applied sciences

MDPI

Article

DEW-YOLO: An Efficient Algorithm for Steel Surface Defect Detection

Junjie Li [1,2] and Mingxia Chen [1,2,*]

[1] Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin 541006, China; 2120221192@glut.edu.cn

[2] Guangxi Engineering Research Center of Intelligent Rubber Equipment, Guilin University of Technology, Guilin 541006, China

* Correspondence: wjunt@sohu.com

Abstract: To address the current steel surface defect detection algorithms in practical applications involving low detection accuracy, an efficient and highly accurate strip steel surface defect detection algorithm, DEW-YOLO, is proposed in this paper. Firstly, by combining the advantages of deformable convolutional networks (DCNs), this paper innovates the C2F module in YOLOv8 and proposes a C2f_DCN module that can flexibly sample features to enhance the abilities of learning and expressing defect features of different sizes and shapes. Secondly, the explicit visual center (EVC) is introduced into the backbone network, which enhances feature extraction capabilities and adaptability and enables the model to better adjust features at different levels and scales. Finally, the original loss function is replaced with the Wise-IoU (WIoU) loss function to accurately measure the similarity between the target frames and improve the defect detection performance of the model. The experimental results on the NEU-DET dataset demonstrate that the algorithms proposed in this paper achieved a mean average precision (mAP) of 80.3% in steel surface defect detection tasks, which was a 3.9% improvement over the original YOLOv8 model. The model's inference speed reached 91 frames per second (FPS). DEW-YOLO effectively enhances the accuracy of steel defect detection and better satisfies industrial inspection requirements.

Keywords: steel surface defect detection; YOLOv8; deformable convolution; explicit visual center; Wise-IoU

Citation: Li, J.; Chen, M. DEW-YOLO: An Efficient Algorithm for Steel Surface Defect Detection. *Appl. Sci.* **2024**, *14*, 5171. https://doi.org/10.3390/app14125171

Academic Editor: Paulo Santos

Received: 19 May 2024
Revised: 10 June 2024
Accepted: 11 June 2024
Published: 14 June 2024

1. Introduction

Steel is an essential and important material for national construction with widespread applications, particularly in infrastructure development, where it plays an irreplaceable role. Currently, the majority of steel materials are produced through pressure processing of raw materials to obtain materials with specific shapes and properties. During the process of pressure processing, steel is inevitably subject to defects due to factors such as cost constraints, equipment limitations, existing technologies, and surface roughness [1]. These defects may include cracks, inclusions, patches, pitting, and other imperfections [2]. Once these defects appear in the steel, its ability to withstand compression, tension, corrosion, and other factors will inevitably be affected to varying degrees. Therefore, defect detection is particularly important before the production of steel materials.

Currently, defect detection methods can be broadly categorized into three types: traditional defect detection, machine vision-based defect detection, and deep learning-based defect detection. Traditional defect detection methods mainly contain manual sampling [3], infrared detection [4], and magnetic flux leakage detection [5]. The artificial sampling method randomly chooses and tests the sample with the naked eye, but there are sampling imbalances, resulting in large errors. This is highly susceptible to human factors and other issues. The infrared detection method detects defects through temperature changes

caused by different surfaces of the steel material, which is limited to the infrared absorption capacity of steel and is unable to accurately classify defects. Magnetic flux leakage detection tests defects by detecting the presence of magnetic features on defective steel surfaces. This technique is not capable of accurate identification of fine and narrow cracks, thus limiting the types of defects which can be detected.

Machine vision inspection technology is an improvement to traditional inspection which reduces staff workload and is not affected by the environment. Machine vision-based defect detection methods are mainly classified into four categories: defect detection based on image preprocessing, the classifier, feature extraction, and image segmentation. Machine vision inspection methods have made tremendous progress in the task of recognizing steel surface defects, which are more efficient and practical than traditional defect detection methods. However, they generally have problems such as poor classification of multiple defect types, the need for manual extraction of features, and performing well in real time but having poor accuracy.

Therefore, deep learning-based defect detection technology has emerged as an important research direction. Deep learning inputs data and automatically extracts features, not only retaining the advantages of no manual labor and avoiding an environmental impact but also eliminating the need to manually extract features. This end-to-end modeling architecture simplifies the industrial production process. The mainstream deep learning target detection algorithms include two-stage [6] methods such as R-CNN [7], Fast R-CNN [8], Faster R-CNN [9], and Mask R-CNN [10], and one-stage [11] methods like SSD [12], the YOLO [13] algorithm, and object detection algorithms based on the Transformer architecture, like DETR [14]. The two-stage target detection algorithm has a relatively high detection accuracy, but it detects targets more slowly than the one-stage target detection algorithm because it generates a series of candidate frames during the detection process. The one-stage target detection algorithm turns them into a regression problem. Zhao et al. [15] proposed a Faster R-CNN algorithm based on deformable convolution and multi-scale fusion. While their proposed model performed well in detecting surface defects on steel, it struggled to meet industrial requirements in terms of detection speed. The accuracy of the one-stage target detection algorithm is relatively low, but it has a high detection speed, making it well suited to the task of detecting steel surface defects in industry. Lin et al. [16] proposed an improved SSD model to enhance the performance of steel surface defect detection and a deep residual network (ResNet) for defect classification. Kou et al. [17] proposed an improved YOLO-V3 defect detection model which utilizes the anchor-free mechanism to optimize the detection performance at multiple scales and shorten the detection time. Ling Wang et al. [18] proposed an improved YOLOv5 method for real-time detection of surface defects on strip steel which uses an attention mechanism and a multi-scale detection block to enhance the detection capability of the model. The detection speed of the model was improved to a certain extent, and it had an mAP value of 72% in the steel surface defect detection task. Kewen Xia et al. [19] introduced the reparameterized C3 module into the C3 module of YOLOv5s and constructed a feature fusion structure using the multipath spatial pyramid pooling module, which resulted in improved model detection accuracy. The model had an mAP of 76.8% on the NEU-DET dataset, and the detection speed was relatively fast. Yi Qu et al. [20] proposed an improved PP-YOLOE model for small targets. The model introduces coordinate attention into the backbone structure and simplifies the traditional PAN + FPN component into an optimized FPN feature pyramid structure, which improved the performance and overall accuracy of the model for small target detection. Zheng et al. [21] proposed the EW-YOLOv7 defect detection model. The model combines the GhostNetV2 module and the Acmix attention machine to effectively optimize the performance of the model's defect detection.

Although these detection methods, compared with traditional defect detection and machine vision defect detection, were improved in terms of detection speed, the detection accuracy still could not meet practical work requirements. Aiming at the existing algorithms that are difficult to practically apply in industrial environments due to factors such as low

accuracy and slow speed in steel defect detection, this paper proposes a new algorithm for detecting surface defects on strip steel, DEW-YOLO, based on the YOLOv8n algorithm. In summary, the main contributions of this paper are as follows:

- The DCN is introduced and combined with the C2F module to form the C2F_DCN module, which enables the model to be more flexible in extracting the features of complex textures and irregularly shaped defects.
- We embed the EVC structure into the backbone network of YOLOv8 to achieve adjustment and optimization for different feature scales. This enables the model to better capture global information and local details at various scales, thereby enhancing the accuracy and robustness of object detection.
- The WIoU loss function is used to replace the original YOLOv8 bounding box loss function, providing more stable and reliable detection results, especially for real industrial inspection scenarios.

