algorithms

MDPI

Article

Optimized Adaptive Multi-Scale Architecture for Surface Defect Recognition

Xueli Chang [1,2], Yue Wang [1,2], Heping Zhang [1,2], Bogdan Adamyk [3,*] and Lingyu Yan [1,2,*]

[1] School of Computer Science, Hubei University of Technology, Wuhan 430068, China; chang99@hbut.edu.cn (X.C.); 20140106@hbut.edu.cn (Y.W.); zhanghping@hbut.edu.cn (H.Z.)
[2] Key Laboratory of Green Intelligent Computing Network in Hubei Province, Wuhan 430068, China
[3] Aston Business School, Aston University, Birmingham B4 7ET, UK
* Correspondence: b.adamyk@aston.ac.uk (B.A.); yanlingyu@hbut.edu.cn (L.Y.)

Abstract

Detection of defects on steel surface is crucial for industrial quality control. To address the issues of structural complexity, high parameter volume, and poor real-time performance in current detection models, this study proposes a lightweight model based on an improved YOLOv11. The model first reconstructs the backbone network by introducing a Reversible Connected Multi-Column Network (RevCol) to effectively preserve multi-level feature information. Second, the lightweight FasterNet is embedded into the C3k2 module, utilizing Partial Convolution (PConv) to reduce computational overhead. Additionally, a Group Convolution-driven EfficientDetect head is designed to maintain high-performance feature extraction while minimizing consumption of computational resources. Finally, a novel WISEPIoU loss function is developed by integrating WISE-IoU and POWERFUL-IoU to accelerate the model convergence and optimize the accuracy of bounding box regression. The experiments on the NEU-DET dataset demonstrate that the improved model achieves a parameter reduction of 39.1% from the baseline and computational complexity of 49.2% reduction in comparison with the baseline, with an mAP@0.5 of 0.758 and real-time performance of 91 FPS. On the DeepPCB dataset, the model exhibits reduction of parameters and computations by 39.1% and 49.2%, respectively, with mAP@0.5 = 0.985 and real-time performance of 64 FPS. The study validates that the proposed lightweight framework effectively balances accuracy and efficiency, and proves to be a practical solution for real-time defect detection in resource-constrained environments.

Keywords: steel surface defect detection; lightweight model; revcol; generalization performance

check for updates

Academic Editor: Md Baharul Islam

Received: 22 July 2025
Revised: 17 August 2025
Accepted: 18 August 2025
Published: 20 August 2025

Citation: Chang, X.; Wang, Y.; Zhang, H.; Adamyk, B.; Yan, L. Optimized Adaptive Multi-Scale Architecture for Surface Defect Recognition. *Algorithms* **2025**, *18*, 529. https://doi.org/10.3390/a18080529

1. Introduction

As an indispensable foundational material in the fields such as construction and industrial manufacturing, the quality characteristics of steel products directly affect the performance and safety reliability of final products. In modern industrial production, systematic detection of surface defects in steel materials enables the scientific evaluation of material quality. Defects such as scratches and cracks on steel surfaces significantly degrade their mechanical properties and service life. With the continuous improvement of standards of quality for industrial products, the technology of surface defect detection has become a critical component of quality control systems. Currently, methods of defect detection of steel surface are primarily split into three categories: manual visual inspection, traditional machine vision-based methods, and deep learning-based methods [1].

Manual defect inspection relies on visual observation by inspectors, but this approach has significant limitations, including high labor costs, low efficiency, and difficulty in real-time monitoring [2]. In contrast, machine vision-based detection technologies reduce labor costs while significantly improving detection accuracy and efficiency. However, it should be noted that these methods still require manual feature extraction, which may constrain system robustness and generalization capabilities [3], especially when handling large-scale defect datasets.

As industrial requirements to detection accuracy continue to rise, traditional machine vision-based methods struggle to meet practical needs, hindering the development of deep learning-based intelligent detection technologies. Jonathan et al. first applied convolutional neural networks (CNNs) to steel surface defect detection [4], demonstrating their superior automatic feature extraction and classification capabilities, as well as the applicability of deep learning in complex industrial scenarios. Current target detection algorithms are mainly divided into two categories: one-stage algorithms (e.g., YOLO series [5–11], SSD [12], DETR [13]), which offer higher detection speed but relatively lower accuracy, and two-stage algorithms (e.g., R-CNN [14], Fast R-CNN [15], Faster R-CNN [16]), which achieve higher accuracy through region proposal mechanisms but suffer from lower computational efficiency. Notably, CNN-based methods (e.g., YOLO) underperform when dealing with complex textures or strong background interference. Recent studies showed that the Transformer-based methods, leveraging their global modeling advantages, exhibit stronger feature learning capabilities. However, their massive computational demands pose significant challenges for industrial applications. Therefore, developing lightweight defect detection models that balance accuracy and computational efficiency has become a key research focus.

This study focuses on the lightweight improvement of the YOLOv11 algorithm. Although YOLOv11 demonstrates outstanding performance in detection speed, computational efficiency, model size, and accuracy, further optimization is needed for industrial defect detection scenarios with stringent real-time requirements and limited computational resources. We propose an improved lightweight steel surface defect detection model, RF-EW-YOLOv11, with the following contributions:

(1) Reversible Connected Multi-Column Network (RevCol) for Backbone Reconstruction: A multi-column reversible connection design preserves low-level texture and high-level semantic features, reducing feature loss and enhancing multi-scale defect representation.

(2) FasterNet Lightweight Module Integration: Introduces partial convolution (PConv) in the C3k2 module, reducing computational complexity to 1/16 of standard convolution and minimizing redundant calculations and memory access overhead.

(3) Group Convolution-Driven EfficientDetect Head: Reconstructs the detection head using group convolution, reducing parameters while maintaining feature extraction performance through cross-channel interaction.

(4) WISEPIoU Hybrid Loss Function Optimization: Combines the dynamic penalty mechanism of WISE-IoU and the geometric constraints of POWERFUL-IoU to accelerate convergence and improve bounding box regression accuracy, particularly for small defects in complex backgrounds.

The remainder of this paper is organized as follows: Section 2 reviews related work on defect detection algorithms. Section 3 describes the proposed RF-EW-YOLOv11 model. Section 4 presents experiments validating the model's effectiveness. Section 5 concludes the paper.

2. Related Work

2.1. Traditional Machine Vision-Based Steel Surface Defect Detection

Traditional machine vision techniques for steel surface defect detection rely on knowledge from optics, physics, and computer science. The process involves image processing, feature extraction, classification, and recognition [17]. In 2018, Xu et al. proposed RNAMlet, a novel method of adaptive multi-scale geometric analysis (MGA) for steel surface defect recognition [18]. By combining asymmetric and non-overlapping models (NAM), RNAMlet optimized feature extraction and improved classification accuracy while maintaining real-time performance. Liu et al. demonstrated that combining LBP feature extraction with ELM classification achieved the best results for cold-rolled strip surface defects [19]. In 2020, Liu et al. introduced NSST-KLPP for aluminum strip defect recognition, combining non-subsampled shearlet transform (NSST) and kernel locality preserving projections (KLPPs) [20]. Sun et al. proposed an unsupervised aluminum plate defect detection method (PPFMNBD) using multi-light illumination, addressing the scarcity of defect samples in industrial settings [21].

However, these methods still face limitations: reliance on manually designed features, sensitivity to environmental factors (e.g., lighting), and poor adaptability to diverse industrial requirements.

2.2. Deep Learning-Based Steel Surface Defect Detection

Deep learning has significantly enhanced defect detection accuracy and automation. Target detection algorithms are divided into one-stage and two-stage approaches. Cha et al. applied Faster R-CNN for structural damage detection [22]. Zhao et al. improved Faster R-CNN with deformable networks for random steel defects [23]. Yang et al. developed a lightweight CNN-based system using MobileNetV2 and CBAM attention [24]. Akhyar et al. proposed FDD based on Cascade R-CNN with deformable convolutions, achieving high accuracy but slow inference speeds [25].

