A lightweight and high-accuracy algorithm for strip steel surface defect detection

J.N. Wang [1,2,3], X. Xie [2,], L. Wu [1,3], W.Y. Bai [1] , W.Y. Guo [1]*
[1] School of Software Engineering, Anyang Normal University, Anyang, 455000, Henan, China;
[2] School of Information and Software Engineering, East China Jiaotong University, Nanchang, 330013, Jiangxi, China;
[3] Engineering Research Center for Intelligent Industry of Oracle Bone Culture in Henan Province, Anyang, 455000, Henan, China

Abstract

Given the dual demands for lightweight design and high precision in detecting surface defects of strip steel within complex industrial environments, this paper develops an improved YOLOv8 algorithm based on the YOLOv8n architecture. To start with, a SlimNeck lightweight structure is integrated into the neck network to reconstruct the feature extraction pathway, allowing the improved YOLOv8 model to markedly reduce its complexity while preserving robust feature expression capabilities. Next, the DySample dynamic upsampling operator is utilized in the neck network instead of the conventional UpSample operator, aiming to boost the model's sensitivity to minute defects on the strip steel surface. Finally, the InnerMPDIoU loss function is incorporated to optimize bounding box regression accuracy, thereby fortifying the model's adaptability to the multi-scale features of strip steel surface defects. Experimental results on the strip steel surface defect dataset, an open-source resource from Northeastern University (NEU-DET), indicate that the improved YOLOv8 algorithm achieves substantial improvements in detection accuracy while retaining its lightweight edge. Compared to the YOLOv8n baseline, it reduces parameters by 6.64 %, lowers computational load by 9.88 %, and maintains a model weight of just 5.9 MB. At the same time, its precision (P) reaches 80.1 % and mean average precision (mAP) hits 78.6 %, with these values being 3 % and 1.2 % higher than those of the baseline respectively. Consequently, the improved YOLOv8 algorithm proposed in this paper successfully balances lightweight design with high-precision detection performance, effectively meeting the dual requirements for strip steel surface defect detection tasks.

Keywords: strip steel surface defect detection, YOLOv8, lightweight model, high accuracy detection, SlimNeck structure, DySample module, InnerMPDIoU loss function.

Citation: Wang JN, Xie X, Wu L, Bai WY, Guo WY. A lightweight and high-accuracy algorithm for strip steel surface defect detection. Computer Optics 2026; 50(5): 1808. doi: 10.18287/COJ1808.

Introduction

As a commonly utilized foundational material in the iron and steel industry, strip steel holds significant importance in numerous key sectors including construction, automobile manufacturing, aerospace, and shipbuilding [1], with its surface quality directly impacting the performance, safety, and service life of end products. When surface defects such as cracks, patches, or scratches appear on strip steel, this not only reduces the material's own corrosion resistance and structural strength but also may trigger a series of quality issues during subsequent processing and use, resulting in substantial economic losses for enterprises. Thus, efficient and precise identification of surface flaws in strip steel stands as a core link in ensuring strip steel production quality and is vital for enhancing the efficiency of the iron and steel industry. However, in practical production, traditional methods for detecting strip steel surface defects still face multiple bottlenecks [2]. For instance, manual inspection is characterized by both inefficiency and a susceptibility to subjective influences like experience and fatigue of inspectors, making it hard to satisfy the quality management requirements of contemporary high-speed manufacturing lines. While technologies like eddy current inspection and magnetic flux leakage detection have advanced in detection principles, they are constrained by issues including slow detection speeds and weak anti-interference capabilities, preventing large-scale industrial application. Traditional machine vision-based detection methods have achieved notable improvements in speed and accuracy, yet they rely heavily on domain experts to manually design features, a reliance that not only leads to a surge in workload but also makes them difficult to adapt to the high standards of modern steel production for defect detection. Against the backdrop of traditional detection methods struggling to overcome these limitations, the swift advancement of deep learning over the past few years has brought revolutionary breakthroughs to industrial defect detection. With its core strengths of great efficiency and precision, deep learning has quickly gained widespread application across various industrial defect detection scenarios, particularly within the domain of object detection where remarkable progress has been made, offering new insights and solutions for strip steel surface defect detection.

Against this backdrop, a growing number of researchers have turned to deep learning methods for defect detection studies. For instance, Li et al. [3] developed a defect detection algorithm based on YOLOv3. This algorithm first employs a weighted K-means clustering approach to optimize and determine prior box parameters, then adjusts YOLOv3's

network architecture to fuse shallow and deep features, thus boosting the network's accuracy in defect detection. However, this model suffers from a relatively high parameter count. Zhou et al. [4] constructed a lightweight strip steel surface defect detection model, YOLOv5s-GCE, based on YOLOv5. This model integrates Ghost modules in place of certain CBS structures within the original YOLOv5s framework, while replacing bottleneck components in C3 with Ghost bottleneck structures. These modifications help minimize the model size and achieve network lightweighting. Additionally, the model incorporates the CA coordinate attention mechanism to strengthen key feature channels and their positional information, boosting the ability to accurately recognize and locate targets. Yet, in situations involving the detection of surface defects on strip steel, the model still has shortcomings in detecting specific defect types. Yu et al. [5] put forward the CABF-FCOS algorithm, an improved version based on FCOS. This algorithm first selects the anchor-free FCOS as its detection architecture to get rid of the impact of anchor box-related hyperparameters. Second, it designs a CAM module to reduce feature information loss during processing. Finally, it replaces the original FPN with the BFFN, making image feature fusion more efficient. After improvement, the network achieves a detection accuracy of 76.68 %, but this accuracy still has space for enhancement, and the model's parameter count remains relatively high, rendering it challenging to be deployed on resource-limited devices like edge terminals. Zou et al. [6] developed a strip steel defect detection algorithm based on YOLOv4-tiny, integrating multi-scale detection and a spatial attention mechanism. Tests indicate that it attains a detection precision of 73.29 % with an FPS of 163. While this algorithm offers high detection speed, its accuracy remains relatively low. Cao et al. [7] built an SPP_Res feature pyramid structure with residual edges based on the YOLOv5 algorithm and introduced the C3_MHSA multi-head attention mechanism. These two improvements effectively boost the precision of detecting surface flaws in steel. However, the algorithm still performs inadequately in detecting some small-scale targets. Deng et al. [8] proposed an improved Faster R-CNN-based detection algorithm capable of efficient, high-precision identification of surface flaws in hot-rolled strip steel. The algorithm's improvements mainly involve three aspects: First, fusing low-level detailed features with high-level semantic features through feature addition. Second, obtaining fixed-size feature vectors via Precise ROI Pooling to avoid feature position deviation. Third, using the mean shift clustering algorithm to cluster the strip steel dataset and obtain prior box sizes suitable for detection of surface defects in hot-rolled strip steel. Experimental results show the algorithm achieves a mAP of 85.34 % on the hot-rolled strip steel surface defect detection dataset, but there is still room to further boost the precision of identification for certain defect types on the strip steel surface. Yang et al. [9] put forward a modified YOLOv7-tiny detection algorithm to tackle problems in the detection of surface defects on strip steel, such as low small-target detection efficiency, insufficient defect positioning accuracy, and a large algorithm parameter count. The improvement strategies include: First, introducing the GSConv structure to replace standard convolution in the neck network and optimizing the ELAN module with GSConv to strengthen the network's feature fusion ability while reducing parameter numbers. Second, embedding the SPDConv module for low-resolution small targets before the prediction head to improve small-defect detection performance through secondary defect feature extraction. Third, employing the MPDIoU loss function to enhance the algorithm's positioning accuracy. Experimental results confirm the improved algorithm's effectiveness, but there is still scope to enhance both real-time detection performance and generalization ability.

