*electronics*

MDPI

*Article*

# Steel Surface Defect Detection Algorithm Based on YOLOv8

**Xuan Song [1], Shuzhen Cao [1], Jingwei Zhang [2,*] and Zhenguo Hou [3,*]**

[1] School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, China; songxuan@zzu.edu.cn (X.S.); caoshuzhen2021@gs.zzu.edu.cn (S.C.)
[2] School of Civil Engineering, Zhengzhou University, Zhengzhou 450001, China
[3] China Construction Seventh Engineering Bureau Co., Ltd., Zhengzhou 450000, China
* Correspondence: zhangjingwei@zzu.edu.cn (J.Z.); houzhenguo@cscec.com (Z.H.)

**Abstract:** To improve the accuracy of steel surface defect detection, an improved model of multi-directional optimization based on the YOLOv8 algorithm was proposed in this study. First, we innovate the CSP Bottleneck with the two convolutions (C2F) module in YOLOv8 by introducing deformable convolution (DCN) technology to enhance the learning and expression ability of complex texture and irregular shape defect features. Secondly, the advanced Bidirectional Feature Pyramid Network (BiFPN) structure is adopted to realize the weight distribution learning of input features of different scales in the feature fusion stage, allowing for more effective integration of multi-level feature information. Next, the BiFormer attention mechanism is embedded in the backbone network, allowing the model to adaptively allocate attention based on target features, such as flexibly and efficiently skipping non-critical areas, and focusing on identifying potentially defective parts. Finally, we adjusted the loss function from Complete-Intersection over Union (CIoU) to Wise-IoUv3 (WIoUv3) and used its dynamic non-monotony focusing property to effectively solve the problem of overfitting the low quality target bounding box. The experimental results show that the mean Average Precision (mAP) of the improved model in the task of steel surface defect detection reaches 84.8%, which depicts a significant improvement of 6.9% compared with the original YOLO8 model. The improved model can quickly and accurately locate and classify all kinds of steel surface defects in practical applications and meet the needs of steel defect detection in industrial production.

**Keywords:** attention mechanism; feature fusion; machine vision; steel surface defect detection



**Citation:** Song, X.; Cao, S.; Zhang, J.; Hou, Z. Steel Surface Defect Detection Algorithm Based on YOLOv8. *Electronics* **2024**, *13*, 988. https://doi.org/10.3390/electronics13050988

Academic Editor: Beiwen Li

Received: 25 January 2024
Revised: 3 March 2024
Accepted: 4 March 2024
Published: 5 March 2024



## 1. Introduction

Steel, as a crucial foundational material, plays a key role in the development of the national economy [1]. It is extensively used in various sectors, including construction [2], manufacturing, transportation, energy, and more. Steel quality directly affects engineering safety and economic efficiency. However, surface defects produced during the production of steel (such as Crazing, Inclusion, Patches, Pitted surface, roll-in Scale, Scratches, etc. [3]) are likely to cause safety hazards. Therefore, accurate and efficient detection of steel surface defects is crucial. Detection methods are mainly divided into three categories: manual detection, traditional photoelectric detection, and advanced machine vision detection [4]. Among them, although manual detection is intuitive, it is limited by factors such as high labor cost and significant differences in subjective judgment, resulting in low accuracy and efficiency. Traditional photoelectric detection methods include eddy current testing [5], magnetic leakage testing [6], infrared testing [7], and laser scanning detection [8]. However, due to their excessive costs, these methods have not been widely adopted.

In recent years, with the advancement of technologies like machine vision, the detection of surface defects in steel has evolved towards automation and artificial intelligence. Utilizing high-resolution cameras, advanced image processing algorithms, and deep learning models, automated detection and classification of surface defects in steel can be achieved. These technologies enhance the accuracy and efficiency of detection while

reducing the influence of human factors. Numerous researchers have explored both traditional machine learning and deep learning techniques for steel defect detection. Traditional machine learning often involves research on feature extraction methods. Park et al. [9] proposed a two-stage statistical-based detection approach using statistical methods for feature processing, followed by classification using Support Vector Machines (SVM) to identify defect objects, demonstrating the detection of most defects. Xu et al. [10] employed multi-scale geometric analysis to decompose images into various detail levels, calculating statistical features for each sub-band to transform them into high-dimensional feature vectors. To reduce dimensionality and extract key features, a graph embedding algorithm was used to dimensionally reduce high-dimensional feature vectors, followed by SVM for steel defect classification. Yun et al. [11] utilized discrete wavelet transform for image processing, optimizing extracted features and locating defects using dynamic programming. Hu et al. [12] proposed a method using BP neural networks and SVM for steel surface defect detection. When binarily processing images of defective steel to extract features and determine defect types, experimental comparisons showed higher classification accuracy for the SVM model, while the BP neural network exhibited faster recognition speeds. Liu et al. [13] introduced a defect classification method using an ensemble of Extreme Learning Machines (ELM) based on locally binary patterns to capture local texture and structural information for feature extraction. ELMs were trained as individual models, and their outputs were aggregated to determine the final classification decision. Ashour et al. [14] proposed a feature extraction method combining Discrete Shearlet Transform (DST) with a Gray-Level Co-occurrence Matrix (GLCM). Firstly, the Shearlet transform is employed to capture texture and edge information from the image. After this, GLCM is applied to the transformed subbands, followed by principal component analysis to reduce the dimensionality of the obtained high-dimensional features. The defect classification is performed using a supervised SVM.

In recent years, deep learning has become increasingly popular, as, unlike traditional machine learning methods, it can learn more complex patterns in large datasets. Deep learning has been used to solve various problems in different fields, such as Unmanned Aerial Vehicle data processing, climate change prediction, and environmental analysis [15–18]. Therefore, deep learning is a powerful tool that has the potential to address many of the world's most pressing problems. As a result, many researchers now utilize deep learning for steel defect detection. Soukup et al. [19] proposed using Convolutional Neural Networks (CNN) training under the fully supervised strategy to improve the detection performance of steel surface defects and further improve the efficiency through regularization method. Yi et al. [20] proposed a method combining symmetrical surround saliency maps with CNN. Symmetrical surround saliency maps segment defect areas, followed by deep CNN for defect recognition. This end-to-end defect recognition approach avoids the separation of feature extraction and image classification present in traditional methods, thereby enhancing detection efficiency and accuracy. Damacharla et al. [21] used the transfer learning strategy and applied ResNet and DenseNet encoders on the basis of the U-NET model to improve the accuracy of the steel defect detection. He et al. [22] introduced an improved Fast R-CNN network model that integrates multi-scale feature maps from different network layers, minimizing feature loss and improving steel defect detection. Uraon et al. [23] presented an FPN+Resnet network model for multi-defect detection in complex backgrounds. Bouguettaya et al. [24] proposed a network model combining MobileNet-V2 and Xception with transfer learning. The strategy of deep ensemble learning was employed to retain the advantage of fast execution while addressing potential issues related to model size in traditional deep learning methods. Akhyar et al. [25] integrated deformable convolution and deformable RoI pooling techniques into the cascaded R-CNN architecture, enhancing the model's adaptability to changes in target shapes. A guided anchor proposal strategy directed the network's attention towards regions that might contain targets. Furthermore, the introduction of random scaling and ultimate scaling techniques aided the model in more accurate target processing. Lan et al. [26] improved the CasMVS Net using a three-dimensional reconstruction network and introduced multi-scale feature enhancement.

Extracting features at various scales and effectively fusing them improved accuracy. Point cloud processing techniques were combined to locate and identify surface defects on steel plates more precisely. Xia et al. [27] proposed an improved YOLOv5s model. A large core C3 module that can be reparametrized is designed innovatively, which enhances the model's ability to perceive and extract features effectively in complex texture environments. The training strategy of convolution kernel of different sizes corresponding to feature maps of different scales is adopted to adapt to defects of different shapes. Raj et al. [28] proposed the YOLOV7-CSF model, which introduced a lightweight and low-cost coordinate attention mechanism into the head structure of YOLOv7, then adopted SCYLLA-Intersection over Union loss function to improve detection efficiency. Huang et al. [29] proposed the WFE-YOLOv8s model based on YOLOv8s, replacing the original C2F module with a new CFN structure, reducing the number of network parameters and GFLOPs, and improving the algorithm accuracy through an EMA attention mechanism, which increased by 4.7 percentage points compared with the mAP of the original model.

