




Article

SCFNet: Lightweight Steel Defect Detection Network Based on Spatial Channel Reorganization and Weighted Jump Fusion

Hongli Li ^{1,2,†}, Zhiqi Yi ^{1,2,†}, Liye Mei ^{3,4,†} , Jia Duan ⁵, Kaimin Sun ⁶ , Mengcheng Li ¹, Wei Yang ⁵ 
and Ying Wang ^{5,*}

¹ School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China

² Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430205, China

³ The Institute of Technological Sciences, Wuhan University, Wuhan 430072, China

⁴ School of Computer Science, Hubei University of Technology, Wuhan 430068, China

⁵ School of Information Science and Engineering, Wuchang Shouyi University, Wuhan 430064, China

⁶ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

* Correspondence: wangying@wsyu.edu.cn

† These authors contributed equally to this work.

Abstract: The goal of steel defect detection is to enhance the recognition accuracy and accelerate the detection speed with fewer parameters. However, challenges arise in steel sample detection due to issues such as feature ambiguity, low contrast, and similarity among inter-class features. Moreover, limited computing capability makes it difficult for small and medium-sized enterprises to deploy and utilize networks effectively. Therefore, we propose a novel lightweight steel detection network (SCFNet), which is based on spatial channel reconstruction and deep feature fusion. The network adopts a lightweight and efficient feature extraction module (LEM) for multi-scale feature extraction, enhancing the capability to extract blurry features. Simultaneously, we adopt spatial and channel reconstruction convolution (ScConv) to reconstruct the spatial and channel features of the feature maps, enhancing the spatial localization and semantic representation of defects. Additionally, we adopt the Weighted Bidirectional Feature Pyramid Network (BiFPN) for defect feature fusion, thereby enhancing the capability of the model in detecting low-contrast defects. Finally, we discuss the impact of different data augmentation methods on the model accuracy. Extensive experiments are conducted on the NEU-DET dataset, resulting in a final model achieving an mAP of 81.2%. Remarkably, this model only required 2.01 M parameters and 5.9 GFLOPs of computation. Compared to state-of-the-art object detection algorithms, our approach achieves a higher detection accuracy while requiring fewer computational resources, effectively balancing the model size and detection accuracy.

Keywords: surface defect detection; feature reconstruction; lightweight network; feature fusion



Citation: Li, H.; Yi, Z.; Mei, L.; Duan, J.; Sun, K.; Li, M.; Yang, W.; Wang, Y. SCFNet: Lightweight Steel Defect Detection Network Based on Spatial Channel Reorganization and Weighted Jump Fusion. *Processes* **2024**, *12*, 931. <https://doi.org/10.3390/pr12050931>

Academic Editor: Chin-Hyung Lee

Received: 8 April 2024

Revised: 27 April 2024

Accepted: 29 April 2024

Published: 2 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Steel is one of the most commonly used metals in manufacturing and is used widely in a variety of applications including construction, bridges, automobiles, and machinery. Due to its excellent performance, steel plays a crucial role in large industries such as metallurgy, geological drilling, and marine exploration. However, quality issues in steel often precipitate safety incidents, significantly compromising engineering integrity and personal safety [1]. As steel production increases, the possibility of defective steel entering the market increases, resulting in increasingly strict quality standards. In industrial manufacturing, the production environment for steel is complex and susceptible to various factors such as temperature and impact [2]. This results in surface defects such as cracks, patches, scratches, and inclusions [3,4]. Steel surface defect detection algorithms are essential for ensuring product quality, steel safety, and controlling production costs.

Typically, different types of defects on steel surfaces exhibit significant differences in terms of shape, size, and distribution. Examples include the following: (A) indistinctive features of defects: defect textures and grayscale are similar to the background [see Figure 1A]; (B) similar defects of different categories: different defects have similar distributions in shape and texture [see Figure 1B]; (C) low-contrast defects: defects have low color contrast with the background [see Figure 1C]; and (D) varied defects within the same category: defects within the same category exhibit significant differences in shape and texture [see Figure 1D]. This presents a considerable challenge to the feature extraction capacity of detectors. Early defect detection relied heavily on manual identification. However, manual identification is often costly, slow, and highly dependent on the experience and working conditions of the identification personnel. With advancements in the computer industry driving the automation sector forward, there arises an urgent demand across various industries for lightweight defect detection algorithms that enable automation while ensuring high-speed and high-precision performance [5–9].

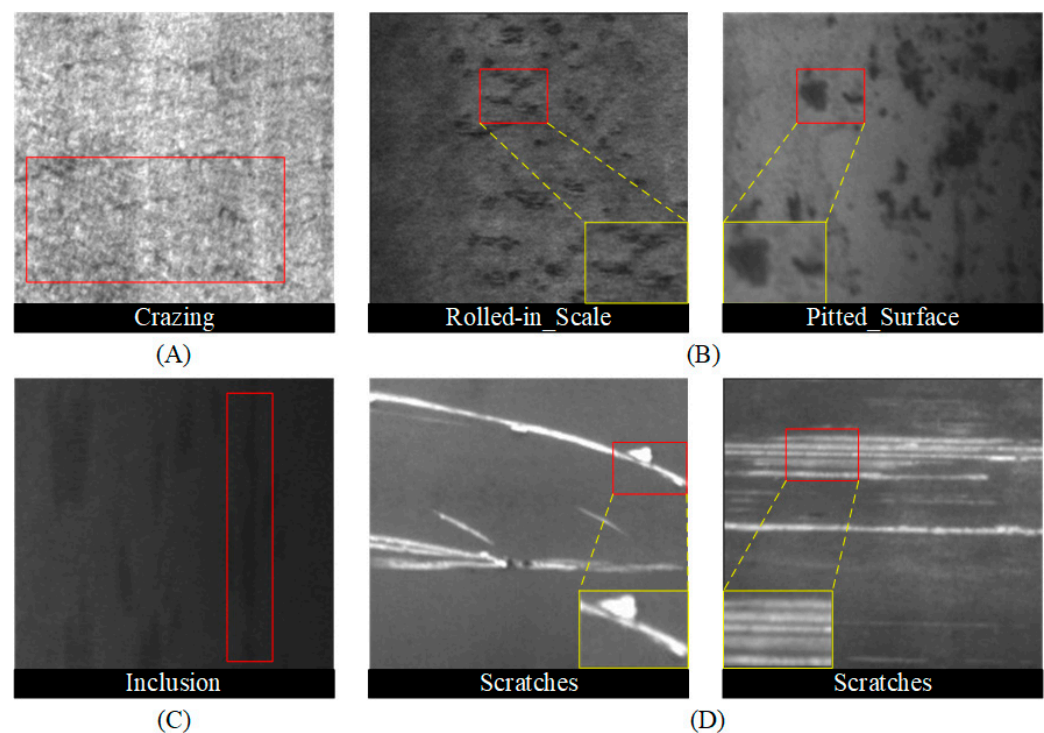


Figure 1. Steel surface defect detection faces a number of challenges, including (from the NEU-DET [10] dataset. The red box represents the defect location, and the yellow box shows the enlarged result.): (A) Defects with indistinct features. (B) Similar defects from different categories. (C) Defects with low contrast. (D) Defects within the same category exhibit significant variations.

The technique of detecting and classifying steel defects automatically is called computer vision-based steel defects detection. Typically, this approach involves extracting the shape, color, and texture information from images to describe and differentiate different types of defects. Techniques such as edge detection, corner detection, and texture analysis are utilized to extract features from images. In order to classify the features once they have been extracted, methods such as Support Vector Machines (SVMs) [11], clustering, Adaboost classifiers, or naive Bayes classifiers are used. However, since feature extractors often rely on manually designed features, this leads to lower model robustness. This makes it highly susceptible to factors such as lighting conditions, shooting angles, and the proportion of the target area [12].

In recent years, deep learning has undergone rapid development, significantly advancing object detection [13–15]. Images are transformed into feature maps through convolu-

tional neural networks. These feature maps typically contain higher-dimensional abstract features that are more targeted than manually designed features. Presently, mainstream object detection algorithms include two-stage algorithms such as R-CNN series [16–19], as well as single-stage algorithms such as SSD [20], You Only Look Once (YOLO) series [21–25], and Transformer-based algorithms such as DETR [26]. However, within the realm of steel defect detection, these deep learning-based object detection models face constraints due to the computational capabilities of terminal devices. Addressing how to optimize these object detection models with large parameters and computational overhead, while meeting task accuracy requirements, to enable deployment on devices with limited computing resources, remains a focal point in the current research on steel defect detection.