While the methods mentioned above have yielded certain outcomes, existing approaches still leave room for enhancement when it comes to meeting the dual demands of lightweight design and the high level of precision necessitated by tasks involving the detection of surface defects in strip steel. Consequently, given that current strip steel surface defect detection methods struggle to satisfy the needs for lightweight and high-precision detection, this paper modifies the YOLOv8n defect detection network and presents the SNDS-YOLO algorithm, which aims to realize high-precision identification of surface defects in strip steel while realizing model lightweighting. Its key innovations are as follows:

1. Lightweight transformation of the neck network: The Slim-Neck lightweight structure is innovatively incorporated into the neck network, which employs lightweight GSConv module instead of traditional Conv convolution to slim down the model. Meanwhile, it utilizes the VoV-GSCSP module in place of the original C2f module, thereby enhancing cross-level feature fusion. The synergistic effect of these two modifications can notably decrease the quantity of model parameters and effectively reduce computational burden, establishing a sturdy basis for the efficient deployment of the model across various devices, particularly terminal devices with limited resources.
2. Dynamic feature adaptation mechanism: Also within the neck network, DySample is integrated to replace the conventional UpSample. Unlike the conventional UpSample that uses a fixed sampling method, DySample can adaptively adjust sampling weights based on the semantic information in the input feature map, enabling the framework to retain high detection sensitivity and accuracy when handling strip steel surface defects of varying scales.
3. Loss function optimization: At the loss function level, the InnerMPDIoU loss function is utilized as a substitute for the conventional CIoU. In comparison to the conventional CIoU, the InnerMPDIoU loss function can more effectively enhance the model's positioning accuracy for tiny defects by reinforcing fine constraints on the overlapping area of bounding boxes. This enhancement can yield more pronounced effects, particularly in scenarios where the edges of strip steel surface defects are relatively blurred.

1. YOLOv8 algorithm

YOLOv8, a YOLO-series object detection algorithm released by Ultralytics [10], has gained widespread use in numerous areas such as target detection and image classification. Its network architecture includes four core modules: the

input module, backbone network, neck structure, and head structure [11]. The input module manages preprocessing and input of image data, the backbone network assumes the core role of feature extraction, the neck structure focuses on fusing multi-scale feature maps output by the backbone network [12], and the head structure specifically carries out detection and classification tasks. Notably, YOLOv8 has developed multiple versions such as YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x based on different parameter counts and computational amounts, with model weight sizes increasing sequentially across these versions. Given the practical demand for model lightweighting in industrial strip steel surface defect detection scenarios, in this paper, YOLOv8n, which has the smallest number of parameters, is selected as the benchmark model for improvement.

2. Improved YOLOv8 algorithm

Over the past few years, the fast-paced progress of deep learning technology has fueled substantial improvements in object detection algorithms in various areas. In this context, the YOLO series algorithms, characterized by their efficiency and precision, have demonstrated outstanding performance in real-time object detection tasks. As an iterative update of this series, YOLOv8 has not only made new breakthroughs in detection speed and accuracy but also significantly boosted its adaptability to diverse application scenarios by offering abundant model configuration options, thereby further expanding its range of applications. Nevertheless, despite the YOLOv8 algorithm exhibits outstanding performance in the domain of object detection, when it is directly applied to the specific scenario of detecting surface defects in strip steel, there are still prominent problems including poor defect detection precision and redundant network structure. To address bottlenecks like insufficient precision and subpar real-time capability in the detection of surface defects on strip steel, this paper presents the SNDS-YOLO algorithm, an improved version based on YOLOv8n for detecting surface flaws in strip steel. Specifically, its improvement ideas are as follows: First, the SlimNeck lightweight structure is incorporated into the neck network, and by reconstructing the feature extraction path, the model's level of intricacy is significantly reduced on the premise that the feature expression ability remains undamaged. Second, for the upsampling link in the neck network, the DySample structure is adopted to replace the traditional UpSample structure, so as to boost the model's perception sensitivity to tiny defects on the strip steel surface. Finally, the InnerMPDIoU is incorporated to refine the bounding box regression procedure, thus further enhancing the bounding box locating precision in defect detection. Through these improvements, the SNDS-YOLO algorithm can better adapt to the dual requirements of lightweight design and high accuracy in detecting surface defects of strip steel, and the architecture of its network is illustrated in Fig. 1.

