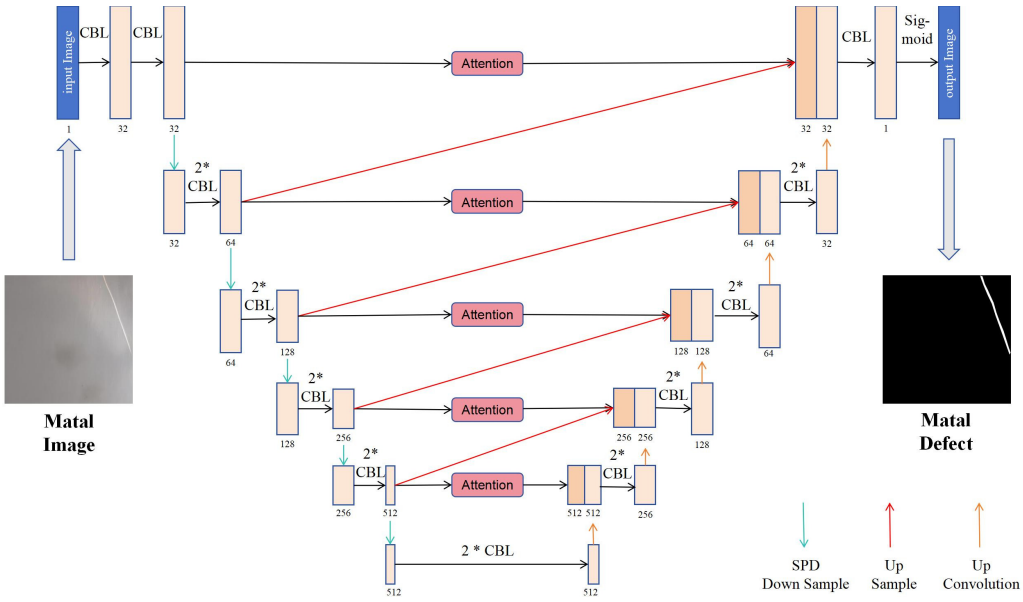

Figure 1. Architecture of the proposed segmentation network. We incorporate an enhanced self-attention block into the feature fusion of these encoding–decoding stages, which acts as the preliminary process of feature fusion. In the encoding process, we utilized an efficient downsampling method based on SPD and LCAM, replacing the traditional max pooling operation, as illustrated by the light green arrow in the figure. Additionally, the red arrow in the figure indicates the upsampling operation, which uses bilinear interpolation, while the orange arrow indicates the Up-Convolution operation.

2. Relevant Works

2.1. Traditional Methods

Early methods of metal defect detection include liquid penetrant testing [4], laser scanning detection, eddy current testing, ultrasonic testing, and magnetic particle inspection. Manual defect detection was once a common method, but it is labour-intensive. With the rapid advancement of the modern manufacturing industry, this manual method is no

longer sufficient to meet the demand for efficient and high-precision defect detection in contemporary industrial production. This is due to its vulnerability to errors and omissions, low efficiency, and excessive reliance on work experience.

2.2. Digital Signal Processing Methods

With the advancement of automation technology, many researchers and technologists have made significant contributions to defect detection which primarily involve traditional digital signal processing methods and deep learning methods.

Regarding over-detection on defect-free surfaces, H. Ono et al. [5] reported a twin illumination and subtraction technique. They developed a prototype system that can detect defects by capturing two images with a slight time difference. Their system is based on strobe lighting and a dual CCD camera. It utilizes a decision tree judgment function that analyses the light/dark patterns of concave defect features. Liu et al. [6] introduced a hardware and software system for online visual inspection. The hardware comprises an upper optical module and a lower optical module. The defect detection algorithm includes an automatic search of the regions of interest (ROI) and defect detection based on the Sobel algorithm. For large steel roll surface defects with fuzzy boundaries, uneven intensity, and complex background textures, Yang et al. [7] proposed a machine vision system based on an edgeless active contour model. This system can accurately segment blurry and non-uniform defects and typically requires less computation time.

Compared to early purely manual detection methods, the above-mentioned approaches utilize traditional image processing and machine learning algorithms for digital signal processing, offering advantages such as objective results and high computational efficiency. However, these approaches rely on specific signal acquisition tools and demonstrate high detection precision only for specific defects, lacking strong generalization ability. Additionally, these approaches require manual feature design, which requires technicians to have specialized prior knowledge of the detection object's properties, such as the metal's acoustic conductivity, as well as its optical refractivity and reflectivity.

2.3. Deep Learning Methods

Compared to digital signal processing methods, deep learning methods, such as convolutional neural networks (CNNs), can automatically extract deep representative features end-to-end from data [8]. In the past decade, deep learning and neural networks have been widely applied in the field of computer vision [9–12] due to their powerful feature extraction capabilities and ability to generalize [13]. This progress has been greatly supported by high-performance graphics cards, cloud computing power, and big data. These advancements have enabled real-time and efficient defect detection in modern industrial production.

In the field of image segmentation, Fully Convolutional Networks (FCNs) [10] and U-Net [11] have provided models for extracting features from segmentation targets more precisely and efficiently. Specifically, U-Net utilizes an encoder–decoder structure with skip connections, enabling the merging and blending of feature maps across different network levels. This process helps the decoder reconstruct feature maps to their original image size, significantly enhancing the ability of a fully convolutional model to preserve details and segment accurately.

Over the past few years, researchers in related fields have made significant progress in utilizing deep learning models to address various challenges encountered in defect segmentation and anomaly detection [14–24]. Ref. [14] proposed a composite vision system to enable simultaneous the 3D depth and 2D grey imaging of the bead surface. This system is designed to detect typical surface defects on aluminium alloy weld beads. Ashkan Moosavian et al. [15] collected a dataset comprising 400 experimental images of crankshafts with structural defects, such as scratches, pitting, and grinding. They simultaneously applied the DexiNed edge detection filter to the training set images and trained a model based on MobileNet to detect these defects accurately. In the detection of scratch defects

on curved metal surfaces, inevitable halos often obscure the defects' features. To address this issue, He et al. [16] used a multiple high-frequency projection illumination imaging method to generate high-contrast images of metal part surfaces. They utilized this method to create a dataset named RMSSC, which includes 50 images of reflective metal surface scratches. Subsequently, they developed a defect detection neural network and enhanced its generalization performance through inverse transfer learning. In response to the need for accurate manual data annotation in supervised training, Ref. [17] proposed a semi-supervised method. This method combines the predictions of an object detector with the segmentation of a zero-shot model, eliminating the need to label a dataset for semantic segmentation.

