DFP-YOLO: An Efficient Algorithm for Detecting Steel Surface Defects

Jiawei Chai, Ziwei Zhou

Abstract—**To enhance the accuracy of steel plate surface defect detection and minimize the incidence of misdetection and leakage, this paper proposes a DFP-YOLO algorithm based on YOLOv8n to achieve efficient detection of steel plate surface defects. Firstly, the C2f module of the backbone network and Neck layer is substituted by the DWR_DRB module to strengthen the ability of capturing defects at various scales and enhance the efficiency of model feature extraction. Secondly, the Feature Pyramid Share Convolution module is devised to extract multi-scale features through convolutional layers with different dilation rates, integrating local details with global contextual information for a better comprehension of complex scenes. Finally, a Powerful-IoU loss function is utilized to control the scale size of the auxiliary boundary to accelerate the detection speed and improve the model accuracy. The experimental results demonstrate that the proposed algorithm in this paper boosts the mean accuracy (mAP) of the steel plate surface defect detection task by 3.4% compared to the original YOLOv8n model, increases the accuracy by 8.8%, and raises the inference speed of the model by 49 frames per second when conducting DFP-YOLO detection on the dataset NEU-DET. Meanwhile, the generalization and robustness of the model are verified through tests on the industrial steel plate surface defects dataset GC10-DET and the larger public benchmark dataset PASCAL VOC 2012.**

Index Terms—***YOLO*v8, steel defect detection, DWR_DRB module,Feature Pyramid Share Convolution module，Powerful IoU loss function**

I. Introduction

Steel plate surface defect detection constitutes an essential research direction within the domain of materials science and engineering, undertaking the responsibility of guaranteeing product quality and enhancing production efficiency [1]. As industrial manufacturing continues to evolve and market demand for high-standard and high-quality products escalates, the surface defect detection technology for steel plates gains increasing significance. The possible defects that may exist on the surface of steel plates encompass cracks, inclusions, pores, and scratches, among others. These defects not only impact the mechanical properties of the material, causing safety risks during utilization, but also may influence the subsequent processing procedures. Hence, it is of particular significance to detect and repair these defects promptly and accurately [2].

The advancement of modern steel plate surface defect detection technology is inextricably linked to the integration of multidisciplinary technologies. Traditional inspection methods, such as visual inspection, magnetic particle inspection, and penetration inspection, although somewhat effective to a certain degree, gradually fall short of meeting the demands of the modern manufacturing industry due to the high requirements of the operating environment, low inspection efficiency, and reliance on manual judgment [3-5]. In contrast, the introduction of computer vision technology significantly enhances the efficiency and accuracy of steel plate surface defect detection.

The principal detection algorithms can be divided into two groups: single-stage detection models exemplified by Faster R-CNN [6] and two-stage detection models represented by YOLO [7] and SSD [8]. Although Faster R-CNN demonstrates superiority in accuracy, it possesses a complex structure that demands the training of two networks (the RPN and the final detection network), leading to slower detection and relatively arduous implementation and debugging [13][16]. Conversely, YOLO has a straightforward structure, considers target detection as a regression problem, and accomplishes the entire detection process via a single network, which enjoys obvious advantages in speed, can handle high-resolution images in real time, and is facile to implement and debug [9][10].

As the YOLO architecture is continuously explored in depth and combined with emerging technologies such as migration learning and self-supervised learning, its future development prospects are becoming increasingly clear. Ongoing innovations will further promote the YOLO algorithm to a higher level and enable it to play a more significant and indispensable role in the field of computer vision.

In recent years, numerous scholars have carried out in-depth and systematic research on deep learning-based target detection algorithms. Chen [14] et al. proposed an online surface defect detection method based on the improved YOLOv3. By employing the lightweight network MobileNetV2 as a feature extractor, the Extended Feature Pyramid Network (EFPN) and the Feature Fusion Module (FFM) were designed. Moreover, the IoU loss function was introduced to effectively address the mismatch between classification and bounding box regression. Qu [15] et al. further simplified the feature pyramid structure, providing a novel idea for resolving the issue of small target detection. Wang [17] et al. put forward a lightweight defect detection

Manuscript received December 6, 2024; revised April 6, 2025.

Jiawei Chai is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan,Liaoning 114051, P. R. China(e-mail:2192520875@qq.com).

Ziwei Zhou is an associate professor of School of Computer Science and Software Engineering ,University of Science and Technology Liaoning, Anshan, Liaoning 114051, P. R. China(e-mail: 381431970@qq.com)

method based on YOLOv5. Literature [18] proposed a surface defect detection model for steel strip based on the improved YOLOv7. The model combines the attention mechanism and replaces the original loss function with the SIoU loss function and redefines the penalty term. These enhancements effectively address the problems of low detection speed and low detection accuracy of traditional methods. The method significantly enhances the detection accuracy and efficiency of small target defects and provides significant technical support for improving the surface quality control of hot rolled strip steel. Song [19] et al. proposed a multi-directional optimization improvement model based on YOLOv8. The model significantly improves the detection capability of complex texture and irregular shape defect features by introducing the deformable convolution technique, the bidirectional feature pyramid network structure, the BiFormer attention mechanism, and adjusting the loss function. Huang [20] et al. proposed a new surface defect detector for steel plates based on the YOLOv8s, which can be optimized and improved by introducing the WIoU loss function, re-designing the CFN module in the backbone network, etc. This effectively resolves the data quality imbalance problem, reduces the computational overhead, and improves the detection accuracy and robustness of the model.

The main focus of this paper is to enhance the average accuracy of steel plate surface defects without augmenting the model parameters, thereby reducing the leakage rate and false detection rate during the detection process.

II. Related work

YOLOv8 is a single-stage target detection algorithm introduced by Ultralytics, which achieves a balance between speed and accuracy through its highly optimized architectural design and is applicable to a wide range of computer vision tasks, including target detection, image segmentation, and image classification. The architecture of YOLOv8 is divided into four principal modules: the input module, the backbone network, the necking network, and the prediction module [11]. The input module conducts data augmentation, adaptive image scaling, and anchor frame optimization to ensure the robustness of the model and its adaptability to diverse inputs. The backbone network utilizes an improved convolution and feature extraction module to extract key features from the input image and extends the receptive field of the features with the SPPF module to further enhance the model's performance in multi-scale target detection. The neck network integrates the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) modules [12] to achieve the deep fusion of multi-scale features, making the feature maps of different scales simultaneously rich in semantic and positional information and thereby enhancing the detection capability of targets of various sizes. The prediction module performs target localization and classification at different scales through multiple detection heads, enabling the model to efficiently identify and accurately locate target boundaries.