2.3. YOLOv11

The iteration of YOLOv11 is based on algorithms such as YOLOv8, while introducing more innovative features to make its design more focused on balancing efficiency and practicality, aiming to solve various challenges encountered in various industries with higher accuracy and efficiency.

In terms of network architecture, YOLOv11 continues the overall framework of YOLOv8, but has made significant improvements in key modules. One of its core innovations is the introduction of a configurable C3k2 module to replace the original C2f module, significantly improving the gradient flow and computational efficiency of the network. The C3k2 module has a unique dynamic switchable characteristic, and its specific behavior can be determined by setting the parameter value of C3k to True or False. When C3k = True, the module will use the C3k structure to replace the traditional Bottleneck structure, which allows the network to perform more in-depth and complex feature extraction. When C3k = False, the module degenerates into a standard C2f structure. This switchable design strategy enables YOLOv11 to dynamically adjust the network's feature extraction capability according to different application scenarios and computing resource requirements, achieving the optimal balance between computational efficiency and feature expression capability.

In the feature fusion section, YOLOv11 introduces the C2PSA module in the Neck layer, which combines the standard C2f module with Pointwise Spatial Attention (PSA) blocks. This mechanism adaptively learns the importance weights of each spatial position in the feature map, allowing the network to more effectively focus on important regions in the image, thereby enabling more accurate detection of targets of different sizes and positions. Experiments have shown that this design enables YOLOv11 to have more robust detection performance for multi-scale targets in complex scenes while maintaining computational efficiency.

Finally, in the design of the detection head, YOLOv11 adopts the same decoupling head design as YOLOv8, that is, its Head layer also contains two independent branches, separating the classification and regression tasks. However, YOLOv11 has made more refined optimizations by replacing the two 3 × 3 convolutions in the classification detection head with two Depthwise Separable Convolution (DWConv) convolutions. This improvement decomposes the calculation of spatial correlation and channel correlation, effectively reducing redundant calculations while maintaining the performance of the model, improving the efficiency of the model, and enabling efficient deployment on resource limited mobile devices.

3. Methodology

We propose a lightweight improved YOLOv11 detection algorithm, which achieves comprehensive performance enhancement through structural optimization and module innovation. Firstly, the backbone network of YOLOv11 is reconstructed using the reversible connection multi-column network (RevCol) idea to extract more features at a lower cost. Secondly, the lightweight and fast network FasterNet is embedded into the C3k2 module of the model, further reducing the model's parameter count, volume, and computational load, without sacrificing accuracy. Additionally, an efficient detection head (EfficientDetect) is proposed, which uses group convolution to perform lightweight modifications on the original three detection heads of YOLOv11, reducing consumption of computational resources, while maintaining high-performance feature extraction capabilities of the detection heads. Finally, the WISEPIoU loss function, combining WISE-IoU and POWERFUL-IoU, is proposed to address the limitations of the original single loss function of YOLOv11. This loss function enables faster model convergence and improves detection accuracy. The improved model is referred to as RF-EW-YOLOv11, and its network model is shown in Figure 1, where C3k2-Faster is FasterNet Integrated Cross-Stage Partial Block with kernel size 2; SPPF is the Spatial Pyramid Pooling Fast module for multi-scale feature aggregation; C2PSA is the Cross Stage Partial block with Position Self-Attention for enhanced feature representation.

3.1. RevCol-Reconstructed Backbone

The current YOLOv11 backbone employs a top-down architecture, which tends to compress or discard features during layer propagation, retaining only the most relevant and necessary information from the input. This approach often leads to the loss of embedded image information during feature extraction, thereby degrading model performance. To address this issue and enable the model to preserve more comprehensive raw information, we proposes a reconstructed backbone network based on the Reversible Connected Multi-Column Network (RevCol) [26].

RevCol adopts a multi-input design, breaking through the traditional paradigm of network information feedforward transmission. Each column starts from low-level information and extracts semantic features through compressed image channels. Due to unknown downstream tasks, some useful features may be prematurely discarded during the feature extraction stage of the training process. To alleviate this issue, RevCol introduced reversible connections between columns, allowing the network to reversibly revisit column inputs. This design ensures lossless data transfer between columns without the need for additional memory storage. In addition, supervision is applied at the end of each column to constrain feature extraction, thereby decoupling low-level texture details from high-level semantic information and enhancing the model's adaptability to multi-scale targets. The specific architecture is shown in Figure 2.

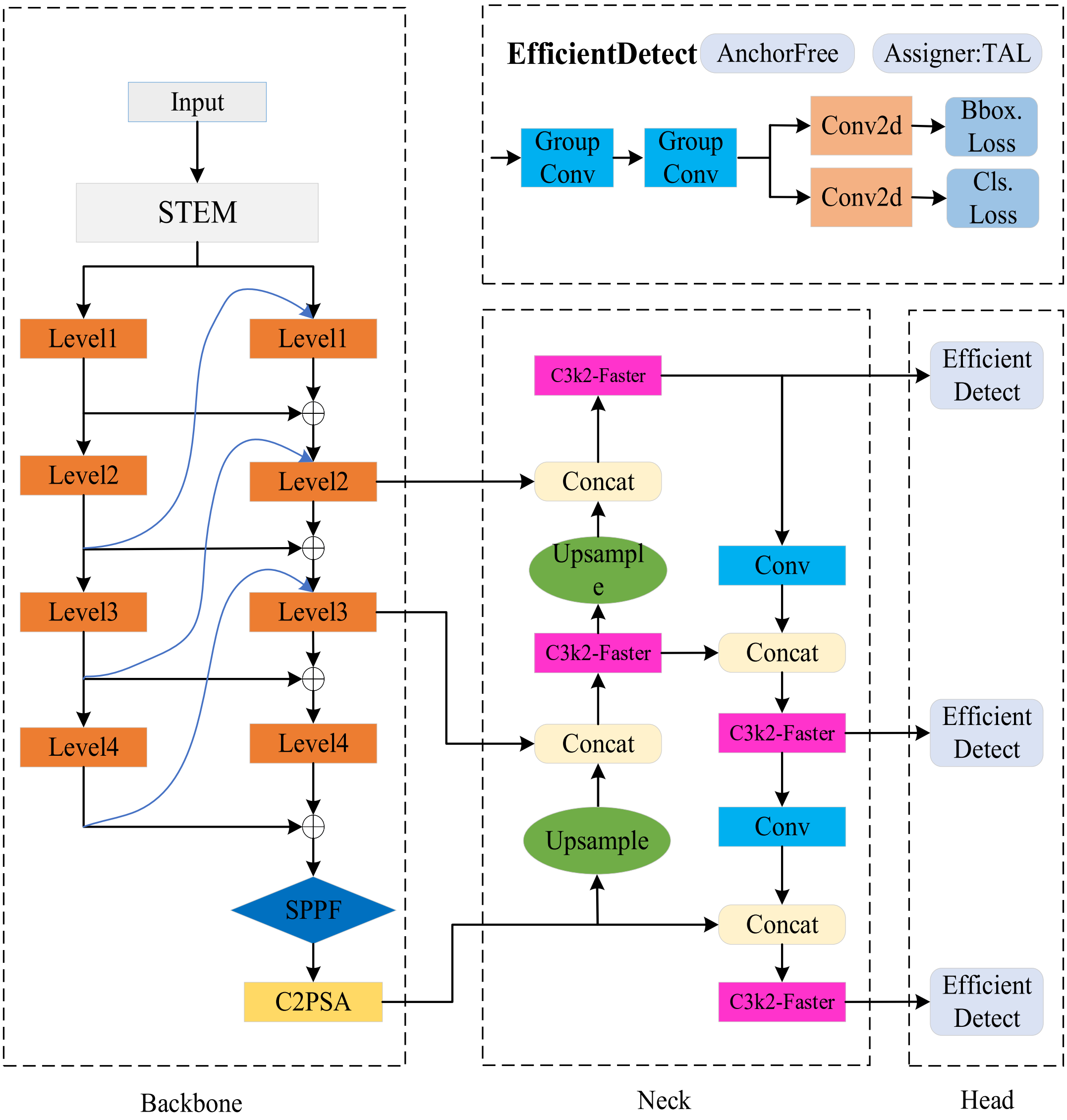

Figure 1. Structure diagram of RF-EW-YOLOv11.