Currently, deep learning methods have achieved good performances in surface defect detection. The aforementioned works have undoubtedly made significant contributions to defect detection methods based on deep learning. However, efficient and high-precision defect detection still faces certain challenges due to the varying sizes of defects in images, complex production environments affecting data sampling, and the inherent foreground–background class imbalance in metal surface defect data.

3. The Proposed Segmentation Network

In this study, we developed an image segmentation network to localize scratch defects. The backbone network, attention module, and loss function of the proposed network are described in the following section.

3.1. Outline of the Model

Inspired by the U-Net segmentation network, we established a scratch defect segmentation system for metal surface data (Figure 1). This system consists of five components: a backbone network based on encoder–decoder architecture, a downsampling module, an attention module, skip connections, and feature fusion.

The backbone network is a modified version of the basic U-Net model. Initially, a new downsampling approach was proposed, based on LCAM and SPD modules, to capture more information. Subsequently, an improved self-attention module was developed based on image frequency decomposition, leading to the design of a lightweight attention module. This lightweight attention module has been integrated into the skip connections between the encoder and decoder to capture dense global contextual information, emphasize defect areas, and assign weights to each channel. The specific modules, methods, and the loss function will be detailed below.

3.2. Basic Framework

U-Net performs well in most image segmentation applications. Therefore, we improved the U-Net model and utilized it as the fundamental backbone network architecture for our proposed method.

The U-Net segmentation network is composed of an encoder and a decoder. In its enhanced basic architecture, each block in the encoder comprises two convolutional layers, batch normalization modules (Batch Normalize), leaky rectified linear units (Leaky ReLUs), and a downsampling module that reduces the feature maps to 1/4 of their original scale. The decoder part upsamples the feature maps using a deconvolutional layer and two convolutional layers, followed by batch normalization modules and Leaky ReLU functions. To reduce the information loss caused by pooling operations, the blocks corresponding to the encoder and decoder paths are used to splice the feature maps together via skip connections. This integration combines coarse semantic information from each section of the encoder with fine semantic information from each section of the decoder.

Thus, the shallow layers of the encoder focus on low-level features such as image contours, textures, and colours, while the deep layers of the encoder, with a large receptive field, learn high-level features. Moreover, skip connections help to recover lost edge infor-

mation as much as possible, enabling the U-Net to retain high-level semantic information from the encoder while maintaining detailed information from the decoder.

3.3. Improved Self-Attention Module

To enhance the accuracy and robustness of the segmentation results, the DeepLabV2 model [13] included an Atrous Spatial Pyramid Pooling (ASPP) module with multi-scale atrous convolutions. However, this approach falls short in capturing dense contextual information, as it only collects contextual details from a limited number of points surrounding each pixel through atrous convolutions. To address this limitation and establish connections between pairs of pixels, Alexey Dosovitskiy et al. [12] incorporated a self-attention mechanism into computer vision. Traditional semantic segmentation models face challenges in fully utilizing the global information of an image due to the local receptive field constraint of convolutional encoders. To address this issue, Danpei Zhao et al. [25] introduced the RSANet, a regional self-attention-based network. This network enhances the model's logical understanding of image content by examining point-to-point relationships between pixels and surrounding regional pixels, thereby improving segmentation accuracy. However, in a convolutional neural network, neighbouring elements in the feature maps of convolutional layers share semantic information spatially. Golnaz Ghiasi et al. [26] pointed out that even if neurons are killed, convolutional neural networks can still obtain the semantic information of a pixel point according to its neighbouring pixels. Similarly, in traditional self-attention modules, the pairwise dependencies between pixels are typically computed repeatedly. Therefore, the self-attention module must be lightweight and anti-redundant because the global feature maps obtained by traditional self-attention modules often contain redundant information.

The wavelet transform is a powerful extension of the Fourier transform, renowned for its ability to capture the instantaneous characteristics of a signal by utilizing local information in both time and frequency domains. In the field of image processing, especially for tasks with local mutable objectives, the wavelet transform is commonly employed to decompose the input image into low-frequency components and high-frequency components in vertical, horizontal, and diagonal directions. The high-frequency components contain local detailed information, especially edge information. On the other hand, the low-frequency components represent the global information of the image layer, such as colour. Consequently, high-frequency components often require local and detailed processing, while low-frequency components require global structural processing.

$$Attention(X) = Vsoftmax(Q^T K) + X \tag{1}$$

$$Q = W^Q X \tag{2}$$

$$K = W^K X \tag{3}$$

$$V = W^V X \tag{4}$$

As depicted in Figure 2, the feature map generated by the convolutional encoder was a combination of high- and low-frequency features. To prepare for the merging of encoding–decoding features, the encoder's extracted feature map was implicitly divided into low-frequency and high-frequency components by introducing a pair of convolutions with a kernel size of $(1, 1)$. Subsequently, the low-frequency components were fed into the self-attention module to extract their overall features. These overall features were then combined with the high-frequency components into a decoder feature map. This methodology draws inspiration from octave convolution [27] and CSP [28] modules. In the self-attention module, calculations were conducted across channel dimensions, as shown in Equation (1), instead of across spatial dimensions. This adjustment aimed to reduce the computational burden of the self-attention mechanism and implicitly encode global information. Consequently, this approach prevented the self-attention block from generating redundant information while establishing relationships among the pixels of low-frequency

components, resulting in the extraction of efficient global features and the creation of dense contextual information. Simultaneously, the high-frequency components highlighted the target areas, aiding in the localization of scratch defect areas and improving the module's ability to segment minor and fine defects.

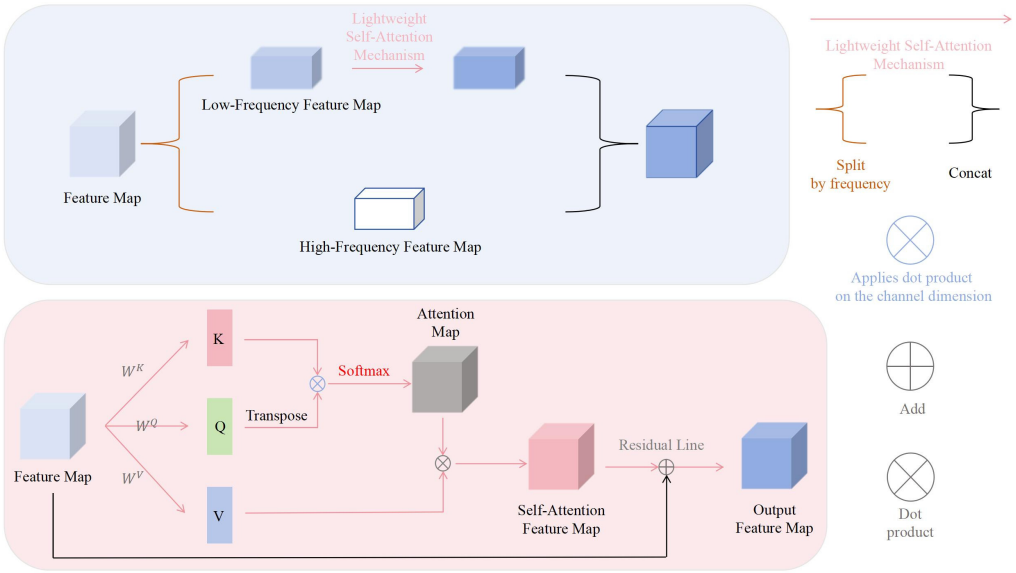

Figure 2. Network structure of the attention module.