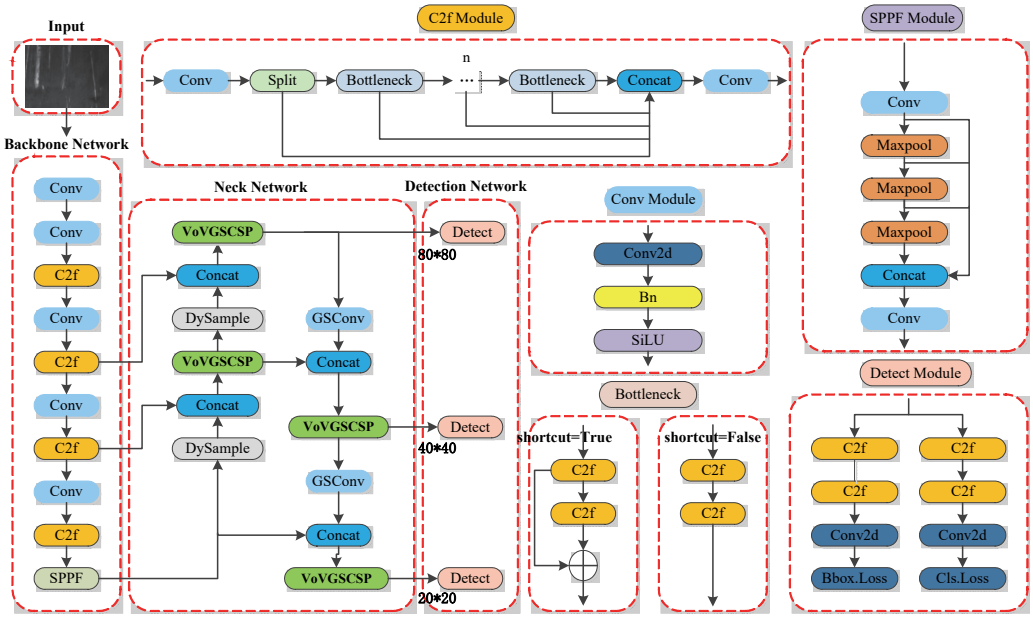


Fig. 1. SNDS-YOLO model network architecture

2.1. SlimNeck structure

Within the realm of machine vision, enhancing a model's computational accuracy does not simply depend on increasing parameter counts but rather relies more on the application of lightweight design strategies. Amid this trend, researchers continue to explore approaches to reduce model complexity while maintaining or even enhancing performance. Depth-wise separable convolution, for instance, is a common lightweight method. However, while it can effectively reduce model parameters and computational time complexity, its ability for feature extraction and fusion is far less robust compared to standard convolution. This practical contradiction creates room for further optimization. The Slim-Neck structure, designed based on the theories of DensNet [13], VoVNet [14], and CSPNet [15], can specifically

tackle this issue, offering new insights for balancing lightweight properties and feature expression capabilities. Given this, to achieve model lightweighting while further improving its detection performance, this paper chooses to draw on the Slim-Neck structure's design ideas for balancing model complexity and feature extraction capabilities, and conducts targeted optimization on the neck network of YOLOv8. The specific optimization measures are as follows.

Firstly, traditional standard convolution is replaced with lightweight GSConv convolution. This convolution is constructed by innovatively integrating standard convolution, depth-wise separable convolution, and shuffle operations. In comparison to traditional convolution, GSConv can reduce computational costs to half of the former while achieving comparable performance improvements. Additionally, it strikes a better balance between model detection accuracy and operating speed, which establishes a foundation for the neck network to maintain or even enhance feature processing capabilities during lightweight transformation. Its architecture is illustrated in Fig. 2.

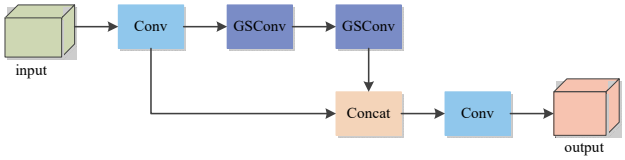


Fig. 2. GSConv network architecture

Secondly, on this basis, considering that the optimization of a single convolution layer is difficult to fully release the lightweight potential of the neck network, the design idea of the GSConv structure is further borrowed to construct the cross-level partial network structure VoV-GSCSP. By integrating the core advantages of cross-level feature interaction mechanism and lightweight convolution, this module can effectively maintain or even strengthen the information integrity in the feature transmission process while simplifying the overall network structure and reducing redundant computations, thus ensuring that the model accuracy is not negatively affected by the lightweight transformation. The design of this module not only continues the lightweight concept of GSConv but also enhances the effectiveness of feature fusion through cross-level connections, providing more comprehensive support for the overall performance improvement of the neck network. Its specific architecture is presented in Fig. 3.

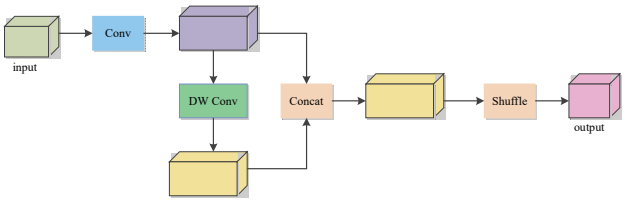


Fig. 3. VoV-GSCSP network architecture

Finally, based on the above step-by-step optimization of the convolution layers and module structure, this paper applies the Slim-Neck lightweight structure to the Neck part of YOLOv8. On the one hand, the lightweight GSConv is used to completely replace the traditional Conv convolution to continue its advantages in streamlining parameters and reducing computational costs. On the other hand, the newly constructed VoV-GSCSP structure is adopted to take the place of the original C2f structure to strengthen the effectiveness of cross-level feature fusion. These two improvements form a complementary and synergistic optimization system, where the former achieves model "slimming" at the level of basic operation units, and the latter ensures efficient transmission of feature information through module architecture innovation. The combined effect of the two enables the improved neck network to significantly decrease the parameter count and computational intricacy while successfully achieving a simultaneous improvement in detection precision rate, ultimately realizing the dual optimization goals of lightweight and high performance.

2.2. Dynamic Sample module

Dynamic Sample[16] has a core advantage over traditional static upsampling operators like bilinear interpolation and CARAFE. This advantage lies in decoupling the mechanisms of sampling point generation and feature interpolation, which can significantly reduce parameter counts and computational overhead while dynamically capturing and focusing on feature details in target regions. This characteristic enables it to perform prominently in lightweight model deployment scenarios, rendering it highly effective for real-time detection in resource-limited environments. Additionally, traditional UpSample uses a fixed sampling method, which often fails to accurately adapt to defect features of varying scales and forms when handling surface defect inspection of strip steel in demanding industrial settings. In contrast, DySample can adaptively adjust sampling weights based on the semantic information contained in input feature maps, allowing the model to retain high detection sensitivity and accuracy under complex industrial conditions. Therefore, to boost model performance while maintaining its lightweight nature, this paper chooses the DySample operator to replace traditional UpSample, thus boosting the enhanced model's defect detection capability. Its basic framework is presented in Fig. 4.

First, grid sampling is performed on a given input feature map vector χ of size $C\times H\times W$ and a sampling set $\boldsymbol{S}$ of size $2g\times sH\times sW$, where 2g represents the coordinates of x and y, i.e., the network coordinate points. The locations within $\boldsymbol{S}$ are

used to perform resampling on the bilinear interpolation estimate χ into a feature map vector χ' of size $C\times sH\times sW$, and its formula can be described as:

$$\chi' = \text{grid_Sample}(\chi, S). \quad (1)$$

In the formula, grid_Sample is a function provided by PyTorch.