The key advantage of this method lies in its ability to retain essential low-level information while maintaining high accuracy during training, which is critical for achieving superior detection performance in subsequent tasks.

As shown in Figure 2, the reversible connected multi-column network (RevCol) first segments the input image into several non-overlapping regions, and then processes them in several different hierarchical modules. The starting position of each column is the lowest level spatial information. As the data in each column propagates (Level 1 throughout Level 4), the semantic information of the features in the input image is gradually extracted. At the same time, reversible connections are introduced between each column to maintain information on different decoupling dimensions. The last column integrates the propagated information in the network structure and is responsible for generating feature maps. In Figure 2, starting from Level 2, the output of the previous column will serve as the input for the next column, and feature alignment at different scales will be achieved through "Fusion blocks". In other words, each layer of the input in the next column contains high-level semantic information from the previous column and low-level spatial information from this column. This is how the model can fully learn decoupled features during forward propagation. The reversible connection lines between each column effectively preserve the information of each column, while ensuring the information is not lost during transmission. This process can be represented by the following formulas.

$$X_t = OP_t(X_{t-1}, X_{t-m+1}) + \theta X_{t-m} \quad (1)$$

$$X_{t-m} = \theta^{-1}[X_t - OP(X_{t-1}, X_{t-m+1})] \quad (2)$$

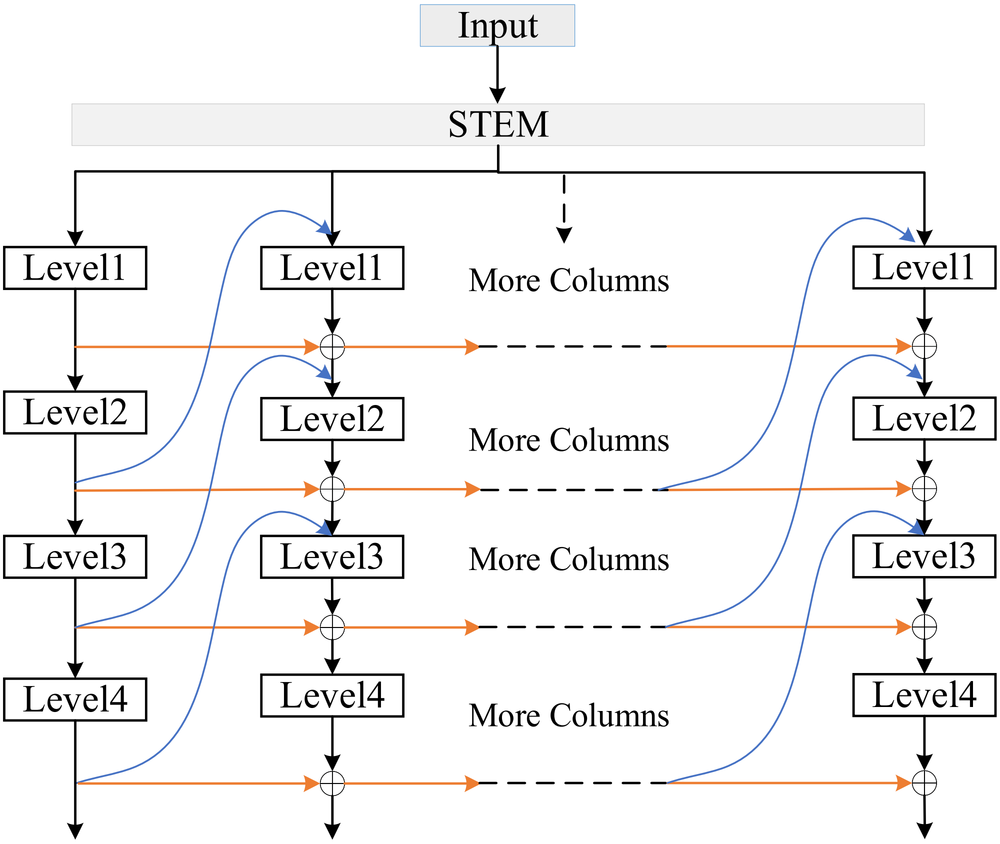

Figure 2. Reversible multi-column network structure (RevCol).

Among them, X_t represents the t-th layer features; X_{t-1} is the input of lower level information in the column; X_{t-m} is the input of information from the same layer as the previous column; X_{t-m+1} is the input of advanced semantic information from the previous column; OP is a non-linear operation; and θ is a reversible operation.

This section is based on the RevCol concept and reconstructs the BackBone of YOLOv11 by replacing the ConvNextBlock structure in the Level module of the RevCol network with C3k2. The microstructure of RevCol before improvement is shown in Figure 3a, where each level layer is feature extracted through downsampling and ConvNextBlock structure. The improved Level module is shown in Figure 3b, and the four different levels of channels set by the improved model are 16, 32, 64, and 128. The depths of the blocks at different levels are 1, 2, 2, and 1. In order to avoid the complexity of the backbone network, which may increase the complexity and number of parameters of the model, this paper set the number of columns in RevCol to 2. The reconstructed backbone network is shown in Figure 4.

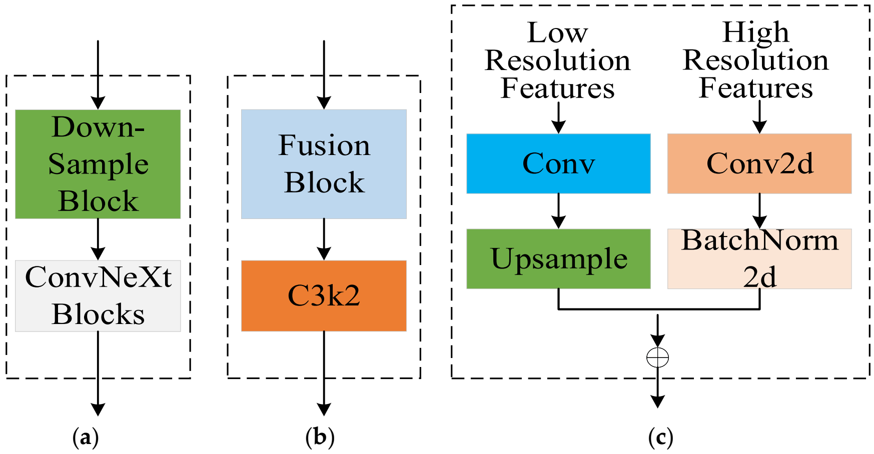

Figure 3. Level structure. (**a**) The original Level structure; (**b**) The optimized Level structure; (**c**) Fusion module structure.