The steel defect detection algorithms mentioned above, although excellent in terms of innovation, have certain limitations that cannot be ignored. Firstly, when faced with a variety of different steel surface defects, machine learning algorithms rely on traditional manual feature extraction, which is relatively weak in generalization, and it is difficult to flexibly respond to variable defect types. Second, although CNN performs well in multiple tasks, it may have shortcomings in global feature extraction, which in turn limits its ability to effectively identify small, detail-rich defects. Furthermore, in the industrial environment with limited computing resources, the existing complex network architecture may not be conducive to real-time monitoring and online feedback due to the slow computing speed, which affects the operating efficiency of the entire production line and product quality control. To solve the above problem, this paper proposes an improved YOLO8 algorithm, which aims to improve the accuracy of steel surface defect detection on the basis of ensuring real-time performance. The primary contributions of this paper are as follows:

1. Adding deformable convolutions in the backbone network to enhance the adaptability to target shapes or local structures. This improvement significantly improved mAP by about 3.2 percentage points.
2. Substituting the original model's feature fusion structure with BiFPN to capture feature information from different scales more effectively, thereby improving detection accuracy. Building on the previous step, approximately a 1.6 percentage point increase in mAP.
3. Integrating BiFormer into the backbone network for adaptive attention to targets and allocation of computational resources, leading to improved detection accuracy. The mAP again grew by about one percentage point.
4. Implementing WIoUv3 to enhance the accuracy of predicting bounding boxes for targets and their generalization. The mAP gained another 1.1 percentage points.

## 2. YOLOv8 Algorithm Introduction

Faster R-CNN, RetinaNet, SSD, and YOLO series algorithms are all classic target detection algorithms. Faster R-CNN is a two-stage target detection algorithm with high precision but limited speed. RetinaNet introduces Focal Loss to improve small target detection, which is fast but has room for improvement in accuracy. SSD realizes real-time and efficient detection with single-stage detection and multi-scale feature mapping, but it has poor recognition accuracy for small targets. YOLOv8 inherits the features of the YOLO series, and, compared with the previous version, it has optimized the network structure and improved the loss function, the sample matching strategy, and the training strategy. The following will introduce the improvement of each part in detail. After the above key optimization measures, YOLOv8 not only greatly improves the reasoning speed but also improves the detection accuracy to some extent. YOLOv8 currently has YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, and other versions, which are mainly different in model size, parameter number, computing resource requirements, and

detection accuracy. The model size is arranged according to the above positions. These releases are designed to meet the needs of different tasks, hardware, and performance. Embedded devices or edge computing platforms in industrial environments often have strict limits on computing resources, while the YOLOv8n model is small, requires relatively low memory and computing power, and is suitable for deployment to hardware with limited resources. Although the performance of YOLOv8 largely depends on the quality and diversity of training data, it is still a relatively good choice for the application scenarios of steel defect detection, so this paper chooses to improve on YOLOv8n. The network architecture of YOLOv8 is depicted in Figure 1.

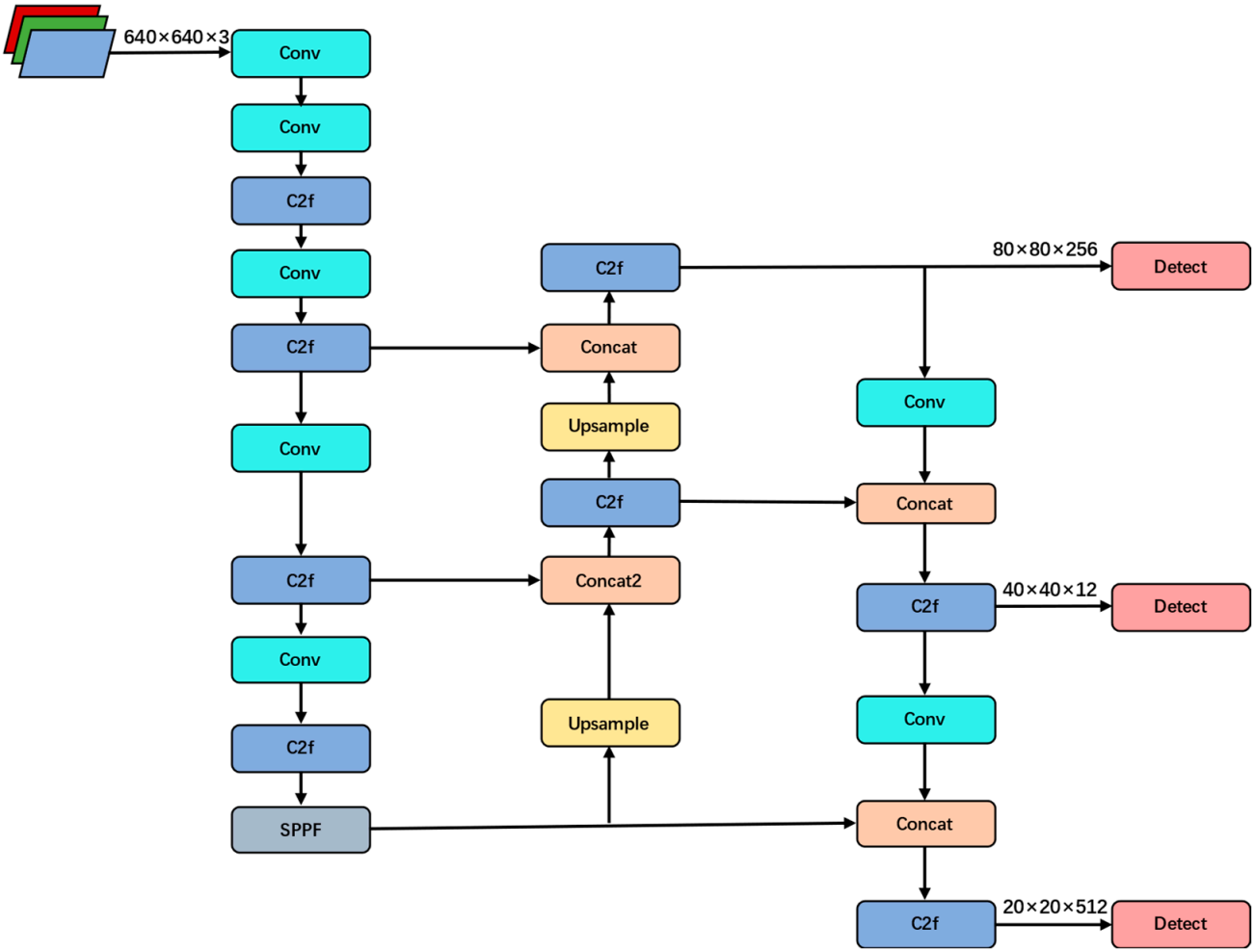


**Figure 1.** Network structure diagram of YOLOv8 algorithm.

The YOLOv8 algorithm consists of four main components: Input, Backbone, Neck, and Head. Mosaic was used for data enhancement at the input side, which increased the diversity of the training set, effectively reduced the overfitting phenomenon, and improved the generalization ability of the model in new scenarios. To prevent excessive data enhancement and excessive data augmentation, this operation is disabled in the last 10 epochs of training. The Backbone module, responsible for feature extraction, includes modules such as C2F and Spatial Pyramid Pooling Fusion (SPPF). C2F comprises convolutional layers, split operations, and multiple Bottleneck units. The design of the C2F module is inspired by the advanced concept of Efficient Layer Aggregation Network (ELAN), an innovative architecture designed to improve the computational efficiency and performance of networks in the fields of computer vision and deep learning. At the same time, the C2F module also draws on the design characteristics of C3 and effectively collects multi-level feature information by building a connection mechanism between multiple branches,

therefore obtaining more accurate and rich gradient signals. This design shows a good effect on the small target detection task, because it can alleviate the situation of weak features and difficult localization of small targets, improving the detection accuracy and stability of the model. SPPF dynamically adjusts the size of the input image through max pooling operations, ensuring consistent feature sizes and increasing receptive fields. It also enables the fusion of local and global features to help the network learn multiple layers of semantic information.

The Neck module utilizes the Path Aggregation Feature Pyramid Network (PAFPN) structure for feature fusion. This network combines the Feature Pyramid Network (FPN) [30] and the Path Aggregation Network (PANet) [31]. FPN employs a multi-level feature pyramid network in a top-down manner to fuse low-level and high-level image features, producing feature maps at different scales. PANet introduces an additional pathway in a bottom-up manner after extracting features from the original input image, significantly reducing the computational cost of feature propagation. By combining FPN and PANet, the feature maps are adaptively fused, addressing the issue of scale differences to improve the detection performance of various sizes of targets and enhancing the network's feature representation capability. The Head module adopts the decoupled head paradigm [32], removing the previous objectness branch and retaining separate classification and regression branches for predicting class labels and bounding boxes, respectively. This approach enables better focus on category and boundary information.