3.4. Efficient Downsampling Module

Relying solely on skip connections for feature fusion between the encoder and decoder is far from sufficient for recovering lost features, especially in the initial stage of network training. Before training and evaluating AlexNet, VGGNet16, and ResNet-50, Koziarski et al. [29] reduced the size of the input image in the luminance channel and then filled it again using the bilateral interpolation method. They found that when the resolution of the images used for ResNet-50 training was reduced to 1/5 and 1/8 of its original resolution, the classification accuracy of ResNet-50 dropped to 75% and 50% of its original accuracy, respectively.

To address this issue, Chen et al. [30] proposed a global context-aware progressive aggregation network based on the U-Net framework. This network consists of four modules: Feature Interweaving Aggregation (FIA), Self-Refinement (SR), Head Attention (HA), and Global Context Flow (GCF). These modules interweave and fuse the global contextual information, low-level detail information, and high-level semantic information of the U-Net, obtaining richer feature information. Zhao et al. [31] proposed an edge-guided network (EGNet) for salient object detection, which progressively integrates and complements local edge information and global position information to extract prominent edge features. However, we believe that these approaches do not tackle the core issue; instead, the focus should be on alleviating the loss of detailed information at the source.

In this paper, a downsampling convolution block based on an SPD module and a lightweight channel attention module was proposed and embedded into the encoder to reduce the contextual information loss caused by the multiple convolutions and pooling operations conducted in the encoder. This approach prevented the loss of valuable information at its source, thereby capturing more information.

3.4.1. Space-to-Depth Module

A feature map X with dimensions (S, S, C_1) is sliced as follows, and as displayed in Figure 3:

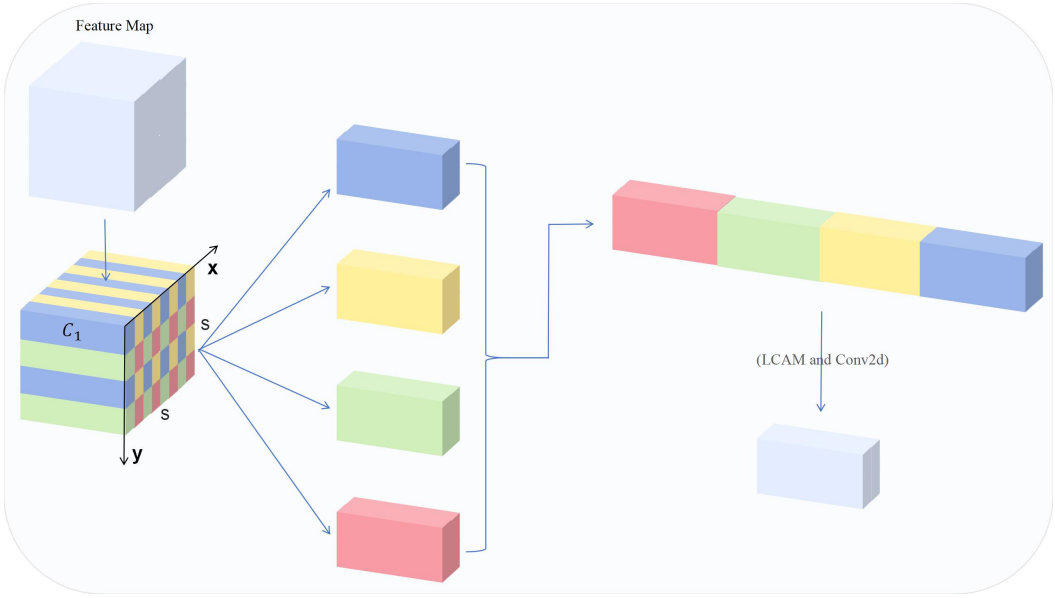

Figure 3. Framework of the downsampling module.

$$
\begin{aligned}
&f_{0,0} = X[0:S:scale, 0:S:scale],\\
&f_{1,0} = X[1:S:scale, 0:S:scale],\\
&f_{scale-1,0} = X[scale-1:S:scale, 0:S:scale];\\
&f_{0,1} = X[1:S:scale, 0:S:scale],\\
&f_{1,1} = ...,\\
&f_{scale-1,1} = X[scale-1:S:scale, 1:S:scale];\\
&f_{0,scale-1} = X[0:S:scale, scale-1:S:scale],\\
&f_{1,scale-1} = ...,\\
&f_{scale-1,scale-1} = X[scale-1:S:scale, scale-1:S:scale];
\end{aligned}
$$

3.4.2. Lightweight Channel Attention Module

On feature maps, each channel corresponds to the extraction result of a specialized feature extractor. However, not all channels have features of the same importance. Each channel needs to be assigned a weight. Generally, for an input, its channel weight is determined based on Equation (5), where MLP represents a multi-layer perceptron with shared weights, and GAP and GMP represent the global average pooling and global max pooling, respectively.

To downsample efficiently, the channel attention module must be lightweight. Therefore, an MLP was designed with a single hidden layer containing only one neuron, and pointwise convolution layers were used instead of fully connected layers. The structure of this module is illustrated in Figure 4.

$$W = \sigma\{MLP[GAP(x)] \oplus MLP[GMP(x)]\} \tag{5}$$

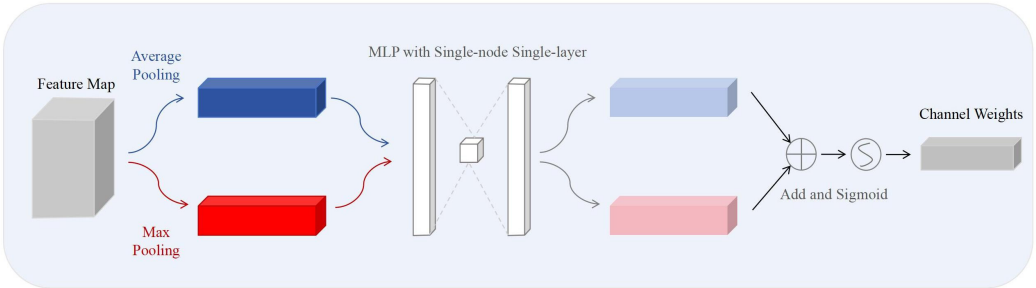

Figure 4. Framework of the LCAM.