For detecting and recognizing defects on industrial steel surfaces, traditional machine learning methods have played a vital role in the early stages, usually involving image preprocessing, thresholding, and feature extraction. Traditional algorithms include LBP [10], HOG [27], and GLCM [28]. A number of studies [29,30] have developed more complex feature extractors by combining other methods in order to extract more accurate features of steel surface defects. Zhao et al. [31] utilized vector regularized kernel approximation and SVM for defect detection. Gong et al. [32] proposed developing a new Multi-Hypersphere SVM (MHSVM+) algorithm to provide additional information for detection tasks. Chu et al. [33] developed Multi-Information Siamese SVMs (MTSVMs), which are based on binary Siamese SVMs. Zhang et al. [34] proposed a method for identifying and diagnosing defects by merging Gaussian functions fitted to histograms of test images with membership matrices. However, traditional machine learning methods have significant limitations. The features used in these methods are manually designed, making them susceptible to changes in imaging environments and exhibiting poor robustness. Additionally, these methods often require extensive computational resources, resulting in slow processing speeds and difficulty in real-time detection.

Neural networks possess the capability to automatically extract features, fit models, and dynamically update parameters through learning processes, thereby allowing deep learning methods to excel across a multitude of tasks [35–39]. Upon entering samples into the network, it is capable of automatically classifying defect types and predicting defect locations. In practical steel surface defect detection, defects vary in size and shape, and the complex background makes them difficult to detect. Furthermore, smaller defects exhibit relatively minor changes in texture and color, making it difficult to distinguish between them. Using RDD-YOLO, Zhao et al. [40] integrated Res2Net blocks into the backbone network in order to enhance neck feature extraction and reuse shallow feature maps. Additionally, this method separates regression and classification with decoupled detection heads, improving detection accuracy. According to Wang et al. [41], YOLOv7 can be improved by integrating ConvNeXt modules into the backbone network and incorporating attention mechanisms in the pooling layers. The Diagonal Feature Fusion Network (DFN) strategy introduced by Yu et al. [42] matches multi-scale feature information without sacrificing speed, thereby significantly reducing the model size. Liu et al. [43] proposed DLF-YOLOF, using anchor-free detectors to reduce hyperparameters and expand contextual information in feature maps using deformable convolution networks and local spatial attention modules. Using a multi-scale lightweight network, Shao et al. [44] proposed a steel defect detection model that reduces the parameter count while improving the model accuracy. The aforementioned algorithms have made significant contributions in terms of both accuracy and speed. Nonetheless, these methods do not take into account the loss of information during the layer-by-layer feature extraction and spatial transformation of data, which is crucial for the detection of steel defects.

In order to further improve the detection accuracy while ensuring the lightweight of the model, we propose a lightweight and efficient steel defect detection algorithm called SCFNet. Specifically, we adopt an efficient and lightweight feature extraction module, LEM, to deeply excavate the defect information within the steel. And ScConv is applied in the deep network to reconstruct the spatial and channel information of feature maps, enhanc-

ing the representation of the defect features while reducing the generation of redundant information. Additionally, this article utilizes BiFPN for feature fusion, integrating deep semantic information and shallow spatial textures into one feature map, thereby preserving more complete spatial details and richer semantic features of the defect targets. We outline the contributions of this article as follows.

1. We propose a lightweight and efficient steel defect detection network, namely SCFNet. This network utilizes an LEM to extract feature information. The LEM, based on Depth-Wise convolution and channel-weighted fusion, can better extract ambiguous features.
2. Considering the low-contrast defects present in steel materials, we introduce the ScConv module into the LEM. By reconstructing the spatial information and channel features of the feature map, ScConv effectively reduces redundant features while enhancing key features in steel, thus making the defect area more clearly and accurately represented in the feature map.
3. We introduce the BiFPN module for feature fusion, leveraging its unique skip connection structure to minimize feature information loss during the convolution process. BiFPN ensures the preservation of crucial texture features, making it easier for the network to identify low-contrast defects.
4. We apply data augmentation techniques on the steel defect dataset and discuss the impact of different data augmentation methods on the detection accuracy. Ultimately, the proposed SCFNet demonstrates strong detection performance, outperforming state-of-the-art detectors in steel defect detection.

2. Materials and Methods

In practical steel defect detection, owing to complex backgrounds and the indistinct features of certain defects, detectors are susceptible to false positives and false negatives. We have noted that existing mainstream object detection networks lack sufficient capability in extracting ambiguous and low-contrast features. To optimize the defect detection performance in steel materials, we propose the SCFNet network model, as illustrated in Figure 2, which consists of three main components: the feature extraction module, neck fusion module, and detection head module. An image's deep features are extracted using the feature extraction module. Next, these features are forwarded to the neck fusion module. The neck fusion module is capable of constructing a feature pyramid network from top to bottom, transmitting the semantic information features of the fused feature maps, and then propagating the fused texture features from bottom to top. The neck network generates three feature maps with different spatial sizes, which are then fed into the detection heads separately. This allows the model to better detect objects on large, medium, and small scales, thereby alleviating the issue of inconsistent target scales. Specifically, when the image is input into the LEM consisting of three convolutional layers, 16 Mobile Inverted Bottleneck Convolution (MBConv) modules, 1 spatial channel recombination convolution module (comprising spatial recombination module SRU and channel recombination module CRU), and 3 feature maps with different spatial sizes and channel numbers, C3, C4, and C5, are obtained. Among them, C3 represents the shallow feature map with more texture information, C4 represents the middle feature map with certain texture information and semantic information, and C5 represents the deep feature map with more semantic information. A neck fusion network integrates the information from three different depths to coordinate and enrich the semantic and texture information of the three feature maps. Finally, the detection heads operate on the three feature maps separately to obtain the output information.

In the SCFNet network architecture, the feature extraction module is the LEM, which is extremely lightweight yet possesses strong feature extraction capabilities. As a result, it is able to extract deeper features from steel materials and adapt to defects that are not readily apparent. The neck module adopts a BiFPN for feature fusion. Compared to mainstream fusion networks like PANet [45], this fusion network features a unique skip

connection structure, minimizing the loss of spatial information and thereby enhancing the detector's performance.

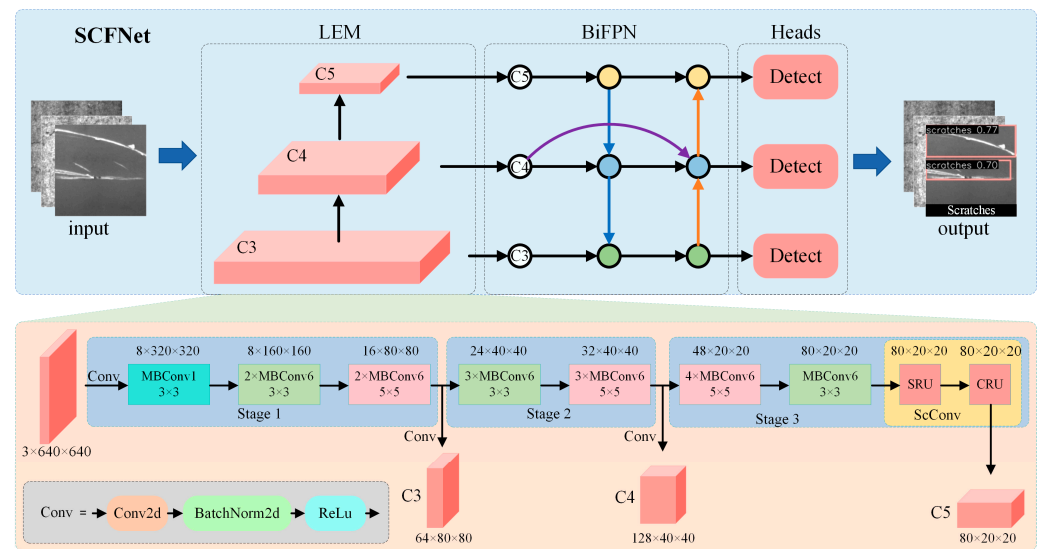


Figure 2. Architecture of the proposed SCFNet. The yellow, blue, and green circles represent feature maps of different scales.

2.1. Lightweight and Efficient Feature Extraction Module

Given the challenge of ambiguously extracting the target features of steel defects and the computational constraints imposed by terminal devices, deploying and utilizing networks present significant challenges. To address this issue, inspired by previous works [46,47], we propose a lightweight feature extraction network. In the past, convolutional neural network models typically optimized the model by adjusting one of three parameters: the input image resolution, network channel width, or depth. Tan's [46] study demonstrates that all three factors significantly impact the final accuracy and proposes a composite scaling method to uniformly adjust the network width, depth, and input image resolution, as shown in Equation (1) [46].

$$\begin{cases} \text{depth} : d = \alpha^\varphi \\ \text{width} : \omega = \beta^\varphi \\ \text{resolution} : r = \gamma^\varphi \\ \text{s.t. } \beta^2 \gamma^2 \alpha \approx 2 \\ \beta \geq 1, \gamma \geq 1, \alpha \geq 1 \end{cases} \quad (1)$$

where α , β , and γ are constants that can be determined by a small grid search. φ is an intuitively defined coefficient used to determine how many extra resources are available to scale the model.