## 3. Improved YOLOv8 Algorithm

The improved network structure diagram is shown in Figure 2. Firstly, C2f_DCNv2, combined with deformable convolution, is used to replace the C2f in the original network, which enhances the ability of the model to capture complex shapes and irregular target features. Then, PAFPN is adjusted to BiFPN, which improves the accuracy of the model in dealing with defects of various sizes, especially in identifying small or complex surface defects. Secondly, the BiFormer attention mechanism is introduced to enhance the model's focus on defect details within images. Finally, the original CIoU is replaced with WIoUv3 to improve the accuracy of bounding box regression. By improving the above four aspects of YOLOv8, the robustness and accuracy of the model for defect type, size, and location changes are improved.

### *3.1. Introducing C2F_DCNv2*

Deformable convolution differs from traditional convolution in that the convolutional positions are not fixed and can be adjusted adaptively [33]. It introduces learnable offsets on top of traditional convolution. Since the offsets can be fractional, bilinear interpolation is used to calculate the positions of pixels and obtain their corresponding feature values. The output formula for deformable convolution is as follows.

$$\mathrm{y}(p_0) = \sum_{p_n \in R} w(p_n) \bullet x(p_0 + p_n + \Delta \mathrm{p_n}) \tag{1}$$

where $w(p_n)$ represents the weight values of the convolutional kernel at $p_n$ position, $x(p_0 + p_n + \Delta \mathrm{p_n})$ represents the feature value of the feature map at $p_0 + p_n + \Delta \mathrm{p_n}$ position, and represents the offset added on top of $p_0 + p_n$ the position.

Using deformable convolution allows for covering objects at different scales, enhancing the model's feature representation capability, and improving detection performance. However, the introduction of offset may lead to covering irrelevant regions, thereby disturbing feature extraction and degrading the overall performance. To address this issue, a team from the University of Science and Technology of China proposed an upgraded version of deformable convolution known as DCNv2 [34]. DCNv2 introduces a modulation mechanism by incorporating a modulation parameter $\Delta \mathrm{m_k} \in [0, 1]$, which learns the weights of sampling points. For uninteresting regions, the weight coefficient $\Delta \mathrm{m_k}$ is assigned a small value. Although additional learning parameters are required, this technique

is worth introducing due to the improvements in model generalization and performance. The output formula for DCNv2 is as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \bullet x(p_0 + p_n + \Delta p_n) \bullet \Delta m_k \quad (2)$$

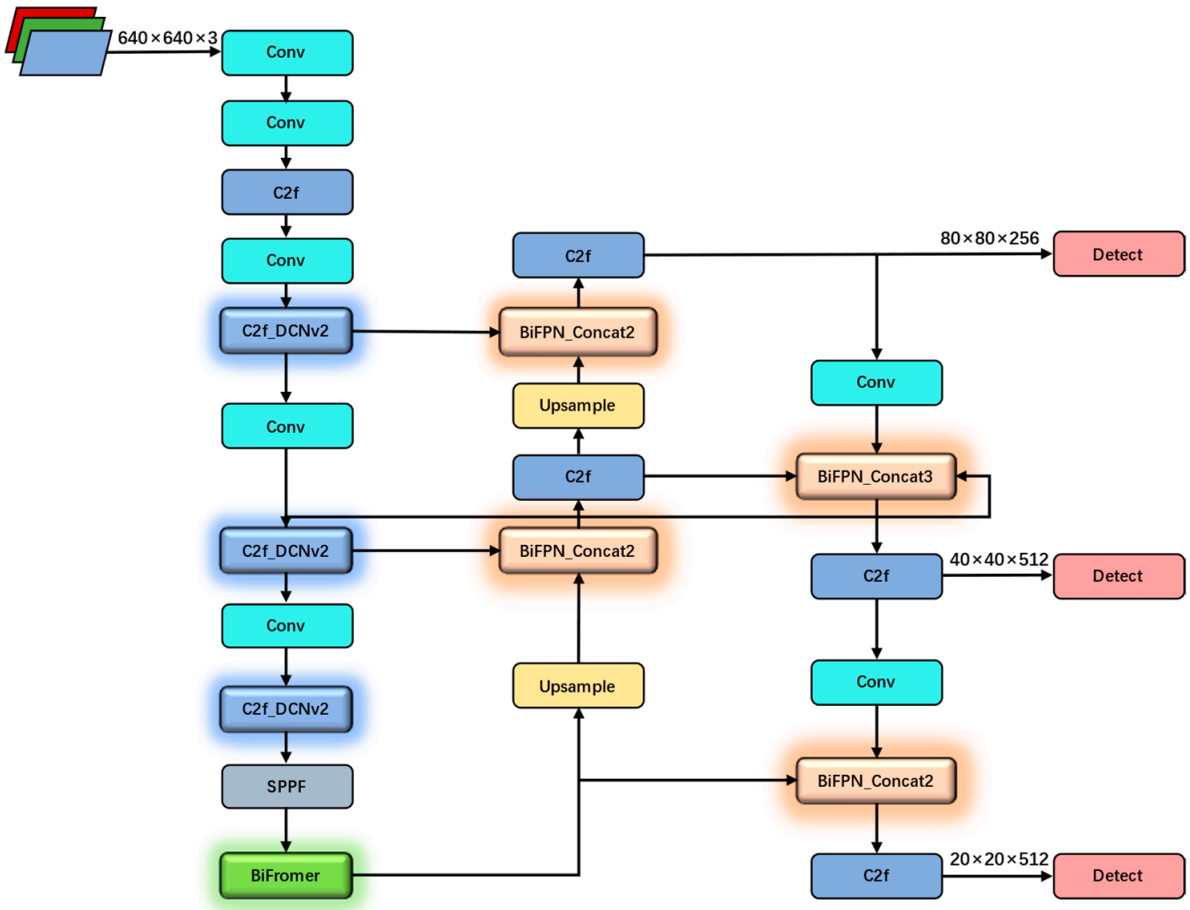


**Figure 2.** Improved network structure diagram.

In this paper, the idea of DCNv2 is integrated into the C2F module of YOLOv8, resulting in the proposed C2F_DCNv2 layer. The backbone network is primarily responsible for extracting low-level features and global information from the original image, and these feature details are crucial for comprehending the overall context of the entire image. Therefore, the last three C2F modules in the backbone network are replaced with C2F_DCNv2 modules. The introduction of DCNv2 allows for an increased effective receptive field and the sampling of effective locations. The Conv operation in the Bottleneck of the C2F module is replaced with DCNv2. After the introduction of DCNv2 in YOLOv8, the model can more accurately capture the detailed information of the boundary and complex shape of the target object, especially for the detection of objects with changeable morphology. The structures of DCNv2 and the modified Bottleneck are illustrated in Figure 3 and Figure 4, respectively.

### *3.2. Bi-Directional Feature Pyramid Network*

In order to better fuse multi-scale features, the PAFPN in YOLOv8 is replaced with the BiFPN [35]. Compared to other structures, BiFPN can effectively fuse features without increasing computational cost. The main idea behind BiFPN is to construct a feature pyramid by utilizing information flow from both the bottom-up and top-down directions while employing a repeated weighted fusion approach at each pyramid level. By leveraging information flow from both directions, BiFPN can fuse features at various levels to better

accommodate objects of assorted sizes. Through the repeated weighted fusion process, BiFPN enhances the accuracy and generalization capability of the model, thus improving object detection performance. Compared to the PAFPN feature fusion network, BiFPN has the following differences: ① Removal of unidirectional input nodes (these nodes do not participate in cross-level feature fusion, and their impact on the overall network performance is relatively small, so removing them simplifies the network structure). ② The original input node and output node of the same layer are connected, so that the feature map of the layer can be better retained and utilized in the process of feature fusion. This can enhance the information transmission and fusion ability of the same layer feature map and improve the perception and recognition ability of the target. ③ By repeating the process, the network gradually fuses features from more levels, resulting in a more comprehensive and semantically expressive final feature representation. ④ Instead of simple feature map stacking or addition, as in traditional fusion methods, BiFPN uses weighted feature fusion. Since features may have different semantic information and resolutions, their fusion is performed with distinct weights to ensure a more comprehensive and accurate feature representation. Due to its complex connection patterns, an accurate training strategy needs to be designed. By combining BiFPN to form BiFPN_Concat2 and BiFPN_Concat3 modules, setting learnable parameters and learning weights for different branches, Concat operations are applied separately to feature maps for two-branch and three-branch configurations. The structural diagrams of PAFPN and BiFPN are shown in Figure 5 and Figure 6, respectively.