3.4.3. Our Downsampling Module

Based on the two preceding modules, we have designed an efficient downsampling module, as detailed in Figure 3. For the input feature map, first, its size is reduced to 1/4 of its original while quadrupling the depth (i.e., the number of channels) using the SPD module, as detailed in the previous section. Subsequently, the LCAM assigns weights to each channel of the feature map, as illustrated in Figure 4. Finally, a convolution block with a kernel size of $(1, 1)$ is utilized to restore the depth of the feature map to its original size.

As shown in Figure 1, we have integrated our downsampling module into our model. Specifically, it is designed to efficiently downsample the feature map during the encoding process, thereby replacing the max pooling layer used in the original U-Net model. Notably, in Figure 1, we have represented the downsampling process with the light green arrows.

3.5. Hybrid Loss Function

In deep learning segmentation tasks, the loss function is used to measure the discrepancy between the predicted segmentation result and the ground truth mask. Its purpose is to guide the model to effectively learn from the data during training, enabling the model to predict segmentation results that closely resemble the true masks. To achieve efficient and accurate defect segmentation, an effective loss function is crucial. Binary Cross-Entropy (BCE) is widely used as an efficient loss function in computer vision tasks such as image classification [32]. Its expression is as follows:

$$L_{BCE} = -\sum -(1 - t_i) log(1 - \hat{p}_i) + t_i log \hat{p}_i \tag{6}$$

where p_i and t_i represent the predicted value and the true value, respectively.

In contrast to most segmentation tasks, a scratch defect pixel area accounts for a very low proportion of a full image's pixels. The segmentation of scratch defects on a metal surface presents a typical foreground–background class imbalance challenge. The loss for the entire sample is expressed as follows:

$$Loss = \frac{\sum_{i=1}^{N} L(y_i, \hat{p}_i)}{N} \tag{7}$$

For the binary classification of metal scratches, the loss expression is presented as follows:

$$Loss = \frac{\sum_{i=1}^{m} -log(\hat{p}) + \sum_{i=0}^{n} -log(1 - \hat{p})}{N} \tag{8}$$

where $N = m + n$ represents the total number of samples, with m and n denoting the numbers of positive and negative samples, respectively. When $m \ll n$ (namely, when negative samples dominate), this indicates that the model tends to learn from the class with more samples, severely impacting its ability to recognize the class with fewer samples.

As such, the BCE is severely affected by class imbalance issues. Therefore, we adopted the focal loss function, as displayed in Equation (9), to train the model for improved classification performance.

$$L_{focal} = -(1 - p_t)^{\gamma} log(p_t) \tag{9}$$

where $p_t = \begin{cases} \hat{p} & , if\ y = 1 \\ 1 - \hat{p} & , otherwise \end{cases}$

In Equation (9), when $\gamma > 0$, samples difficult to identify are weighted and treated as hyperparameters. When γ is set to 0, L_{focal} is degraded into the original CE (Cross-Entropy) loss function. $(1 - p_t)^{\gamma}$ can reduce the weights of samples that are easy to identify and increase the weights of samples that are difficult to identify, thereby addressing class imbalance issues [2].

For a segmentation model, it is necessary to evaluate the similarity between the areas segmented by the model and the true scratch areas, in addition to measuring the classification effect of the model using a loss function. Hence, we introduced the D_{ice} loss function to assess the similarity between the areas segmented by the model and the ground truth masks [3], thus supervising the segmentation network's learning effect. The D_{ice} loss function is defined as follows:

$$L_{D_{ice}} = \frac{2\sum_{i=1}^{N} p_i t_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} t_i^2} \tag{10}$$

where N is the total number of pixels, p_i is the value of the i-th pixel in the result segmented by the model, and t_i is the value of the i-th pixel in the ground truth scratch mask.

To train the model to segment metal scratches more efficiently, we introduced a hybrid loss function that combines the characteristics of both the focal and D_{ice} loss functions. By weighting and summing these two losses, their respective strengths were effectively leveraged. The hybrid loss function was finalized and expressed as follows:

$$Total\ Loss = \alpha L_{focal} + L_{D_{ice}} \tag{11}$$

where α serves as a hyperparameter and is used to balance the focal loss and D_{ice} loss. As such, the hybrid loss function possesses the sensitivity of BCE towards pixel misclassification, the sensitivity of D_{ice} loss towards the discrepancy between segmented results and true scratch areas, and the feature of focal loss to weight samples that are difficult to classify.

4. Experiment and Analysis

4.1. Experimental Environment

The proposed network was implemented using PyTorch 1.13.1 and the OpenCV 4.8.1.78 extension library. The GPU utilized was the NVIDIA GeForce RTX 3050 Laptop, and the CPU was the AMD Ryzen 7 5800H with Radeon Graphics. The initial settings were as follows: a learning rate of 0.01, a decay of 0.001, a momentum of 0.88, a batch size of 32, and a maximum of 150 iterations. Additionally, the weight α for the focal function in the hybrid loss function was set to 0.5, and the γ factor in the focal loss function was set to 2.0.

4.2. Dataset

Our work aims to enhance the segmentation capabilities of the model with respect to fine and micro targets. In this study, MSDD-UNet is employed to detect scratches on metal surfaces, addressing issues such as the inherent foreground–background imbalance in the scratch segmentation dataset. To this end, we specifically created a dataset for segmenting fine and micro scratches. In total, we collected 1575 metal surface images and extracted the sections containing scratch defects (see Figure 5 for a sample of these images).

As shown in Figure 5, the dataset comprises images of scratches on various metal surfaces captured under different lighting conditions. The scratches exhibit a range of shapes and colours and are predominantly elongated, with some shorter ones as well. Recognizing and segmenting scratches in such a dataset proves challenging for the model, as it faces several difficulties:

(1) The scratches vary in colour and shape, and the surface features of different metals are quite different.

(2) The images contain elongated scratches, which makes it difficult for the model to accurately segment the tips of these targets.

(3) The dataset includes various metals with diverse surface textures. These textures may lead the model to misidentify them as scratches, causing disturbances.

(4) The scratch segmentation dataset inherently suffers from an imbalance between foreground and background, with scratch pixels constituting a relatively small proportion of the overall image.

After preprocessing, the images were uniformly resized to $96 \times 96 \times 3$. We used LabelMe to annotate scratch defects and generate corresponding mask labels. The dataset was split into a training set and a test set at an 8:2 ratio, with the training set comprising 1260 images and the test set comprising 315 images. Before initiating training, we randomly subjected the training set to horizontal, vertical, and combined horizontal and vertical flips to augment the data.