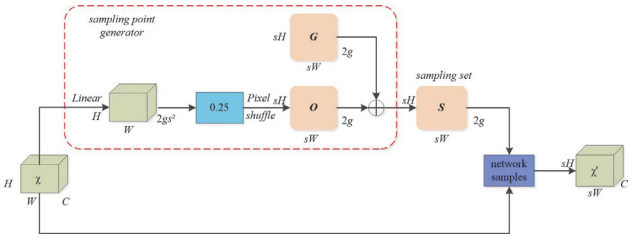


Fig. 4. Dynamic Upsampling Framework

Second, taking scaling parameter s and feature tensor χ ($C\times H\times W$), a linear layer maps C channels to $2s^2$ channels, producing offset $\boldsymbol{O}$ ($2s^2\times H\times W$). Following pixel shuffle to $2\times sH\times sW$ dimensions, $\boldsymbol{S}$ is computed by adding $\boldsymbol{O}$ to grid $\boldsymbol{G}$, and its formula is as follows:

$$\boldsymbol{O} = \text{linear}(\chi), \quad (2)$$
$$\boldsymbol{S} = \boldsymbol{G} + \boldsymbol{O}. \quad (3)$$

In addition, the normalization layer typically constrains output values between -1 and 1, which may cause local s^2 sampling positions to overlap. To alleviate this situation, a predefined range coefficient is added, that is, the offset is multiplied by a coefficient of 0.25, so that the sampling positions are subject to local constraints, and its formula is:

$$\boldsymbol{O} = 0.25\text{linear}(\chi). \quad (4)$$

Finally, the flexibility of the offset can also be increased by generating point-wise dynamic range scaling parameters via linear transformation of input features. The approach utilizes a Sigmoid activation and a constant 0.5 multiplier., and its formula is as follows:

$$\boldsymbol{O} = 0.5Sigmoid\big(linear_1(\chi)\big) \cdot linear_2(\chi). \quad (5)$$

2.3. InnerMPDIoU loss function

The loss function evaluates prediction errors against true measurements, the smaller its value, the closer the model's predictions are to reality. Among IoU-based loss functions, different types have distinct focuses. GIoU [17], for instance, incorporates consideration of the minimum enclosing region alongside IoU, allowing it to direct the model's refinement of bounding box predictions. DIoU [18], along with attention to prediction-ground truth box overlaps, emphasizes characterizing their relative positional relationship, addressing GIoU's insensitivity to overlap changes in certain scenarios. CIoU [19] comprehensively combines constraints on overlapping region, midpoint distance, and dimensional ratio, performing better when handling targets of varying shapes and sizes. EIoU [20] strengthens attention to width-height ratios based on CIoU, enhancing the model's adaptability to targets of different forms. However, these loss functions share a common limitation: when the detection and reference boxes with identical aspect ratios but varying dimensions, they struggle to effectively distinguish this deviation. To resolve this issue, the MPDIoU was proposed [21]. This function calculates loss based on the minimum point distance of horizontal rectangles, comprehensively accounting for overlapping area, center point distance, and width-height deviation to enable more precise model loss quantification. For this reason, this paper introduces the MPDIoU to improve the model's regression efficiency and precision, and further incorporates Inner-IoU to enhance model performance. The calculation diagram of the MPDIoU is presented in Fig. 5.

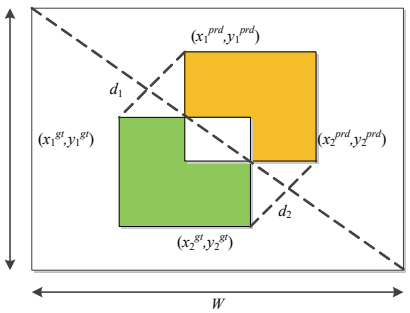


Fig. 5. MPDIoU Calculation Diagram

The MPDIoU loss function is computed as:

$$L_{MPDIoU} = 1 - MPDIoU, \tag{6}$$

$$MPDIoU = IoU - \frac{d_1^2+d_2^2}{W^2+H^2}, \tag{7}$$

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \tag{8}$$

$$d_1^2 + d_2^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2 + \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2. \tag{9}$$

The MPDIoU loss function is computed as:

In the formula, IoU represents the intersection over union, W and H denote the input image's width and height dimensions, x_1^{prd}, x_2^{prd}, y_1^{prd}, y_2^{prd} delineate the predicted region, x_1^{gt}, x_2^{gt}, y_1^{gt}, y_2^{gt} define the ground truth box coordinates, d_1 measures the top-left corner displacement between predicted and ground truth boxes, while d_2 quantifies their bottom-right corner separation.

The MPDIoU loss function incorporates distances between both corner points of the bounding boxes, enabling more precise measurement of prediction errors. It shows strong performance in differentiating boxes with identical aspect ratios but varying dimensions or locations. The core idea of the Inner-IoU loss function is to introduce an auxiliary bounding box to participate in the loss calculation [22], and to rescale the auxiliary bounding box via a proportional factor, so as to adapt to different datasets and detection models. Given a ground truth box B^{gt} and an anchor box B, the Inner-IoU is defined as:

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} \cdot ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} \cdot ratio}{2}, \tag{10}$$

$$b_t^{gt} = y_c^{gt} - \frac{b^{gt} \cdot ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{b^{gt} \cdot ratio}{2}. \tag{11}$$

In the formula, x_c^{gt}, and y_c^{gt} denote the horizontal and vertical coordinates of B^{gt}'s center point, while w^{gt} and b^{gt} represent its width and height dimensions. The value range of ratio is usually [0.5, 1.5]. The coordinates b_l^{gt}, b_r^{gt}, b_t^{gt}, and b_b^{gt} define the left, right, top, and bottom edges of B^{gt}'s auxiliary bounding box.

$$b_l = x_c - \frac{W \cdot ratio}{2}, b_r = x_c + \frac{W \cdot ratio}{2}, \tag{12}$$

$$b_t = y_c - \frac{b \cdot ratio}{2}, b_b = y_c + \frac{b \cdot ratio}{2}. \tag{13}$$

In the formula, x_c and y_c denote the horizontal and vertical positions of B's centroid, while b specifies its height. The auxiliary box margins are given by b_l, b_r, b_t, and b_b for left, right, top, and bottom edges respectively.

$$\text{inter} = \left(\min\left(b_r^{gt}, b_r\right) - \max\left(b_l^{gt}, b_l\right)\right), \left(\min\left(b_b^{gt}, b_b\right) - \max\left(b_t^{gt}, b_t\right)\right), \tag{14}$$

$$union = (w^{gt} \cdot b^{gt})(ratio)^2 + (wb)(ratio)^2 - inter, \tag{15}$$

$$IoU^{inner} = \frac{inter}{union}. \tag{16}$$

In the formula, the terms inter and union denote the overlapping region and combined area of both auxiliary boxes. IoUinner denotes the intersection over union of Inner-IoU.