YOLOv8 offers several versions for different application requirements, such as lightweight YOLOv8s and YOLOv8n, medium-sized YOLOv8m, larger precision YOLOv8l, and high-precision YOLOv8x [22]. The structures of these versions range from lightweight to complex and are suitable for different application environments ranging from mobile devices to high-performance computing platforms. Users can choose the appropriate model version based on specific computational resources and application requirements to achieve the best balance between resource efficiency and detection performance. For example, YOLOv8s is suitable for resource-constrained devices, such as mobile devices and embedded devices, to achieve real-time detection without excessive loss of accuracy, while YOLOv8x is suitable for high-precision scenarios requiring high detection accuracy, such as autonomous driving and industrial inspection tasks [23].

With its flexible structural design and outstanding performance, YOLOv8 has extensive applications in domains such as real-time video surveillance, drone tracking, autonomous driving, industrial quality control, medical image analysis, and the like. In autonomous driving, YOLOv8 can identify and locate vehicles, pedestrians, traffic signs, etc. in real time to support safe driving; in industrial inspection, it can be employed for defect identification and object detection to enhance production efficiency; in medical image analysis, YOLOv8 assists in detecting lesion areas to provide doctors with auxiliary diagnoses. In conclusion, the efficient architecture of YOLOv8 enables it to excel in speed, accuracy, and resource efficiency, which offers robust support for computer vision technology in practical applications [24].

III. Improved model

In this experiment, we chose YOLOv8n as the benchmark model and made several targeted improvements based on it, as shown in Fig. 1. First, to improve the network performance, we introduce the Dilated reparam block modul (DRB) into the Dilation-wise residual module module (DWR) to form the improved DWR_DRB module.This module aims to improve the feature extraction capability by increasing the depth and width of the network. Next, we embed the DWR_DRB module on top of the original C2f module at the backbone network layer to enhance the richness of feature representation. In addition, to further enhance feature fusion and multi-scale feature learning, we also apply the DWR_DRB module at the neck layer to facilitate cross-layer transfer of information. Subsequently, we designed the Feature Pyramid Share Conv (FPSC) module to replace the SPPF module in the original benchmark model to enhance the fusion and representation of multi-scale features. In addition, we adjust the loss function from CIoU to Powerful Intersection over Union (Powerful-IoU, PIoU) to more effectively match the location and shape of the real bounding box [14], which in turn improves the accuracy of the detection box and ultimately enhances the overall detection accuracy.

A. DWR_DRB Model

In the YOLOv8 model, the C2f module is one of the key components of feature extraction The C2f module achieves feature fusion by stitching features from different branches

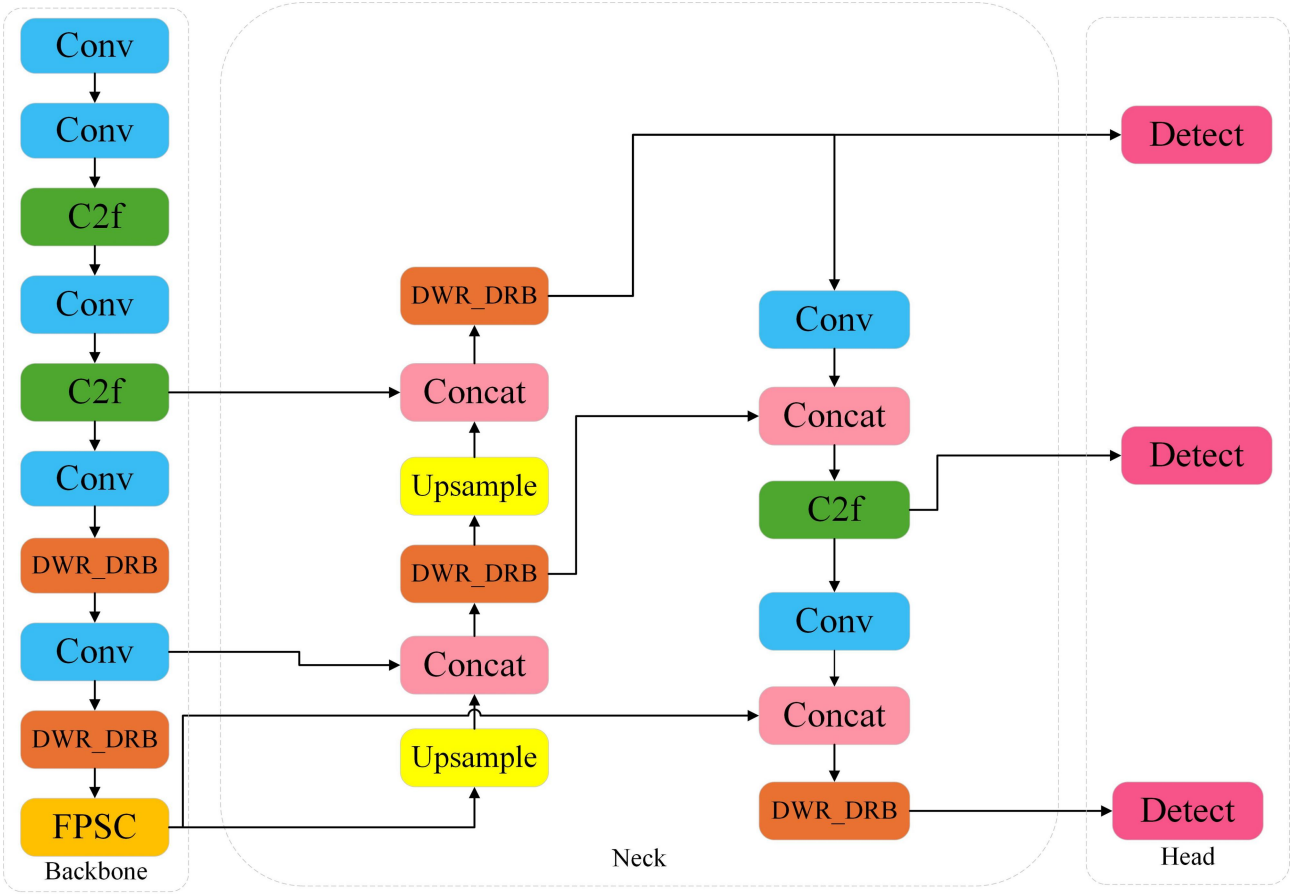


Fig. 1. DFP-YOLO model

in the channel dimension. This splicing operation allows the fused features to contain information from multiple branches, which enhances the expressive capability of the features and thus improves the overall performance of the model in the target detection task. However, in the steel plate surface defect detection task, the types and sizes of defects show significant diversity, including tiny cracks, holes, scratches and other defects at different scales. The presence of these multi-scale features puts higher demands on feature fusion. If they are not handled properly in the fusion process, it may result in some critical local detail information being ignored or lost. Specifically, small-scale defects are easily suppressed or masked when fused with large-scale features, leading to a decrease in the accuracy and reliability of the model in detecting these small defects.