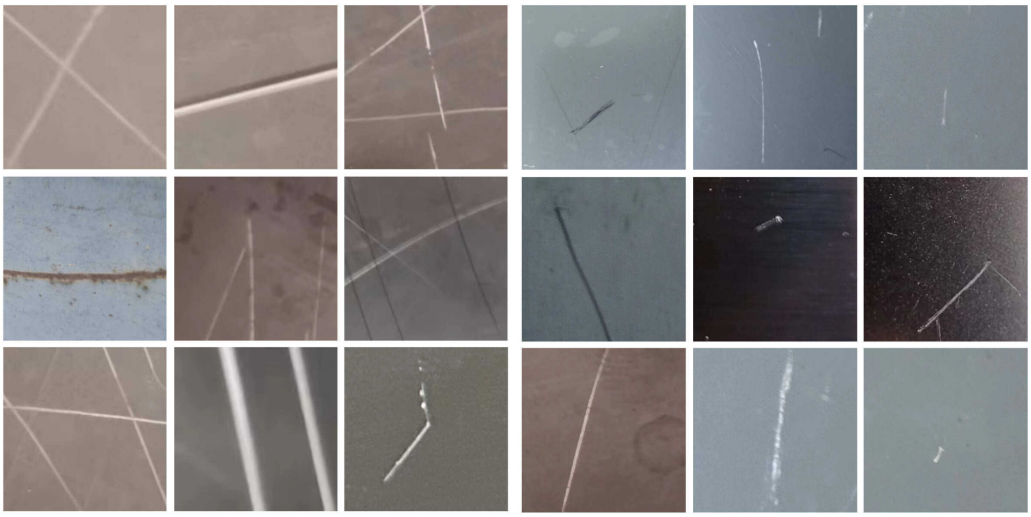

Figure 5. Examples of metal surface defects. This figure displays some of the metal surfaces and scratches of different shapes and colours present in the dataset. Notably, we have specifically highlighted some shorter scratches alongside some elongated ones. Additionally, some metallic surface textures can also be observed in the figure.

4.3. Indicator Assessment

During this assessment, the following four types of samples may be generated:

(1) TP (True Positive): this is used to represent the pixels that are labelled as scratch defects and detected as scratch defects by the model.
(2) TN (True Negative): this is used to represent the pixels that are labelled as non-scratch defects and detected as non-scratch defects by the model.
(3) FP (False Positive): this is used to represent the pixels that are labelled as non-scratch defects but detected as scratch defects by the model, indicating a false detection.
(4) FN (False Negative): this is used to represent the pixels that are labelled as scratch defects but detected as non-scratch defects by the model, indicating a missed detection.

In the experiments, we utilized the Intersection over Union (IoU) as the primary performance indicator to evaluate the proposed method. The calculation formula for the IoU is shown below:

$$IoU = \frac{TP}{TP + FP + FN} \tag{12}$$

In addition, the model assessment also involved the following three indicators for supplementary evaluations:

(1) Precision: This refers to the proportion of samples labelled as scratch defects among all scratch defect samples segmented by the model. A higher precision value indicates a more precise and reliable segmentation of scratch defects by the model. The expression of this indicator is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

(2) Recall: This refers to the proportion of pixels detected as scratch defects by the model among the pixels labelled as scratch defects in the samples. A higher recall value indicates that scratch defects were segmented more comprehensively by the model. The formula for this metric is provided below:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

(3) F_1 Score: This takes into account both the precision and recall of the model, defined as their harmonic mean. The calculation formula is as follows:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{15}$$

4.4. Ablation Experiment

To validate the effects of the loss functions and modules used in this method on the U-Net model, ablation experiments were conducted. The experimental setup and results are listed in Table 1.

Table 1. By comparing precision, recall, F1 score, and IoU, we demonstrate the benefits of each step in our model improvement.

	Ablations			**Metrics**			
Experiment	**Hybrid Loss**	**Our Downsampling**	**Our Attention**	**Precision**	**Recall**	**F_1 Score**	**IoU**
NO.1				0.8332	0.8990	0.8649	0.7501
NO.2	✓			0.8587	0.8642	0.8614	0.7665
NO.3	✓	✓		0.8776	0.8796	0.8786	0.7934
NO.4	✓		✓	0.8717	0.9012	0.8862	0.8027
NO.5	✓	✓	✓	0.8740	0.9046	0.8890	0.8039

In Experiment 1, the original U-Net model was trained on the metal scratch dataset 150 times. In Experiment 2, the proposed hybrid loss function was introduced in addition to the steps of Experiment 1. Comparatively, Experiment 2 showed a slight improvement in both precision and IoU. Notably, the IoU increased by 1.64%.

Based on Experiment 2, Experiment 3 incorporated our downsampling module. This module was based on SPD, fundamentally mitigating the loss of edge information caused by convolutions and pooling operations. Subsequently, a lightweight channel attention module computed the channel weights, followed by convolutions with a kernel size of 1×1 for the final processing. According to the experimental data, Experiment 3 maintained the recall as much as possible while improving the precision. Therefore, when compared to Experiment 1 and Experiment 2, the model from Experiment 3 achieved a higher F1 score, reaching 0.8786. Compared to Experiment 2, which only added the loss function, Experiment 3 achieved a further 2.69% increase in the IoU.

In Experiment 4, we integrated our attention mechanism into the skip connections between the encoder and decoder, in addition to the procedures outlined in Experiment 2. Based on frequency decomposition and lightweight self-attention, this approach largely preserved useful and detailed information and encoded global features implicitly. Compared to Experiments 1 and 2, Experiment 4 not only improved the precision and IoU but also exhibited a significant increase in recall. In terms of F1 score, it reached above 0.88 for the first time, specifically 0.8862. Meanwhile, the IoU was improved by 3.62% from that obtained in Experiment 2.

In Experiment 5, both the downsampling module and attention module were incorporated into the model while the steps of Experiment 2 were conducted. As a result, the model ensured high precision while achieving the highest defect recall among the five experiments, reaching the highest F1 score. Moreover, the IoU achieved in Experiment 5

reached the highest among all five experiments, presenting a 3.74% improvement from Experiment 2 and a 5.38% improvement from Experiment 1. These results demonstrate that our proposed method is an effective algorithm for enhancing the U-Net.

4.5. Contrast Experiment

4.5.1. Contrast Experiment on Scratch Data

To further validate the effects of the U-Net improvement in our proposed method on segmenting metal scratch defects, we compared this model with several common semantic segmentation models. The experimental results are provided in Table 2, with parameters, Multiply-Adds (MAdds), and floating-point operations (FLOPs) included as additional assessment indicators.