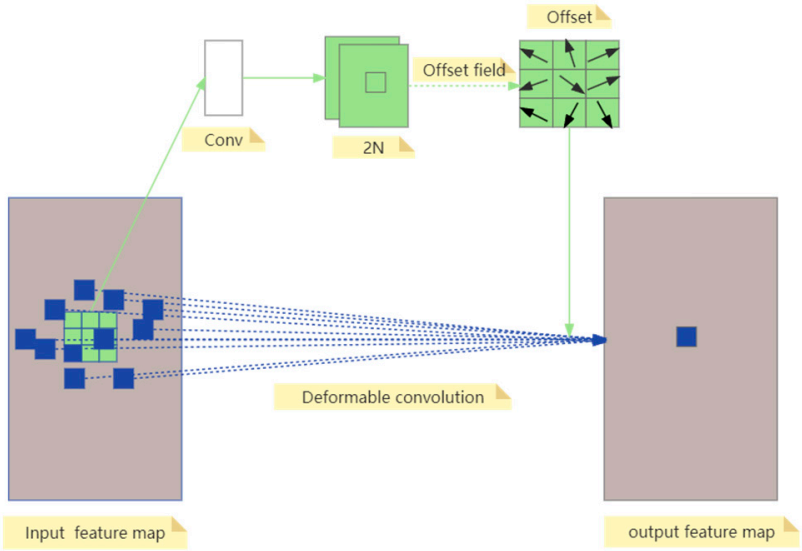


**Figure 3.** Network structure of the DCNv2.

### *3.3. Biformer Attention Mechanism*

To improve the feature extraction capability of YOLOv8 for small objects, the BiFormer attention mechanism [36] is introduced at the end of the Backbone. In the YOLOv8 model, the backbone serves as the feature extraction component and the neck serves as the feature fusion part; therefore, adding the BiFormer attention mechanism at the end of the backbone can better extract feature information from the input images. BiFormer utilizes sparse sampling to preserve fine-grained feature information, enabling better feature representation on smaller feature maps and addressing the low accuracy issue in small object detection. The core component of BiFormer is the BiFormerBlock, which consists of three parts: Region partition and input projection, region-to-region routing with directed graph, and token-to-token attention. First, an input feature map is linearly projected to obtain the query (Q), key (K), and value (V). The calculation formulas for Q, K, and V are as follows:

$$\mathrm{Q} = \mathrm{X}^{\mathrm{r}}\mathrm{W}^{\mathrm{q}}\ \mathrm{K} = \mathrm{X}^{\mathrm{r}}\mathrm{W}^{k}\ \mathrm{V} = \mathrm{X}^{\mathrm{r}}\mathrm{W}^{v} \tag{3}$$

In the above formula, $X \in R^{H \times W \times C}$. The input feature map is divided into $S \times S$ regions, with each region containing $H \times W/S^2$ feature vectors. $X$ is reshaped into $X^r \in R^{S^2 \times H \times W/S^2 \times C}$, and $W^q$ $W^k$ $W^v$ represent the mapping weights for query, key, and value, respectively.

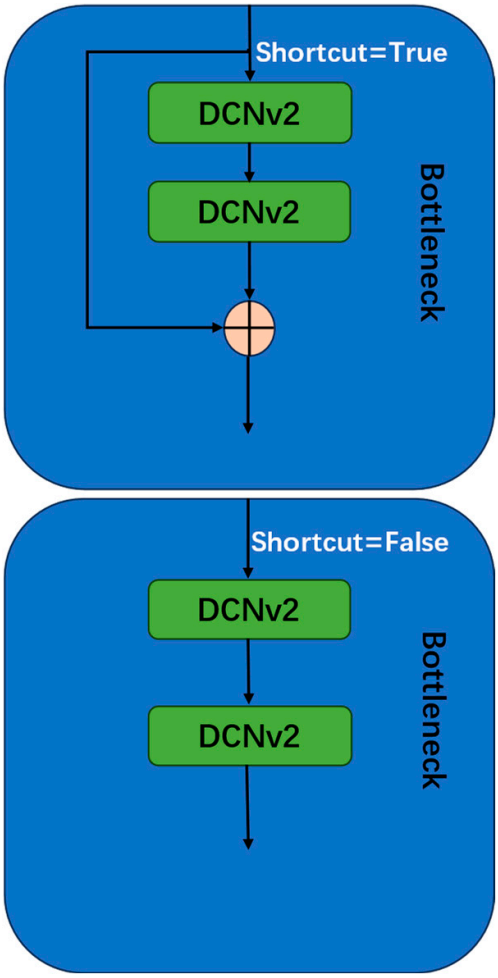


**Figure 4.** The Bottleneck network structure of this paper.

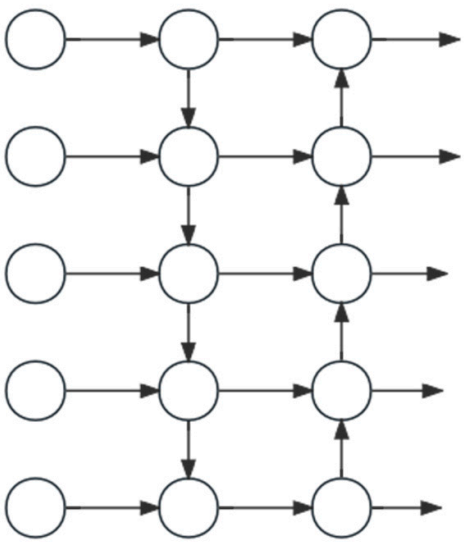

**Figure 5.** PAFPN Network structure.

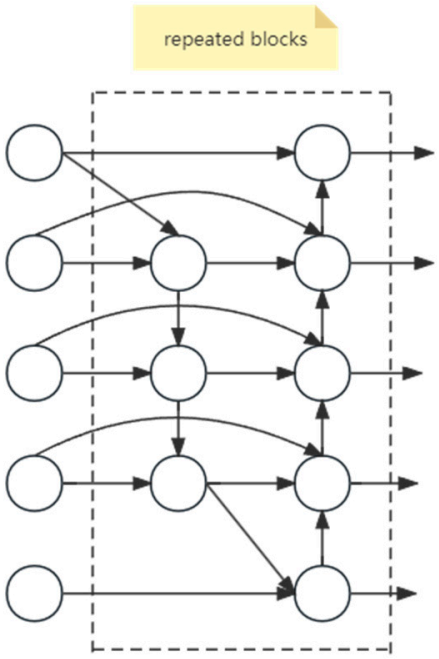


**Figure 6.** BiFPN Network structure.

Then, a directed graph is constructed using an adjacency matrix to determine which key-value pairs should be involved in the attention computation, i.e., which regions should be focused on for a given region. Next, the region-to-region routing index matrix is used to determine the remaining candidate regions, referred to as routing regions. Finally, fine-grained token-to-token attention is applied to interact with tokens in the routing regions, resulting in attention outputs. It is a dual-level routing dynamic sparse attention mechanism. Compared to traditional global attention mechanisms, this approach can filter at the coarse-grained region level and apply fine-grained token-to-token attention within the routing regions. It achieves flexibility in computation allocation and improves computational efficiency by selectively attending to relevant parts of tokens using adaptive querying, skipping irrelevant regions. BiFormer adopts a four-level pyramid structure with a downsampling factor of 32. Specifically, the first stage uses overlapping block embedding, and the second to fourth stages use block merging modules to reduce the input spatial resolution and increase the number of channels. Although BiFormer optimizes the computational efficiency, the introduction of this attention slightly increases the computational overhead of the overall algorithm. The network structure of BiFormer is shown in Figure 7.

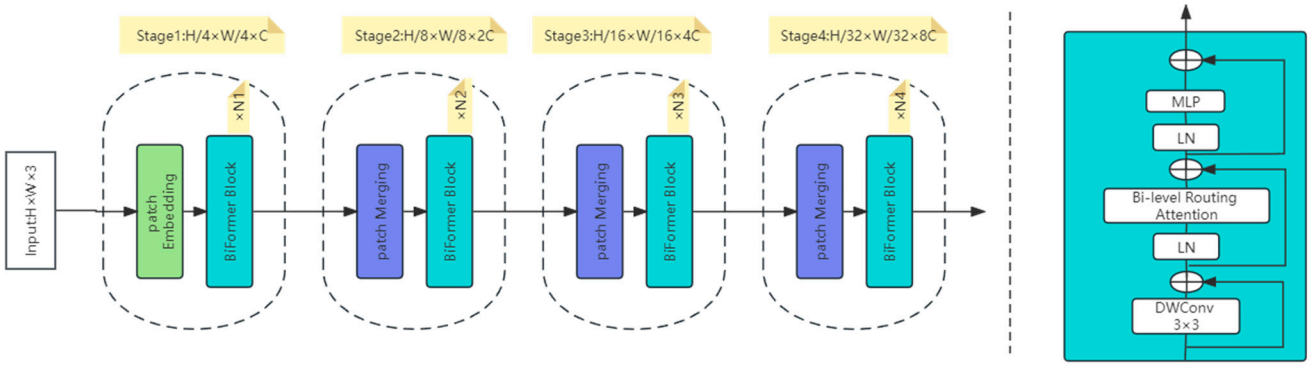


**Figure 7.** BiFormer network structure diagram.

### *3.4. Wise-IoU*

In object detection tasks, bounding boxes are commonly used to represent the position and size of the targets. Bounding boxes are usually represented by four coordinate values, i.e., the coordinates of the top-left and bottom-right corners. Traditional IoU loss function evaluates the overlap between two boxes by computing the ratio of their intersection to their union. However, the IoU loss function has some limitations. For example, it may inaccurately assess the overlap between large and small objects, it may be unfair for boxes of different scales, and it cannot distinguish cases where there is a high overlap but the target is not properly localized. In YOLOv8, to address the bounding box regression problem, the CIoU loss [37] is used as the loss function. CIoU loss is an improved loss function that considers factors such as position offset, scale difference, and aspect ratio, providing a more accurate assessment of the similarity between predicted and ground truth boxes. The loss function is defined as follows:

$$\mathcal{L}_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2\left(\mathrm{b}, \mathrm{b}^{\mathrm{gt}}\right)}{\mathrm{c}^2} + \alpha \mathrm{V} \tag{4}$$