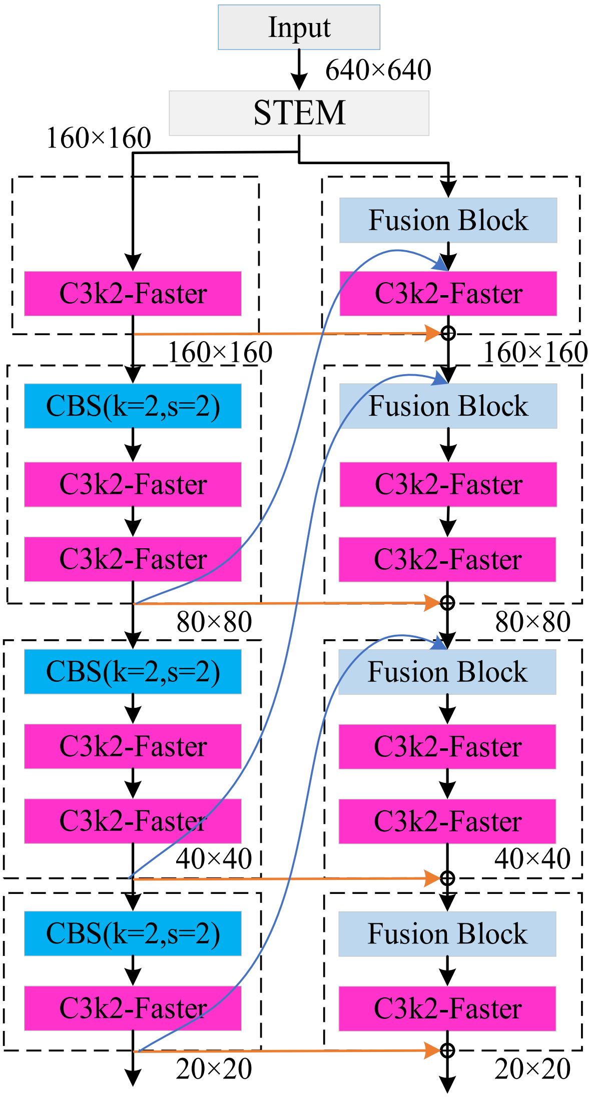

Figure 4. Reconstructed backbone network.

3.2. *FasterNet Integration*

Depthwise Separable Convolution (DWConv), as a variant of Ordinary Convolution (Conv), is widely used in key modules of Convolutional Neural Networks. Although

DWConv can effectively reduce frequent memory access of operators, it cannot directly replace Ordinary Convolution due to its inherent problems of insufficient feature interaction between channels and weakened spatial information extraction ability, resulting in a serious loss of accuracy. To balance efficiency and accuracy, Partial Convolution (PConv) convolves only a subset of the input channels, reducing redundant calculations while preserving key features. Therefore, the lightweight fast network FasterNet [27], whose design is based on the idea of partial convolution, will be embedded into the C3k2 module of YOLOv11 to improve algorithm performance by reducing redundant calculations and frequent memory access, and named C3k2-Faster.

FasterNet introduces an innovative Partial Convolution (PConv) strategy that primarily performs regular convolutions on certain channels of the input feature map while keeping the remaining channels unchanged. Then, it uses 1 × 1 convolution or pointwise convolution to further extract information. The detailed process of this strategy is shown in Figure 5b. This method not only reduces computational complexity, but also considers information extraction from all channels, opening up new directions for lightweight network design.

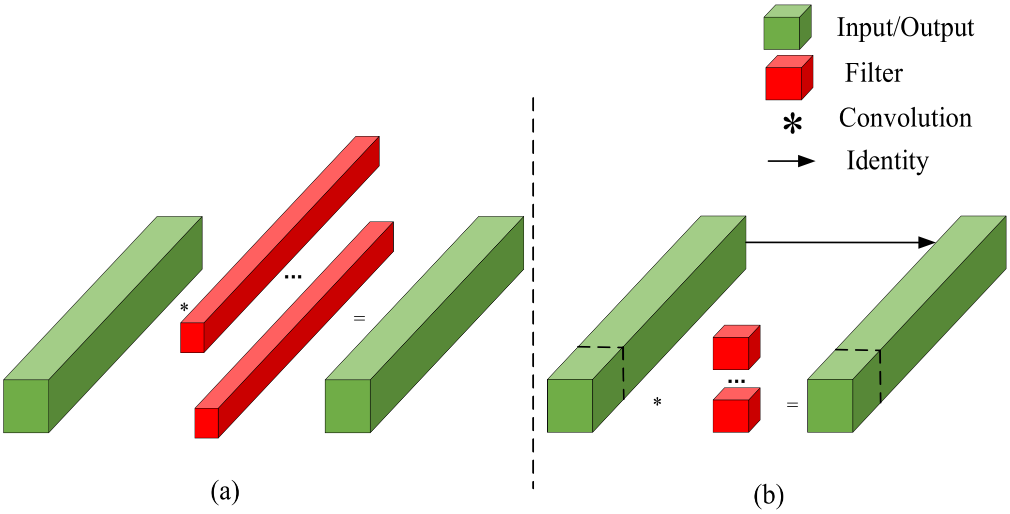

Figure 5. Structural comparison between convolution and partial convolution. (**a**) Convolution and (**b**) Pconv.

In the case of a partial ratio of r = 1/4, PConv only uses 1/4 of the input channels for convolution, while the remaining 3/4 channels remain unchanged, and the subsequent 1 × 1 convolution has a very low computational overhead. The computational overhead of partial convolution is only 1/16 of that of ordinary convolution.

3.3. EfficientDetect Head

The detection head of YOLOv11 is similar to that of YOLOv8, both using a dual branch structure for object classification and bounding box regression. YOLOv11 introduces depthwise separable convolution (DWConv) on the classification branch, which significantly reduces computational resource consumption while maintaining high classification performance. However, due to the superposition effect of the number of channels and convolution kernel parameters, this process still generates a large number of parameters, which limits the computational efficiency of the model. Therefore, in order to further optimize the structural design of the detection head, reduce redundant parameters, and improve model performance, this section designs an efficient EfficientDetect detection head module, as shown in the upper right corner of Figure 1.

EfficientDetect Head enhances multi-scale object detection capability by dynamically adjusting the weight allocation of different levels in the feature pyramid, and also uses spatial attention pooling operation to enable the model to focus on key areas of the image. EfficientDetect uses two group convolutions to form a shared convolutional layer module. The design of this module aims to reduce the number of parameters and computational complexity through the characteristics of grouped convolution, while achieving efficient interaction of cross-channel information. Specifically, W, H, and C correspond to the width, height, and number of channels of the feature map, and the first grouped convolution has the same number of input and output channels, a convolution kernel size of 3 × 3, and a grouping size of g = c/16, where c is the number of input channels used to reduce parameter and computational complexity while preserving feature expression ability and k is the size of the convolution kernel. The configuration of the second grouped convolution is the same as that of the first one to ensure consistency in feature extraction. The shared convolution module achieves cross-channel information fusion and interaction by concatenating the output features of grouped convolutions, thereby integrating the originally independent classification and regression branches into a unified feature processing flow. This design not only significantly reduces the parameter and computational complexity of the model, but also enables the extraction of key information from input features. Finally, the output features of the shared convolution module are split into two independent branches for object category prediction and bounding box regression. The grouped convolution structure is shown in Figure 6, where arrow means genetation.

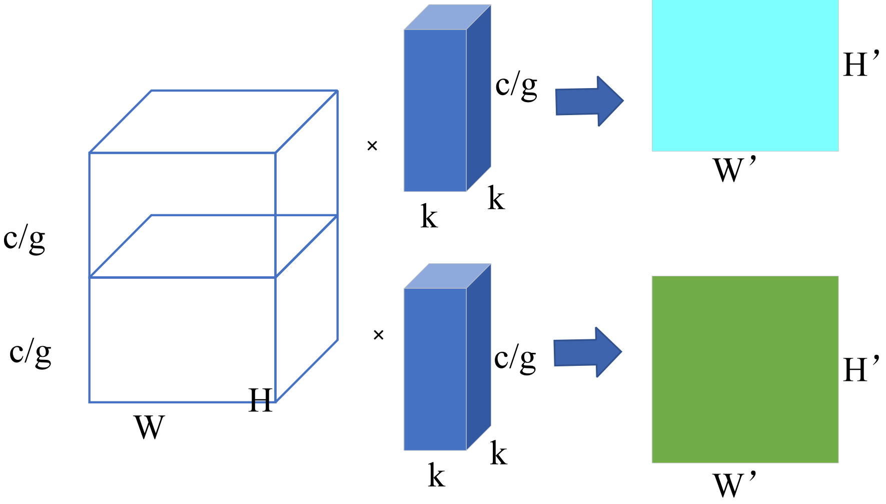

Figure 6. Grouping convolution.