Table 2. This table presents a comparison of four selected models from the U-Net series and our model, demonstrating the superiority of our model in terms of its computational efficiency and IoU, among other factors.

Model	Params	MAdds	FLOPs	IoU
U-Net	1.97 G	3.93 G	1.97 G	0.7501
U-Net++ [33]	4.87 G	9.72 G	4.87 G	0.7834
U-Net with CBAM	8,599,178	5.03 G	2.52 G	0.7829
U-Net with Self-Attention	8,850,850	5.03 G	2.52 G	0.7940
Ours	8,502,210	4.95 G	2.48 G	0.8039

As shown in Table 2, our proposed model improved the IoU by 5.38% compared to the classic U-Net model, by 2.05% compared to the dense connection-based U-Net++, by 2.10% compared to the U-Net model fused with a Convolutional Block Attention Module (CBAM), and by 0.99% compared to the U-Net model fused with traditional self-attention mechanisms. Additionally, our proposed model was not only lightweight but also surpassed the U-Net model with traditional self-attention alone in its scratch segmentation IoU.

The IoU–epoch curves and loss–epoch curves generated by each model during training are illustrated in Figure 6. It is evident that our model consistently outperformed U-Net and U-Net++ in IoU throughout the training process. Due to foreground–background class imbalance and similar issues, all models exhibited low IoU values at the beginning of their training, but the loss–epoch curves show the faster convergence of our model. The U-Net fused with traditional self-attention began to overfit after the IoU reached its peak. However, our model incorporated a series of design improvements such as lightweighting the self-attention mechanism. Additionally, in the loss function, we applied weighted adjustments to hard-to-train samples. As a result, our model did not exhibit significant signs of overfitting in the later stages of training.

Figure 7(1) depicts the precision-recall curves of each model. As observed, our model yielded a higher precision in detecting defects than the other four models at a high recall. At a high recall, our model not only maintained a high precision but also experienced less precision decay than the other models, which is obvious in Figure 7(1). The experimental results combining Figure 7(1) and Table 2 indicate that the proposed improved U-Net model can achieve a higher IoU, recall, and precision.

Figure 7(2) provides the IoU–epoch curves generated during the training of our proposed model using BCE, D_{ice} loss, and our hybrid loss function, respectively. Notably, the model trained with the hybrid loss function achieved the best training performance in terms of the IoU. The BCE focuses solely on pixel classification errors, regardless of the discrepancy between the model's segmentation results and the true scratches. Therefore, it is necessary to introduce the D_{ice} loss function [2]. Additionally, as implied in Figure 7(2) and Table 1, it is essential to apply weighting to the difficult-to-classify samples when training the model, even though the D_{ice} function is less susceptible to foreground–background class imbalance.

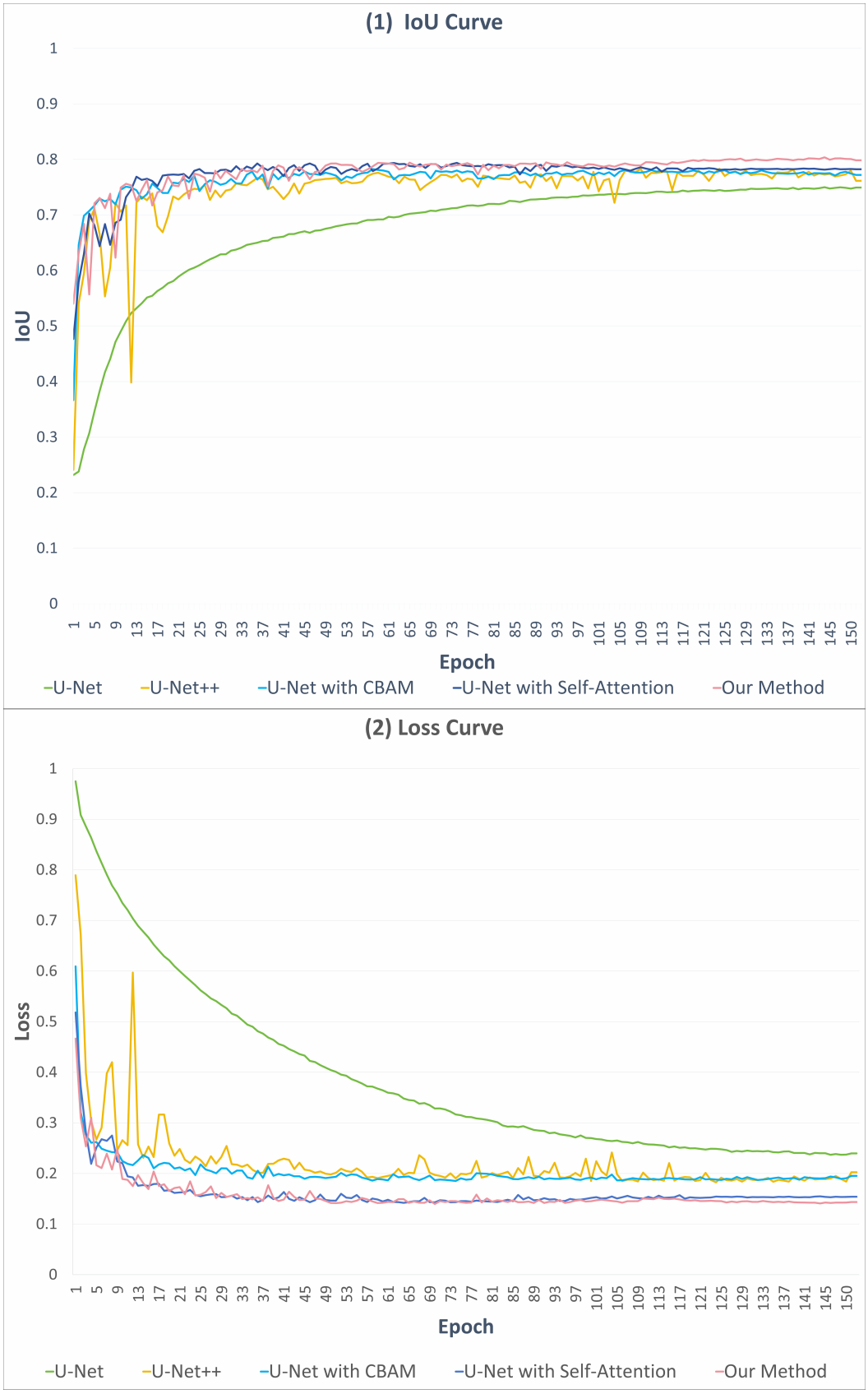

Figure 6. IoU and loss curve.