Setting $\varphi = 1$ and based on the constraints in Equation (1), a grid search was performed. This led to $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$, resulting in the basic feature extraction module, LEM. The LEM has a relatively small parameter count and operates at a faster speed, making it highly suitable for lightweight detection tasks.

Figure 2 illustrates the LEM model structure composed of 3 convolutional layers, 16 MBConv modules, and 1 ScConv module. This model possesses strong feature extraction capabilities. Upon putting images into the network, the dimensions of the output feature maps increase gradually while the image size decreases. The deep feature maps harbor abundant semantic information, enabling the network to extract a broader range of classification features. In contrast, shallow feature maps contain a high level of texture information, which allows the network to retain certain texture characteristics and, as a result, place bounding boxes around the target objects in a more accurate manner. Similar

to other mainstream single-stage object detection models, SCFNet's feature extraction module outputs three layers of feature maps. These three layers of feature maps undergo interaction in the neck network, complementing each other's feature information, before being separately input into the detection heads for detection.

Figure 3 shows the MBConv module structure. This module mainly consists of two regular convolutions, one Depth-Wise convolution, one Squeeze-and-Excitation (SE) module, and a Dropout layer. The first convolution aims to increase the dimensionality, which helps in extracting features from deeper layers. In this context, MBConv1 signifies that the first convolution does not augment the dimensionality, whereas MBConv6 denotes a six-fold increase in the dimensionality. Depth-Wise convolution performs grouped convolutions, where each channel of the input is convolved separately without altering the number of channels in the feature map. A convolution following the SE module is a pointwise convolution, which uses only 1×1 convolutional kernels, operates on all channels, and can change the number of channels. By using Depth-Wise convolutions and pointwise convolutions, it is possible to construct deeper networks with smaller convolutional kernels and fewer parameters. This makes the model more lightweight without sacrificing accuracy. As a learnable attention mechanism, the SE module determines the importance of each channel by learning weights, thus guiding the model attention to more significant features.

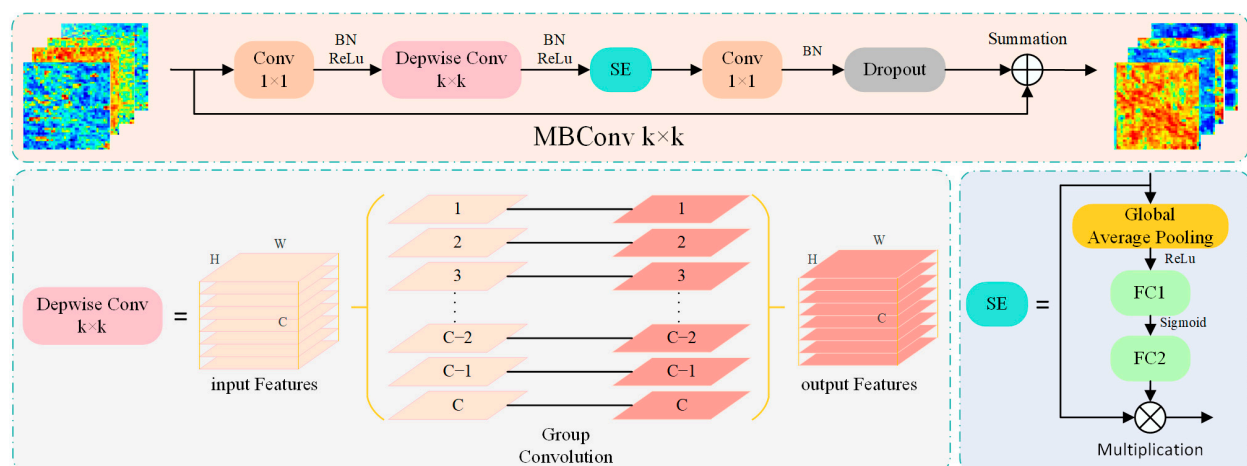


Figure 3. Structure diagram of the MBConv. The input and output feature maps are represented using heatmaps, where warm colors indicate strong features and cool colors indicate weak features.

2.2. Spatial and Channel Reconstruction Convolution

Due to the existence of similar features between different defect categories and defects with low contrast in steel defects, this poses a challenge to the feature expression capability of detectors. The ability of the feature extraction module to obtain representative features directly impacts the final results of the entire network. To enhance the representational capacity of the network, we propagate deep feature maps through the spatial and channel reconstruction convolution (ScConv) module. The ScConv module structure, as shown in Figure 4, consists of two units: the Spatial Reconstruction Unit (SRU) and the Channel Reconstruction Unit (CRU), which are sequentially placed in the module.

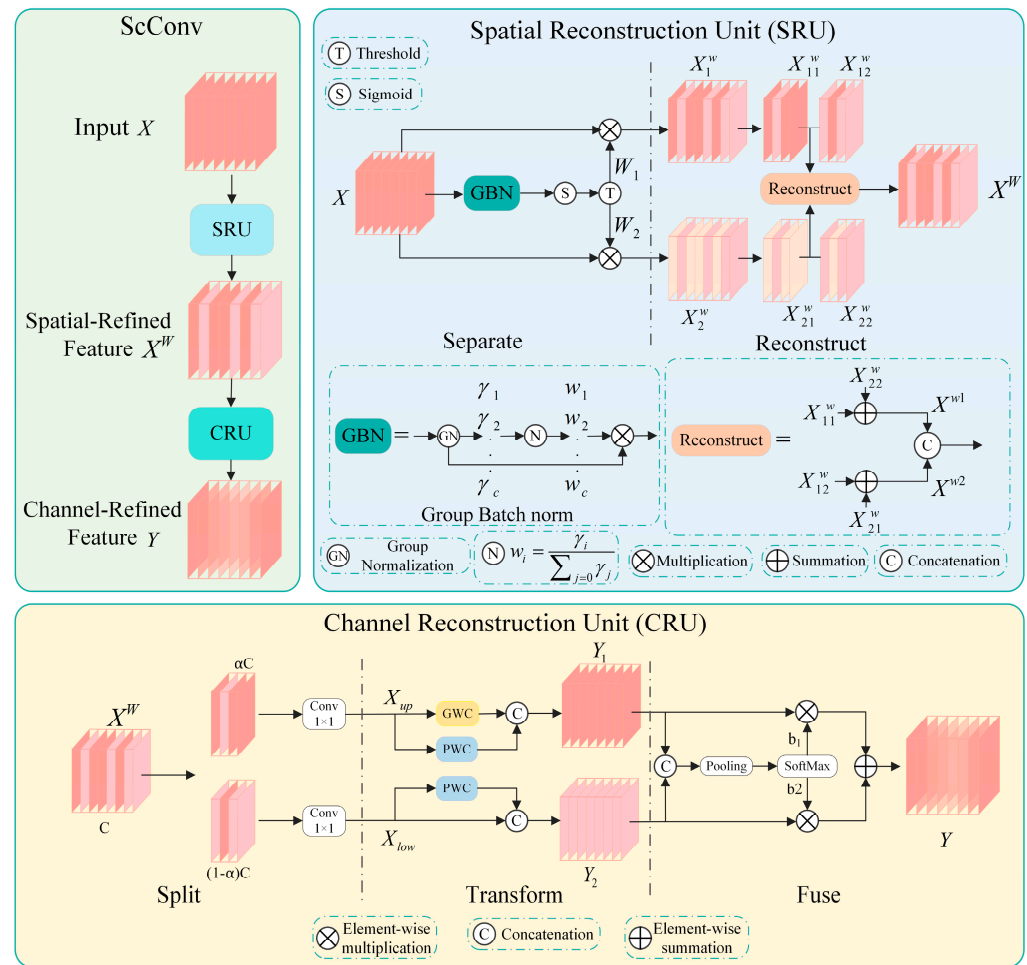


Figure 4. The ScConv module structure diagram. Dark red indicates feature maps with rich information, while light red indicates feature maps with less information.

The ScConv module can utilize spatial and channel redundancies between features to enhance feature map feature representation. The output feature map of the last MBConv6 in the feature extraction module serves as the input to the ScConv module. Firstly, it passes through the SRU to obtain spatial-refined features X^W , then it utilizes the CRU to obtain channel-refined features Y . The SRU separates parts of the feature map that contain rich spatial information from those with relatively less spatial content. Specifically, it evaluates the information content of different feature maps using the Group Batch Normalization (GBN) module. Given an input feature map $X \in R^{N \times C \times H \times W}$, where N is the batch axis, C is the channel axis, and H and W are the height and width axes of the feature map, the operation of the Group Normalization (GN) module is as shown in Equation (2) [47]:

$$X_{out} = GN(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (2)$$

where μ and σ are the mean and standard deviation of X , ϵ is a small natural number, and X and β are trainable affine transformations. Subsequently, the normalized correlation weights of w_c are calculated, which represent the importance of spatial information at different positions in the feature map. Then, the weight coefficients are multiplied by the feature map, normalized using the Sigmoid function, and thresholded to separate them. Those weights normalized above the threshold are set to 1 to obtain the information-rich weight W_1 , while those below the threshold are set to 0 to obtain the weight W_2 with less information. Then, W_1 and W_2 are, respectively, multiplied with feature map X , resulting in feature map X_1^W rich in information and feature map X_2^W with less information. To further

compress the spatial redundancy, the two feature maps are cross-reconstructed by fully combining their information through addition before being connected, resulting in the spatially refined feature map X^W . This approach, superior to directly adding the feature maps, enables a tighter interaction of spatial information between the two feature maps.