To address this problem and further enhance the ability of the network model to extract and utilise multi-scale contextual information, we add the DWR_DRB module to the original C2f module.The introduction of the DWR_DRB module helps to retain detailed information of small-size defects during multi-scale feature fusion, thus ensuring accurate detection of defects at different scales. This improvement not only improves the model's multi-scale sensing capability, but also significantly improves the model's performance in steel plate surface defect detection, which can more effectively cope with the complexity and challenges of the steel plate defect detection task.

The DWR module is designed by means of the residual method, and the network module of DWR is illustrated in Fig. 2. Within the residual path, we extract and fuse multi-scale contextual information efficiently by means of a two-step approach and combine different receptive fields into a final feature map, thereby enhancing the feature representation capacity of the model. The traditional single-step multi-scale feature extraction is divided into two steps: regional residualization and semantic residualization. This division not only reduces the redundant receptive fields in the traditional single-step approach but also effectively improves the feature representation and generalization performance through the fusion of multi-scale contextual information.

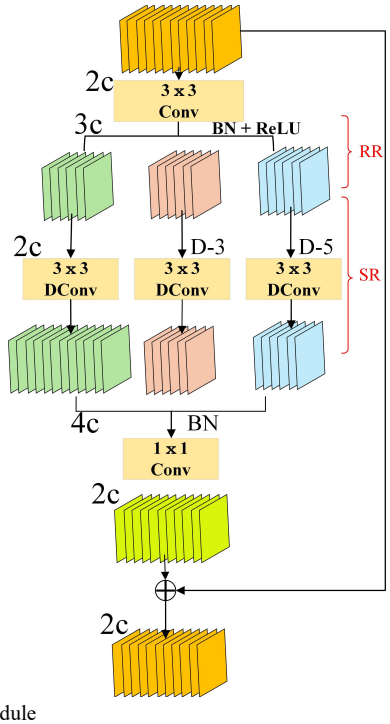


Fig. 2.DWR module

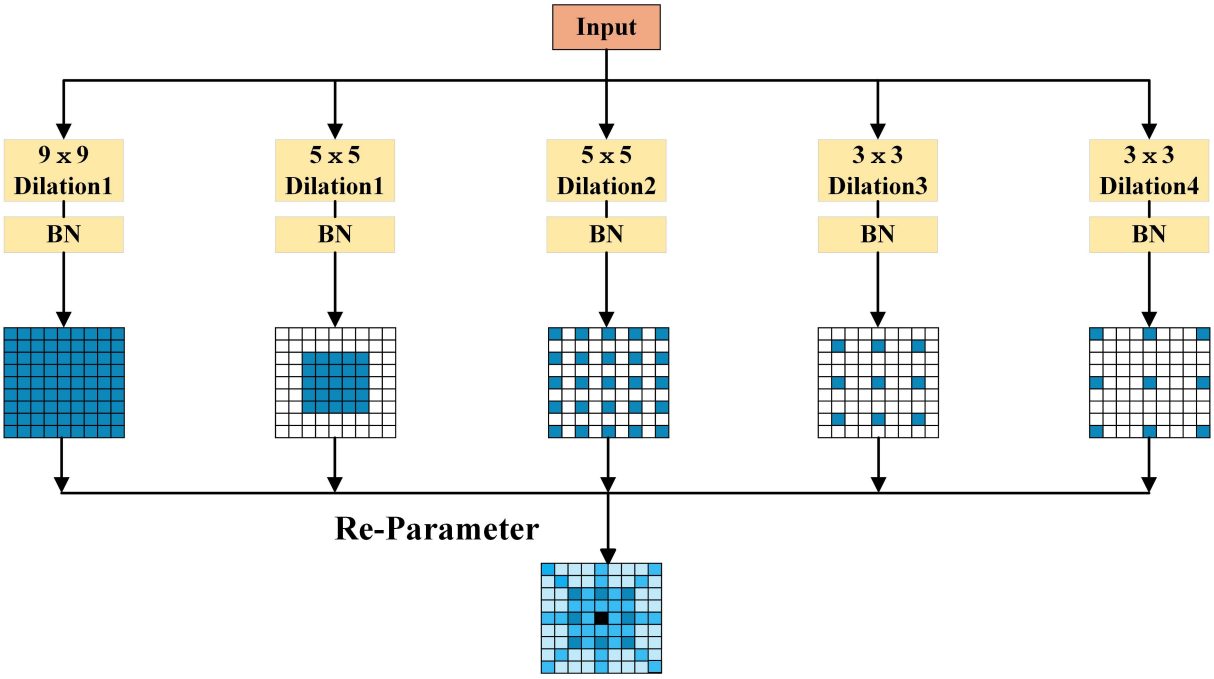


Fig. 3.DRB module

In the first step, we generate parsimonious feature maps with different region representations through a 3x3 convolutional operation combined with batch normalisation (BN) and ReLU activation functions. These feature maps provide a solid foundation for the morphological filtering in the second step, which enables the feature extraction to better cover different region sizes, thus improving the regional descriptive capability of the features.

In the second step, semantic-based morphological filtering is accomplished by applying a depth-separable convolution to each regional feature map. This is carried out to prevent the introduction of unnecessary redundant receptive fields and to guarantee the efficient extraction of semantic information under multi-scale circumstances. Through this procedure, we can acquire semantically relevant features more effectively, reduce the information redundancy in the feature extraction process, and effectively integrate the feature information at different scales, which significantly enhances the expressive and context-aware capabilities of the features.

The DRB module intends to enhance the feature capturing capacity of large convolutional kernels by employing convolutions with different expansion rates in parallel, and performs exceptionally well in capturing sparse patterns; the network module of DRB is presented in Fig. 3. The mechanism of dilation convolution effectively expands the receptive field by setting the dilation rate, enabling the convolutional layer to scan the input feature map to capture patterns between distant pixels, rather than merely the relationship between neighboring pixels. This mechanism proves useful for recognizing sparse but significant features in an image, particularly in cases where a pixel in the feature map might be associated with multiple pixels at a distance.

In the inference phase, to reduce the additional computational overhead, the DRB combines all the inflated convolutional layers into a single non-inflated convolutional layer through an equivalent transformation. This conversion is equivalent to expanding the convolutional kernel of the inflated convolution into a sparse large convolutional kernel that maintains the same receptive field while significantly reducing the computational complexity. It has been demonstrated that the large convolutional kernel is excellent in capturing global patterns, but it needs to be used in combination with a parallel small convolutional kernel, which is better at capturing small-scale detailed features during training.