Finally, integrating Inner-IoU into the MPDIoU produces the composite InnerMPDIoU expression:

$$L_{InnerMPDIoU} = L_{MPDIoU} + IoU - IoU^{inner}, \tag{17}$$

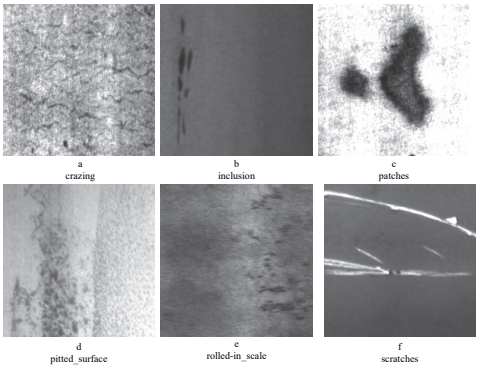


Fig. 6. Defect samples from the NEU-DET dataset

3. Experimental results and analysis

In this study, the NEU-DET strip steel surface defect dataset, an open-source resource from Northeastern University, was utilized [23]. This dataset comprises six common defect types: crazing, inclusion, patches, pitted_surface, rolled-

in_scale, and scratches, with sample images of these defects displayed in Fig. 6. Each defect type in the dataset includes 300 images, amounting to a total of 1800 images, and each image is accompanied by labels in xml file format that contain details such as image dimensions, defect positions, and categories. During the data preprocessing phase, the NEU-DET dataset was converted into YOLO format to meet the input specifications of YOLO series models. Following this, the dataset was divided randomly into training (1440 images), validation (180 images), and test (180 images) sets following an 8:1:1 distribution ratio.

To guarantee unbiased and comparable experimental outcomes, this paper conducts experiments in a unified experimental environment. Firstly, all experiments are tested using the hardware and software environments shown in Table 1. Secondly, consistent hyperparameter configurations are adopted in the model training process, including the use of the SGD optimizer, an initial learning rate set to 0.01, a fixed training batch size of 16, and a unified training number of 300 epochs.

Tab. 1. Experimental setup arrangement

Item	Description
Operating System	Ubuntu 20.04
CPU	AMD EPYC 9K84
GPU	NVIDIA H20-NVLink-96GB
Programming Language	Python 3.8.10
Deep Learning Framework	PyTorch 2.0.0
Memory	150 GB
CUDA Version	CUDA 11.8

3.1. Experimental evaluation metrics

This study adopts multiple evaluation criteria to thoroughly analyze the enhanced model's capabilities [24, 25]. Detection accuracy is measured through Precision (P), Recall (R), and mean Average Precision (mAP) to assess strip steel defect identification performance. For operational efficiency, practical deployment considerations are evaluated using model parameters (Params), computational load (GFLOPs), and storage requirements (Weight size). The mathematical formulations are:

$$P = \frac{TP}{TP+FP}, \tag{18}$$

$$R = \frac{TP}{TP+FN}, \tag{19}$$

$$AP = \int_0^1 P(R)d(R), \tag{20}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} A\,P_i. \tag{21}$$

Where, P is Precision, R is Recall, TP counts true detections, FP false alarms, and FN missed detections. AP_i represents class-wise precision, and mAP the macro-average over N categories.

3.2. Ablation experiment

Five ablation studies were conducted to validate the enhancements in the SNDS-YOLO algorithm, and implemented by gradually incorporating improvement modules. These experiments employed Params, GFLOPs, Weight size, Precision, and mAP@0.5 as evaluation criteria, with the results presented in Tab. 2. Here, M1 denotes the YOLOv8n baseline model; M2 refers to the model with the SlimNeck module added to M1; M3 is the model integrating the DySample module into M1; M4 represents the combined improvement of M2 and M3, meaning both the SlimNeck and DySample modules are introduced; and M5 further adds the InnerMPDIoU loss function to M4, forming the SNDS-YOLO algorithm that includes three improvement modules: the SlimNeck module, the DySample module, and the InnerMPDIoU loss function. An analysis of the data in Tab. 2 reveals that after adding the SlimNeck module, the M2 model achieved lightweighting. Compared with the YOLOv8n model, M2 saw a 6.98 % reduction in parameters, a 9.88 % decrease in computational load, a model weight size of only 5.9 MB, along with a 1.4 % increase in P value and a 0.7 % rise in mAP value. The M3 model showed no performance improvement after introducing the DySample module, but when the SlimNeck and DySample modules were further combined to form M4, the model's performance in various aspects improved significantly compared with using a single module alone. Relative to the YOLOv8n model, M4 had 6.64 % fewer parameters, 9.88 % lower computational load, a 6.35 % reduction in model weight size, as well as a 1.5 % increase in P value and a 0.6 % growth in mAP value, indicating that the combination of the two modules produced a positive synergistic effect. The M5 model is formed by adding the InnerMPDIoU loss function to M4. Compared with M2 and M3, its P value increased by 1.6 % and 3.1 % respectively, and its mAP value rose by 0.5 % and 3.2 % respectively; compared with M4, its P value was 1.5 % higher and its mAP value was 0.6 % higher. Finally, relative to the YOLOv8n model, the proposed SNDS-YOLO model exhibited a 6.64% reduction in parameters, a 9.88 % decrease in computational load, a 6.35 % shrinkage in model weight size, while the P value increased significantly by 3 % and the mAP value rose notably by 1.2 %, demonstrating that the improved SNDS-YOLO model achieved both lightweighting and high-precision

detection of strip steel surface defects. Furthermore, the SNDS-YOLO model achieved an FPS of 230.20, exceeding the industrial production line standard of 30 to 100 FPS, thereby fulfilling the requirement for real-time detection of surface defects on steel strips. Therefore, overall, the SNDS-YOLO model proposed in this paper outperforms other models in terms of overall performance.