3.4. *WISEPIoU Loss Function*

Boundary box regression is one of the core steps in object detection tasks, and its optimization effect directly affects the accuracy of object localization. YOLOv11 continues the design of YOLOv8, still using DFL Loss (Distribution Focal Loss) and CIoU Loss (Complete Intersection over Union Loss) as regression loss functions in the branches responsible for bounding box regression and confidence prediction. Among them, DFL Loss improves regression accuracy by optimizing the discretization representation of bounding box distribution. CIoU Loss, on the basis of traditional IoU loss, further introduces constraints on center point distance and aspect ratio to more comprehensively measure the matching degree between predicted boxes and real boxes. The CIoU loss function is represented by the following formulas:

$$L_{CIoU} = L_{IoU} + \frac{d^2}{c^2} + \alpha \times \nu, 0 \leq L_{CIoU} \leq 2 \tag{3}$$

$$\alpha = \frac{\nu}{1 - IoU + \nu}; v = \frac{4}{\pi^2}(\arctan\frac{W_{gt}}{h_{gt}} - \arctan\frac{W}{h})^2 \tag{4}$$

Formula (3) for the CIoU loss function consists of three parts, which are used to measure the consistency of the overlap area, center point distance, and aspect ratio between the predicted box and the true box. Among them, d represents the Euclidean distance between the center point of the predicted box and the center point of the real box, and c represents the diagonal length of the minimum bounding rectangle between the predicted box and the real box. It is a weight coefficient used to evaluate the shape consistency between the predicted box and the real box.

This study replaced CIoU with WISEIOU, which is a combination of WISE IoU and POWERFUL IoU. The loss functions of WISE IoU and POWERFUL IoU are shown in Formulas (5) and (6), respectively.

$$L_{WIoU} = L_{IoU} \times R_{WIoU}, R_{WIoU} = \exp(\frac{d^2}{c^2}) \tag{5}$$

$$L_{PIoU} = L_{IoU} + R_{PIoU}; R_{PIoU} = 1 - \exp(-(\frac{\frac{dw_1}{w_{gt}} + \frac{dw_2}{w_{gt}} + \frac{dh_1}{h_{gt}} + \frac{dh_2}{h_{gt}}}{4})^2) \tag{6}$$

From the above formula, it can be seen that the WISE *IoU* loss function is a loss function multiplied by a penalty factor R_{PIoU} that can promptly correct deviations from the original loss function; POWERFUL IoU is a loss function plus a penalty factor R_{PIoU}, where dw_1, dw_2, dh_1, and dh_2 represent the edge distance between the predicted box and the target box, and w_{gt} and h_{gt} are the width and height of the target box, respectively. Therefore, based on the characteristics of the two IoU loss functions, the WISPEOU loss function formula is shown in Formula (7).

$$L_{WisePIoU} = L_{IoU} \times R_{WPIoU} + R_{PIoU} \tag{7}$$

4. Experiments and Analysis

4.1. Datasets and Experimental Setup

This study used the NEU-DET [28] and deepPCB [29] datasets to validate the effectiveness of ES-BiCF-YOLOv8 in defect detection direction. The NEU-DET dataset contains six types of defects on steel surfaces, namely cracks (CRs), inclusions (INs), patches (PAs), pitting surfaces (PSs), rolled oxide scales (RSs), and scratches (SCs). The dataset consists of 1800 images and is divided into training, validation, and testing sets in an 8:1:1 ratio. The deepPCB dataset contains six types of defects on the surface of PCB boards, namely open circuit, short circuit, mouse bite, missing hole, burr, and spurious copper, totaling 1500 images. These images are divided into training, validation, and testing sets in a ratio of 7:1:2.

We used the deep learning framework pytorch for model training and validation, with CUDA version 11.2 and an experimental environment based on Ubuntu 16.04.7 operating system. The experimental parameter settings are shown in Table 1, with no dropout and no early stopping occurred.

Table 1. Experimental parameters.

Condition Parameters	Parameter Setting
Images	640 × 640
Optimiser	SGD
Batchsize	16
Epoch	300
Initial learning rate	0.01
Momentum factor	0.937
Weighting of decay	0.0005
Threshold values for IoU	0.5

4.2. Evaluation Metrics

In order to better evaluate the performance of defect detection algorithms, this paper constructed evaluation indicators for defect target detection system. We used average detection accuracy, model parameter count, GFLOPs, and FPS (frames per second) as performance evaluation metrics for the model.

MAP (Mean Accuracy): mAP reflects the average detection accuracy of the model across all categories, with higher values indicating better model performance. The formula is as follows.

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{8}$$

Among them, n is the total number of categories in the dataset, i is the number of detections, and AP (Average Precision) refers to the area under the Precision Recall Curve, which is the average accuracy of a single category. The mAP value is obtained by taking the average of the AP values of all categories.

(1) Parameter quantity (Params): Parameter quantity is used to evaluate the complexity of the model, and a smaller parameter quantity indicates that the model is lighter.

(2) Calculation volume (GFLOPs): The calculation volume measures the computational efficiency of the model, and the smaller the GFLOPs value, the higher the computational efficiency of the model.

(3) FPS (number of images processed per second): FPS represents the model's ability to process images per unit of time, with higher values indicating a better real-time performance of the model. The calculation method is shown in Formula (9), where Frameum represents the total number of detected images and ElapsedTime represents the total time spent on detection.

$$FPS = \frac{Framenum}{ElapsedTime} \tag{9}$$

(4) Precision measures the accuracy of positive predictions and reflects the model's ability to control false detections. It is calculated as follows:

$$Precision\ (\%) = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

where *TP* (True Positive) denotes the number of correctly detected objects, and *FP* (False Positive) represents the number of incorrectly detected objects.

Recall evaluates the completeness of object detection and reflects the model's capability to avoid missed detections. It is defined as follows:

$$Recall(\%) = \frac{TP}{TP + FN} \times 100\% \tag{11}$$

These indicators together constitute a comprehensive evaluation of the performance of proposed method, which helps us to fully understand the performance of the model in defect recognition tasks.

4.3. Effectiveness of Loss Function Optimization

In order to understand the impact of loss function on network performance, we conducted comparative experiments. Several mainstream bounding box regression loss functions were selected for analysis, including CIoU, PIoU, SIoU, WIoU, Shape IoU, and Wise PIoU. The performance comparison of different loss functions is shown in Table 2.

Table 2. Performance comparison of different loss functions on NEU-DET dataset.

Loss Function	mAP@0.5	Params (M)	GFLOPs	FPS
CIoU	0.741	1.57	3.2	91
PIoU	0.734	1.57	3.2	91
SIoU	0.725	1.57	3.2	91
WIoU	0.726	1.57	3.2	91
Shape-IoU	0.749	1.57	3.2	91
Wise-PIoU	0.758	1.57	3.2	91

The experimental results show that Wise PIoU outperforms all other loss functions in terms of accuracy and recall, demonstrating significant advantages. Wise PIoU has a fast convergence and dynamic focusing mechanism. Through an adaptive weight allocation strategy, it effectively suppresses the harmful gradient of low-quality anchor points, while enhancing the learning ability of medium quality anchor points, thereby improving the detection accuracy of the model.

4.4. Ablation Experiment and Result Analysis

For the convenience of displaying experimental results, we named the experimental models as shown in Table 3.

