Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 17, No. 2, 2026

RCL-YOLO: A Reparameterized Network for Strip Steel Surface Defect Detection

Jiacheng Jiang, Yuebin Su

College of Mathematics and Statistics, Sichuan University of Science and Engineering, Zigong, Sichuan, 643000, China

Abstract: Strip steel surface defect detection is essential for intelligent quality control in modern steel manufacturing. Defects such as crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches usually have weak texture, low contrast, irregular morphology, and large scale variation, which makes accurate real-time detection difficult. This paper proposes RCL-YOLO,a lightweight strip steel surface defect detector based on YOLOv8n, to deal with these issues. A RepStem-based downsampling module is introduced to strengthen shallow local feature extraction and blueuce the loss of fine defect details during spatial resolution blueuction. C2f-RMBC (Reparameterized Mobile Bottleneck Convolution C2f) is then built to improve local structural modeling for weak and irregular defect patterns while keeping an efficient reparameterized inference structure. In addition, a Lighten Cross-Attention Neck is used to improve the interaction between high-resolution texture features and low-resolution semantic features, which helps multi-scale defect representation and blueuces background interference. Experiments are conducted on the NEU-DET surface defect dataset. The results show that RCL-YOLO achieves an mAP@0.5 of 78.20%, surpassing the YOLOv8n baseline by 2.54 percentage points. RCL-YOLO uses only 2.84M parameters and 7.6 GFLOPs, and its end-to-end inference speed reaches 129.05 FPS. Taken together, these results indicate that the proposed method maintains a good balance between detection accuracy,a lightweight structure, and real-time performance, which makes it suitable for industrial strip steel surface defect inspection.

Keywords: Strip Steel Surface Defect Detection; YOLOv8n; RepStem; C2f-RMBC; Lighten Cross-Attention.

1. Introduction

Surface defect inspection is an essential part of strip steel manufacturing, because defects produced during rolling, cooling, transportation, and storage can directly affect product quality, mechanical performance, and the reliability of later processing [1,2]. Traditional manual inspection is often subjective, requires substantial labor, and is hard to sustain under high-speed industrial production conditions. Deep learning has changed this. Automatic visual inspection is now a main technical approach in intelligent steel manufacturing and industrial quality control [3,4].

These observations show that the difficulty of strip-steel inspection is not limited to detecting objects at different scales. Useful evidence may appear as a faint boundary, a slight change in gray-level distribution, or a broken texture merged into the rolling background. Because of this,the detector needs to keep weak signals when the feature map is first blueuced,separate defect cues from repeated metallic patterns,and use both local detail and broader context without drowning out small regions. These points lead directly to the changes made here to the stem, backbone,and neck of the baseline network.

Deep object detectors for industrial inspection usually follow one of two routes: proposal-based detection or direct pblueiction. R-CNN [5],Fast R-CNN [6],Faster R-CNN [7],and Cascade R-CNN [8] first refine candidate regions and then perform classification and localization. This can improve box accuracy,but it also adds several processing steps. SSD [9],EfficientDet [10],and the YOLO family [11-14] predict classes and boxes in a single forward pass,so they fit better with production lines where detection time is limited. For this reason,YOLOv8n was chosen as the baseline. It has a compact backbone,C2f feature-reuse blocks,an FPN-PAN neck,and an anchor-free detection head [14-16]. Its deployment efficiency is a clear advantage. Still,its standard downsampling and feature fusion are not designed for low-contrast and irregular strip-steel defects.

This paper proposes RCL-YOLO, a reparameterized convolution and Lighten Cross-Attention enhanced detector for strip steel surface defect detection. The method modifies YOLOv8n at three stages: early downsampling,backbone feature extraction,and neck feature fusion. First,a RepStem-based downsampling module is introduced to improve shallow local representation and blueuce the loss of fine defect information during spatial resolution blueuction. Second,a RMBC-enhanced C2f module is built to strengthen the backbone's ability to model local structure while keeping an efficient reparameterized inference form. Third,a Lighten Cross-Attention Neck is added to improve cross-scale interaction between high-resolution detail features and low-resolution semantic features,which helps multi-scale defect representation and blueuces background interference.The main contributions of this paper are summarized as follows.