2. Materials and Methods

2.1. Datasets

This study employed the NEU-DET [22] dataset from Northeastern University, which consists of six common steel surface defects: crazing (Cr), inclusion (In), patches (Pa), a pitted surface (PS), a rolled-in scale (RS), and scratches (Sc). There are 300 images for each category, totaling 1800 images with an image size of 200 × 200. The dataset is provided with annotations labeling the location of the defective image and category information. For better experimentation, 1440 images were randomly selected for the training set, with 180 images for the validation set and 180 images for the testing set (i.e., the dataset was allocated at a ratio of 8:1:1. The categories of steel surface defects in the NEU-DET dataset are illustrated in Figure 1.

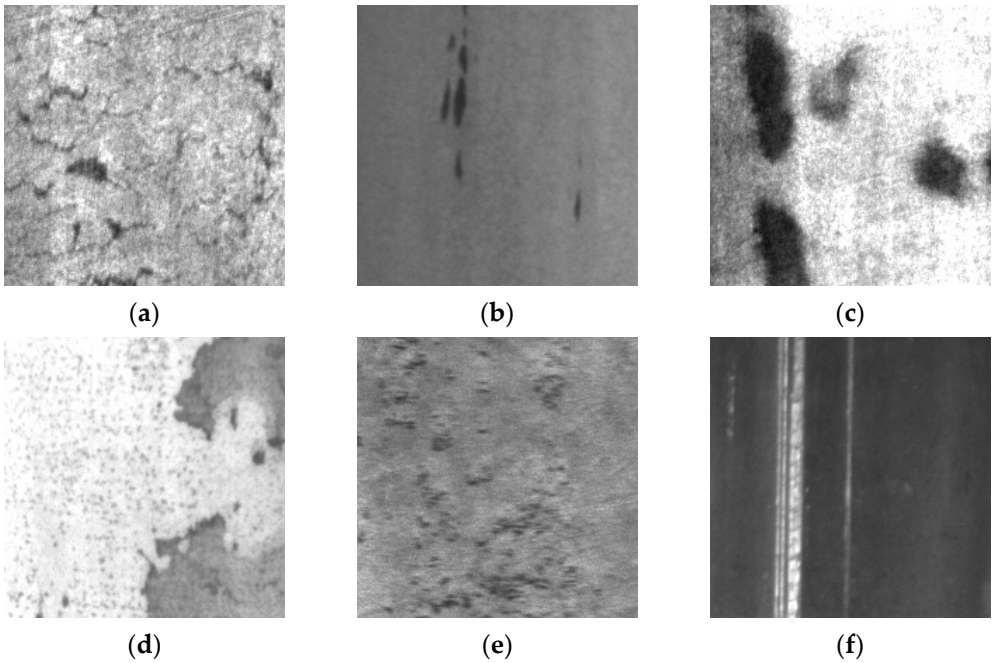


Figure 1. Six types of surface defects on steel strips: (**a**) crazing; (**b**) inclusion; (**c**) patches; (**d**) pitted surface; (**e**) rolled-in scale; (**f**) scratches.

2.2. YOLO Algorithm

A real-time object detection technique called You Only Look Once (YOLO) splits an image into grid cells and predicts the bounding boxes and classes for items in each cell at the same time, making object detection quick and precise. YOLOv1 is the first version of the YOLO algorithm, which utilizes a single-scale fully convolutional neural network for

object detection, known for its real-time and high efficiency characteristics. Building upon YOLOv1, YOLOv2 [23] introduced multiscale feature maps and anchor boxes, enhancing detection accuracy and robustness. YOLOv3 [24] further improved the network structure by adopting the deeper Darknet-53 structure as the feature extraction network and introducing multiscale prediction and cross-scale connections, thereby enhancing detection speed and accuracy. YOLOv4 [25] introduced the CSPDarknet53 and SPP structures based on YOLOv3, further improving detection performance and speed. YOLOv5 [26] enhanced the detection speed and robustness by introducing a lighter network structure and multiscale training. YOLOv7 [27] introduced strategies such as the E-ELAN, a scalable and efficient layer aggregation network, an innovative transition module, and a parameterized structure to enhance the capability of feature extraction and semantic information representation, further optimizing the target detection results.

The YOLO algorithm has undergone iterations and has been upgraded to YOLOv8, offering models of different sizes, including N, S, M, L, and X scales based on the scaling factors. The architecture of the YOLOv8 algorithm draws inspiration from the design philosophy of the YOLOv7 ELAN. It incorporates the C2f structure from YOLOv5 into the backbone and neck networks to obtain a richer gradient flow. In the head network, YOLOv8 separates the classification and detection heads with a decoupled head structure. In addition, YOLOv8 employs the TaskAlignedAssigner strategy for positive sample assignment and introduces the distribution focal loss as a regression loss. With these improvements, YOLOv8 has made significant progress in terms of loss computation and network structure. It is evident that the official Ultralytics team has made considerable efforts in achieving a balance between model lightweighting and performance. Figure 2 illustrates the network architecture of YOLOv8.

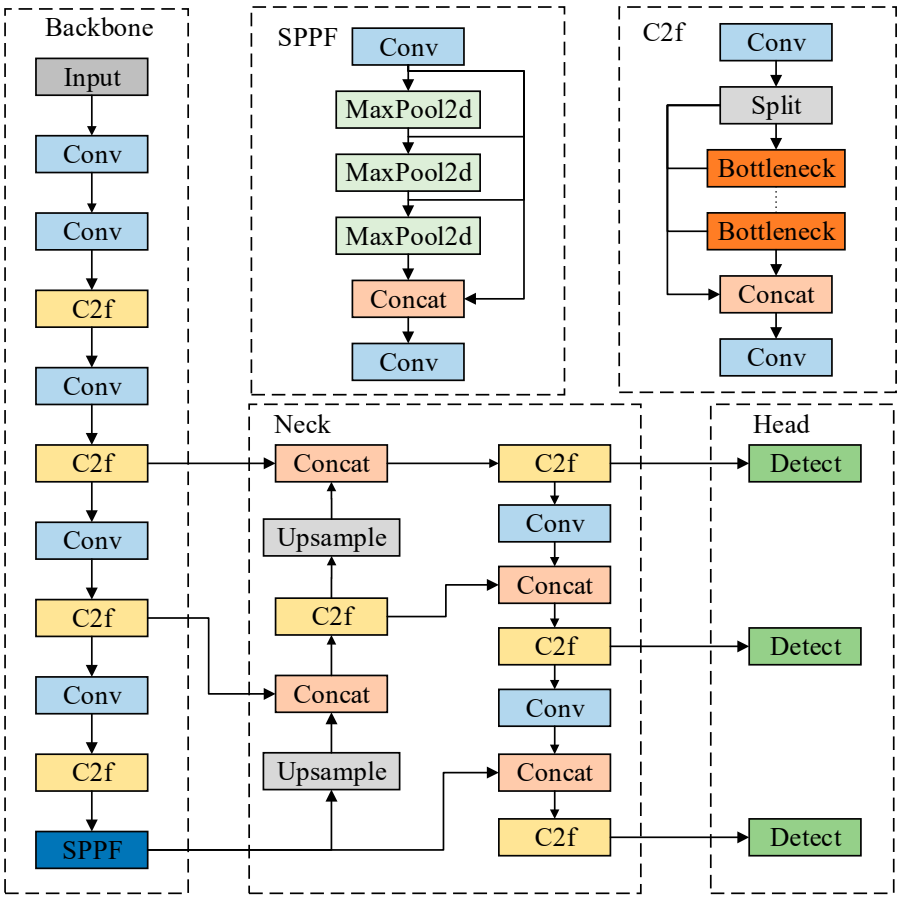


Figure 2. Network structure diagram of YOLOv8 algorithm.