In the CIoU loss function, b represents the center coordinates of the predicted box, $\mathrm{b}^{\mathrm{gt}}$ represents the center coordinates of the ground truth box, and $\rho$ is the Euclidean distance between the two center points, i.e., the straight-line distance between the center point of the predicted box and the center point of the ground truth box. c represents the diagonal distance of the minimum enclosing box that contains both the predicted box and the ground truth box. $\alpha$ is a positive weight parameter, and V is a parameter that measures the similarity of aspect ratios between the predicted box and the ground truth box. It is used to penalize cases where there is a significant difference in aspect ratio between the predicted box and the ground truth box.

Although the CIoU loss outperforms traditional IoU calculation in addressing issues such as bounding box offset and aspect ratio imbalance in object detection, it is mainly used to enhance the fitting capability of bounding box regression. The presence of low-quality data in the dataset, however, can lead to overfitting if the bounding box regression is overly emphasized for these low-quality examples, thereby reducing the detection performance of the model. In this paper, Wise-IoUv3 [38] is used to replace CIoU and solve this problem by using a dynamic non-monotonic focal mechanism that effectively leverages the potential of the non-monotonic focal mechanism.

WIoUv1 is an attention-based bounding box loss that constructs distance attention based on distance metrics. The formula is as follows:

$$\mathcal{L}_{\mathrm{WIoUv1}} = \mathcal{R}_{\mathrm{WIoU}} \mathcal{L}_{\mathrm{IoU}} \tag{5}$$

$$\mathcal{R}_{\mathrm{WIoU}} = \exp\left(\frac{\left(x - x_{\mathrm{gt}}\right)^2 + \left(\mathrm{y} - \mathrm{y}_{\mathrm{gt}}\right)^2}{\left(\mathrm{W}_{\mathrm{g}}^2 + \mathrm{H}_{\mathrm{g}}^2\right)^*}\right) \tag{6}$$

In the formula, $(x, y)$ and $(x_{gt}, y_{gt})$ represent the center coordinates of the ground truth box and the predicted box, respectively. $W_g, H_g$ represent the widths and the heights of the minimum enclosing region that simultaneously contains both the ground truth and predicted boxes.

Based on WIoUv1, WIoUv3 uses gradient gain as the focusing coefficient and non-monotonic dynamic focusing coefficient to consider the allocation of loss function so that the model pays more attention to those samples that are difficult to accurately match the target, thereby enhancing detection accuracy and robustness. The computational formula for WIoUv3 is as follows.

$$\beta = \left(\frac{\mathcal{L}_{\mathrm{IoU}}^*}{\overline{\mathcal{L}_{\mathrm{IoU}}}}\right)^{\gamma} \in [0, +\infty) \tag{7}$$

$$\mathcal{L}_{\mathrm{WIoUv3}} = \mathrm{r}\mathcal{L}_{\mathrm{WIoUv1}} \tag{8}$$

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{9}$$

where $\overline{\mathcal{L}_{IoU}}$ represents the sliding average of the momentum m and $\delta$ is a variable parameter when $\beta = \delta$, $r = 1$. A small outlier $\beta$ usually means that the anchor frame matches the real target to a higher degree, so the anchor frame quality is relatively good. Assigning a smaller gradient gain to this type of high-quality anchor frame during training ensures that the optimization process pays more attention to those anchors of ordinary quality, thus making the bounding frame regression more accurate. For anchors with large outliers, their matching degree with the actual marked frames is often low, indicating that their quality is poor. At this point, a small gradient gain is assigned to prevent these low-quality examples from producing excessive harmful gradients in the process of backpropagation, affecting the overall model convergence and optimization efficiency. In terms of computational principles, replacing CIoU with Wise-IoUv3 in YOLOv8 can reduce harmful gradients while maintaining attention to these samples, which is beneficial for better learning of the model and improving its localization performance.

## 4. Experimental Results and Analysis

### *4.1. Experimental Environment and Dataset*

The experimental environment used in this study employed the Windows 10 operating system with 32 GB of memory. The computer's CPU was an i5-12400F with a clock frequency of 2.50 GHz, while the GPU was an NVIDIA GeForce RTX 3090. Python version 3.8.16 was utilized, along with the PyTorch 2.0.0 deep learning framework and CUDA 11.7 for accelerated computations.

The dataset used in the experiment is NEU-DET, which is a steel surface defect detection dataset created by the team led by Professor Ke-Chen Song from Northeastern University. This dataset consists of six different types of defects on steel plates, including Crazing (fine cracks or fractures on the surface of the steel plate), Inclusion (impurities or foreign substances present on the surface of the steel plate), Patches (large patches or uneven areas on the surface of the steel plate), Pitted_surface (small pits or corrosion spots on the surface of the steel plate), Rolled-in_scale (presence of oxidation or rolled-in scales on the surface of the steel plate), and Scratches (scratches or scrapes on the surface of the steel plate). The dataset contains a total of 1800 images of steel plate surface defects, with 300 samples for each type of defect. Each image is 200 $\times$ 200 pixels in size and is provided as a grayscale image. The defect areas in the image are marked to facilitate the training and evaluation of defect detection. The NEU-DET data set is divided into training set, test set and verification set according to the ratio of 8:1:1. The training set contains 1440 images, the test set contains 180 images, and the verification set contains 180 images. Such a partition ratio can maintain the diversity and representative of the data set while providing a sufficient sample size for training, testing, and verifying the performance of the algorithm.

### *4.2. Evaluation Metrics*

In this paper, precision (P), recall (R), mAP@0.5, GFLOPS, Params and FPS are used as evaluation indicators. The accuracy rate P is expressed as the proportion of the model that is actually true in the predicted true class sample. The recall rate R represents the proportion of all actual true class samples that the model successfully predicts to be true. Meanwhile, mAP represents mean average precision, and mAP@0.5 means that the IoU threshold is set to 0.5 during mAP calculation. That is, the detection frame is correct only when the IoU between the detection frame and the real target is greater than 0.5 (this is a common threshold setting for object detection tasks). GFLOPS refers to the number of floating point operations performed by the model per second, Params refers to the number of parameters in the model, and Params can evaluate the complexity and scale of the model. FPS stands for frames per second and represents the number of frames a model is capable

of in regard to inferences or predictive operations per unit of time. The calculation formula of accuracy P, recall R and mAP is as follows.

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

$$\text{mAP} = \frac{1}{\text{c}} \sum_{\text{i}=1}^{\text{c}} \text{AP}_\text{i} \tag{12}$$

In the above formula, *TP* represents the true case, *FP* represents the false positive case, and *FN* represents the false negative case.

### *4.3. Ablation Experiments*

To evaluate and validate the effectiveness of the proposed improvements, five ablation experiments were conducted. Under consistent environments and training parameters, training or testing was performed using experimental groups and control groups, and the corresponding results were recorded. The results of the ablation experiments are presented in Table 1.