The CRU plays a pivotal role in harnessing channel information redundancy to further refine and enhance the channel features of the feature maps. The CRU primarily comprises three modules: Split, Transform, and Fuse. The Split module first divides the given spatially refined feature map into two feature maps with channel numbers denoted as αC and $(1 - \alpha)C$, respectively. Then, both feature maps undergo 1×1 convolutions to adjust the channel numbers to half of the original input feature map, resulting in outputs X_{up} and X_{low} . The Transform module takes X_{up} as the input and processes it through a “strong feature extractor”. The “strong feature extractor” employs Group-wise Convolution (GWC) and pointwise convolution (PWC) instead of regular convolutions. GWC has fewer parameters and computations compared to conventional convolutions but lacks inter-channel information flow, while PWC supplements the inter-channel information flow. The outputs of both operations are summed to obtain Y_1 . Meanwhile, X_{low} is passed into the “weak feature extractor”, which only employs 1×1 PWC to extract some detailed features. Afterwards, it undergoes residual connections to yield Y_2 . The Fuse module combines the two feature maps by concatenating Y_1 and Y_2 along the channel dimension. To extract the global spatial information, the concatenated feature map undergoes global average pooling. This information is utilized to generate feature vectors β_1 and β_2 using SoftMax. These vectors are then multiplied and added to Y_1 and Y_2 , respectively, to obtain the channel-refined feature map Y . The feature maps processed through the ScConv module are enhanced in their representation of important features, significantly improving the detection of steel defects with less prominent characteristics. The overall computation formula for global spatial information $S_m \in R^{c \times 1 \times 1}$ is described in Equation (3) [47].

$$\begin{cases} S_m = Pooling(Y_m) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W Y_c(i, j), m = 1, 2 \\ \beta_1 = \frac{e^{S_1}}{e^{S_1} + e^{S_2}}, \beta_2 = \frac{e^{S_2}}{e^{S_1} + e^{S_2}}, \beta_1 + \beta_2 = 1 \\ Y = \beta_1 Y_1 + \beta_2 Y_2 \end{cases} \quad (3)$$

where S_1 and S_2 represent global channel descriptors, while β_1 and β_2 denote feature vectors. After passing through the ScConv module, the feature representation is enhanced. At this point, the LEM sends the last layer feature map C_5 along with C_4 and C_3 to the neck network for feature fusion. In summary, the proposed LEM is lightweight yet possesses strong feature extraction capabilities. Additionally, the ScConv module utilizes spatial and channel redundancies to enhance feature representation, thereby improving the model learning capability and detection accuracy.

2.3. Feature Pyramid Fusion with a Weighted Bidirectional Approach

Considering the significant scale variations and indistinct features of defects in steel, to enable the model to address the issue of large-scale variations in objects within images, we separately input three feature maps into the detection heads to detect objects at large, medium, and small scales. Generally, shallow feature maps possess higher spatial resolution and carry abundant spatial and positional information but lack distinct semantic features. Conversely, deep feature maps contain rich semantic information but lack sufficient spatial details. Deep feature maps provide the model with abundant semantic information that is used to categorize objects, while shallow feature maps provide the model with abundant texture information that is used to locate objects. Both are crucial for object detection tasks. To further compensate for the resulting accuracy loss, inspired by [48], we employ a BiFPN based on weighted fusion to interactively fuse the three feature maps. Through

weighted fusion, local details, spatial positions, and semantic information are amalgamated, bolstering the representational capacity of semantic features.

As shown in Figure 5, the BiFPN module comprises a set of learnable weight parameters. After receiving feature maps with the same spatial channel size, the module performs weighted summation on each feature map, followed by activation processing using SiLu, and finally convolutional output. BiFPN utilizes a feature propagation structure similar to the Path Aggregation Network (PAN) [45], sequentially transmitting feature information from deep feature maps to shallow ones, and then propagating the fused shallow feature maps back to the deep feature maps. Specifically, BiFPN first processes the deep feature map C_5 through convolution and upsampling to match the shape of C_4 , then performs weighted fusion. Taking the intermediate feature maps C_4 and P_4 as an example, the fusion process is as described by Equation (4) [48].

$$\begin{cases} P_4^{td} = \text{Conv}\left(\frac{w_1 \cdot C_4 + w_2 \cdot \text{Resize}(C_5)}{w_1 + w_2 + \epsilon}\right) \\ P_4 = \text{Conv}\left(\frac{w'_1 \cdot C_4 + w'_2 \cdot P_4^{td} + w'_3 \cdot \text{Resize}(P_3)}{w'_1 + w'_2 + w'_3 + \epsilon}\right) \end{cases} \quad (4)$$

where P_4^{td} is the intermediate feature from the fourth level of the top-down path, P_4 is the output feature from the fourth level of the bottom-up path, w represents the learnable feature fusion coefficient, and ϵ is a very small constant (in this experiment, this coefficient is 0.0001) to prevent division by zero errors. (\cdot) denotes the SiLu activation function. This fusion method allows feature fusion with minimal feature loss and fewer parameters, enabling the network to fully integrate the feature map information while ensuring a lightweight design, which is beneficial for detecting subtle defects in steel materials. The fused three-layer feature maps are then summed through the Cross Stage Partial (CSP) module. The CSP module divides the input into two parts, where one part undergoes two convolution operations and is then concatenated with the other part. This structure amplifies the CNN learning capability and diminishes computational bottlenecks, making it suitable for industrial applications. After enhancing the features through the CSP, the three feature maps are used as inputs to the detection head module. In summary, SCFNet utilizes a weighted BiFPN for feature fusion, carefully controlling the parameter count increases to maintain a lightweight model structure. Furthermore, the experimental results validate the feasibility and efficacy of this approach.

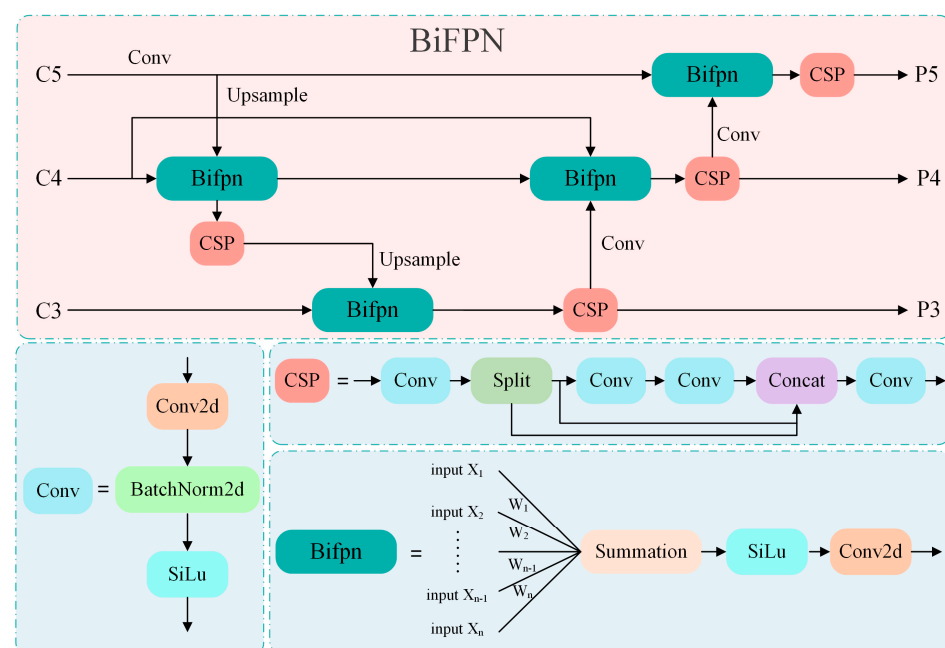


Figure 5. Structure diagram of the BiFPN module.