3. The Proposed Method

To increase the performance, efficiency, and generalization capability of the defect detection model, this paper improves upon the YOLOv8n model and proposes a new model for the detection of steel strips named DEW-YOLO, whose network structure is illustrated in Figure 3. Firstly, the C2f_DCN module takes the place of the original C2f module in the backbone network of YOLOv8n, enhancing the model's capability to capture the features of complex shapes and irregular objects. Then, the ECV module is introduced. By implementing global centralization adjustment at different levels, the model can better utilize global information from the deepest layers to adjust shallow features, thereby obtaining richer and more global feature representations. Finally, WIoU replaces the original loss function, which effectively improves the model detection capability and reduces leakage and misdetection. By improving YOLOv8n in the aforementioned three aspects, the model's accuracy in detecting surface defects on steel has been enhanced.

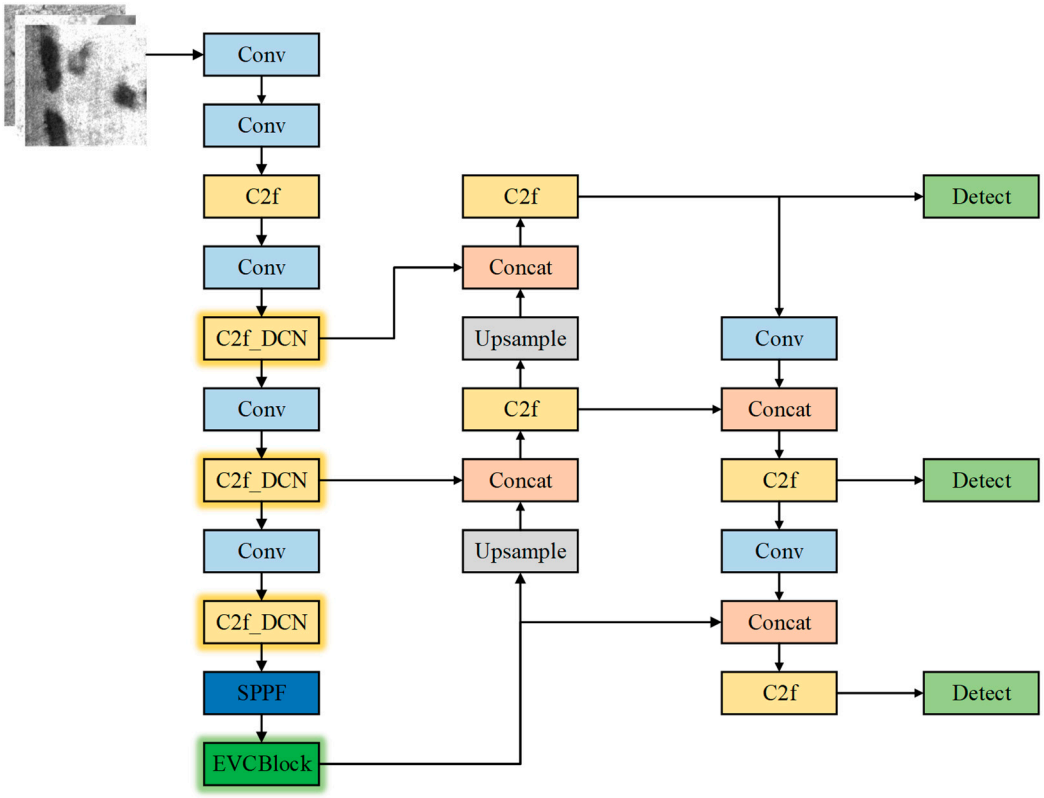


Figure 3. Network structure diagram of DEW-YOLO.

3.1. C2F_DCN

Deformable convolution is a type of convolution operation based on spatial deformations. It is commonly used in convolutional neural networks to handle images with spatial deformation characteristics [28]. Compared with traditional convolution operations, deformable convolution has stronger adaptability and better control over the receptive field. It can perceive and locate irregular shapes of objects more accurately. In deformable convolution, sampling positions are not confined to regular grid positions. Instead, an additional offset is introduced, allowing the convolution kernel to adaptively adjust based on the spatial variations of the input feature map. The output formula for deformable convolution is as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n), \quad (1)$$

where Δp_n represents the offset, $w(p_n)$ denotes the weight of the convolutional kernel at position p_n, and $x(p_0 + p_n + \Delta p_n)$, and $y(p_0)$ represent the mapping relationship between the original image and the feature map obtained by convolution. The operation process of deformable convolution is illustrated in Figure 4.

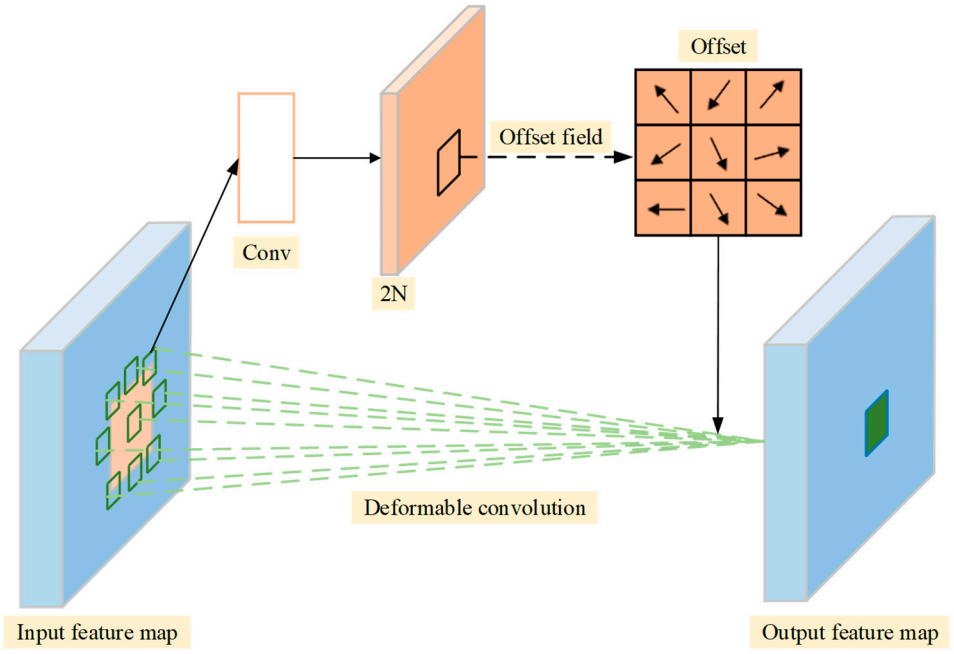


Figure 4. Structure diagram of deformable convolution.

First, feature maps are extracted using traditional convolutional kernels, resulting in a feature response map. The obtained feature response map from the previous step is used as the input for the deformable convolution operation. A new convolutional layer is applied to it, generating a feature map with 2N channels, where N represents the size of the convolutional kernel. Each pair of channels represents a coordinate offset, which describes the displacement distance of the convolutional kernel at that position. These offsets can be continuously adjusted during training to adapt to different input images. Then, bilinear interpolation on the offset feature map is performed to upsample it to the same size as the input feature map, enabling the use of the offset feature map to adjust the position and size of the convolutional kernel later on. Finally, by combining the upsampled offset feature map with the original input feature, both are learned simultaneously through optimization of the backpropagation algorithm to obtain the ability to dynamically adjust the position size of the deformable convolutional kernel.

To enhance the multi-scale feature extraction capability, the C2f module needs to integrate feature maps of different scales. Deformable convolution can learn nonlinear deformation fields, which facilitate better capturing of the correspondence between feature points across different scales. This is advantageous for extracting contextual information from multi-scale features. Deformable convolution can establish more complex correspondence relationships between feature points across different scales through deformable convolution kernels. This is more suitable for the C2f module to learn the transformations of feature points across scales compared with ordinary convolution.