Tab. 2. Ablation experiment results

Model	YOLOv8n	SlimNeck	DySample	Inner MPDIoU	Params /M	GFLOPs	Weight size/MB	P/%	mAP @0.5/%	FPS
M1	√	-	-	-	3.01	8.10	6.30	77.10	77.40	318.70
M2	√	√	-	-	2.80	7.30	5.90	78.50	78.10	245.70
M3	√	-	√	-	3.02	8.10	6.30	77.00	75.40	276.70
M4	√	√	√	-	2.81	7.30	5.90	78.60	78.00	216.30
M5	√	√	√	√	2.81	7.30	5.90	80.10	78.60	230.20

3.3. Loss function analysis experiment

Since the selection of the ratio value in the InnerMPDIoU loss function may affect the improvement effect of the model on defect detection accuracy, conducting comparative experiments on different ratio values can more scientifically screen out the optimal parameter configuration. For this reason, this paper further explores the impact of the ratio value on model performance, and the experimental results are shown in Tab. 3. It can be clearly seen from the data in Tab. 3 that when the ratio value is 1.5, the comprehensive performance of the model reaches the optimal. Therefore, the ratio value adopted in this paper is 1.5.

Tab. 3. Ablation experiment with ratio

ratio	P/%	R/%	mAP @0.5/%
0.7	83.9	68.3	78
1.2	82.0	69.7	77.9
1.5	80.1	72.2	78.6
1.7	73.3	73.5	77.9

3.4. Comparative experiments

To further verify the performance advantages of the proposed SNDS-YOLO algorithm, we adopted several key evaluation metrics, including Params, GFLOPs, Weight size, Precision, Recall, and mAP@0.5, and conducted a detailed comparison with mainstream object detection algorithms such as SSD, RTDETR, Faster RCNN, YOLOv3, YOLOv5n, YOLOv6, YOLOv7, YOLOv8n, YOLOv9s, and YOLOv10n on the NEU-DET dataset. All comparative experiments used consistent system configurations, model parameters, and dataset partitions. Table 4 presents the experimental outcomes, from which it can be clearly seen that the SNDS-YOLO algorithm performs excellently in the three metrics of Params, GFLOPs, and Weight size, significantly refining the model's lightweight level and computational efficiency. Specifically, the SNDS-YOLO algorithm has only 2.81M in the Params metric, which is 21.47M lower than SSD, 17.07M lower than RTDETR, 58.74M lower than YOLOv3, and 1.42M, 34.41M, 0.2M, and 4.36M lower than YOLOv6, YOLOv7, YOLOv8n, and YOLOv9s respectively. In the GFLOPs metric, SNDS-YOLO algorithm is 7.30, which is 53.95 lower than SSD, 49.70 lower than RTDETR, 148.00 lower than YOLOv3, and 4.50, 97.90, 0.80, and 19.4 lower than YOLOv6, YOLOv7, YOLOv8n, and YOLOv9s respectively. In the Weight size metric, the SNDS-YOLO algorithm is 5.90MB, which is 87.20MB lower than SSD, 34.60MB lower than RTDETR, 117.60MB lower than YOLOv3, and 2.80MB, 69.00MB, 0.40MB, and 15.3MB lower than YOLOv6, YOLOv7, YOLOv8n, and YOLOv9s respectively. It is worth noting that although the SNDS-YOLO algorithm has a slight increase in the Params, GFLOPs, and Weight size metrics compared with the YOLOv5n and YOLOv10n models, it has significant improvements in P, Recall, and mAP, with increases of 6.9 %, 2.9 %, and 3.7 % respectively compared with the YOLOv5n model, and 6.0 %, 2.5 %, and 4.6 % respectively compared with the YOLOv10n model. Finally, compared with the YOLOv8n baseline model, the SNDS-YOLO algorithm has a 6.64 % reduction in the number of parameters, a 9.88 % decrease in computational load, and a model weight size of only 5.9 MB. At the same time, the P and mAP of the SNDS-YOLO algorithm are increased by 3 % and 1.2 % respectively. This further verifies that the SNDS-YOLO algorithm can provide more accurate detection results while ensuring efficient operation.

In summary, in comparison to other models, the SNDS-YOLO algorithm shows marked improvement in overall effectiveness. Experiments further indicate that the enhanced SNDS-YOLO algorithm effectively boosts the model's detection accuracy and speed while achieving lightweighting, fulfilling the need for efficient yet accurate defect identification in industrial steel production settings.

3.5. Visual comparison of defect detection results

To intuitively demonstrate the improvement effect after optimizing model performance, the study offers comparative defect detection visualization between the YOLOv8n baseline model and the SNDS-YOLO model, with relevant results

shown in Fig. 7. Here, Fig. 7*a* is the original defect image, Fig. 7*b* displays the detection performance of the YOLOv8n baseline model, and Fig. 7*c* shows the detection results of the proposed SNDS-YOLO model. It can be clearly seen from Fig. 7 that the SNDS-YOLO model performs significantly better than the YOLOv8n baseline model in detecting four defect types: crazing, pitted_surface, rolled-in_scale, and scratches. For the remaining two defect types, inclusion and patches, the detection capabilities of the two models are quite similar. This indicates that the proposed SNDS-YOLO model has higher flaw identification accuracy and holds distinct advantages for steel strip surface flaw identification.

Tab. 4. Comparison of experimental results

Model	Params /M	GFLOPs	Weight size /MB	P/%	R/%	mAP @0.5%
SSD	24.28	61.25	93.10	66.58	76.38	75.11
RTDETR	19.88	57.00	40.50	77.20	65.20	73.10
Faster RCNN[26]	-	-	-	75.90	66.60	72.20
YOLOv3	61.55	155.30	123.50	67.10	70.60	72.10
YOLOv5n	2.50	7.10	5.30	73.20	69.30	74.90
YOLOv6	4.23	11.80	8.70	73.30	71.80	76.60
YOLOv7	37.22	105.20	74.90	77.90	66.30	74.50
YOLOv8n	3.01	8.10	6.30	77.10	73.70	77.40
YOLOv9s	7.17	26.70	15.30	79.20	71.80	77.70
YOLOv10n	2.27	6.50	5.80	74.10	69.70	74.30
SNDS-YOLO(ours)	2.81	7.30	5.90	80.10	72.20	78.60