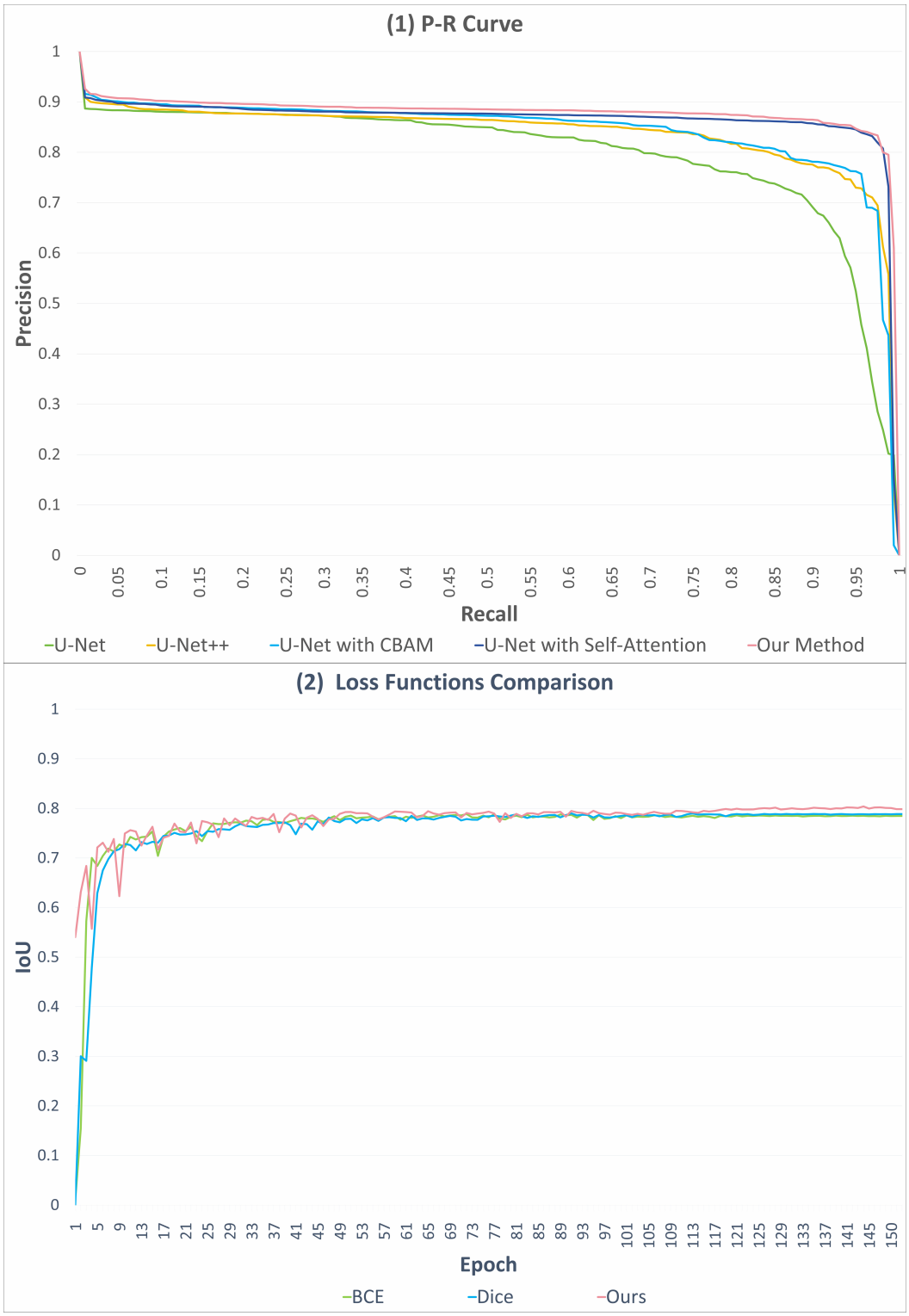

Figure 7. P-R curve and comparison of loss functions.

4.5.2. Contrast Experiment on Hot-Rolled Steel Strip Surface Defect Data

To further validate the segmentation capability of our model, we compared it with five of the most commonly used models in image segmentation. We utilized the NEU-Seg dataset [34] to evaluate the segmentation abilities of each model. The NEU-Seg dataset is a public segmentation dataset of surface defects in hot-rolled steel strips [34]. Compared to our dataset, the NEU-Seg dataset features a wider variety of targets, enabling us to assess whether our model's ability to segment targets extends beyond just scratches. This dataset

comprises three types of defects, with 300 examples of each type given. In our experiment, we did not differentiate among the three types of defects. That is to say, we considered the three types of defects as 1, and non-defective areas as 0.

We selected UNet, UNet++, DeepLabv3+ [35], and PSPNet [36] to conduct a comparative experiment with our proposed model. Prior to the experiment, the models underwent a 100-epoch pretraining phase utilizing our dataset of metal surface scratch defects. The experimental results are shown in Table 3. Notably, DeepLabv3+$_{(1)}$ denotes the use of MobileNetV2 [37] as its backbone, whereas DeepLabv3+$_{(2)}$ denotes the use of ResNet101 [38] as its backbone. As shown in Table 3, our model achieves the highest IoU compared to other segmentation methods, without requiring extensive hardware resources or high computational power. The segmentation IoU of our model is significantly higher than that of UNet and UNet++. Compared with DeepLabv3+$_{(1)}$, which uses MobileNetV2 as its backbone, our model achieves a 4.98% higher IoU score. In comparison to DeepLabv3+$_{(2)}$, which has ResNet101 as its backbone, our model consumes fewer hardware resources, has lower computational complexity, and achieves a higher IoU. When compared to PSPNet, our model utilizes 70% of its hardware resources and only 1/3 of its computation, resulting in a segmentation IoU that is 1.14 percent higher.

Table 3. Based on the NEU-Seg dataset, we compared our model to the five most commonly used models to further verify the advantages of our model in terms of computational efficiency and IoU. This table shows the results of the experiment.

Model	Memory	FLOPs	IoU
U-Net	28.09 MB	1.97 G	0.6991
U-Net++	56.78 MB	4.87 G	0.7082
DeepLabv3+$_{(1)}$	22.10 MB	629.65 M	0.7097
DeepLabv3+$_{(2)}$	35.18 MB	2.79 G	0.7392
PSPNet	48.16 MB	7.14 G	0.7481
Ours	34.45 MB	2.48 G	0.7595

To sum up, our proposed MSDD-UNet demonstrated a competitive performance in metal scratch segmentation and other similar segmentation tasks. The proposed downsampling module, attention mechanism, and loss functions can also be utilized in similar segmentation tasks, facilitating the development of an end-to-end segmentation network.