2.4. Loss Function

The network loss consists of two components [49]: object classification loss L_{cls} and bounding box loss L_{cls} .

$$L_{total} = c_1 L_{cls} + c_2 L_{bbox} \quad (5)$$

where c_1 and c_2 represent the weights of the loss functions. A Binary Cross-Entropy Loss (BCE) is used to calculate the classification loss, while CIoU and distribution focal losses (DFLs) are used to compute the bounding box loss. The formulas for calculation are as follows [49]:

$$L_{cls}(y, p) = y \log(1 - p) - y \log(p) - \log(1 - p) \quad (6)$$

where y represents the actual class of the target, taking values of 0 or 1, and p represents the predicted class of the target, ranging from 0 to 1.

$$L_{bbox} = \lambda_1 L_{CIoU} + \lambda_2 L_{DFL} \quad (7)$$

$$\begin{cases} L_{CIoU} = IoU - \left(\frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \right) \\ v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \\ \alpha = \frac{v}{(1 - IoU) + v} \end{cases} \quad (8)$$

$$L_{DFL}(y_i, y_{i+1}) = -(i + 1 - y) \log(y_i) - (y - i) \log(y_{i+1}) \quad (9)$$

where λ_1 and λ_2 represent the weighting coefficients of the loss. IoU stands for Intersection over Union, ρ denotes the Euclidean distance between the centers of the predicted bounding box and the ground truth bounding box, while c represents the distance between the predicted bounding box and the closest point to the ground truth bounding box's enclosing rectangle. αv stands for the aspect ratio, which is the ratio of width to height, between the predicted bounding box and the ground truth bounding box. (b, b^{gt}) represent the center coordinates of both the predicted and ground truth bounding boxes, while w, h, w^{gt}, h^{gt} denote their respective widths and heights. y denotes the actual label.

3. Experiments

3.1. Datasets

Our proposed defect detection method is evaluated using the NEU-DET [10] dataset to assess its accuracy, robustness, and generalizability. Developed by Northeastern University researchers, the NEU-DET dataset includes six common surface defects in steel. During the manufacturing process of steel plates, six different types of surface defects are commonly encountered. These defects include Scratches (Sc), Inclusion (In), Craze (Cr), Pitted Surface (PS), Patches (Pa), and Rolled-in Scales (RS). There are 300 images of each defect type, each with a resolution of 200×200 pixels, adding up to 1800 images in total.

3.2. Implementation Details

In this article, we conducted experiments using a 16 GB Nvidia RTX 4060 Ti GPU. The deep learning framework utilized was PyTorch 2.0.1. The ratio of the training data, validation data, and testing data was set to 8:1:1. We employed the SGD optimizer with a momentum of 0.937 and a learning rate of 0.01. There was a BatchSize of 32, and the training was conducted for 400 epochs. The code has been open-sourced at <https://github.com/LazyShark2001/SCFNet> (accessed on 25 April 2024).

3.3. Evaluation Metrics

Selecting appropriate evaluation metrics is crucial for assessing the algorithm performance in defect detection. Evaluation metrics should be chosen in a way that objectively measure the algorithm's accuracy and robustness. In practical industrial production, both the accuracy of defect detection and the size of the model are crucial. When the detection accuracy of defects is too low, machines may make incorrect judgments, failing to accu-

rately identify defective workpieces. Additionally, large model sizes can pose deployment challenges on terminal devices. Precision (P), Recall (R), and Mean Average Precision (mAP) are commonly used as metrics to evaluate algorithm performance [4]. Furthermore, to evaluate the complexity and size of the model, we can consider the number of Floating-point Operations (FLOPs) and the number of parameters (Params). FLOPs quantify the computational workload required for inference, while Params represent the number of trainable parameters in the model. These metrics offer insights into the computational efficiency and model complexity, which are essential considerations for deployment on terminal devices and real-world applications.

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$mAP = \frac{\sum_{i=1}^c \int_0^1 P(R) dR}{c} \quad (12)$$

where TP represents the number of correctly classified as positive samples; FP represents the number of incorrectly classified as positive samples; and FN represents the number of incorrectly classified as negative samples. Precision and recall are, respectively, denoted as P and R .

3.4. Comparison with State-of-the-Art Models

We conducted comparative experiments with several mainstream detection algorithms to validate the superiority of our proposed model, including two-stage algorithms such as Faster R-CNN, as well as one-stage algorithms such as YOLOv5s, YOLOv7-tiny, YOLOv8s, CG-Net, and FCCv5s.

In Figure 6, we visually compare our SCFNet (right) with other models on the NEU-DET [10] dataset. Specifically, in the “Crazing” category, our model accurately detects defects. Due to the indistinct features of the targets, other models such as Faster R-CNN and YOLOv5s often lose significant texture information during feature extraction and transformation. This can result in unreliable feature learning and lead to false alarms. SSD and CenterNet models have weak feature extraction capabilities, resulting in missed detections. In the “Inclusion” category, our model accurately detects two defects with high confidence. Our algorithm achieves good visual results in “Patches,” “Pitted Surface,” “Rolled-in Scale,” and “Scratches” without missing detections or false alarms. Compared to other networks, our model successfully identifies defects with ambiguous features (Crazing) and detects low-contrast defects (Inclusion in the sixth row of Figure 6) better, demonstrating its outstanding capability in defect detection.

Table 1 presents the results. In our experimental results, it has been demonstrated that our proposed lightweight and highly efficient steel surface defect detection network, SCFNet, performs better on the NEU-DET dataset when analyzing the P , mAP_{50} , $mAP_{50:95}$, model parameter count, and model computation complexity for the NEU-DET dataset, with values of 0.876, 0.812, 0.469, 5.9, and 2, respectively. Among them, metrics P , mAP_{50} , and $mAP_{50:95}$ perform the best, while the model parameter count and model computational complexity rank second. Compared to the current mainstream detectors, our proposed model achieves a balance between lightweight design and high accuracy in steel defect detection, achieving optimal precision with minimal model parameters and computational complexity.

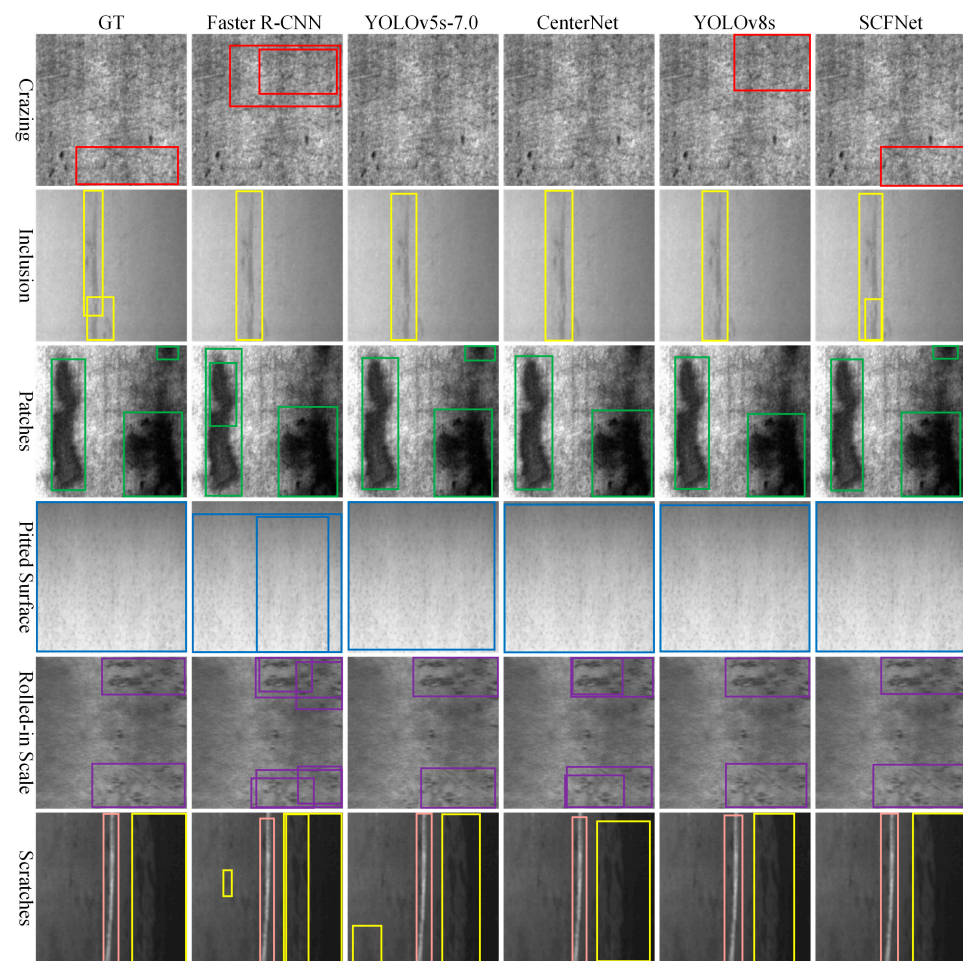


Figure 6. Results of different models compared to SCFNet on NEU-DET [10] dataset. In the picture, the red box represents “Craziing”, the yellow box represents “Inclusion”, the green box represents “Patches”, the blue box represents “Pitted Surface”, the purple box represents “Rolled-in Scales”, and the pink box represents “Scratches”.