In deep networks, the deformation characteristics of objects are often more complex. Therefore, this paper improves the C2f structure in the YOLOv8 backbone network by replacing the ordinary convolution in the bottleneck structure of the C2f structure with deformable convolution. The improved C2f_DCN structure is illustrated in Figure 5.

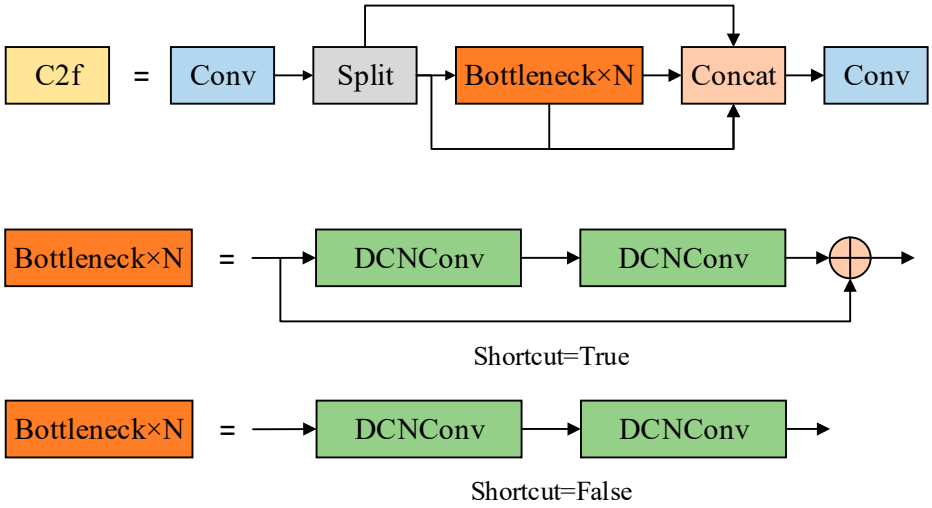


Figure 5. Structure diagram of C2f_DCN.

In the improved bottleneck structure, two layers of deformable convolution are introduced, allowing the model to freely sample the input feature map. This enables better learning of information related to the scale, background, and deformation of the target objects. After the convolutional layers, the output and input feature maps are fused to use residual connections. This allows the improved module to adaptively learn objects of different sizes and better understand the characteristics of different steel defects, thereby possessing stronger robustness and generalization ability.

3.2. Explicit Visual Center

The explicit visual center is composed of the multilayer perceptron (MLP) and the learnable visual center (LVC), whose architecture is shown in Figure 6. While the lightweight MLP architecture is for capturing long-range dependencies, the LVC is for aggregating local corner regions of the input image [29]. In addition, for smoothing the feature, a Stem block, contributed by a 7×7 convolution, a batch normalization (BN) layer, and an activation function layer, is added between these two parallel blocks.

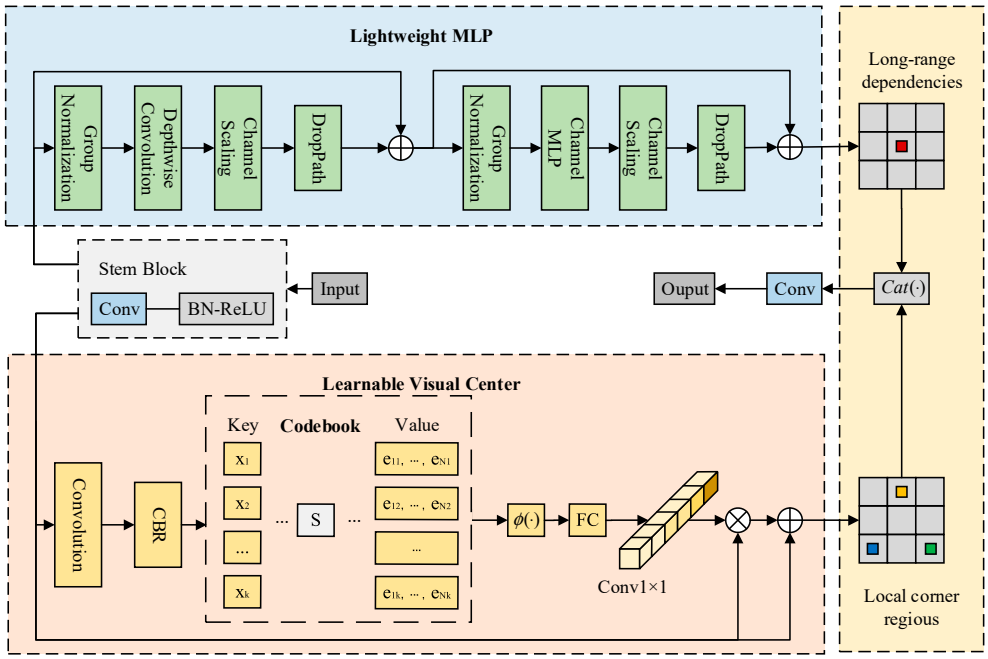


Figure 6. Structure of explicit visual center.

The MLP is composed of two residual modules: a depth-separable convolution-based module and a channel MLP-based module. In this case, the input of the MLP module is the output of the depth-separable convolution module. Both modules undergo channel scaling and tuning operations to improve feature generalization and robustness. The input for the depth-separable convolution module is the group-normalized feature map $\widetilde{X}_{in}$, and the output result $\widetilde{X}_{in}$ can be expressed as

$$\widetilde{X}_{in} = DConv(GN(X_{in})) + X_{in} \tag{2}$$

The input for the channel MLP module is the output of the depth-separable convolution module, which is group-normalized before the channel MLP operation. The process can be expressed as

$$MLP(X_{in}) = CMLP\left(GN\left(\widetilde{X}_{in}\right)\right) + \widetilde{X}_{in} \tag{3}$$

The LVC is an encoder with an intrinsic dictionary, consisting of an intrinsic codebook and a set of learnable vision-centered scale factors. The processing of LVC consists of two steps. The input features are convolved using a convolutional layer with CBR processing firstly, and then the features processed in the first step are combined with the codebook by means of a learnable scale factor, which maps the position information to $\widetilde{X}_i$ and b_k. The formula is as follows:

$$e_k = \sum_{i=1}^{N} \frac{e^{-S_k\left\|\widetilde{X}_i - b_k\right\|^2}}{\sum_{j=1}^{K} e^{-S_k\left\|\widetilde{X}_i - b_k\right\|^2}} \left(\widetilde{X}_i - b_k\right), \tag{4}$$

where S is the scale factor, $\widetilde{X}_i$ is the ith pixel, and b_k is the kth visual code. The ReLU and BN computations are implemented using ϕ with all e_k combined, followed by channel multiplication and channel addition of the input X_{in} and the local features Z. The computational formulas are given below:

$$e = \sum_{k=1}^{K} \phi(e_k), \tag{5}$$

$$Z = X_{in} \otimes (\delta(Conv_{1\times1}(e))), \tag{6}$$

$$LVC(X_{in}) = X_{in} \oplus Z \tag{7}$$

Using the channel dimension, the feature maps of the two modules are stitched together as the output of the EVC. The output expression is as follows:

$$X = cat(MLP(X_{in}); LVC(X_{in})), \tag{8}$$

where $MLP(X_{in})$ represents the long-range dependencies and $LVC(X_{in})$ represents the local corner regions. The integrated signature encompasses the strengths of both modules, allowing the detection model to learn a full range of discriminative feature representations. ECV assigns the features in the feature pyramid to different targets, and it achieves this by computing information such as the size, shape, and position of each target, assigning a region of interest (ROI) to each target, and then weighting the fusion of these ROIs with features from the feature pyramid. This enables better detection of targets of different sizes.