These parallel small and large convolutional kernels are processed in separate batch normalization (BN) layers and then summed up to integrate large- and small-scale features. After training, the BN layer is incorporated into the convolutional layer through a structural reparameterization strategy, so that the features of the small convolutional kernels can be equivalently integrated into the large convolutional kernel during inference, further optimizing the model performance and inference efficiency.

B. Feature Pyramid Share Convolution Model

Spatial Pyramid Pooling (SPP) is a technique in deep learning, especially used in Convolutional Neural Networks (CNNs) to process input images of different sizes and scales, as shown in Fig. 4.

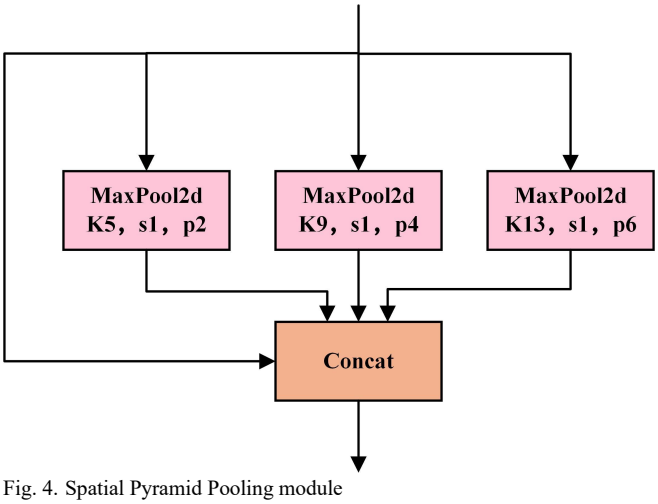


Fig. 4. Spatial Pyramid Pooling module

Atrous Spatial Pyramid Pooling (ASPP) is an improved technique proposed on the basis of SPP, as depicted in Fig. 5. ASPP captures multiscale information across different receptive fields by employing dilated convolution with various dilation rates. Additionally, the ASPP module typically encompasses a global average pooling layer, which is utilized to generate image-level features. These features are subsequently fused with the output of the dilated convolution layer through up-sampling (e.g., bilinear interpolation) to Introduce global contextual information.

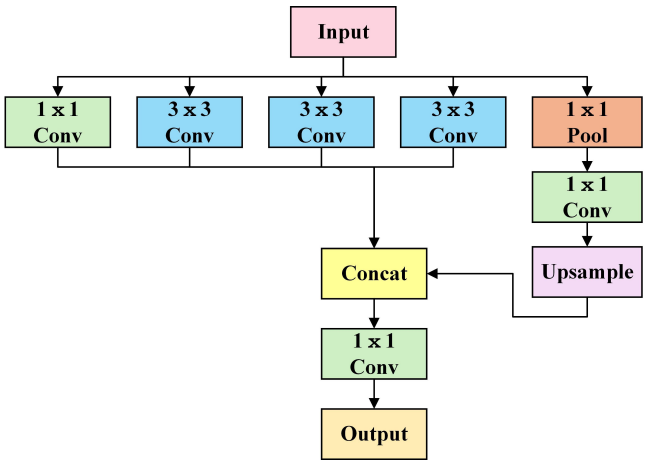


Fig. 5.Atrous Spatial Pyramid Pooling module

Inspired by the aforementioned module, this paper designs the Feature Pyramid Shared Convolution (FPSC) module, which can effectively extract multi-scale feature information by employing convolutional layers with different expansion rates, thereby enhancing the ability to capture different scales and contextual information in an image. Convolutional layers with a low expansion rate contribute to capturing local details, while those with a high expansion rate focus on obtaining global contextual information to achieve comprehensive feature representation. The Feature Pyramid Shared Convolution model is presented in Fig. 6.

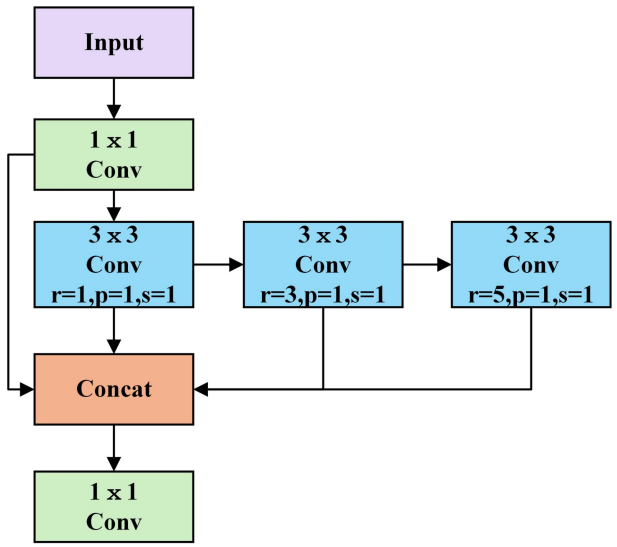


Fig. 6.Feature Pyramid Share Convolution module

Firstly, multi-scale feature extraction is one of the core strengths of the module. Convolutional layers with varying expansion rates are capable of extracting features at different scales, which is highly beneficial for capturing complex details and contexts in the image, considering both subtle local information and focusing on the overall global features. Additionally, the module employs shared convolutional layers to reduce the number of model parameters. Compared with setting independent convolutional layers for each expansion rate, the parameter sharing approach significantly reduces redundancy and markedly improves the computational and storage efficiency of the model.

Secondly, through the efficient channel transformation of the 1x1 convolutional layer, the module can flexibly regulate the number of channels of the feature map to achieve more effective feature fusion. The 1x1 convolution not only reduces computational resources and prevents overfitting, but also enhances the computational efficiency of the network while retaining important feature information to ensure the full expression and integration of features.

Ultimately, in comparison with the SPPF module based on pooling operation, this module attains more fine-grained feature extraction by means of convolutional operation. Pooling operation may sacrifice some details during the feature extraction process, whereas convolutional operation enjoys higher flexibility and expressive capacity to better capture the details and complex patterns in the image, thereby enhancing the model's overall comprehension and expression of the features.