Table 3. Description of the experimental models.

Model Name	RevCol Backbone	FasterNet Network	EfficientDetect Detection Head	WISEPIOU
R-YOLOv11	√			
RF-YOLOv11	√	√		
RC-E-YOLOv11	√	√	√	
RC-EW-YOLOv11	√	√	√	√

(1) The YOLOv11 model introduced into the RevCol backbone network is referred to as R-YOLOv11;

(2) The YOLOv11 model incorporating RevCol backbone network and FasterNet network is referred to as RF-YOLOv11;

(3) The YOLOv11 model incorporating RevCol backbone network, FasterNet network, and EfficientDetect detection head is referred to as RC-E-YOLOv11;

(4) The YOLOv11 model incorporating RevCol backbone network, FasterNet network, EfficientDetect detection head, and WISPEOU loss function is referred to as RC-EW-YOLOv11.

4.4.1. Ablation Study on NEU-DET

This section first designed an ablation experiment on the NEU-DET dataset of steel surface defects, and the experimental results are shown in Table 4. The experimental results show that the improved RF-EW-YOLOv11 model exhibits good performance on the evaluation metrics set in this paper. Although its detection accuracy slightly decreased compared to the benchmark model YOLOv11, it exhibits significant advantages in terms of parameter count, computational complexity, and FPS. Specifically, compared with the original model, the mAP value of the model decreased by 1.4%, the number of parameters decreased by 1.01 M, the GFLOPs decreased by 3.1, and the FPS value was basically the same as the original model.

Table 4. Ablation experiments on the NEU-DET dataset.

Solution	mAP@0.5	Params (M)	GFLOPs	FPS
YOLOv11 *n* (baseline)	0.769	2.58	6.3	93
R-YOLOv11	0.766	2.09	4.9	88
RF-YOLOv11	0.760	1.84	4.3	90
RF-E-YOLOv11	0.746	1.57	3.2	87
RF-EW-YOLOv11	0.758	1.57	3.2	91

The R-YOLOv11 model reconstructed by RevCol did not show significant changes in detection accuracy, but its parameter and computational complexity decreased by 0.49 M and 1.4 GFLOPs, respectively, and FPS decreased by 5. This indicates that RevCol can obtain more effective information with fewer parameter calculations, ensuring the excellent feature extraction ability of the model. Embedding FasterNet network in C3k2 of R-YOLOv8 model can further reduce the computational cost of the model while ensuring the ability to extract feature information. From Table 4, it can be seen that the RF-YOLOv11 model embedded in the FasterNet network has a mAP value that is basically the same as the original model, with a decrease of 0.74 M in parameters and 2 GFLOPs in computation. This result validates the advantage of FasterNet network in saving computing resources, while also taking into account the feature extraction of all channels, ensuring that detection accuracy is not affected. Compared with the R-YOLOv11 model, the FPS value has increased by 4, indicating that the FasterNet network can accelerate the detection speed of the model to a certain extent. The addition of this module has made the model perform very well. Due to the large proportion of computation and parameter count in the original model detection head, the introduction of EfficientDetect detection head reduces the computational complexity of the model's parameter count. From the table, it can be seen that the RF-E-YOLOv11 with the introduction of EfficientDetect detection head has a 2.9% decrease in mAP value compared to the original model, and a 1.01 M and 3.1 GFLOPs decrease in parameter and computational complexity, respectively. Although the detection accuracy of the model is affected, the parameter and computational complexity of the model have been greatly improved, which is in line with the purpose of lightweight improvement. In addition, by introducing the loss function WISPEOU, the detection accuracy of the RF-EW-YOLOv11 model slightly decreased compared to the original model, but this improvement partially compensated for the accuracy loss caused by the introduction of the EfficientDetect detection head. Overall, the improved model still maintains good detection performance, and the reduction in FPS is not severe, still meeting the requirements of real-time detection and achieving the expected goal of lightweight design. Figure 7 shows the detection performance of different models on the NEU-DET dataset. The color corresponds to different types of defects, from left to right are cracking, inclusion, patches, pitted sturface. The type of detect is together with corresponding box.

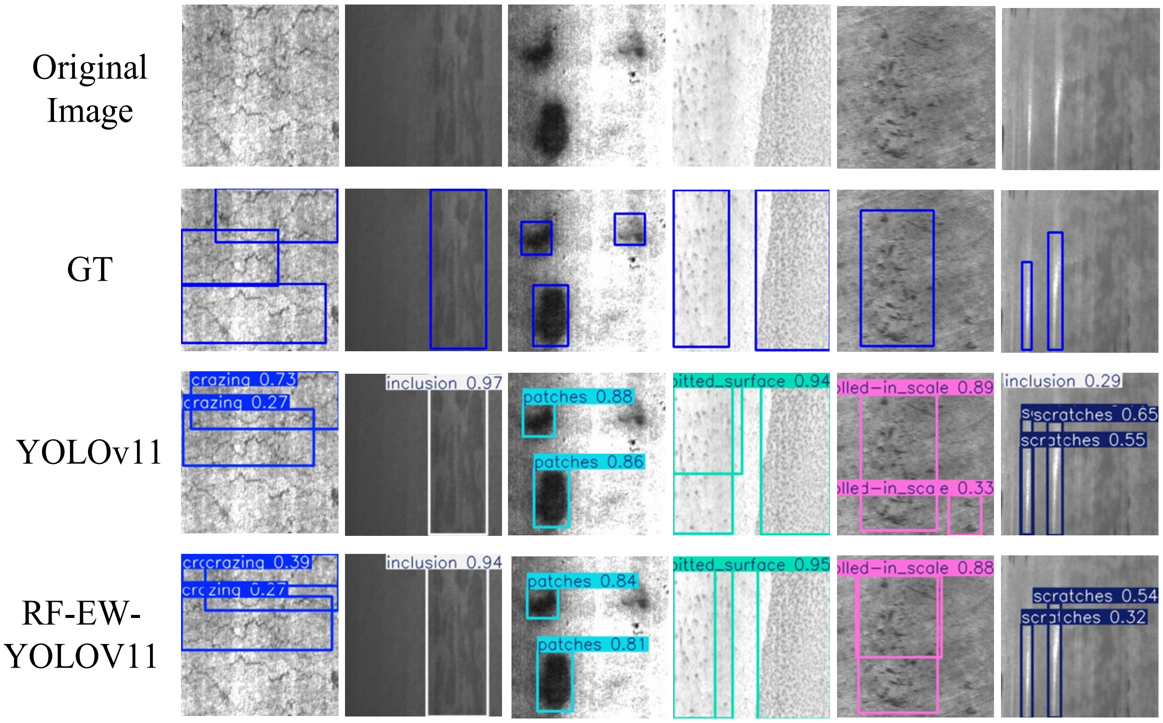

Figure 7. Detection performance on the NEU-DET dataset.

4.4.2. Ablation Study on DeepPCB

Due to the limited sample size and single scene of NEU-DET, as well as the diverse and fuzzy boundaries of defects, it may lead to a lack of diversity and generalization of defects. In order to verify the generalization ability of the model, this study also conducted ablation experiments on the deepPCB dataset. The experimental results are shown in Table 5. The results indicate that RF-EW-YOLOv11 significantly improves the efficiency and inference speed of the model while maintaining high detection accuracy, demonstrating excellent comprehensive performance. Compared with the baseline model YOLOv11, RF-EW-YOLOv11 maintained the same accuracy on mAP (0.985), but the parameter count significantly decreased from 2.58 M to 1.57 M, the computational complexity decreased from 6.3 GFLOPs to 3.2 GFLOPs, and the inference speed increased from 62.9 FPS to 64 FPS. This result indicates that RF-EW-YOLOv11 not only successfully compressed the volume and computational complexity of the model, but also further improved the computational efficiency and detection speed of the model. In summary, the model RF-EW-YOLOv11 proposed performs well on the deepPCB dataset, demonstrating its excellent generalization ability. Figure 8 shows the detection performance of different models on the deepPCB dataset.