**Table 1.** Improved point ablation experiment.

| Sequence | C2F_DCNv2 | BiFPN | BiFormer | WIoUv3 | Params (M) | GFLOPs | mAP@0.5 | FPS |
|---|---|---|---|---|---|---|---|---|
| 1 | - | - | - | - | 3.0 | 8.1 | 0.779 | 200.0 |
| 2 | √ | - | - | - | 3.1 | 7.6 | 0.812 | 186.3 |
| 3 | √ | √ | - | - | 3.1 | 8.3 | 0.828 | 178.0 |
| 4 | √ | √ | √ | - | 3.4 | 18.5 | 0.837 | 153.5 |
| 5 | √ | √ | √ | √ | 3.4 | 18.5 | 0.848 | 142.8 |

The first group demonstrated the original, unmodified YOLOv8 algorithm with a mAP value of 0.779 on the steel surface defect detection task. In the second group, the C2F_DCNv2 module is introduced to replace the original C2F structure. This change makes the model more flexible to deal with complex shapes and irregular target features, thus improving the detection accuracy. The experimental results show that the mAP value rises to 0.812. On this basis, the third group further replaced the PAFPN structure with the more efficient BIFPN for feature fusion. This change helps the model to better integrate multi-scale feature information, especially when identifying small or complex surface defects, showing advantages. The experimental results show that the mAP value is further increased to 0.828. Subsequently, the fourth group added the BiFormer attention mechanism to enhance the correlation between the model's understanding of the global image structure and local details. Although this resulted in an increase in Params and GFLOPs, the experimental results proved that the improvement effectively improved the detection performance, and the mAP value reached 0.837. Finally, the fifth group applied the WIoUv3 loss function to the optimized network structure to guide the bounding box regression process more accurately. The final experimental results showed that the cumulative effect of these improvements was significant, and the mAP value was increased to 0.848. While each improvement will reduce the FPS somewhat, the fifth group has a minimum of 142.8 frames per second, which is an acceptable range that meets the real-time needs of industrial inspection. Through a series of ablation experiments, we can clearly see the positive contribution of each improvement point to the overall detection effect, thus proving the effectiveness of these improvement points.

### *4.4. Comparative Experiments*

Figure 8 shows the confusion matrix for the improved model. The horizontal axis represents the true value, and the vertical axis represents the predicted value. It can be seen that most of the predicted values correspond to the real values, so the model has a good prediction performance.

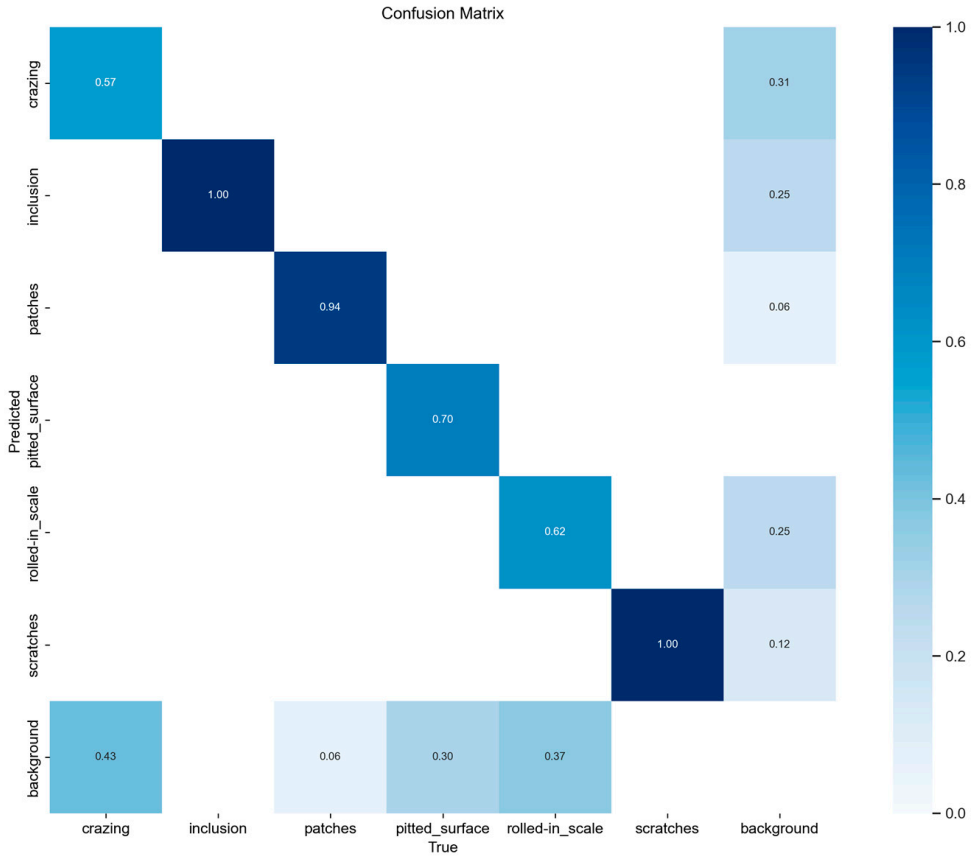


**Figure 8.** Confusion matrix graph of the improved YOLOv8 algorithm.

The PR curve graphs in Figures 9 and 10 illustrate the experimental results of YOLOv8 and the improved YOLOv8 under the same conditions. The figures display the mAP@0.5 values for each category as well as the overall mAP@0.5. From the graphs, it can be observed that the improved algorithm has increased the mAP from 77.9% to 84.8%, resulting in a 6.9 percentage point improvement. It is worth noting that, in the YOLOv8 detection, the mAP value for the "rolled-in_scale" category was only 0.465, whereas after the improvement, it reached 0.716. This demonstrates a significant enhancement compared to the baseline model for this particular category.

Figure 11 shows the prediction results before and after the improvement of the Yolov8 model. By comparing Figure 11, it is evident that the improved algorithm achieves more accurate object localization and a certain increase in confidence scores.

To validate the effectiveness of the proposed algorithm, the algorithm in this paper is compared with mainstream object detection models, including SSD, Fast RCNN, DETR, YOLOv5s, YOLOv7, and YOLOv8n, on the NEU-DET dataset. The experimental results are presented in Table 2 for comparison.

Table 2 displays the values of map@0.5 and fps for different models across various defects. From Table 2, it can be observed that there is a slight difference in detection accuracy between the SSD and Fast R-CNN algorithms for steel surface defect detection. Although Fast R-CNN may have a slightly better detection performance, as a two-stage detection algorithm, it has a relatively large number of parameters and noticeably slower detection speed. DETR introduced transformer architecture for target detection. When

dealing with fine-grained steel surface defects, due to the global self-attention mechanism, it may encounter problems of high memory consumption and slow convergence speed, so the detection effect of DETER is poor. The detection effect of YOLOv5s and YOLOv7 is also unsatisfactory in the case of small defect size and dense distribution. YOLOv8 is the fastest detection algorithm with relatively good detection performance. The improved YOLOv8 algorithm demonstrates overall improvement in detection performance while only slightly decreasing the detection speed. The number of parameters is also maintained at a satisfactory level. Furthermore, it exhibits significant improvement in detecting certain challenging defect types, such as cracks and rolled-in scale. Based on the above findings, for the task of steel surface defect detection, the proposed algorithm in this paper outperforms other algorithms and performs better in completing the detection task.