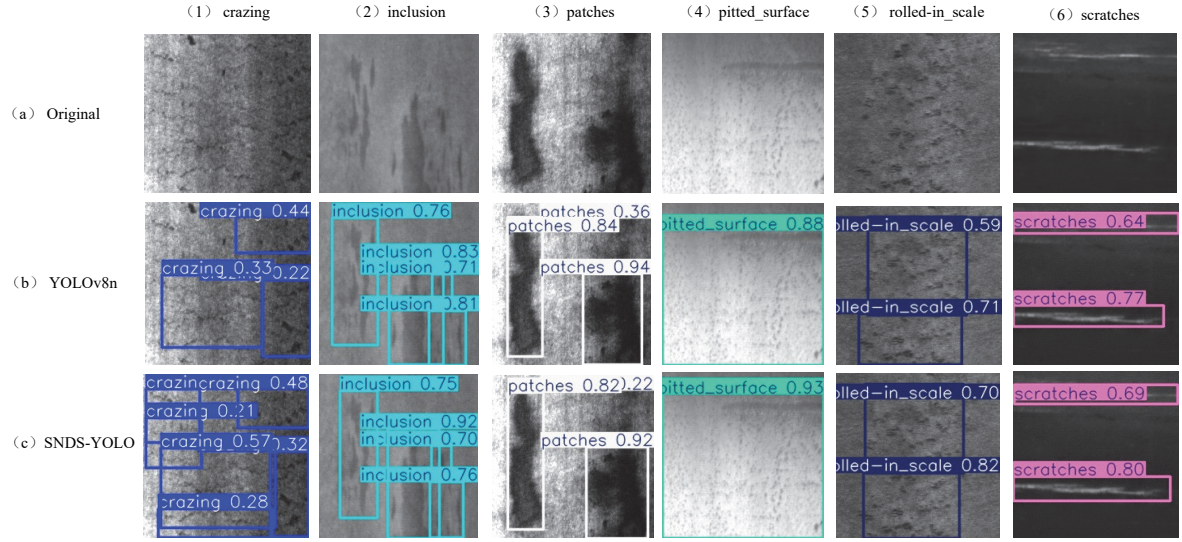


Fig. 7. Visual comparison between YOLOv8n baseline model and SNDS-YOLO model

3.6. Comparative experiments on defect types

To verify the detection performance of the proposed SNDS-YOLO improved model across different defect categories, we conducted a detailed comparison based on the evaluation metrics of precision (P) and mean average precision (mAP) for six common types of strip surface defects. The comparison results are shown in Tab. 5. For the crazing defect, the SNDS-YOLO model achieved a precision of 59.7 %, which represents a significant improvement over the 56.1 % obtained by the YOLOv8n model. Meanwhile, the mAP also increased considerably from 39.9 % to 43.1 %. However, in detecting inclusion and patches defects, the performance of SNDS-YOLO was slightly inferior to that of the YOLOv8n baseline model. In contrast, for pitted_surface defects, SNDS-YOLO exhibited strong performance, with the precision rising from 89.0 % to 92.8 % and the mAP increasing from 87.3 % to 91.3 %. Similarly, for rolled_in_scale defects, the precision improved from 67.8 % to 76.0 %, and the mAP increased from 62.2 % to 64.1 %. Finally, in detecting scratches, the SNDS-YOLO model achieved a precision of 87.6 %, compared with 74.6 % for YOLOv8n, while the mAP improved from 93.3 % to 95.6 %. Overall, SNDS-YOLO demonstrated superior performance in most defect detection tasks, indicating that our improvements upon the YOLOv8n baseline model are effective.

Tab. 5. Comparison of specific types of defects

Model	evaluation metrics	crazing	inclusion	patches	pitted_surface	rolled-in_scale	scratches
YOLOv8n	P%	56.1	85.0	90.1	89.0	67.8	74.6
	mAP@0.5/%	39.9	85.9	95.7	87.3	62.2	93.3
SNDS-YOLO(ours)	P%	**59.7**	80.0	84.7	**92.8**	**76.0**	**87.6**
	mAP@0.5/%	**43.1**	83.8	93.6	**91.3**	**64.1**	**95.6**

4. Discussion

This study has achieved certain advancements in steel strip surface quality inspection. On the NEU-DET dataset, the proposed SNDS-YOLO model exhibits significant advantages over the YOLOv8n baseline model across several key performance indicators. However, it is noteworthy that in terms of Recall rate, the SNDS-YOLO model reaches 72.20 %, which is lower than that of the YOLOv8n baseline model. This result indicates that although the SNDS-YOLO model performs well in lightweighting and accuracy, it still has shortcomings in detecting all actual targets, with a certain rate of missed detections. This could be attributed to the fact that in the pursuit of accuracy, the SNDS-YOLO algorithm is not sufficiently sensitive to the recognition of certain boundary or blurred targets, which leads to the omission of some actual targets. Moreover, this study has not yet explored the robustness of the model under conditions such as noise interference and lighting variations, nor verified its generalization ability on datasets from other production lines. Therefore, future research should focus not only on enhancing the algorithm’s capability to detect surface defects on steel strips in order to further improve the Recall rate, but also on fully considering the robustness of the model in environments with noise interference and its effectiveness on datasets from other production lines.

Conclusions

The SNDS-YOLO algorithm proposed in this paper for strip steel surface defect detection can notably boost the model’s detection precision while retaining the advantage of lightweight, successfully addressing the accuracy-efficiency trade-off in steel strip defect detection. Specifically, the SNDS-YOLO algorithm uses YOLOv8n as the baseline model. First, the SlimNeck lightweight architecture is utilized in the neck network to reconstruct the feature extraction path, which markedly reduces model complexity while ensuring feature expression ability. Second, the innovative introduction of the DySample dynamic upsampling operator in the neck network amplifies the model’s detection sensitivity to tiny defects. Finally, combined with the InnerMPDIoU loss function, the model’s adaptability to multi-scale defects is optimized from the perspective of bounding box regression. Experimental results on the NEU-DET dataset demonstrate that compared with the baseline YOLOv8n model, the SNDS-YOLO algorithm shows excellent performance in multiple dimensions. In terms of model efficiency, the proposed SNDS-YOLO improved algorithm achieves all-round slimming, achieving both a 6.64 % parameter reduction and 9.88 % computation decrease, and the model weight size is compressed to 5.9MB. In terms of detection accuracy, the SNDS-YOLO improved model realizes high-precision detection of strip steel surface defects, with a precision of 80.1% and a mAP of 78.6 %, surpassing the YOLOv8n baseline by 3 % and 1.2 % respectively. Finally, the SNDS-YOLO algorithm proposed in this paper provides a solution with both lightweight and high precision for strip steel surface defect detection in complex industrial environments, and it holds considerable importance in boosting the intelligent progress of detecting surface defects on strip steel.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 62162026), the Natural Science Foundation of Jiangxi Province (No. 20242BAB26019).