Table 5. Ablation experiments on the DeepPCB dataset.

Solution	mAP@0.5	Params (M)	GFLOPs	FPS
YOLOv11 *n* (baseline)	0.985	2.58	6.3	62.9
R-YOLOv11	0.976	2.09	4.9	56.1
RF-YOLOv11	0.982	1.84	4.3	63.6
RF-E-YOLOv11	0.970	1.57	3.2	61.2
RF-EW-YOLOv11	0.985	1.57	3.2	64

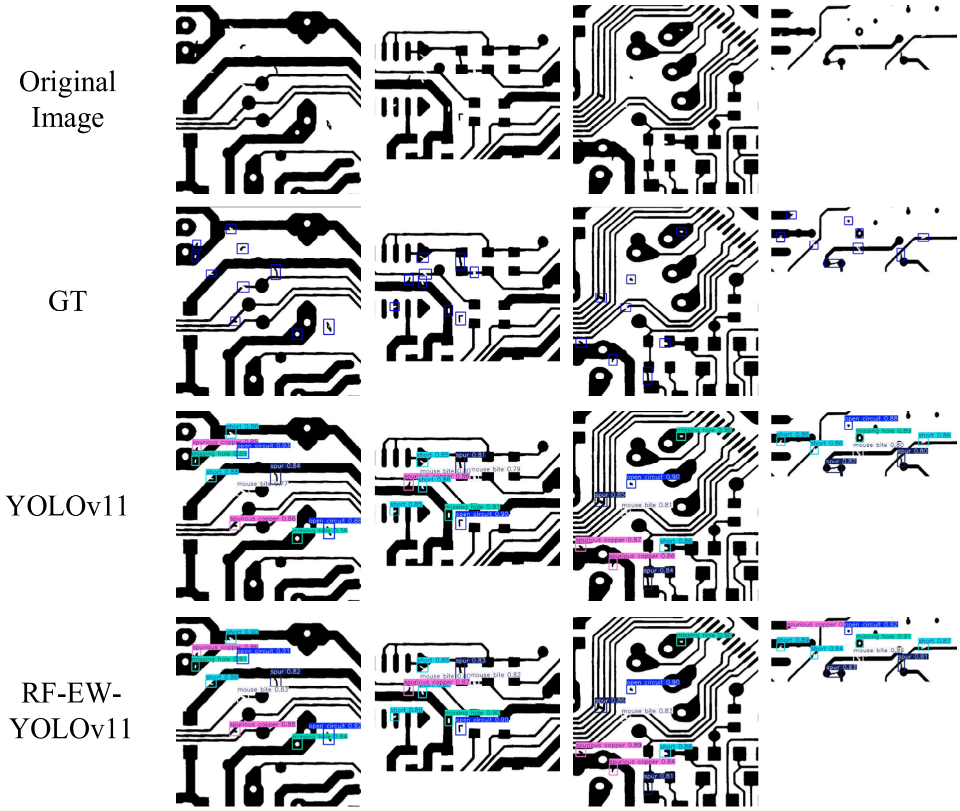

Figure 8. Detection performance on the deepPCB dataset.

4.5. Comparative Experimental Analysis and Result Analysis

4.5.1. Comparative Experiments on NEU-DET Dataset

In order to further verify the feasibility and superiority of the proposed model RF-EW-YOLOv11, this section selected YOLO series algorithms as well as Faster RCNN, SSD, and other algorithms for comparative experiments on the NEU-DET dataset. The experimental results are shown in Table 6.

Table 6. Comparative experiments on the NEU-DET dataset.

Solution	Precision (%)	Recall (%)	mAP@0.5	Params (M)	GFLOPs	FPS
YOLOv5	66.9	58.4	0.746	2.50	7.1	90
YOLOv6	67.4	65.3	0.728	4.23	11.8	113
YOLOv7	66.5	61.2	0.709	6.02	13.1	46
YOLOv8	67.2	64.8	0.734	3.01	8.1	81
YOLOv9	69.4	67.1	0.717	25.5	103.7	67
YOLOv10	70.7	69.5	0.745	2.69	8.4	85
YOLOv11	72.3	70.6	0.769	2.58	6.3	93
YOLOv12	73.8	71.7	0.770	2.57	6.5	50
Faster RCNN [30]	69.6	64.8	0.722	28.33	137.09	-
SSD [31]	65.4	68.9	0.610	41.10	145.3	41
RF-EW-YOLOv11	75.1	70.6	0.758	1.57	3.2	91

From the analysis of the data in Table 6, it can be concluded that the model RF-EW-YOLOv11 achieves optimal parameter count and GFLOPs, with values of 1.57 and 3.2, respectively. In terms of the parameter count of the model, the RF-EW-YOLOv11 model is 1.57 M, which achieves the best performance compared to all other models, demonstrating its outstanding performance in lightweight design and is very suitable for deployment on resource limited devices. Meanwhile, RF-EW-YOLOv11 only requires 3.2 GFLOPs, indicating its significant advantage in computational efficiency and the ability to achieve high detection accuracy at lower computational costs. In terms of model detection speed (FPS), RF-EW-YOLOv11 has an FPS of 91, second only to YOLOv6 (113) and YOLOv11 (93), and better than YOLOv12 (50), with good real-time detection performance. Overall, RF-EW-YOLOv11 has demonstrated significant advantages in balancing high precision and lightweight, efficient computing efficiency, and excellent inference speed. It is particularly suitable for resource limited and real-time demanding application scenarios, and has a wide range of application potential.

4.5.2. Comparative Experiments on deepPCB Dataset

In order to further verify the universal performance of RF-EW-YOLOv11, this section selected YOLO series algorithms, and the new PCB board defect detection model EffNet PCB for experiments on the deepPCB dataset. The experimental results of the EffNet PCB model are from reference [32]. The comparative experimental results on the deepPCB dataset are shown in Table 7.

Table 7. Comparative experiments on deepPCB.

Solution	Precision (%)	Recall (%)	mAP@0.5	Params (M)	GFLOPs	FPS
YOLOv5	97.4	97.5	0.983	2.50	7.1	93
YOLOv6	97.6	96.8	0.979	4.23	11.8	102
YOLOv7	96.3	93.9	0.950	6.02	13.1	48
YOLOv8	97.6	97.4	0.984	3.01	8.1	85
YOLOv9	96.2	95.7	0.963	25.5	103.7	61.7
YOLOv10	97.0	95.3	0.980	2.69	8.4	83
YOLOv11	98.1	96.6	0.985	2.58	6.3	62.9
YOLOv12	98.4	97.1	0.987	2.57	6.5	58
EffNet-PCB [32]	98.2	96.9	0.990	6.40	22.2	-
RF-EW-YOLOv11	98.3	96.7	0.985	1.57	3.2	64

From the results in Table 7, it can be seen that RF-EW-YOLOv11 achieves detection accuracy comparable to mainstream models with a lower parameter count (1.57 M) and computational complexity (3.2 GFLOPs) (mAP value of 0.985), especially in terms of model lightweighting, with a parameter count of only 6.2% of YOLOv9, but maintains higher accuracy. In terms of detection accuracy, the mAP value of RF-EW-YOLOv11 is only slightly lower than YOLOv12 (0.987) and the dedicated PCB detection method EffNet PCB (0.990), indicating its strong robustness in defect detection tasks. In terms of model lightweighting, RF-EW-YOLOv11 performs particularly well, with only 1.57 M parameters, far lower than YOLOv7 (6.02), YOLOv9 (25.5), and EffNet PCB (6.40) methods, and even more streamlined than YOLOv5 (2.50). In terms of computational complexity, the GFLOPs of model RF-EW-YOLOv11 are only 3.2, indicating its high deployment value in industrial scenarios with limited computing resources. However, RF-EW-YOLOv11 still has certain limitations, with a detection speed FPS of only 64, far lower than models such as YOLOv5, YOLOv6, and YOLOv8, indicating that its architecture design still has room for optimization in the real-time detection performance of PCB board defects. Overall, RF-EW-YOLOv11 has achieved a good balance between lightweight and detection accuracy, and is also suitable for industrial PCB quality inspection scenarios with limited computing resources. Its generalization performance is good, but its inference speed has not yet reached its optimal level. In the future, its real-time detection performance on PCB board defects can be further improved through network pruning, quantization, or more efficient attention mechanisms.