C. Powerful IoU Loss Function

In target detection tasks, bounding boxes are frequently employed to represent the location and size of a target. The traditional IoU loss function assesses the overlap between two boxes by computing the ratio of the intersection to the union of the two boxes.

$$\mathrm{IoU}(B_a, B_b) = \frac{|B_a \cap B_b|}{|B_a \cup B_b|} \tag{1}$$

Where B_a and B_b denote the prediction frame and the true frame, respectively. The loss function is defined as:

$$L_{IoU} = 1 - \mathrm{IoU}(B_a, B_b) \tag{2}$$

In YOLOv8, to address the bounding box regression issue, the CIoU loss is adopted as the loss function. The CIoU loss is an enhanced loss function that considers factors such as positional offsets, scale differences, and aspect ratios, enabling a more precise assessment of the similarity between the predicted and actual ground truth boxes. The loss function is defined as follows:

$$L_{CIoU} = 1 - \mathrm{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\upsilon \tag{3}$$

$$\upsilon = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right) \tag{4}$$

$$\alpha = \frac{\upsilon}{1 - \mathrm{IoU} + \upsilon} \tag{5}$$

Where: w^{gt} and h^{gt} denote the width and height of the ground truth frame, respectively, and w and h denote the width and height of the prediction frame.c denotes the length

of the diagonal of the minimum bounding rectangle of the prediction frame and the ground truth frame, α denotes the weights, and υ is a parameter that measures the similarity of the aspect ratio between the prediction frame and the ground truth frame. It is Utilized to penalize the case where there is a significant difference in the aspect ratio between the predicted frame and the ground truth frame. $\rho^2(b, b^{gt})$ denotes the Euclidean distance between the center point of the predicted frame and the center point of the ground truth frame. Its calculation formula is:

$$\rho^2(b, b^{gt}) = \left(x_b - x_{gt}\right)^2 + \left(y_b - y_{gt}\right)^2 \quad (6)$$

Although CIoU loss surpasses traditional IOU computation in addressing issues such as bounding box offset and aspect ratio imbalance in target detection, in certain cases, CIoU might cause undue expansion of the anchor box (anchor box) during the regression process. Despite the fact that CIoU introduces penalty terms for centroid distance and aspect ratio, the complex computation of CIoU may not precisely reflect the disparity between the anchor box and the target box in situations where the two boxes do not overlap, thereby increasing the number of parameters and computational complexity of the model, which could result in prolonged convergence time of the regression process.

To tackle these issues, this paper utilizes PIoU as a substitute for CIoU. PIoU integrates a penalty factor that employs the size of the target frame as the denominator and takes into account the adaptation to the quality of the anchor frame [21]. This approach ensures that the anchor frames are regressed along the most efficient path, thereby accelerating model convergence and enhancing detection accuracy. Specifically, the penalty factor P adjusted to the target size is defined as follows:

$$P = \left(\frac{dw_1}{w_{gt}} + \frac{dw_2}{w_{gt}} + \frac{dh_1}{h_{gt}} + \frac{dh_2}{h_{gt}}\right)/4 \quad (7)$$

Where dw_1, dw_2, dh_1, dh_2 are the absolute distances from the corresponding edges of the prediction frame to the target frame, and w_{gt}, h_{gt} are the width and height of the target frame.

Employing P as a penalty factor in the loss function prevents the expansion of the anchor box. This is because the denominator P solely depends on the size of the target box and is not influenced by the size of the anchor box or the minimum closed box of the target. Unlike other penalty factors in the loss function, P does not change with the increase of the anchor box. Additionally, P is zero only when the anchor frame completely overlaps the target frame. P also adapts to the size of the target frame. Therefore, we utilize a penalty function that adjusts in accordance with the quality of the anchor frame.

$$f(x) = 1 - e^{-x^2} \quad (8)$$

$$PIoU = IoU - f(p), -1 \le PIoU \le 1 \quad (9)$$

$$L_{PIoU} = 1 - PIoU = L_{IoU} + f(P), 0 \le L_{PIoU} \le 2 \quad (10)$$

PIoU offers a more precise metric that reflects the disparity between the anchor and target frames by directly minimizing the distance between the four boundaries of the anchor and target frames. PIoU is devised with a succinct formula, which reduces unnecessary computations and enhances the parametric and computational efficiency of the model. By substituting the PIoU loss function for CIoU, the model not only considerably improves the convergence speed but also boosts the regression accuracy and detection performance. Meanwhile, PIoU effectively averts the problem of anchor frame enlargement, providing a more accurate and efficient solution for the target detection task.

IV. Results and Analysis of Results

A. Dataset

The NEU-DET dataset is a special dataset specifically used for surface defect detection on steel plates, generated by research work conducted by Northeastern University. As shown in Figure 7, the dataset contains a total of 1,800 images, covering six types of defect samples, with each sample containing 300 images.Specific defect types encompass: rolled, patches, crazing, scratches, pitted_surfaces, and inclusion. The images within the dataset all have a size of 200 × 200 pixels. To facilitate the training, validation, and testing of the model, these 1800 images were randomly assigned in an 8:1:1 ratio to create a sample consisting of 1440 training samples, 180 test samples, and 180 validation samples.

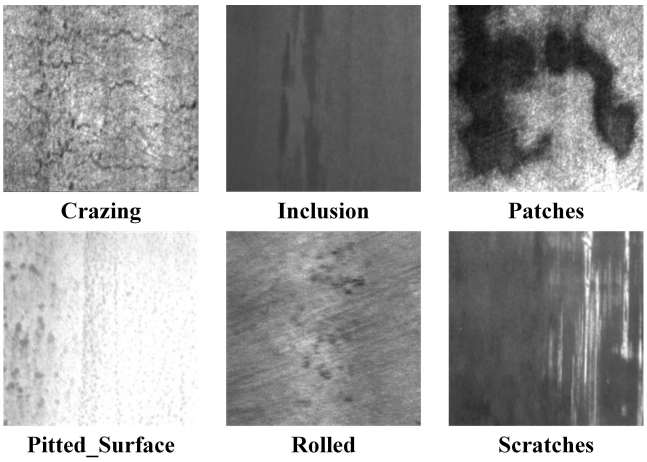


Fig. 7.Six categories of defect samples

B. Experimental Environment

The operating system employed for the experiments in this paper is Windows 11, the CPU is a 13th Gen Intel(R)

TABLE I
Experimental parameter settings

Parameter	Setting
Epoches	200
Input image size	640 × 640
Batch size	16
Initial learning rate	0.01
Momentum	0.937
Optimizer	SGD

Core(TM) i5-13600KF, the GPU is an NVIDIA GeForce RTX 4060 Ti, and the RAM is 16GB (16GB video memory). The deep learning training architecture utilized was Pytorch 2.3.1, and the Python version was 3.9.19. The experimental parameter settings are presented in Table 1.

C. Evaluation Metrics

In this paper, a variety of metrics are adopted to evaluate the performance of the model, including precision (P), recall (R), mean average precision (mAP), frames per second (FPS), and the size of model parameters.