Table 1. SCFNet algorithm performance comparison with other object detection algorithms on NEU-DET [10] dataset.

Methods	P	R	mAP ₅₀	mAP _{50:95}	GFLOPs	Params/M
Faster R-CNN [18]	0.615	0.865	0.76	0.377	135	41.75
CenterNet [50]	0.712	0.749	0.764	0.412	123	32.12
YOLOv5n-7.0 [51]	0.694	0.694	0.746	0.422	4.2	1.77
YOLOv5s-7.0 [51]	0.745	0.719	0.761	0.429	15.8	7.03
YOLOv7-tiny [25]	0.645	0.775	0.753	0.399	13.1	6.02
YOLOv8s [49]	0.768	0.726	0.795	0.467	28.4	11.13
YOLOX-tiny [52]	0.746	0.768	0.76	0.357	7.58	5.03
MRF-YOLO [53]	0.761	0.707	0.768	-	29.7	14.9
YOLOv5s-FCC [54]	-	-	0.795	-	-	13.35
WFRE-YOLOv8s [55]	0.759	0.736	0.794	0.425	32.6	13.78
CG-Net [56]	0.734	0.687	0.759	0.399	6.5	2.3
ACD-YOLO [57]	-	-	0.793	-	21.3	-
YOLOv5-ESS [58]	-	0.764	0.788	-	-	7.07
PMSA-DyTr [2]	-	-	0.812	-	-	-
MED-YOLO [4]	-	-	0.731	0.376	18	9.54
MAR-YOLO [15]	-	-	0.785	-	20.1	-
SCFNet	0.786	0.715	0.812	0.469	5.9	2

Red bold indicates the top-ranking performance, while **blue** bold indicates the second-ranking performance.

Further validating our proposed SCFNet across different defect categories, we conducted comparison experiments with mainstream detection algorithms on the GC10-DET dataset [59]. The specific experimental results and performance are shown in Table 2.

Table 2. Performance comparison of SCFNet algorithm and other object detection algorithms on GC10-DET dataset [59].

Methods	P	R	mAP ₅₀	mAP _{50:95}	GFLOPs	Params/M
Faster R-CNN [18]	0.579	0.656	0.652	0.293	135	41.75
YOLOv5n-7.0 [51]	0.729	0.666	0.699	0.366	4.2	1.77
YOLOv7-tiny [25]	0.707	0.657	0.697	0.344	13.1	6.02
CenterNet [50]	0.726	0.619	0.665	0.308	78.66	32.12
YOLOv8n [49]	0.704	0.65	0.684	0.365	8.1	3.01
YOLOX-tiny [52]	0.659	0.546	0.611	0.259	7.58	5.03
MAR-YOLO [15]	-	-	0.673	-	20.1	-
SCFNet	0.713	0.68	0.704	0.366	5.9	2

Red bold indicates the top-ranking performance, while **blue** bold indicates the second-ranking performance.

GC10-DET is a dataset of steel surface defects obtained from real industrial environments. This dataset contains 3570 grayscale images of defects in steel plates. The experimental setup is consistent with Section 3.2. According to Table 2, our proposed SCFNet achieves high performance on the GC10-DET dataset, with the model parameter count and computational cost only second to YOLOv5n. The SCFNet upholds detection accuracy while possessing a smaller model size and lower computational cost, rendering it well suited for deployment on terminal detection devices with limited computing capability.

3.5. Data Augmentation Module Discussion

Considering the limited availability and scale of publicly available datasets on industrial steel surface defects, training networks with limited data may result in lower robustness and difficulty in detecting blurry samples. In order to investigate the impact of various augmentation techniques on the accuracy of the model, we conducted data augmentation on the steel surface defect dataset. The data are augmented by six different techniques, as illustrated in Table 3, including flipping transformation, shifting transformation, adding noise transformation, adjusting brightness transformation, rotating transformation, and combining the above techniques. Each augmentation technique doubled the dataset, increasing the original training set of 1440 images to 2880 images.

Table 3. Data augmentation results.

Methods	Augment	mAP ₅₀	mAP _{50:95}
SCFNet	Original	0.778	0.448
SCFNet	Shift	0.785	0.45
SCFNet	Noise	0.781	0.441
SCFNet	Brightness	0.785	0.45
SCFNet	Rotation	0.767	0.454
SCFNet	Flip	0.812	0.469
SCFNet	All	0.797	0.458

Red bold indicates the top-ranking performance, while **blue** bold indicates the second-ranking performance.

In Table 3, most data augmentation techniques resulted in varying degrees of improvements in the model performance, whereas rotation augmentation reduced the model accuracy. This discrepancy could arise from inconsistencies in size ratios between the rotated images and the original ones, resulting in the distortion of targets when forcibly resized to a consistent size during network preprocessing. However, other data augmentation methods showed improvements in results. Among them, flipping augmentation achieved the highest accuracy improvement, with an mAP₅₀ reaching 0.812. This might be

because in steel defect detection, where defect features may not be prominent, techniques like adding noise, adjusting brightness, and others might make it challenging for the model to propagate gradients correctly; shift could alter image sizes, potentially causing feature loss around the targets. However, flipping augmentation does not cause these issues. Therefore, flip augmentation appears to maximize the detection performance of models on the NEU-DET [10] dataset.

3.6. Ablation Study

To confirm the roles of each module, we conducted ablation studies on the NEU-DET [10] dataset. Using YOLOv8n as a baseline, we replaced the backbone network for feature extraction with the LEM to reduce the model complexity. As a final layer in the feature extraction module, we introduce the ScConv module to enhance the ability to extract features. We also employed BiFPN as a feature fusion network, retaining more original information. As this network is a lightweight detector, ablation studies on the BiFPN and ScConv modules are conducted on the LEM. Table 4 shows the experimental results.

Table 4. Ablation experiment results on the NEU-DET [10] dataset.

Model	LEM	ScConv	BiFPN	mAP ₅₀	mAP _{50:95}	GFLOPs	Params/M
Baseline	-	-	-	0.773	0.444	8.1	3.01
Baseline	✓	-	-	0.783	0.457	5.7	1.9
Baseline	✓	✓	-	0.787	0.455	5.7	1.91
Baseline	✓	-	✓	0.793	0.455	5.9	1.99
Baseline	✓	✓	✓	0.8	0.456	5.9	2

Red bold indicates the top-ranking performance, while **blue** bold indicates the second-ranking performance.

LEM: By replacing the feature extraction module of YOLOv8n with the LEM, the number of model parameters decreased from 3.01 M to 1.9 M, while the gigaflops (GFLOPs) decreased from 8.1 to 5.7. Additionally, mAP₅₀ increased from 0.773 to 0.783, and mAP_{50:95} increased from 0.444 to 0.457. The LEM utilizes Depth-Wise convolution and SE modules for feature extraction, with fewer connections between different blocks and the avoidance of branching structures. A replacement of the backbone network of YOLOv8n with the LEM improves the model detection accuracy while maintaining a lightweight design and reducing the computational requirements.

ScConv Module: ScConv operates on the deepest layer of feature maps, removing redundant spatial and channel information from feature maps and enhancing their representational capacity. Steel surface defect features are not prominent, leading to potential false positives or negatives. By strengthening the representational capacity of the feature maps through the ScConv module, the model can more easily detect steel surface defects. Figure 7 illustrates a comparison of heatmaps with and without the ScConv module. Heatmaps depict the model prediction results for each pixel, typically using colors to indicate the level of confidence associated with each pixel. Warmer tones, such as red, are used to represent pixels with higher confidence, while cooler tones, such as blue, are used to represent pixels with lower confidence. Additionally, heatmaps aid in analyzing model detection results, highlighting areas that are easier to detect or overlook. Features of defects such as crazing and patches are not prominent, making them difficult for the model to recognize. With the addition of the ScConv module, however, the representational capacity of the feature maps is enhanced, thereby improving the model detection ability. In the ablation experiments, adding the ScConv module increased the model mAP₅₀ from 0.783 to 0.787, with a minimal increase in the model parameters and computational load. As a result, the ScConv module has a low number of model parameters and a low computational load, but significantly enhances the network's feature representation capability, resulting in a more accurate model.