5. Conclusions

This paper proposes an RF-EW-YOLOv11 algorithm based on lightweight improvement to meet the application requirements of the YOLOv11 model in the field of industrial defect detection. The aim is to ensure the detection speed of the model and reduce the number of parameters and computation, while maintaining high detection accuracy. We propose a new detection model RF-EW-YOLOv11 based on the YOLOv11 detection model. Firstly, the backbone network of the original model is reconstructed using a reversible multi-column network (RevCol). By utilizing RevCol's multi-column design and reversible connections, low-level spatial information and high-level semantic information are effectively preserved, thereby reducing model complexity while ensuring high-level feature extraction capability. At the same time, embedding a FasterNet network in the C3k2 module of the YOLOv11 network reduces the model's parameter and computational complexity, improves the model's inference speed, and obtains more useful feature information at a lower cost. In addition, considering that the detection head still generates a large number of redundant parameters during computation, we introduced the EfficientDetect detection head based on Group Conv to further reduce the model's parameter and computational complexity. Finally, replacing the loss function of YOLOv11 itself with WISPEIoU enables the model to converge faster and improve its detection accuracy. The experimental results show that the improved RF-EW-YOLOv11 exhibits excellent performance on both NEU-DET and DeepPCB datasets: on the NEU-DET dataset, RF-EW-YOLOv11 has an mAP of 0.758, a parameter size of only 1.57 M, a computational complexity of 3.2 GFLOPs, and an FPS of 91. Compared with the benchmark model YOLOv11, it reduces the parameter size and computational complexity by 1.01 and 3.1, respectively, while ensuring excellent real-time detection performance. In addition, the generalization ability of RF-EW-YOLOv11 was further validated on the deepPCB dataset, and good results were achieved on the selected evaluation metrics in this paper. Through ablation experiments and comparative experiments, we validated the effectiveness of RevCol backbone network, FasterNet network, EfficientDetect detection head, and loss function improvement in model lightweighting and performance enhancement. Compared with the YOLO series and other mainstream object detection models such as Faster RCNN and SSD, RF-EW-YOLOv11 has significant advantages in terms of parameter count, computational complexity, and inference speed. It is particularly suitable for industrial defect detection scenarios with limited resources and high real-time requirements, providing an efficient and practical solution for the field of industrial defect detection.

Although the proposed adaptive multi-scale architecture captures defect features at different scales by dynamically adjusting the receptive field, this process introduces additional computational overhead, resulting in a slower inference speed than lightweight models. Future research could focus on model lightweighting and acceleration technology. Firstly, consider pruning redundant channels or branches in multi-scale architectures, such as removing specific scale features that contribute less to small defect detection. Secondly, consider using dynamic sparse attention to only calculate local attention related to the defect area, reducing computational complexity.

Author Contributions: Conceptualization, X.C., Y.W. and H.Z.; methodology, B.A.; validation, Y.W., H.Z. and B.A.; writing—original draft preparation, H.Z. and B.A.; writing—review and editing, L.Y.; visualization, H.Z.; supervision, B.A. and L.Y.; project administration, B.A.; funding acquisition, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant (62472149).

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

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

References

1. Wen, X.; Shan, J.; He, Y.; Song, K. Steel surface defect recognition: A survey. *Coatings* **2022**, *13*, 17. [CrossRef]
2. Ren, Z.; Fang, F.; Yan, N.; Wu, Y. State of the art in defect detection based on machine vision. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2022**, *9*, 661–691. [CrossRef]
3. Zhou, A.; Zheng, H.; Li, M.; Shao, W. Defect inspection algorithm of metal surface based on machine vision. In Proceedings of the 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Phuket, Thailand, 28–29 February 2020; IEEE: New York, NY, USA; pp. 45–49.
4. Masci, J.; Meier, U.; Ciresan, D.; Schmidhuber, J.; Fricout, G. Steel defect classification with max-pooling convolutional neural networks. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, Australia, 10–15 June 2012; IEEE: New York, NY, USA; pp. 1–6.
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA; pp. 779–788.
6. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
8. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
9. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
10. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; Springer Nature: Cham, Switzerland, 2024; pp. 1–21.
11. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024; Volume 37, pp. 107984–108011.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; Volume 14, pp. 21–37.
13. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
14. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
15. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
17. Tang, B.; Chen, L.; Sun, W.; Lin, Z.-k. Review of surface defect detection of steel products based on machine vision. *IET Image Process.* **2023**, *17*, 303–322. [CrossRef]
18. Xu, K.; Xu, Y.; Zhou, P.; Wang, L. Application of RNAMlet to surface defect identification of steels. *Opt. Lasers Eng.* **2018**, *105*, 110–117. [CrossRef]
19. Liu, Y.; Xu, K.; Wang, D. Online surface defect identification of cold rolled strips based on local binary pattern and extreme learning machine. *Metals* **2018**, *8*, 197. [CrossRef]
20. Liu, X.; Xu, K.; Zhou, P.; Zhou, D.; Zhou, Y. Surface defect identification of aluminium strips with non-subsampled shearlet transform. *Opt. Lasers Eng.* **2020**, *127*, 105986. [CrossRef]
21. Sun, Q.; Xu, K.; Liu, H.; Wang, J. Unsupervised surface defect detection of aluminum sheets with combined bright-field and dark-field illumination. *Opt. Lasers Eng.* **2023**, *168*, 107674. [CrossRef]

22. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [CrossRef]
23. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W.; Versaci, M. A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 5592878. [CrossRef] [PubMed]
24. Yang, L.; Huang, X.; Ren, Y.; Huang, Y. Steel plate surface defect detection based on dataset enhancement and lightweight convolution neural network. *Machines* **2022**, *10*, 523. [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. Cai, Y.; Zhou, Y.; Han, Q.; Sun, J.; Kong, X.; Li, J.; Zhang, X. Reversible column networks. *arXiv* **2022**, arXiv:2212.11696.
27. Chen, J.; Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
28. Zhao, Z.; Li, B.; Dong, R.; Zhao, P. A surface defect detection method based on positive samples. In Proceedings of the PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, 28–31 August 2018; Springer International Publishing: Berlin/Heidelberg, Germany, 2018. Part II 15. pp. 473–481.
29. Lv, S.; Ouyang, B.; Deng, Z.; Liang, T.; Jiang, S.; Zhang, K.; Chen, J.; Li, Z. A dataset for deep learning based detection of printed circuit board surface defect. *Sci. Data* **2024**, *11*, 811. [CrossRef] [PubMed]
30. Feng, Y.; Liu, W. Steel Surface Defect Detection Algorithm Based on HPDE-YOLO. *J. Shenyang Ligong Univ.* **2025**, *44*, 31–38.
31. Zhang, Z.; Wei, G.; Zhang, Z.; Cai, J. Surface Defect Detection Method Based on Improved YOLOv5. *Control Eng. China* **2025**, *32*, 943–951.
32. Hou, Y.; Zhang, X. A lightweight and high-accuracy framework for Printed Circuit Board defect detection. *Eng. Appl. Artif. Intell.* **2025**, *148*, 110375. [CrossRef]

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