Precision is the ratio of the number of samples accurately predicted as true positives by the model to the total number of samples predicted as true positives by the model, and is computed as follows.

$$\text{Precison} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Recall is the ratio of the number of samples precisely predicted as true positives by the model to the number of all samples that were actually true positives, and can be computed by the following formula.

$$\text{Reecall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

mAP and AP are metrics employed to evaluate multi-category classification problems. mAP is the average of the AP values for all categories, while AP is computed separately for each category. They are computed as follows.

$$\text{AP} = \int_0^1 \text{P(R)dR} \tag{13}$$

$$\text{mAP} = \frac{\sum_{j=1}^{S} \text{AP(j)}}{\text{S}} \tag{14}$$

Where S represents the total number of categories. FPS indicates the number of frames processed per second by the model, and model size represents the size of the storage space occupied by the model. These metrics are crucial for evaluating the performance and adaptability of the model.

D. Experimental Results and Analysis

To validate the effectiveness of each module introduced in this paper, the following ablation experiments were carried out. The experiments are based on YOLOv8n as the baseline and the results are presented in Table 1. In the table, DWR_DRB, FPSC, and PIOU represent the three improvement points proposed in this paper. The symbol √ is employed to indicate that the improvement point is adopted in this ablation experiment.

By using the DWR_DRB module, although there is a slight decline in recall, other metrics such as mAP and FPS are enhanced. Specifically, mAP increases by 3.1%, Precision by 6.3%, and the frame rate by 25 frames/sec. This is attributed to the combined application of the DWR and DRB modules, which fully exploits the outstanding ability of the DWR module to extract multi-scale contextual information and the capacity of the DRB module to detect defects on small-scale patterns. The introduction of the FPSC module leads to a 2.6% improvement in the model's mAp and a 3.8% increase in Precision, with a frame rate of 167 fps. This is because the convolutional layers with different expansion rates can effectively capture both global and local information. With the implementation of the PIoU loss function, there is a remarkable improvement in the accuracy and frame rate of the model. Finally, after integrating the DWR_DRB, FPSC, and PIoU modules, the DFP-YOLO model surpasses the baseline YOLOv8n model in terms of accuracy, mean accuracy, parameters, and frame rate. Under the same dataset, the mean accuracy (mAP) of DFP-YOLO improves by 3.4% compared to that of YOLOv8n, the accuracy rises by 8.8%, and the inference speed increases by 49 frames/sec. Therefore, the DFP-YOLO model proposed in this paper demonstrates excellent detection results in the steel plate surface defect detection task.

To verify the validity of the models, YOLOv8n and DFP-YOLO are tested using the NEU-DET dataset. The experimental results are presented in Table 2. The symbol "↑8.0" in the table indicates that for the defect type of Inclusion, the average accuracy of the DFP-YOLO model is 8% higher than that of the benchmark model. In the table, although the average accuracy of our proposed model decreases by 0.2% in detecting the defect Cr, the average accuracy of the remaining five types of defects are all (mAP) improved. Among them, In and Ro are improved by 8% and 9.7%, respectively, compared with the baseline model, which represents the most significant improvement.

TABLE III
Performance of DFP-YOLO on NEU-DET

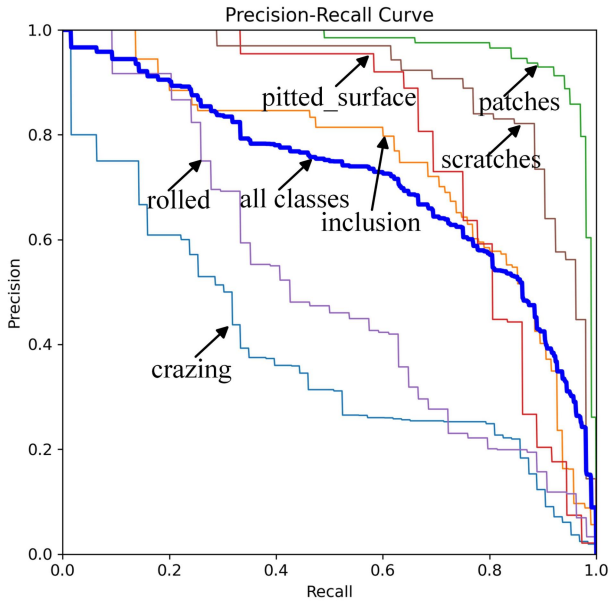
Model	Detect types	P/%	R/%	mAP/%
YOLOv8n	Cr	57.2	23.8	38.3
	In	60.0	84.2	74.2
	Pa	78.8	99.0	96.4
	Ps	72.5	73.1	79.0
	Ro	47.8	57.4	51.2
	Sc	68.7	88.5	90.3
DFP-YOLO（Ours）	Cr	66.2	27.6	38.1（↓0.2）
	In	67.1	72.6	82.2（↑8.0）
	Pa	85.8	96.9	97.9（↑1.5）
	Ps	88.5	66.7	80.2（↑1.2）
	Ro	54.3	40.7	60.9（↑9.7）
	Sc	76.0	88.5	90.8（↑0.5）

Note: Cr, Pa, Ro, Sc, In and Ps respectively denote crazing, patches, rolled, scratches, inclusion and pitted surface.

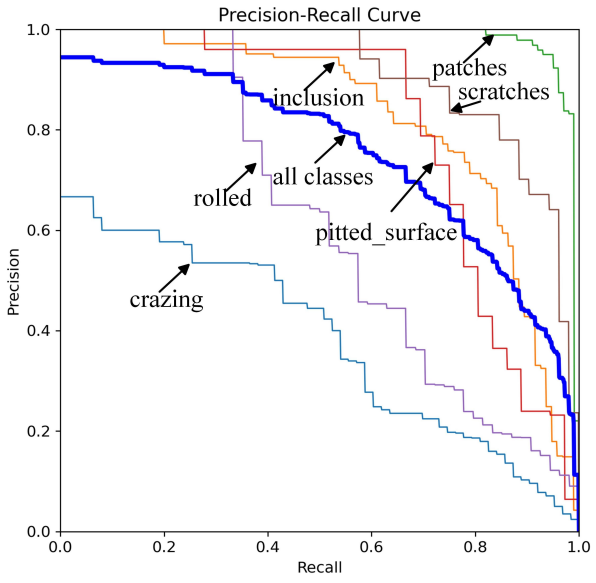
To further illustrate the detection and synthesis capabilities of our proposed model, Fig. 8(a) and (b) respectively display the P-R curves of the YOLOv8n and DFP-YOLO models tested on the NEU-DET dataset. The blue color in the figure represents the mAP curve when Iou is 0.5, and the larger the area enclosed by the curve, the better the overall performance of the model. The area enclosed by our proposed model is evidently larger than that of the benchmark model, which provides an effective validation of the effectiveness of our proposed model. To verify the actual detection effect of the model, we conducted a comparison experiment between the benchmark model and the proposed DFP-YOLO model on a unified dataset, and the experimental results are presented in Fig. 9(a) and (b).