3.3. Wise-IoU Loss

In the YOLOv8 model, DFL loss and CIoU loss [30] are applied as the marginal regression loss functions. The DFL loss function helps focus the detection network faster on the defect target location as well as the neighboring regions, while the CIoU loss function

evaluates the accuracy of the bounding box prediction in the defect detection model. The CIoU formula is as follows:

$$L_{CIoU} = L_{IoU} + \frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)}, \tag{9}$$

$$\alpha = \frac{v}{L_{IoU} + v}, \tag{10}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w}{h} - arctan\frac{w_{gt}}{h_{gt}}\right)^2, \tag{11}$$

$$L_{IoU} = \frac{W_i H_i}{wh + w_{gt}h_{gt} - W_i H_i} \tag{12}$$

The expression for L_{IoU} is illustrated in Figure 7. In this equation, α represents the weight function, used for balancing parameters, v is the aspect ratio measure function, used to assess the aspect ratio consistency. w, h, and (x, y) represent the width and height dimensions and the center coordinates of the prediction box, respectively, w_{gt}, h_{gt}, and (x_{gt}, y_{gt}) represent the width and height dimensions and the center coordinates of the real box, respectively, W_i and H_i represent the intersection width and height, respectively, and W_g and H_g represent the minimum border width and height, respectively. In the prediction frame regression process, there are cases where the penalty term of the CIoU degrades to zero when the prediction frame is linear in the aspect ratio of the real frame, while either high- or low-quality anchor frames can be harmful to the regression losses.

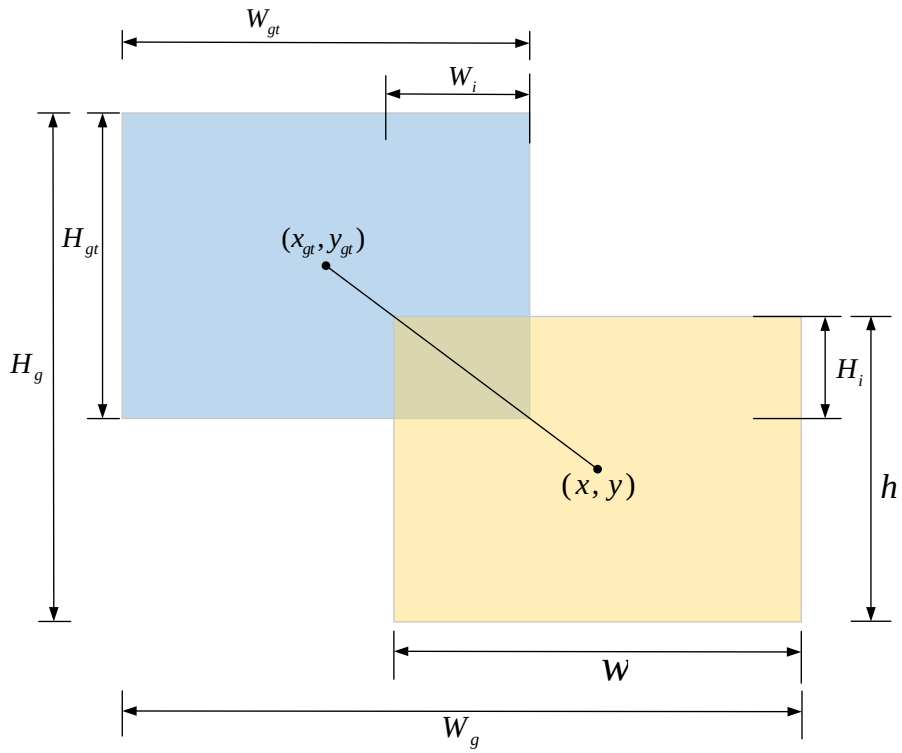


Figure 7. The area of intersection and union of the real box and the prediction box.

The accuracy of the model's bounding box prediction can be affected by the diversity of target sizes and shapes of various defect types in the strip surface defect dataset, as well as the distribution differences in the images. To solve this problem, the WioU [31] is introduced in this paper to replace the CIoU loss function, improving the precision of the

common mass anchor frame and enhance the fitting ability of the bounding box loss. The WIoU loss function is shown below:

$$L_{WIoU} = rR_{WIoU}L_{IoU}, R_{WIoU} \in [1, e),\ L_{IoU} \in [0, 1]. \tag{13}$$

The distance-focusing mechanism R_{WIoU} in the formula is used to amplify the L_{IoU} of ordinary appropriate anchor boxes, and the non-monotonic focusing factor r is used to focus on anchor boxes of normal quality. The expressions for R_{WIoU}, r, and the outlier factor β are as follows:

$$R_{WIoU} = exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*}\right), \tag{14}$$

$$r = \frac{\beta}{\delta\alpha^{\beta-\delta}}, \tag{15}$$

$$\beta = \frac{L_{IoU}^*}{\overline{L_{IoU}}} \in [0, +\infty), \tag{16}$$

where $\overline{L_{IoU}}$ represents the dynamic moving average. Relatively small gradient gains are allocated for both large and small values, reducing the impact on bounding box regression. Meanwhile, α and δ represent hyperparameters. By lowering the contribution of high-quality samples to the loss value, r dynamically assigns gradient gains to bounding boxes, decreasing the harmful gradients produced by low-quality anchor boxes in the later stages of training. Moreover, it focuses on ordinary quality anchor frames to improve model localization.

WIoU removes the aspect ratio penalty term from the CIoU while balancing the impact of both high-quality and ordinary-quality anchor boxes on model regression, which increases the model's generalization capability and overall performance. Therefore, this paper adopts WIoU to replace the original model's boundary loss function for optimizing the model's loss function.

4. Experiments

4.1. Experimental Environment and Parameter Configuration

This experimental environment used the Windows 11 operating system, 16 GiB of RAM, an Intel(R) Core(TM) i7-13700H @ 2.40 GHz CPU (Intel, Santa Clara, CA, USA), an NVIDIA GeForce RTX 4060 Laptop graphics card (NVIDIA, Santa Clara, CA, USA) with 8 GB of video memory, and PyTorch 2.0.1 as the deep learning framework. The Python version was 3.8.18, the CUDA version was 11.7, and the cuDNN version was 11.7. The experimental parameters were set as shown in Table 1.

Table 1. Experimental parameter settings.

Parameter	Setting
Epochs	300
Input image size	640 × 640
Batch size	16
Initial learning rate	0.01
Momentum	0.937
Optimizer	SGD

4.2. Evaluation Indicators

The commonly used evaluation indexes for assessing the detection accuracy and speed of the strip steel surface defect detection model are the precision (P), recall (R), and mAP. The specific calculations are shown in the following equations:

$$Precision = \frac{TP}{TP + FP}, \tag{17}$$

$$Recall = \frac{TP}{TP + FN}, \tag{18}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i, \tag{19}$$

In the equations, TP represents the true cases, FN represents the false negative cases, FP represents the false positive cases, and n represents the total number of cases.

Additionally, evaluation metrics also include the model computational complexity (GFLOPs), model parameter count (parameters), and frames per second (FPS). Model computational complexity and model parameter count are the key metrics for assessing model efficiency and capacity, respectively, while the detection speed evaluates the ability to process image frames every second.

5. Experiment Results and Analysis

5.1. Ablation Experiments

To verify the effectiveness of the proposed improvement strategies, a set of comparative ablation experiments were conducted. The experiments took YOLOv8n as the baseline model, and the results are presented in Table 2. In the table, C2f_DCN, EVC, and WIoU represent the three proposed improvement points in this study. The symbol √ denotes the adoption of the improvement point.