5. Conclusions

On account of metal surface scratch datasets containing class imbalance, this paper proposes an MSDD-UNet-based scratch defect localization algorithm for pixel-level segmentation. Extensive analyses and contrast experiments validated that our proposed segmentation network can accurately identify and localize scratch defects that contain class imbalance. The main contributions of this paper are as follows: (1) a downsampling mode based on an SPD module and LCAM is proposed to overcome the loss of contextual information after multiple convolutions and pooling operations; (2) based on image frequency decomposition and a lightweight self-attention module, the attention module is improved and then integrated into the skip connections to improve the model's precision in localizing scratch defects; (3) to address the foreground–background class imbalance problem in scratches, a hybrid loss function integrating the advantages of both the focal and D_{ice} functions is introduced to realize precise segmentation training.

Author Contributions: Y.L.: methodology and research design; Y.Q.: resources; Z.L.: verification and manuscript writing; H.X.: resources; C.W.: verification and software. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article. The data supporting the findings of this study are available upon request from the first author, [Liu], upon reasonable request.

Acknowledgments: Throughout the writing of this dissertation, I have received a great deal of support and assistance. First and foremost, I would like to express my gratitude to my two supervisors, Qin and Xia, whose expertise was invaluable in formulating my research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would particularly like to acknowledge my team members, Lin and Wang, for their exceptional collaboration and patient support. Additionally, I would like to thank my families for their wise counsel and sympathetic ear. Last but not least, I could not have completed this dissertation without the support of my friends, who provided stimulating discussions as well as happy distractions to rest my mind outside of my research. You are always there for me.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2022.
2. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
3. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016.
4. Metals Engineering Institute. *Fundamentals of Nondestructive Testing: Lessons 10–15[M]*; American Society of Metals: Newbury, OH, USA, 1972.
5. Ono, H.; Ogawa, A.; Yamasaki, T.; Koshihara, T.; Kodama, T.; Iizuka, Y.; Oshige, T. Twin-illumination and subtraction technique for detection of concave and convex defects on steel pipes in hot condition. *ISIJ Int.* **2019**, *59*, 1820–1827. [CrossRef]
6. Liu, H.W.; Lan, Y.Y.; Lee, H.W.; Liu, D.K. Steel surface in-line inspection using machine vision. *First Int. Workshop Pattern Recognit.* **2016**, *10011*, 187–191.
7. Yang, J.; Li, X.; Xu, J.; Cao, Y.; Zhang, Y.; Wang, L.; Jiang, S. Development of an optical defect inspection algorithm based on an active contour model for large steel roller surfaces. *Appl. Opt.* **2018**, *57*, 2490–2498. [CrossRef] [PubMed]
8. Yi, L.; Li, G.; Jiang, M. An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *Steel Res. Int.* **2017**, *88*, 1600068. [CrossRef]
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
10. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
11. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Proceedings, part III 18; Springer International Publishing: Cham, Switzerland, 2015.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
13. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef]
14. Yan, Z.; Shi, B.; Sun, L.; Xiao, J. Surface defect detection of aluminum alloy welds with 3D depth image and 2D gray image. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 741–752. [CrossRef]
15. Moosavian, A.; Bagheri, E.; Yazdanijoo, A.; Barshooi, A.H. An Improved U-Net Image Segmentation Network for Crankshaft Surface Defect Detection. In Proceedings of the 2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP), Tehran, Iran, 6–7 March 2024.
16. He, Z.; Zhao, J.; Zhao, X. Scratch Defects Detection of Curved Metal Surface based on Multiple High Frequency Projection and Inverse Transfer Learning. *IEEE Trans. Instrum. Meas.* **2024**. [CrossRef]
17. Lema, D.G.; Usamentiaga, R.; García, D.F. Enhancing automated inspection in metal industries: Zero-shot segmentation of surface defects using bounding box prompts. *Meas. Sci. Technol.* **2024**, *35*, 085604. [CrossRef]
18. Song, Y.; Xia, W.; Li, Y.; Li, H.; Yuan, M.; Zhang, Q. AnomalySeg: Deep Learning-Based Fast Anomaly Segmentation Approach for Surface Defect Detection. *Electronics* **2024**, *13*, 284. [CrossRef]
19. Arafin, P.; Billah, A.M.; Issa, A. Deep learning-based concrete defects classification and detection using semantic segmentation. *Struct. Health Monit.* **2024**, *23*, 383–409. [CrossRef] [PubMed]

20. Ardiyanto, I. Edge devices-oriented surface defect segmentation by GhostNet Fusion Block and Global Auxiliary Layer. *J. Real-Time Image Process.* **2024**, *21*, 13. [CrossRef]
21. Kong, D.; Hu, X.; Gong, Z.; Zhang, D. Segmentation of void defects in X-ray images of chip solder joints based on PCB-DeepLabV3 algorithm. *Sci. Rep.* **2024**, *14*, 11925. [CrossRef] [PubMed]
22. Feng, H.; Song, K.; Cui, W.; Zhang, Y.; Yan, Y. Cross position aggregation network for few-shot strip steel surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [CrossRef]
23. Zhou, Z.; Yan, L.; Zhang, J.; Zheng, Y.; Gong, C.; Yang, H.; Deng, E. Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement. *Constr. Build. Mater.* **2023**, *387*, 131621. [CrossRef]
24. Kumar, D.D.; Fang, C.; Zheng, Y.; Gao, Y. Semi-supervised transfer learning-based automatic weld defect detection and visual inspection. *Eng. Struct.* **2023**, *292*, 116580. [CrossRef]
25. Zhao, D.; Wang, C.; Gao, Y.; Shi, Z.; Xie, F. Semantic Segmentation of Remote Sensing Image Based on Regional Self-Attention Mechanism. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 8010305. [CrossRef]
26. Ghiasi, G.; Lin, T.Y.; Le, Q.V. DropBlock: A regularization method for convolutional networks. *Neural Inf. Process. Syst.* **2018**, *31*, 10750–10760.
27. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Feng, J. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
28. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020.
29. Koziarski, M.; Cyganek, B. Impact of low resolution on image recognition with deep neural networks: An experimental study. *Int. J. Appl. Math. Comput. Sci.* **2018**, *28*, 735–744. [CrossRef]
30. Chen, Z.; Xu, Q.; Cong, R.; Huang, Q. Global context-aware progressive aggregation network for salient object detection. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 10599–10606. [CrossRef]
31. Zhao, J.X.; Liu, J.J.; Fan, D.P.; Cao, Y.; Yang, J.; Cheng, M.M. EGNet: Edge Guidance Network for Salient Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8778–8787. [CrossRef]
32. Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
33. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Proceedings of the 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, 20 September 2018*; Proceedings 4; Springer International Publishing: Cham, Switzerland, 2018.
34. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
35. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
36. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
37. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.