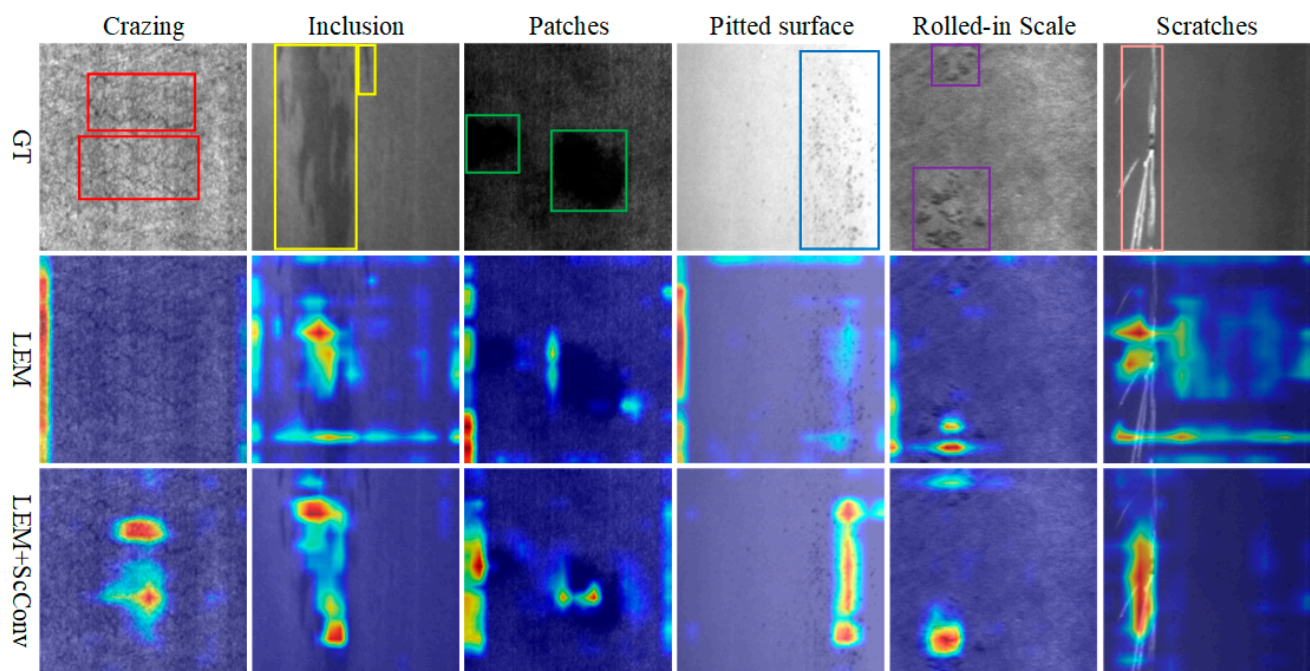


Figure 7. Comparison of heatmaps under ablation of the ScConv module (from the NEU-DET [10] dataset). In the picture, the red box represents “Crazing”, the yellow box represents “Inclusion”, the green box represents “Patches”, the blue box represents “Pitted Surface”, the purple box represents “Rolled-in Scales”, and the pink box represents “Scratches”.

BiFPN: In Table 4, the comparison between the second and fifth rows clearly demonstrates the effectiveness of using BiFPN. The mAP_{50} increased from 0.783 to 0.793 (from the second to the fourth row with BiFPN) and from 0.787 to 0.8 (from the third to the fifth row with BiFPN), while the increase in the model parameters and computational load is minimal. The BiFPN uses unique skip connections and weighted feature fusion mechanisms, allowing the neck network to reuse feature maps and better combine semantic and texture features. This improvement enhances the detection accuracy. Using fewer parameters, BiFPN significantly improves the accuracy by slightly increasing the computational load and parameter count, resulting in a better balance between lightweight design and accuracy.

4. Discussion

Some defective images restrict the detection performance, as depicted in Figure 8 showing cases of detection failure. Defects with low contrast and unclear features in steel materials can lead to missed detections (see Case 1 and 2 of Figure 8). Additionally, there exist defects in steel materials that are highly similar to the background, which can result in false detections (see Case 3 and 4 of Figure 8). In our future work, we intend to incorporate a learnable image enhancement module into the model to improve the detection accuracy of defects with low contrast. Furthermore, we plan to continue researching more effective feature extraction modules to enhance the effectiveness of our approach.

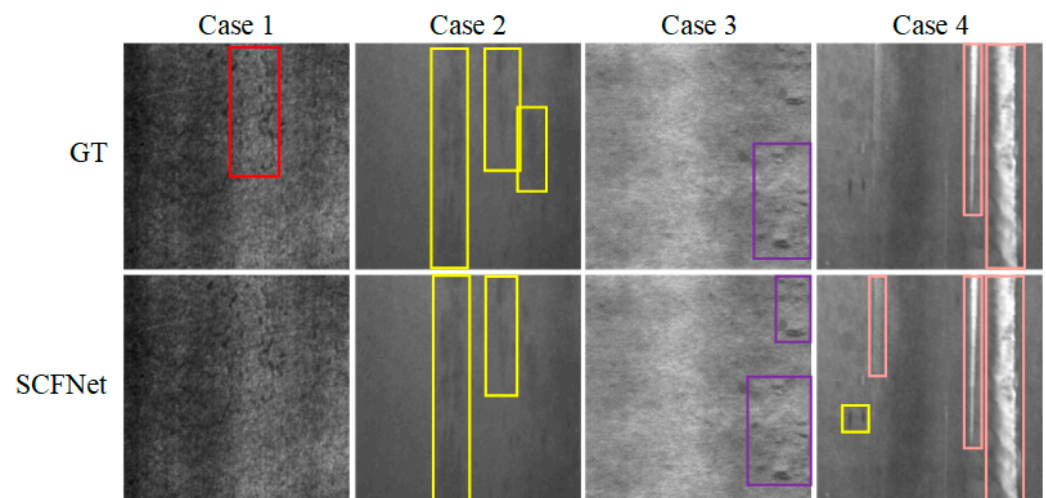


Figure 8. Some failure cases of SCFNet (from the NEU-DET [10] dataset). In the picture, the red box represents “Crazing”, the yellow box represents “Inclusion”, the purple box represents “Rolled-in Scales”, and the pink box represents “Scratches”.

5. Conclusions

Addressing ambiguous defects and low-contrast defects in steel, while accurately identifying defects with similar features but different categories, is crucial for modern industrial production. This article proposes a lightweight steel defect detection algorithm called SCFNet to tackle the aforementioned challenges. To achieve a lightweight defect detection model, SCFNet utilizes the LEM as a feature extraction module. This module is based on Depth-Wise convolution with channel weighting, resulting in stronger capabilities in extracting ambiguous features. We use convolutional structures based on spatial and channel recombination to process the deepest layer feature maps, reducing redundancy and enhancing the model feature representation capability. This module facilitates effective feature representation while disregarding noise information. To preserve more defect texture information, a weighted bidirectional feature pyramid fusion structure is adopted in the neck of the network for feature fusion. In addition, it retains more original content by employing a more effective information propagation mechanism. The experimental results show that on the NEU-DET dataset, compared with most deep learning detection methods, the SCFNet algorithm achieves the highest mAP_{50} metric of 81.2%, the highest $mAP_{50:95}$ metric of 46.9%, the smallest model parameters of 2 M, and the least model computation of 5.9 GFLOPs. SCFNet also achieves the highest accuracy and the smallest computation and model parameters on the GC10-DET dataset. SCFNet demonstrates excellent performance, making it more suitable for practical applications in industrial production.

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

Funding: This research is funded by Hubei Province Young Science and Technology Talent Morning Hight Lift Project (202319); Open Research Fund Program of LIESMARS (Grant No. 21E02); Doctoral Starting up Foundation of Hubei University of Technology (XJ2023007301); Natural Science Foundation of Hubei Province (2022CFB501); University Student Innovation and Entrepreneurship Training Program Project (202210500028).

Data Availability Statement: Data are contained within the article.