Table 2. Ablation experiment results.

Sequence	C2f_DCN	EVC	WIoU	mAP @ 0.5	FPS	Params (M)	GFLOPs
1				76.4	132	3.01	8.2
2	√			78.7	104	3.16	7.6
3		√		77.9	101	7.29	11.6
4			√	77.0	156	3.01	8.2
5	√	√		79.6	96	7.45	11.0
6	√	√	√	80.3	91	7.45	11.0

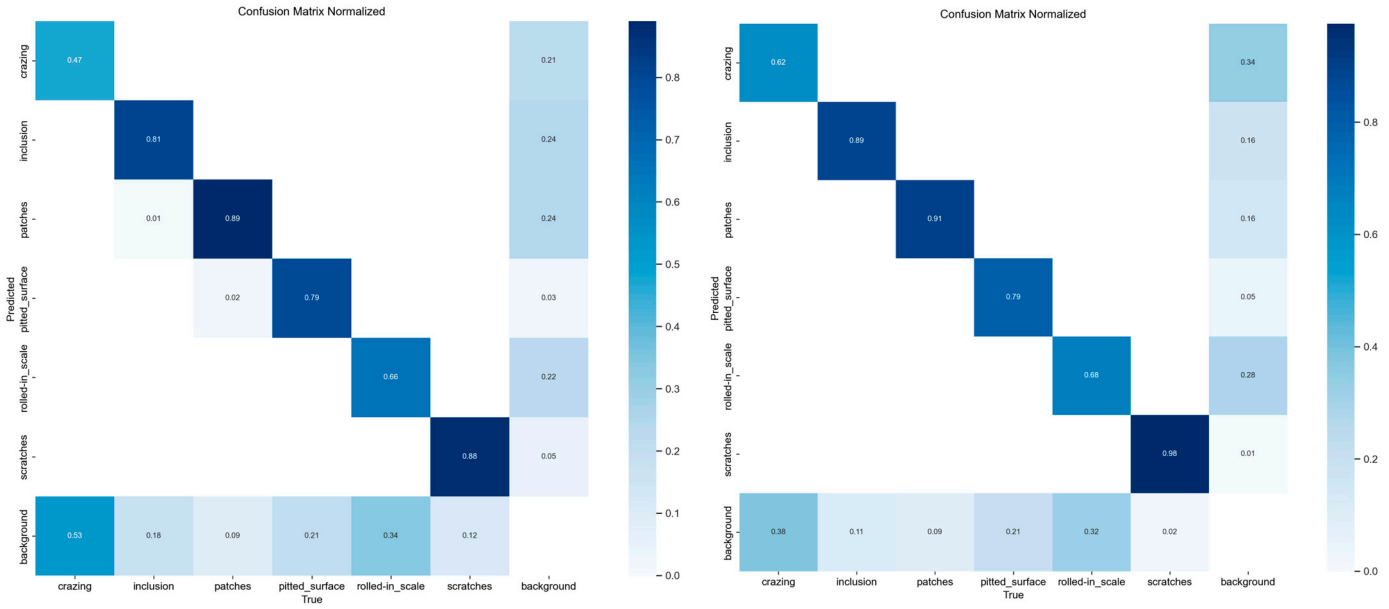
According to the ablation experiment results in Table 2, the first group directly employed YOLOv8n with an mAP value of 76.4%. In the second group, the C2F_DCN module network replaced the C2F module in the backbone. This enhancement allowed the model to strengthen the fusion of multi-level features and effectively adjust the positions and sizes of receptive fields, enabling more flexible handling of complex surface defects on steel materials, which optimized the detection accuracy of the model, with the mAP value increasing to 78.7%. The third group embedded the EVC module into the backbone network of YOLOv8n, and the model fused the displayed explicit visual center of CFPNet to capture the long-range dependence of features and improve the detection of small targets. The mAP value of the model was improved by 1.5%, indicating that the EVC module could effectively improve the model's detection accuracy. The fourth group replaced the loss function of YOLOv8n with the WIoU loss function. The number of parameters of the model and the amount of computation were basically unchanged, and the accuracy slightly improved, but the detection speed of the model was improved by 18%, which proves that the addition of WIoU can make the model more accurate in localization and classification. The fifth group added the EVC module to the second group to enhance the

model's adjustment and optimization for different feature scales. This allowed the model to better capture both global information and local details across different scales. Although adding the EVC module resulted in an increase in Params and GFLOPs, the experimental results prove that the improvement effectively increased the detection accuracy, resulting in an mAP value of 79.6%. Finally, the sixth group replaced the original loss function with a WIoU loss function based on the fifth group, which reduced the difference in gradient gain between the high-quality and low-quality samples, strengthened the model's localization performance, and improved its generalization ability, with an mAP value of 80.3%.

The sixth experiment's minimum frame rate was 91 frames per second, which is within an acceptable range and can satisfy the real-time needs of industrial inspection, even though each improvement may somewhat lower the FPS. The above ablation experiments demonstrate the effectiveness of the three improvement points and also show the contribution that each one makes to the model's detection effectiveness.

5.2. Comparative Experiments

To demonstrate the detection effect of the algorithm proposed in this paper visionally, the YOLOv8n model was compared with the confusion matrix of the proposed model in this paper in a comparison experiment, and the comparison results are shown in Figure 8. The true and predicted categories are represented by the rows and columns of the confusion matrix, respectively. The proportions of successfully predicted categories are represented by the values in the diagonal regions, while the proportions of wrongly predicted categories are represented by the values in the other regions.



(**a**) (**b**)

Figure 8. (**a**) Confusion matrix plot of YOLOv8n. (**b**) Confusion matrix plot of DEW-YOLO.

The comparison results in Figure 8 show that the diagonal region of the DEW-YOLO confusion matrix is darker than the diagonal region of the YOLOv8n confusion matrix, and the values in the DEW-YOLO diagonal region were also larger than those in the YOLOv8n diagonal region. This suggests that the model suggested in this paper is much better at accurately predicting the object categories. In addition, DEW-YOLO effectively reduced the false positive rate compared with YOLOv8n.

As shown in Figures 9 and 10, the PR curve graphs depict a comparison of mAP values for YOLOv8n and DEW-YOLO, respectively. The PR curve graph reflects the mAP values for various defect categories, where "all-classes" denotes the average mAP value across all categories. Compared with the results before improvement, the overall mAP value

increased from 76.4% to 80.3%, showing an improvement of 3.9%. The results prove that the improved model can enhance the detection accuracy across multiple defect categories. Except for the rolled-in scale defect, which had a slight decrease of 0.1% in its mAP value, the other five defects had different levels of improvement, especially patches, which had a 6.7% increase in its mAP value from 83.9% to 90.6%, and scratches, which had the highest mAP value, reaching 97.3%.