References

[1] Yang W, Yang J, Xu CY. Strip surface defect detection based on improved YOLOv5. Acta Metrologica Sinica 2024; 45(11): 1671-1680. DOI: 10.3969/1000-1158.2024.11.11.
[2] Dai LH, Li YS, Shi R. YOLOv8-SSDW: Strip surface defect detection algorithm based on YOLOv8. Journal of Chongqing Technology and Business University 2025; 42(4): 44-52. DOI: 10.16055/1672-058X.2025.0004.006.
[3] Li WG, Ye X, Zhao YT, Wang WB. Strip steel surface defect detection based on improved YOLOv3 algorithm. Acta Electronica Sinica 2020; 48(7): 1284-1292. DOI: 10.3969/0372-2112.2020.07.006.
[4] Zhou S, Zeng Y, Li S. Surface defect detection of rolled steel based on lightweight model. Applied Sciences 2022; 12(17): 8905. DOI: 10.3390/app12178905.
[5] Yu JB, Cheng X, Li QF. Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion. IEEE Transactions on Instrumentation and Measurement 2022; 71: 5000710.
[6] Zou W, Ji C. Real-time detection method of surface defects of hot-rolled strip via improved YOLOv4-tiny model. Mechanical Science and Technology for Aerospace Engineering 2023; 42(6): 883-889. DOI: 10.13433/1003-8728.20230034.
[7] Cao YQ, Wu ML, Xu L. Steel surface defect detection based on improved YOLOv5 algorithm. Journal of Graphics 2023; 44(2): 335-345.
[8] Deng H, Zeng L. Surface defect detection of hot-rolled strip steel based on improved Faster R-CNN. Control Engineering of China 2024; 31(4): 752-759. DOI: 10.14107/20220095.
[9] Yang LS, Li MJ, Hu JW, Wang DX. Strip steel surface defect detection algorithm based on improved YOLOv7-tiny. Computer Engineering 2025; 51(1): 208-215. DOI: 10.19678/1000-3428.0068397.
[10] Reis D, Kupec J, Hong J, et al. Real-time flying object detection with YOLOv8. arXiv preprint arXiv:2305.09972; 2023. DOI: 10.48550/arXiv.2305.09972.
[11] Shan F, Li H, Sun H. Pavement distress identification method based on improved SimAM-YOLOv8. Journal of Jilin University 2024: 1-14.
[12] Xu D, Wang S, Wang Z. Improved YOLOv8 urban vehicle target detection algorithm. Computer Engineering and Applications 2024; 60(18): 136-146.

[13] Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2017: 4700-4708. DOI: 10.1109/CVPR.2017.243.
[14] Lee Y, Hwang J, Lee S, Bae Y, Park J. An energy- and GPU-computation-efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE; 2019: 752-760. DOI: 10.1109/CVPRW.2019.00103.
[15] Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: A new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE; 2020: 390-391. DOI: 10.1109/CVPRW50498.2020.00203.
[16] Liu W, Lu H, Fu H. Learning to upsample by learning to sample. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. IEEE; 2023: 6027-6037. DOI: 10.1109/ICCV51070.2023.00557.
[17] Rezatofighi H, Tsoi N, Gwak JY, Sadeghian A, Reid I, Savarese S. Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE; 2019: 658-666. DOI: 10.1109/CVPR.2019.00075.
[18] Zheng ZH, Wang P, Liu W, Li J, Ye RG, Ren DW. Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press; 2020; 34(7): 12993-13000. DOI: 10.1609/aaai.v34i07.6999.
[19] Zheng ZH, Wang P, Ren DW, Liu W, Ye RG, Hu QH. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE Transactions on Cybernetics 2022; 52(8): 8574-8586. DOI: 10.1109/TCYB.2021.3095305.
[20] Qi W, Chen H, Ye Y, Yu G. Indoor object recognition based on YOLOv5 with EIoU loss function. In: Proceedings of SPIE 12799, Fourth International Conference on Machine Learning and Computer Application. SPIE; 2023: 1279907. DOI: 10.1117/12.3005836.
[21] Ma SL, Xu Y. MPDIoU: A loss for efficient and accurate bounding box regression. arXiv preprint arXiv:2307.07662; 2023. DOI: 10.48550/arXiv.2307.07662.
[22] Hao Z. Inner-IoU: More effective intersection over union loss with auxiliary bounding box. arXiv preprint arXiv:2311.02877; 2023. DOI: 10.48550/arXiv.2311.02877.
[23] Chu Y, Yu X, Rong X. A lightweight strip steel surface defect detection network based on improved YOLOv8. Sensors 2024; 24(19): 6495. DOI: 10.3390/s24196495.
[24] Zhou X, Zhu J, Wang Y. PCB bare board defect detection based on improved YOLOv7 algorithm. Radio Engineering 2023; 53(12): 2791-2797.
[25] Liu SX, Zhang L. DFW-YOLO: A small insulator target defect detection algorithm based on improved YOLOv8s. Computer Optics 2025; 49(5): 835-843. DOI: 10.18287/2412-6179-CO-1600.
[26] Zhao Z, Zhao S. KAN-YOLOv8: A method for surface defect detection of strip steel. Control Engineering of China 2025: 1-8.

About authors

Jianan Wang (b. 1990) is currently pursuing a Ph.D. degree at East China Jiaotong University. He received the Master degree in Computer Software and Theory from East China Jiaotong University in 2018. Since 2018, he has been a lecturer with school of software engineering, Anyang Normal University. His research interests include deep learning, object detection and railway wireless communications. E-mail: *wangjianan@aynu.edu.cn*

Xin Xie (b. 1969) is currently a Professor in the School of Information and Software Engineering, East China Jiaotong University. He received his Master of degree in Control Theory and Control Engineering from Nanchang University in 2001. His research interests are computer networks and machine vision. E-mail: *xienew@gmail.com*

Liang Wu (b. 1982) is currently an associate professor in the School of Software Engineering, Anyang Normal University. He graduated from Xidian University in 2009, majoring in Computer Science and Technology. Research interests are Image processing and Information Safety. E-mail: *02031@aynu.edu.cn*

Wenya Bai (b. 2005) is currently an undergraduate student at the School of Software Engineering, Anyang Normal University. Her research interests include deep learning and object detection. E-mail: *wenyabai@163.com*

Wenya Guo (b. 2004) is currently an undergraduate student at the School of Software Engineering, Anyang Normal University. Her research interests include deep learning and object detection. E-mail: *guo_wenya@163.com*

Received August 18, 2025. *The final version – December 16, 2025.*