TABLE II
Ablation experiment results

Experiment	DWR_DRB	FPSC	PIoU	P/%	R/%	mAP/%	Params/M	FPS
1				64.2	71.0	71.6	3.15	147
2	√			70.5	66.5	74.7	2.99	172
3		√		68.0	70.1	74.2	3.01	167
4			√	64.4	68.3	73.7	3.15	156
5	√	√		67.1	70.0	74.3	2.84	192
6	√		√	66.9	66.9	73.2	2.99	169
7		√	√	68.5	68.4	74.3	3.01	157
8	√	√	√	73.0	65.5	75.0	2.84	196



（a）YOLOv8n



（b）DFP-YOLO

Fig. 8. P-R curve on the NEU-DET dataset

During the process of steel plate surface defect detection, the benchmark model is interfered by the defective background, resulting in a severe leakage phenomenon and thereby affecting the accuracy and overall precision of the recognition. In contrast, the DFP-YOLO model proposed in this paper significantly enhances the recognition and detection of steel plate surface defects by providing a richer feature representation. This not only improves the detection accuracy but also effectively reduces the probability of leakage and misdetection. The results indicate that the DFP-YOLO model has stronger adaptability and reliability in practical applications.

E. Robustness and Stability Verification Experiments

To further verify the generality and robustness of the DFP-YOLO algorithm, the baseline algorithm and DFP-YOLO were tested on the GC10-DET and PASCAL VOC 2012 public datasets. The GC10-DET dataset contains 2,294 images of real industrial steel plate surface defects covering 10 types of defects, such as punched holes, weld lines, crescent gaps, water spots, oil spots, inclusions, roll pits, creases, and waist folds. The PASCAL VOC 2012 dataset is a widely utilized benchmark dataset that covers a variety of natural scenes and 20 types of targets, including people, animals, vehicles, and indoor objects. It is highly diverse and complex and offers a standardized test platform for the performance evaluation of target detection algorithms. The experimental results are presented in Table 4-5.

TABLE IV
Versatility and robustness verification experiments on GC10-DET dataset.

Model	P/%	R/%	mAP/%	FPS
YOLOv8n	58.1	58.8	59.6	135
DFP-YOLO	60.2	59.7	62.3	113

TABLE V
Versatility and robustness verification experiments on VOC 2012 dataset.

Model	P/%	R/%	mAP/%	FPS
YOLOv8n	68.4	56.8	60.5	155
DFP-YOLO	72.6	59.1	64.1	148

As can be observed from Table 4, the algorithm DFP-YOLO proposed in this paper demonstrates significant advantages in the task of detecting surface defects on industrial steel plates. Compared with the baseline model, mAP increases by 2.7%, Precision rises by 2.1%, and Recall ascends by 0.9%. This indicates that the algorithm proposed in this paper can be adapted to different types of metal defect detection tasks.

Furthermore, from the experimental results in Table 5, it can be discovered that on the larger-scale dataset PASCAL VOC 2012, which has more detection types and a more complex background, the mAP increases by 4.2%, the Precision rises by 2.3%, and the Recall ascends by 3.6% compared to the baseline model. This indicates that the DFP-YOLO algorithm is capable of learning and capturing the target features better, demonstrating its robustness and good generalization.

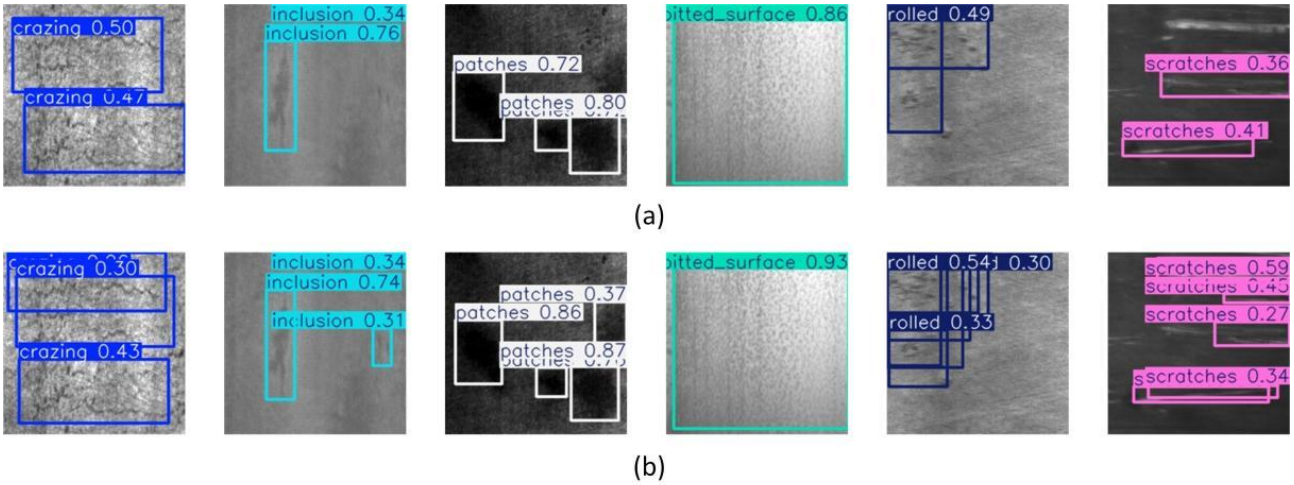


Fig. 9. The actual detection effect of the model（a）The detection performance of the YOLOv8n model（b）The detection performance of the DFP -YOLO model

F. Comparison with Mainstream Models

To verify the performance of the improved method presented in this paper, the results of the improved model proposed herein are compared with those of six mainstream models of SSD, RT-DETR, and YOLO series under the same dataset conditions.

TABLE VI
Comparison with mainstream models

Model	P/%	R/%	mAP/%	Params/M	FPS
SSD	71.8	65.1	71.5	21.6	52
RT-DETR	69.6	67.3	70.0	31.99	44
YOLOv3tiny	57.4	67.9	68.1	12.14	88
YOLOv5n	69.1	63.3	69.2	7.3	142
YOLOv7tiny	67.7	49.7	66.5	6.03	144
YOLOv8n	64.2	71.0	71.6	3.15	147
Ours	73.0	65.5	75.0	2.84	196

As indicated by the data in Table IV, the DFP-YOLO algorithm proposed in this paper surpasses the other models in both P-value (73.0%) and mAP-value (75.0%), manifesting its outstanding performance in the target detection task. Particularly, DFP-YOLO exhibits a distinct lead in the mAP metric, suggesting that the model is more balanced regarding its ability to detect different categories. Although there is a reduction in the R-value of our model, it still fulfills the actual detection requirements. Additionally, DFP-YOLO has a considerably lower number of parameters (2.84M) than many mainstream models (e.g., SSD, RT-DETR, and YOLOv3tiny, etc.), which renders it more efficient in deployment and operation. In terms of frame rate (FPS), our model performs exceptionally well, reaching 196 frames per second, which far exceeds other models. This high frame rate endows DFP-YOLO with a significant advantage in real-time detection scenarios.