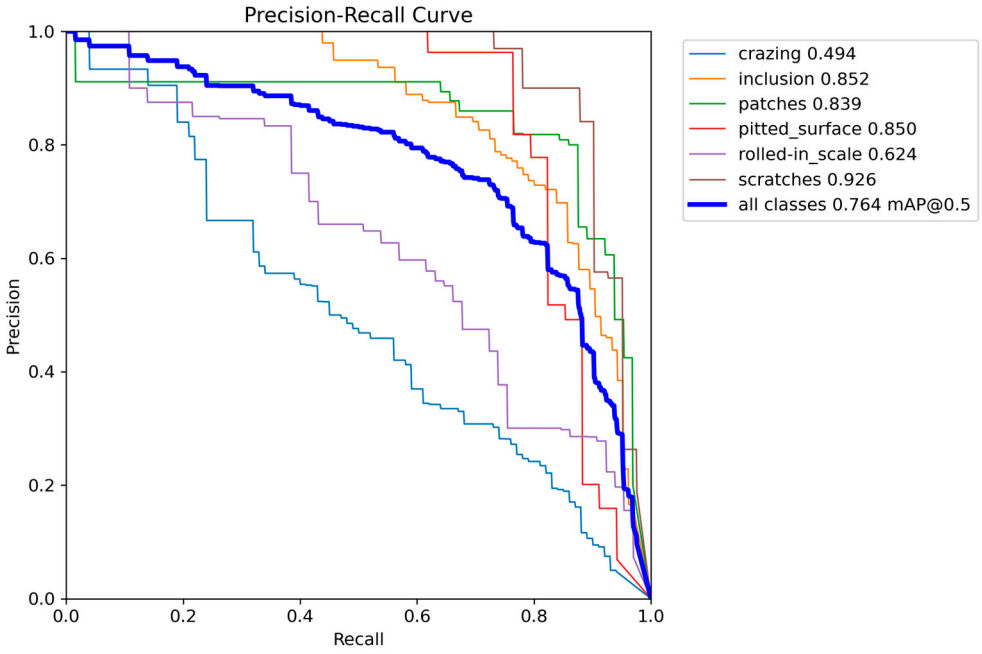


Figure 9. P-R curve of YOLOv8n algorithm.

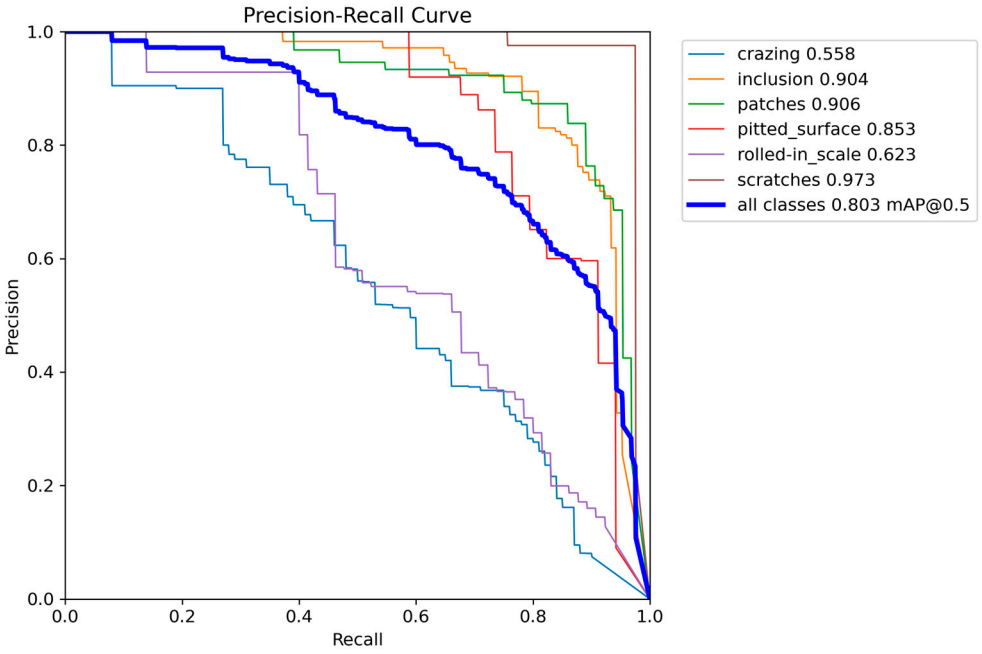


Figure 10. P-R curve of DEW-YOLO algorithm.

5.3. Visualization of the Results

To compare the effectiveness of the proposed algorithm with the YOLOv8n algorithm in detecting surface defects on strip steel more intuitively, the DEW-YOLO model and the YOLOv8n model were applied to detect defects in the same test set, and some of the defect detection effects are shown in Figure 11. As can be gleaned from the figure, the original

YOLOv8n model suffered from the problem of missed detection or low accuracy, with some categories of detection performance being especially poor, while the DEW-YOLO model was significantly better than the original YOLOv8n model in terms of effect; it could more comprehensively identify the defective targets when detecting some of them, and the detection accuracy also significantly improved. The DEW-YOLO model detected defects that were missed by the baseline model and had a higher confidence level than the baseline model in detecting defects in the same location. The test results verify the effectiveness of the DEW-YOLO model for defect detection and provide solid support for quality control in the strip production process.

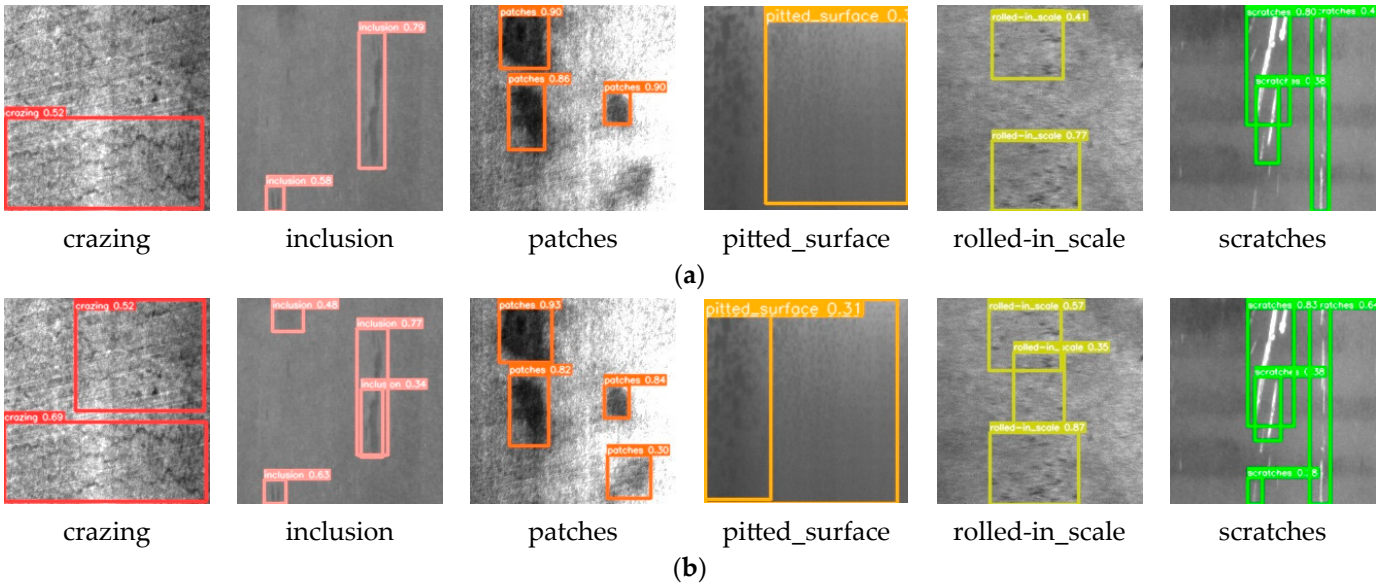


Figure 11. Comparison of detection results. (**a**) Detection effect of YOLOv8n model. (**b**) Detection effect of DEW-YOLO model.

5.4. Comparison Experiment with Different Models

The ablation experiments demonstrated the feasibility of the proposed algorithm. To further validate its performance, the results of the improved model proposed in this paper were compared with those of six mainstream models, namely SSD, Faster R-CNN, DETR, and the YOLO series models, under the same dataset partition conditions. The experimental results can be seen in Table 3.