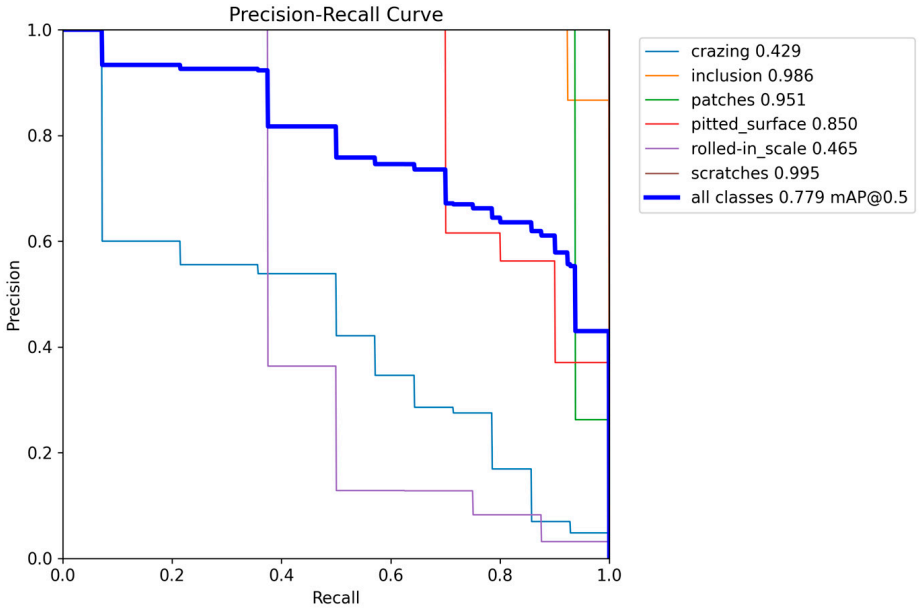


**Figure 9.** FP-R curve of YOLOv8 algorithm.

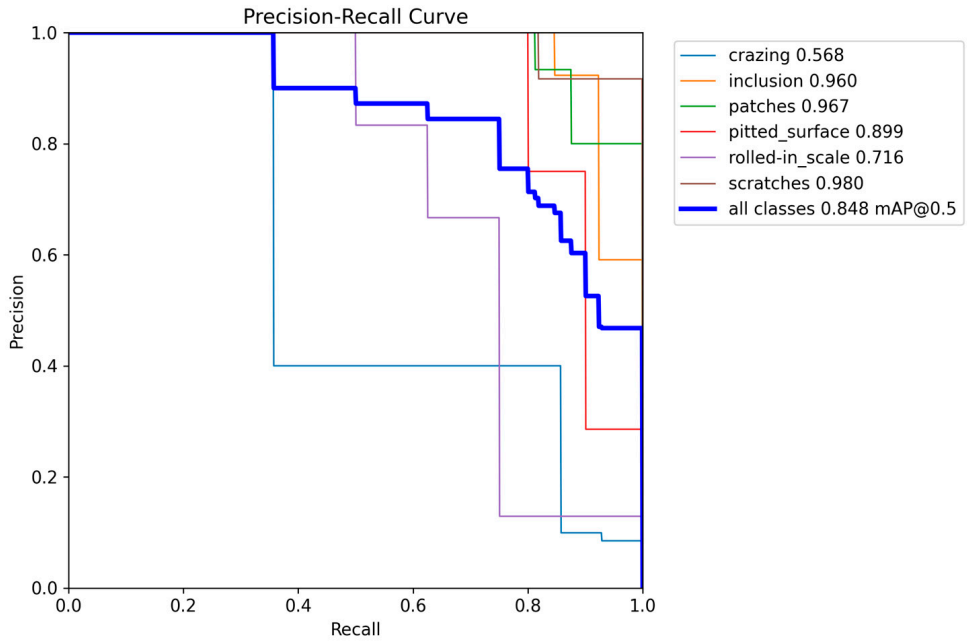


**Figure 10.** P-R curve of the improved YOLOv8 algorithm.

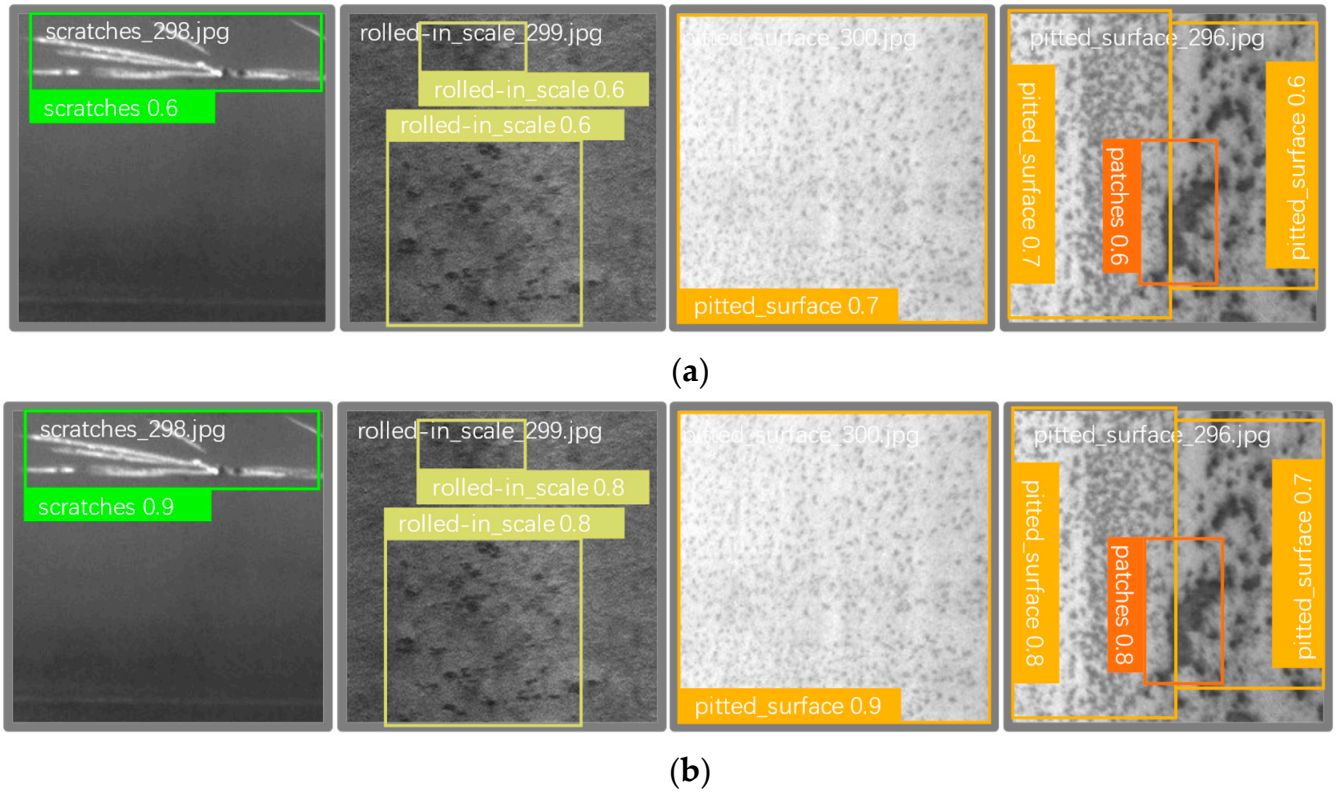


**Figure 11.** Comparison of detection results. (**a**) Detection effect of YOLOv8 model; (**b**) Improved detection effect of YOLOv8 model.

**Table 2.** Comparison of Detection Performance of Different Algorithms.

| Types | SSD | Fast RCNN | DETR | YOLOv5s | Yolov7 | Yolov8n | OURS |
|---|---|---|---|---|---|---|---|
| crazing | 0.411 | 0.421 | 0.261 | 0.201 | 0.185 | 0.429 | 0.568 |
| inclusion | 0.773 | 0.773 | 0.655 | 0.697 | 0.762 | 0.986 | 0.960 |
| patches | 0.922 | 0.919 | 0.898 | 0.931 | 0.908 | 0.951 | 0.967 |
| pitted_surface | 0.792 | 0.866 | 0.706 | 0.706 | 0.534 | 0.850 | 0.899 |
| rolled-in_scale | 0.695 | 0.654 | 0.565 | 0.397 | 0.539 | 0.465 | 0.716 |
| scratches | 0.729 | 0.969 | 0.910 | 0.875 | 0.778 | 0.995 | 0.980 |
| mAP | 0.720 | 0.767 | 0.666 | 0.635 | 0.618 | 0.779 | 0.848 |
| FPS | 142.1 | 42.4 | 28.7 | 94.8 | 70.1 | 200.0 | 142.8 |
| Params (M) | 24.3 | 136.8 | 36.7 | 7.1 | 37.2 | 3.0 | 3.4 |

## 5. Conclusions and Future Outlook

In addressing the issue of steel surface defect detection, this paper proposes an improved YOLOv8 algorithm. The algorithm replaces the convolutions in Bottleneck with DCNv2 to enlarge the receptive field and enhance feature extraction capabilities. Additionally, PAFPN is adjusted to BiFPN for improved feature fusion, thereby enhancing the accuracy of detecting various classes. The introduction of the BiFormer attention mechanism in the backbone network strengthens feature enhancement for improved detection outcomes. Moreover, CIoU is replaced with WIoUv3, employing a dynamic non-monotonic focus mechanism to concentrate on anchors of ordinary quality, thereby enhancing overall detection performance. The mAP of the improved model in this paper can reach 84.8% on the NEU-DET dataset. Compared with the WFE-Yolov8S algorithm proposed by Huang et al., mAP is about five percentage points higher. The effectiveness of the improved model was verified by ablation experiment and comparison experiment.

In this study, we focused on the specific challenge of steel surface defect detection, designing and implementing a series of optimizations for this task to significantly improve the model's accuracy in identifying small targets. Although our core work is closely focused on steel surface defects, the proposed improvement strategies do have broad prospects and universal value for cross-field applications. These strategies not only show excellent results in solving the problem of small and complex defects detection on steel surfaces, but also the core ideas and technical means behind them are also applicable to other diverse scenarios involving small target detection (for example, in fields such as electronic component defect detection and subtle lesion detection in medical image diagnosis).

Although the work in this paper has made some progress in improving mAP, it is worth pointing out that the study is conducted in a supervised way, and it would require high labor costs to obtain adequate steel defect labeling data in practical applications. In future research, we can focus on applying semi-supervised learning to the field of steel defect detection. Due to the wide variety and shape of steel surface defects, how to mine and learn effective feature representation in the absence of labels is a challenge. However, despite these challenges, the application of semi-supervised learning also presents clear advantages. On the one hand, it reduces the dependence on large-scale annotation data, thus saving the high cost of manual annotation; on the other hand, by combining labeled and unlabeled data for training, the model may improve the generalization performance of unencountered or less common defect types, enhancing the adaptability in real scenarios. This has certain research significance for the scenario of limited label resources and ever-changing defect categories in the industrial field.