V. Conclusion

In this paper, several enhancements based on the YOLOv8n model are carried out with the objective of boosting the performance of steel plate surface defect detection. These enhancements involve the incorporation of the DWR_DRB module in the backbone network and Neck layer to enhance the accuracy and efficiency of feature extraction. Additionally, the original SPPF module is substituted with the FPSC module to strengthen the model's capability of representing complex targets, and the original CIoU loss function is replaced by the PIoU loss function to enhance the model's regression accuracy and convergence speed.

Specifically, the addition of the DWR_DRB module can markedly enhance the feature extraction ability of the model at different scales, thereby improving the overall detection accuracy and reducing the computational volume. The FPSC module achieves better capture of image context information by employing convolutional layers with different expansion rates, which can capture the detailed information of the target more effectively, especially when dealing with complex targets, and performs excellently. After replacing the CIoU loss function with the PIoU loss function, the regression accuracy of the model is enhanced, and the convergence speed of training is accelerated, which contributes to improving the accuracy of localization and the robustness of the model.

The experimental results demonstrate that the model following these improvements has attained better detection accuracy, speed and robustness in the steel plate surface defect detection task, which validates the effectiveness and practical application value of the improved method.

References

[1] H. Dong, Y. Liu, L. Wang, X. Li, Z. Tian, Y. Huang, and C. McDonald, "*Roadmap of China steel industry in the past 70 years," Ironmaking & Steelmaking*, vol. 46, no. 10, pp. 922–927, 2019.

[2] B. Tang, L. Chen, W. Sun, and Z.-k. Lin, "Review of surface defect detection of steel products based on machine vision," *IET Image Processing*, vol. 17, pp. 303-322, 2023.

[3] C. D. Soukup and R. Huber-Mörk, "Convolutional neural networks for steel surface defect detection from photometric stereo images," in *Proc. Int. Symp. Visual Computing*, Cham: Springer International Publishing, 2014, pp. 668–677.

[4] Z. X. Zhang, W. H. Cui, Y. Tao, and T. W. Shi, “Road Damage Detection Algorithm Based on Multi-scale Feature Extraction,” *Engineering Letters*, vol. 32, no. 1, pp. 151-159, 2024..

[5] J. Yu, X. Cheng, and Q. Li, "Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion," *IEEE Trans. Instrum. Meas*, vol. 71, pp. 1–10, 2021.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 39, pp. 1137–1149, 2016.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016, pp. 779–788.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Part I*, Cham, Switzerland: Springer International Publishing, Oct. 11–14, 2016, pp. 21–37.
[9] X. Zhang, Y. Wang, and H. Fang, "Steel surface defect detection algorithm based on ESI-YOLOv8," *Materials Research Express*, vol. 11, no. 056509, pp. 276–284, 2024.
[10] Y. Z. Fu, L. Qiu, X. Kong, and H. F. Xu, "Deep Learning-Based Online Surface Defect Detection Method for Door Trim Panel," *Engineering Letters*, vol. 32, no. 5, pp. 939-948, 2024.
[11] J. Wang, Y. Wang, A. Sun, and Y. Zhang, "A lightweight network FLA-Detect for steel surface defect detection," Preprint, 10 July 2024.
[12] Z. Mi, Y. Gao, X. Xu, and J. Tang, "Steel strip surface defect detection based on multiscale feature sensing and adaptive feature fusion," *AIP Advances*, vol. 14, no. 4, 045005, April 2024.
[13] F. Selamet, S. Cakar, and M. Kotan, "Automatic Detection and Classification of Defective Areas on Metal Parts by Using Adaptive Fusion of Faster R-CNN and Shape From Shading," *IEEE Access*, vol. 10, pp. 126030-126045, 2022.
[14] X. Chen, J. Lv, Y. Fang, and S. Du, "Online Detection of Surface Defects Based on Improved YOLOV3," *Sensors*, vol. 22, no. 3, p. 817, 2022.
[15] Y. Qu, B. Wan, C. Wang, H. Ju, J. Yu, Y. Kong, and X. Chen, "Optimization Algorithm for Steel Surface Defect Detection Based on PP-YOLOE," *Electronics*, vol. 12, no. 4161, pp. 1–18, 2023.
[16] R. Wei, Y. Song, and Y. Zhang, "Enhanced Faster Region Convolutional Neural Networks for Steel Surface Defect Detection," *ISIJ International*, vol. 60, no. 3, pp. 539-545, 2020.
[17] H. Wang, X. Yang, B. Zhou, Z. Shi, D. Zhan, R. Huang, J. Lin, Z. Wu, D. Long, "Strip Surface Defect Detection Algorithm Based on YOLOv5," *Materials*, vol. 16, no. 7, p. 2811, 2023.
[18] Y. Wang, H. Wang, and Z. Xin, "Efficient detection model of steel strip surface defects based on YOLO-V7," *IEEE Access*, vol. 10, pp. 133936-133944, 2022.
[19] X. Song, S. Cao, J. Zhang, and Z. Hou, "Steel Surface Defect Detection Algorithm Based on YOLOv8," *Electronics*, vol. 13, no. 5, p. 988, 2024.
[20] Y. Huang, W. Tan, L. Li, L. Wu, "WFRE-YOLOv8s: A New Type of Defect Detector for Steel Surfaces," *Coatings*, vol. 13, no. 12, p. 2011, 2023.
[21] C. Liu, K. Wang, Q. Li, F. Zhao, K. Zhao, H. Ma, "Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism," *Neural Networks*, vol. 170, pp. 276–284, 2024.
[22] X. Jiang, Y. Cui, Y. Cui, R. Xu, J. Yang, and J. Zhou, "Optimization Algorithm of Steel Surface Defect Detection Based on YOLOv8n-SDEC," *IEEE Access*, vol. 12, pp. 95106–95117, 2024.
[23] Z. Li, X. Wei, M. Hassaballah, Y. Li, and X. Jiang, "A deep learning model for steel surface defect detection," *Complex & Intelligent Systems*, vol. 10, pp. 885–897, 2024.
[24] X. Li and Y. Zhang, "Improved Road Damage Detection Algorithm Based on YOLOv8n," *IAENG International Journal of Computer Science*, vol. 51, no. 11, pp. 1720-1730, 2024.