Table 3. Comparison of experimental results of different network models.

Type	SSD	Fast R-CNN	DETR	YOLOv5s	YOLOv7	YOLOv8n	DEW-YOLO
Crazing	0.452	0.474	0.293	0.445	0.417	0.494	0.558
Inclusion	0.818	0.849	0.746	0.861	0.864	0.852	0.904
Patches	0.880	0.916	0.876	0.835	0.859	0.839	0.906
Pitted surface	0.842	0.851	0.814	0.852	0.825	0.85	0.853
Rolled-in scale	0.602	0.541	0.523	0.52	0.566	0.624	0.623
Scratches	0.722	0.947	0.925	0.906	0.887	0.926	0.973
mAP (%)	71.9	76.3	67.2	73.7	73.6	76.4	80.3
FPS	69	19	12	108	38	132	91
GFLOPs	62.7	370.2	100.9	15.8	103.2	8.2	11.0

As can be seen from the experimental results in Table 3, the mAP value of the DEW-YOLO model proposed in this paper improved by 8.4%, 4%, and 13.1% compared with SSD, Fast R-CNN, and DETR, respectively, with a significant improvement in accuracy. The SSD and Faster R-CNN detection models have slow detection speeds and are computationally

intensive, and thus they do not redound to deployment and utilization on devices with limited computational resources. DETR is a Transformer-based target detection algorithm. When detecting multi-class steel defects, this model may encounter problems such as high memory consumption and slow convergence due to the global self-attention mechanism, which makes the DETR model less effective, and its mAP value was the lowest, being only 67.2%. In the case of diverse types and complex defects in steel, the detection accuracy of YOLOv5s and YOLOv7 was also unsatisfactory. Compared with YOLOv5s and YOLOv7, the mAP values of the DEW-YOLO model improved by 6.6% and 6.7%, respectively, resulting in a significant improvement in detection accuracy. And all six types of defect detection accuracies of the DEW-YOLO model were higher than YOLOv5s vs. YOLOv7. YOLOv8n was the fastest detection algorithm with a relatively small number of calculations, but its detection accuracy still needs to be improved. The DEW-YOLO algorithm achieved comprehensive improvements in detection performance. Although there was a slight decrease in detection speed, it still met the detection requirements in industrial applications, and the calculation count remained at a satisfactory level. In summary, the DEW-YOLO algorithm proposed in this paper outperformed the other algorithms in the task of detecting defects on steel surfaces, effectively recognized a wide range of defect classes, and met the terminal detection speed requirements.

6. Conclusions

In order to solve problems such as low detection accuracy for current strip steel surface defect detection algorithms in practical applications, this paper proposed an effective steel surface defect detection algorithm, DEW-YOLO, on the basis of YOLOv8. The algorithm introduces deformable convolution into the C2f network structure, which constitutes the C2f-DCN and replaces the C2f in the backbone network with the C2f-DCN. This allows the model to realize adaptive adjustment of the sensory field, which enhances the model's ability to extract features for multiple classes of defects. Then, the model fuses the explicit visual center of CFPNet to capture the long-range dependence of features and improve the model's ability to detect small targets. Finally, the CIoU loss function was replaced with the WIoU loss function to improve the accuracy of the common mass anchor frames and enhance the fitting ability of the bounding box losses. Extensive experiments and comparisons with other algorithms validated the feasibility of the proposed method and highlighted the sophistication of the methodology of this paper, as it effectively improved the detection accuracy and maintained the detection speed. The mAP of this algorithm reached 80.3% on the NEU-DET strip surface defects dataset, which was 3.9% higher than the original algorithm, and a real-time detection speed of 91 frames/s could be realized in the defect detection of strip production, which made it more efficient and flexible in actual strip production. Although the improved model had a slight increase in computations compared with the original model, it offered significantly better detection accuracy and still remained suitable for deployment on mobile devices. The proposed method is applicable to the current demand for surface defect detection in steel strips.

Using computer vision technology to address surface defect detection in steel strips has practical significance and provides reference value for defect detection problems in other industrial sectors. In future work, the detection precision of the proposed method will be further developed, and the model will be optimized for lightweight deployment to enhance detection speed.

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

Funding: This research was funded by the National Natural Science Foundation of China, Grant No. 61863009, and the Guangxi Key R&D Program, Guike-AB22080093.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The dataset can be downloaded here: http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list (accessed on 13 June 2024).

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

References

1. Duspara, M.; Savković, B.; Dudic, B.; Stoić, A. Effective Detection of the Machinability of Stainless Steel from the Aspect of the Roughness of the Machined Surface. *Coatings* **2023**, *13*, 447. [CrossRef]
2. 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]
3. Amin, D.; Akhter, S. Deep learning-based defect detection system in steel sheet surfaces. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; pp. 444–448.
4. Xie, J.; Yang, X.-Y.; Xu, C.-H.; Chen, G.; Ge, S. Infrared thermal images detecting surface defect of steel specimen based on morphological algorithm. *J. China Univ. Pet.* **2012**, *36*, 146–150.
5. Wang, G.; Xiao, Q.; Gao, Z.; Li, W.; Jia, L.; Liang, C.; Yu, X. Multifrequency AC magnetic flux leakage testing for the detection of surface and backside defects in thick steel plates. *IEEE Magn. Lett.* **2022**, *13*, 1–5. [CrossRef]
6. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
7. 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.
8. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
11. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
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.
13. 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; pp. 779–788.
14. 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.
15. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W. A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 5592878. [CrossRef] [PubMed]
16. Lin, C.-Y.; Chen, C.-H.; Yang, C.-Y.; Akhyar, F.; Hsu, C.-Y.; Ng, H.-F. Cascading convolutional neural network for steel surface defect detection. In *Advances in Artificial Intelligence, Software and Systems Engineering: Proceedings of the AHFE 2019 International Conference on Human Factors in Artificial Intelligence and Social Computing, the AHFE International Conference on Human Factors, Software, Service and Systems Engineering, and the AHFE International Conference of Human Factors in Energy, Washington, DC, USA, 24–29 July 2019*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 202–212.
17. Kou, X.; Liu, S.; Cheng, K.; Qian, Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement* **2021**, *182*, 109454. [CrossRef]
18. Wang, L.; Liu, X.; Ma, J.; Su, W.; Li, H. Real-time steel surface defect detection with improved multi-scale YOLO-v5. *Processes* **2023**, *11*, 1357. [CrossRef]
19. 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]
20. Qu, Y.; Wan, B.; Wang, C.; Ju, H.; Yu, J.; Kong, Y.; Chen, X. Optimization algorithm for steel surface defect detection based on PP-YOLOE. *Electronics* **2023**, *12*, 4161. [CrossRef]
21. Zheng, Z.; Chen, N.; Wu, J.; Xv, Z.; Liu, S.; Luo, Z. EW-YOLOv7: A Lightweight and Effective Detection Model for Small Defects in Electrowetting Display. *Processes* **2023**, *11*, 2037. [CrossRef]
22. Bao, Y.; Song, K.; Liu, J.; Wang, Y.; Yan, Y.; Yu, H.; Li, X. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
23. 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.

24. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
25. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
26. Jocher, G.; Nishimura, K.; Minerva, T.; Vilariño, R.J.A.M. YOLOv5. *Code Repos.* **2020**, *7*, 2021. Available online: https://github.com/ultralytics/yolov5 (accessed on 13 June 2024).
27. 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.
28. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
29. Quan, Y.; Zhang, D.; Zhang, L.; Tang, J. Centralized feature pyramid for object detection. *IEEE Trans. Image Process.* **2023**, *32*, 4341–4354. [CrossRef]
30. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef]
31. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.