**Author Contributions:** Conceptualization, X.S. and S.C.; methodology, X.S. and S.C.; software, J.Z.; validation, X.S., S.C. and J.Z.; formal analysis, Z.H.; investigation, S.C.; resources, Z.H.; data curation, X.S. and S.C.; writing—original draft preparation, X.S. and S.C.; writing—review and editing, X.S. and S.C.; visualization, X.S. and S.C.; supervision, J.Z.; project administration, Z.H.; funding acquisition, J.Z. and Z.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by Research and Development Projects of China Construction Seventh Engineering Bureau Co., Ltd. under Grant CSCEC7B-2023-Z-15.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** Zhenguo Hou was employed by the company China Construction Seventh Engineering Bureau Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Dong, H.B.O.; Liu, Y.; Wang, L.J.; Li, X.; Tian, Z.; Huang, Y.; McDonald, C. Roadmap of China steel industry in the past 70 years. *Ironmak. Steelmak.* **2019**, *46*, 922–927. [CrossRef]
2. Wang, X.; Wang, Z.; Guo, C.; Han, Y.; Zhao, J.; Lu, N.; Tang, H. Application and Prospect of New Steel Corrugated Plate Technology in Infrastructure Fields. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *741*, 012099. [CrossRef]
3. Neogi, N.; Mohanta, D.K.; Dutta, P.K. Review of vision-based steel surface inspection systems. *EURASIP J. Image Video Process.* **2014**, *2014*, 50. [CrossRef]
4. Luo, Q.W.; Fang, X.X.; Liu, L.; Yang, C.; Sun, Y. Automated Visual Defect Detection for Flat Steel Surface: A Survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644. [CrossRef]
5. Meng, X.B.; Lu, M.Y.; Yin, W.L.; Bennecer, A.; Kirk, K.J. Evaluation of Coating Thickness Using Lift-Off Insensitivity of Eddy Current Sensor. *Sensors* **2021**, *21*, 419. [CrossRef] [PubMed]
6. Wang, G.; Xiao, Q.; Gao, Z.H.; Li, W.; Jia, L.; Liang, C.; Yu, X. Multifrequency AC Magnetic Flux Leakage Testing for the Detection of Surface and Backside Defects in Thick Steel Plates. *IEEE Magn. Lett.* **2022**, *13*, 8102105. [CrossRef]
7. Jing, X.; Yang, X.-Y.; Xu, C.-H.; Chen, G.; Ge, S. Infrared thermal images detecting surface defect of steel specimen based on morphological algorithm. *J. China Univ. Pet.* **2012**, *36*, 146–150.
8. Liang, Z.; Xu, K.; Xu, J. 3D Detection Technique for Surface Defects of Steel Plates Based on Linear Laser. *J. Univ. Sci. Technol. Beijing* **2004**, *26*, 662–665.

9. Park, C.H.; Bae, H.M.; Yun, J.P.; Yun, S.W. Automated Surface Inspection System for Black Resin Coated Steel. In Proceedings of the 2012 12th International Conference on Control, Automation and Systems, Jeju Island, Republic of Korea, 17–21 October 2012; pp. 1683–1685.
10. Xu, K.; Ai, Y.H.; Wu, X.Y. Application of multi-scale feature extraction to surface defect classification of hot-rolled steels. *Int. J. Min. Metall. Mater.* **2013**, *20*, 37–41. [CrossRef]
11. Yun, J.P.; Choi, D.; Jeon, Y.; Kim, S.W. Defect inspection system for steel wire rods produced by hot rolling process. *Int. J. Adv. Manuf. Technol.* **2014**, *70*, 1625–1634. [CrossRef]
12. Hu, H.J.; Li, Y.X.; Liu, M.F.; Liang, W.H. Steel strip surface defects classification based on machine learning. *Comput. Eng. Des.* **2014**, *35*, 620–624.
13. Liu, Y.; Jin, Y.; Ma, H. Surface Defect Classification of Steels Based on Ensemble of Extreme Learning Machines. In Proceedings of the 2nd World Robot Conference (WRC)/Symposium on Advanced Robotics and Automation (WRC SARA), Beijing, China, 21–22 August 2019.
14. Ashour, M.W.; Khalid, F.; Halin, A.A.; Abdullah, L.N.; Darwish, S.H. Surface Defects Classification of Hot-Rolled Steel Strips Using Multi-directional Shearlet Features. *Arab. J. Sci. Eng.* **2019**, *44*, 2925–2932. [CrossRef]
15. Haq, M.A.; Rahaman, G.; Baral, P.; Ghosh, A. Deep Learning Based Supervised Image Classification Using UAV Images for Forest Areas Classification. *J. Indian Soc. Remote Sens.* **2021**, *49*, 601–606. [CrossRef]
16. Jawaharlalnehru, A.; Sambandham, T.; Sekar, V.; Arunnehru, J.; Loganathan, V.; Kannadasan, R.; Khan, A.A.; Wechtaisong, C.; Haq, M.A.; Alhussen, A.; et al. Target Object Detection from Unmanned Aerial Vehicle (UAV) Images Based on Improved YOLO Algorithm. *Electronics* **2022**, *11*, 2343. [CrossRef]
17. Choutri, K.; Lagha, M.; Meshoul, S.; Batouche, M.; Bouzidi, F.; Charef, W. Fire Detection and Geo-Localization Using UAV's Aerial Images and Yolo-Based Models. *Appl. Sci.* **2023**, *13*, 11548. [CrossRef]
18. Haq, M.A.; Jilani, A.K.; Prabu, P. Deep Learning Based Modeling of Groundwater Storage Change. *Comput. Mater. Contin.* **2022**, *70*, 4599–4617.
19. Soukup, D.; Huber-Mörk, R. Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images. In Proceedings of the 10th International Symposium on Visual Computing (ISVC), Las Vegas, NV, USA, 8–10 December 2014; pp. 668–677.
20. Yi, L.; Li, G.Y.; Jiang, M.M. An End-to-End Steel Strip Surface Defects Recognition System Based on Convolutional Neural Networks. *Steel Res. Int.* **2017**, *88*, 176–187. [CrossRef]
21. Damacharla, P.; Rao, A.M.V.; Ringenberg, J.; Javaid, A.Y. TLU-Net: A Deep Learning Approach for Automatic Steel Surface Defect Detection. In Proceedings of the 2021 International Conference on Applied Artificial Intelligence (ICAPAI), Halden, Norway, 19–21 May 2021; pp. 1–6.
22. He, Y.; Song, K.; Meng, Q.; Yan, Y. An End-to-End Steel Surface Defect Detection Approach Via Fusing Multiple Hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [CrossRef]
23. Uraon, P.K.; Verma, A.; Badholia, A. Steel Sheet Defect Detection using Feature Pyramid Network and RESNET. In Proceedings of the 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 13–15 October 2022; pp. 1543–1550.
24. Bouguettaya, A.; Mentouri, Z.; Zarzour, H. Deep ensemble transfer learning-based approach for classifying hot-rolled steel strips surface defects. *Int. J. Adv. Manuf. Technol.* **2023**, *125*, 5313–5322. [CrossRef]
25. Akhyar, F.; Liu, Y.; Hsu, C.Y.; Shih, T.K.; Lin, C.-Y. FDD: A deep learning-based steel defect detectors. *Int. J. Adv. Manuf. Technol.* **2023**, *126*, 1093–1107. [CrossRef] [PubMed]
26. Lan, H.; Yu, J.-B. Steel surface defect detection based on deep learning 3D reconstruction. *J. Zhejiang Univ. (Eng. Sci.)* **2023**, *57*, 466–476.
27. Xia, K.W.; Lv, Z.L.; Zhou, C.D.; Gu, G.; Zhao, Z.; Liu, K.; Li, Z. Mixed Receptive Fields Augmented YOLO with Multi-Path Spatial Pyramid Pooling for Steel Surface Defect Detection. *Sensors* **2023**, *23*, 5114. [CrossRef]
28. Raj, G.D.; Prabadevi, B. Steel Strip Quality Assurance With YOLOV7-CSF: A Coordinate Attention and SIoU Fusion Approach. *IEEE Access* **2023**, *11*, 129493–129506. [CrossRef]
29. Huang, Y.; Tan, W.Z.; Li, L.; Wu, L. WFRE-YOLOv8s: A New Type of Defect Detector for Steel Surfaces. *Coatings* **2023**, *13*, 2011. [CrossRef]
30. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
31. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
32. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
33. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. *arXiv* **2017**, arXiv:1703.06211.
34. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets v2: More Deformable, Better Results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
35. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2019; pp. 10778–10787.

36. Zhu, L.; Wang, X.; Ke, Z.; Zhang, W.; Lau, R. BiFormer: Vision Transformer with Bi-Level Routing Attention. *arXiv* **2023**, arXiv:2303.08810.
37. Zheng, Z.H.; Wang, P.; Ren, D.W.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef]
38. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.