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

References

- Guo, B.; Wang, Y.; Zhen, S.; Yu, R.; Su, Z. SPEED: Semantic prior and extremely efficient dilated convolution network for real-time metal surface defects detection. *IEEE Trans. Ind. Inform.* **2023**, *19*, 11380–11390. [\[CrossRef\]](#)
- Su, J.; Luo, Q.; Yang, C.; Gui, W.; Silvén, O.; Liu, L. PMSA-DyTr: Prior-Modulated and Semantic-Aligned Dynamic Transformer for Strip Steel Defect Detection. *IEEE Trans. Ind. Inform.* **2024**, *20*, 6684–6695. [\[CrossRef\]](#)
- Luo, Q.; Fang, 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. [\[CrossRef\]](#)
- Li, Z.; Wei, X.; Hassaballah, M.; Li, Y.; Jiang, X. A deep learning model for steel surface defect detection. *Complex Intell. Syst.* **2024**, *10*, 885–897. [\[CrossRef\]](#)
- Sampath, V.; Murtua, I.; Martín, J.J.A.; Rivera, A.; Molina, J.; Gutierrez, A. Attention guided multi-task learning for surface defect identification. *IEEE Trans. Ind. Inform.* **2023**, *19*, 9713–9721. [\[CrossRef\]](#)
- Wen, L.; Wang, Y.; Li, X. A new cycle-consistent adversarial networks with attention mechanism for surface defect classification with small samples. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8988–8998. [\[CrossRef\]](#)
- Lian, J.; Jia, W.; Zareapoor, M.; Zheng, Y.; Luo, R.; Jain, D.K.; Kumar, N. Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. *IEEE Trans. Ind. Inform.* **2019**, *16*, 1343–1351. [\[CrossRef\]](#)
- Zhang, D.; Song, K.; Wang, Q.; He, Y.; Wen, X.; Yan, Y. Two deep learning networks for rail surface defect inspection of limited samples with line-level label. *IEEE Trans. Ind. Inform.* **2020**, *17*, 6731–6741. [\[CrossRef\]](#)
- Shen, K.; Zhou, X.; Liu, Z. MINet: Multiscale Interactive Network for Real-Time Salient Object Detection of Strip Steel Surface Defects. *IEEE Trans. Ind. Inform.* **2024**, 1–11. [\[CrossRef\]](#)
- Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [\[CrossRef\]](#)
- Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [\[CrossRef\]](#)
- Luo, Q.; Su, J.; Yang, C.; Silvén, O.; Liu, L. Scale-selective and noise-robust extended local binary pattern for texture classification. *Pattern Recognit.* **2022**, *132*, 108901. [\[CrossRef\]](#)
- Lu, H.-P.; Su, C.-T. CNNs combined with a conditional GAN for Mura defect classification in TFT-LCDs. *IEEE Trans. Semicond. Manuf.* **2021**, *34*, 25–33. [\[CrossRef\]](#)
- Wen, L.; Li, X.; Gao, L. A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification. *IEEE Trans. Ind. Electron.* **2020**, *68*, 12890–12900. [\[CrossRef\]](#)
- Zhang, H.; Li, S.; Miao, Q.; Fang, R.; Xue, S.; Hu, Q.; Hu, J.; Chan, S. Surface defect detection of hot rolled steel based on multi-scale feature fusion and attention mechanism residual block. *Sci. Rep.* **2024**, *14*, 7671. [\[CrossRef\]](#)
- 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, 24–27 June 2014; pp. 580–587.
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
- 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 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- 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; pp. 21–37.
- 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, 26 June–1 July 2016; pp. 779–788.
- Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
- Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
- Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
- 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, 18–22 June 2023; pp. 7464–7475.
- Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
- Shumin, D.; Zhoufeng, L.; Chunlei, L. AdaBoost learning for fabric defect detection based on HOG and SVM. In Proceedings of the 2011 International Conference on Multimedia Technology, Hangzhou, China, 26–28 July 2011; pp. 2903–2906.
- Chondronasios, A.; Popov, I.; Jordanov, I. Feature selection for surface defect classification of extruded aluminum profiles. *Int. J. Adv. Manuf. Technol.* **2016**, *83*, 33–41. [\[CrossRef\]](#)

29. Liang, Y.; Xu, K.; Zhou, P. Mask gradient response-based threshold segmentation for surface defect detection of milled aluminum ingot. *Sensors* **2020**, *20*, 4519. [CrossRef]
30. Wu, X.-Y.; Xu, K.; Xu, J.-W. Application of undecimated wavelet transform to surface defect detection of hot rolled steel plates. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; pp. 528–532.
31. Zhao, T.; Chen, X.; Yang, L. IPCA-SVM based real-time wrinkling detection approaches for strip steel production process. *Int. J. Wirel. Mob. Comput.* **2019**, *16*, 160–165. [CrossRef]
32. Gong, R.; Wu, C.; Chu, M. Steel surface defect classification using multiple hyper-spheres support vector machine with additional information. *Chemom. Intell. Lab. Syst.* **2018**, *172*, 109–117. [CrossRef]
33. Chu, M.; Liu, X.; Gong, R.; Liu, L. Multi-class classification method using twin support vector machines with multi-information for steel surface defects. *Chemom. Intell. Lab. Syst.* **2018**, *176*, 108–118. [CrossRef]
34. Zhang, J.; Wang, H.; Tian, Y.; Liu, K. An accurate fuzzy measure-based detection method for various types of defects on strip steel surfaces. *Comput. Ind.* **2020**, *122*, 103231. [CrossRef]
35. Mei, L.; Hu, X.; Ye, Z.; Tang, L.; Wang, Y.; Li, D.; Liu, Y.; Hao, X.; Lei, C.; Xu, C. GTMFuse: Group-Attention Transformer-Driven Multiscale Dense Feature-Enhanced Network for Infrared and Visible Image Fusion. *Knowl. Based Syst.* **2024**, *293*, 111658.
36. Xu, C.; Ye, Z.; Mei, L.; Yu, H.; Liu, J.; Yalikun, Y.; Jin, S.; Liu, S.; Yang, W.; Lei, C. Hybrid Attention-Aware Transformer Network Collaborative Multiscale Feature Alignment for Building Change Detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–14. [CrossRef]
37. Yang, W.; Mei, L.; Ye, Z.; Wang, Y.; Hu, X.; Zhang, Y.; Yao, Y. Adjacent Self-Similarity Three-dimensional Convolution for Multi-modal Image Registration. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 6002505. [CrossRef]
38. Xu, C.; Ye, Z.; Mei, L.; Shen, S.; Sun, S.; Wang, Y.; Yang, W. Cross-Attention Guided Group Aggregation Network for Cropland Change Detection. *IEEE Sens. J.* **2023**, *23*, 13680–13691. [CrossRef]
39. Yang, W.; Shen, P.; Ye, Z.; Zhu, Z.; Xu, C.; Liu, Y.; Mei, L. Adversarial Training Collaborating Multi-Path Context Feature Aggregation Network for Maize Disease Density Prediction. *Processes* **2023**, *11*, 1132. [CrossRef]
40. Zhao, C.; Shu, X.; Yan, X.; Zuo, X.; Zhu, F. RDD-YOLO: A modified YOLO for detection of steel surface defects. *Measurement* **2023**, *214*, 112776. [CrossRef]
41. Wang, R.; Liang, F.; Mou, X.; Chen, L.; Yu, X.; Peng, Z.; Chen, H. Development of an improved yolov7-based model for detecting defects on strip steel surfaces. *Coatings* **2023**, *13*, 536. [CrossRef]
42. Yu, Z.; Wu, Y.; Wei, B.; Ding, Z.; Luo, F. A lightweight and efficient model for surface tiny defect detection. *Appl. Intell.* **2023**, *53*, 6344–6353. [CrossRef]
43. Liu, G.-H.; Chu, M.-X.; Gong, R.-F.; Zheng, Z.-H. DLF-YOLOF: An improved YOLOF-based surface defect detection for steel plate. *J. Iron Steel Res. Int.* **2023**, *31*, 442–451. [CrossRef]
44. Shao, Y.; Fan, S.; Sun, H.; Tan, Z.; Cai, Y.; Zhang, C.; Zhang, L. Multi-Scale Lightweight Neural Network for Steel Surface Defect Detection. *Coatings* **2023**, *13*, 1202. [CrossRef]
45. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
46. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
47. Li, J.; Wen, Y.; He, L. Sconv: Spatial and channel reconstruction convolution for feature redundancy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6153–6162.
48. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
49. Ultralytics/Ultralytics. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 29 October 2023).
50. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
51. Jocher, G. Stoken YOLOv5. Available online: <https://github.com/ultralytics/yolov5/releases/tag/v7.0> (accessed on 20 November 2023).
52. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
53. Xia, K.; Lv, Z.; Zhou, C.; 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] [PubMed]
54. Yu, J.; Wang, C.; Xi, T.; Ju, H.; Qu, Y.; Kong, Y.; Chen, X. Development of an Algorithm for Detecting Real-Time Defects in Steel. *Electronics* **2023**, *12*, 4422. [CrossRef]
55. Huang, Y.; Tan, W.; Li, L.; Wu, L. WFRE-YOLOv8s: A New Type of Defect Detector for Steel Surfaces. *Coatings* **2023**, *13*, 2011. [CrossRef]
56. Wang, H.; Yang, X.; Zhou, B.; Shi, Z.; Zhan, D.; Huang, R.; Lin, J.; Wu, Z.; Long, D. Strip surface defect detection algorithm based on YOLOV5. *Materials* **2023**, *16*, 2811. [CrossRef]
57. Fan, J.; Wang, M.; Li, B.; Liu, M.; Shen, D. ACD-YOLO: Improved YOLOv5-based method for steel surface defects detection. *IET Image Process.* **2024**, *18*, 761–771. [CrossRef]

-
58. Ren, F.; Fei, J.; Li, H.; Doma, B.T. Steel Surface Defect Detection Using Improved Deep Learning Algorithm: ECA-SimSPPF-SIoU-Yolov5. *IEEE Access* **2024**, *12*, 32545–32553. [[CrossRef](#)]
 59. Lv, X.; Duan, F.; Jiang, J.-J.; Fu, X.; Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **2020**, *20*, 1562. [[CrossRef](#)] [[PubMed](#)]

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.