RCL-YOLO changes three parts of YOLOv8n rather than enlarging the detector uniformly. RepStem replaces the first downsampling operation and uses a multi-branch training structure to preserve defect-sensitive responses before spatial resolution is substantially reduced. C2f-RMBC substitutes the conventional bottleneck inside C2f with a reparameterizable mobile block, retaining the feature-reuse pattern of C2f while enriching local transformations during optimization. In the neck, Lighten Cross-Attention introduces semantic-to-detail guidance before conventional feature concatenation, so that high-resolution responses are fused according to information from deeper features. On NEU-DET, the complete model reaches 78.20% mAP@0.5 with 2.84M parameters and 7.6 GFLOPs. Its measured end-to-end speed is 129.05 FPS, indicating that the accuracy gain over YOLOv8n is obtained without losing real-time capability under the reported test

conditions.

2. Related Work

2.1. Strip Steel Surface Defect Detection

Strip-steel surface defects show clear differences in their shapes and visual characteristics. Crazing and scratches usually appear as narrow, small, or discontinuous structures; in contrast, patches, pitted surface, and rolled-in scale may have gray-level characteristics that are similar to the normal steel background. Inclusion defects also show differences in their sizes and local contrast. Because of these properties, the detection process depends on both the preservation of fine texture information and the discrimination of different defect categories [4]. The NEU-DET dataset contains 1,800 grayscale images and covers six common defect categories; it is often used to compare different detection models under the same small-image setting [17]. The number of images in each category is relatively balanced, which is useful for analyzing the detection results of different classes; however, the image resolution is limited, so the loss of shallow features during the early stage may have a greater influence on the final detection results.

Previous studies have improved strip-steel surface inspection through several methods, such as using stronger backbone networks, improving multi-scale feature aggregation, adding attention-based feature refinement, and designing lightweight models. Proposal-based detectors can usually provide relatively detailed localization results, but these detectors need additional region generation and refinement processes, which also increase the computational burden. YOLO-based methods are more commonly used for online industrial inspection because the model can complete category prediction and bounding-box regression in one forward pass [14]. However, many existing architectural improvements mainly modify only one part of the detector; weak defect information may be reduced during initial downsampling, local feature transformation, and cross-scale feature fusion. Therefore, this study uses YOLOv8n as the baseline model and introduces modifications at these three positions; the purpose is to improve the feature representation quality while keeping the overall model compact enough for real-time detection.

2.2. Structural Reparameterization

Structural reparameterization uses different computational structures during the training and deployment stages. During training, parallel convolution branches or identity branches provide more feature transformation paths; after the training process is completed, the compatible linear branches are fused algebraically into an equivalent convolution. Therefore, the additional branches used in training do not remain in the final deployed model [18, 19]. This method can be applied to both the shallow and intermediate feature extraction stages of strip-steel surface defect detection. In the stem part, multiple training branches allow weak edges and texture information to pass through more than one local transformation before downsampling; this process may reduce the loss of some weak defect features. In the backbone network, reparameterized blocks can improve the modeling of local defect structures, while the model still keeps a relatively compact form during inference.

However, the actual inference speed is not only determined by the number of parameters or GFLOPs; memory movement, tensor reshaping, and operator implementation may also affect practical throughput [20, 21]. Based on this consideration, RCL-YOLO applies structural reparameterization to RepStem and C2f-RMBC. These two modules are located in different feature extraction stages, and their purpose is to improve local feature learning without directly assuming that lower theoretical computation will always result in higher measured FPS.

2.3. Attention-Based Feature Fusion

Attention mechanisms assign different levels of importance to different feature responses, and they are commonly used to reduce irrelevant feature activations caused by background information [22, 23]. In strip-steel images, this selective weighting is useful because a defect may occupy only a small part of the feature map, while the surrounding metallic texture can generate responses in a much larger area. As a result, the background features may interfere with the identification of weak defect regions.

YOLOv8n transfers feature information through the top-down and bottom-up paths of the FPN-PAN neck; however, simple concatenation does not clearly describe how deep semantic information should influence shallow detail features. Cross-attention provides a more direct feature interaction method, in which one feature level generates queries, while another feature level supplies keys and values [24]. Based on this idea, the Lighten Cross-Attention Neck uses low-resolution semantic information to refine high-resolution detail features before the two feature streams are fused; in this way, selective cross-scale guidance is introduced, while the overall neck topology is not completely replaced.

3. Proposed Method

The overall architecture of the RCL-YOLO is shown in Fig. 1. The model follows the basic detection pipeline of YOLOv8n but introduces three key modifications. First, the original early downsampling operation is replaced by a RepStem-based downsampling module. Second, the C2f module in the backbone is enhanced by RMBC to form a C2f-RMBC block. Third, the neck is improved with Lighten Cross-Attention for adaptive multi-scale fusion.

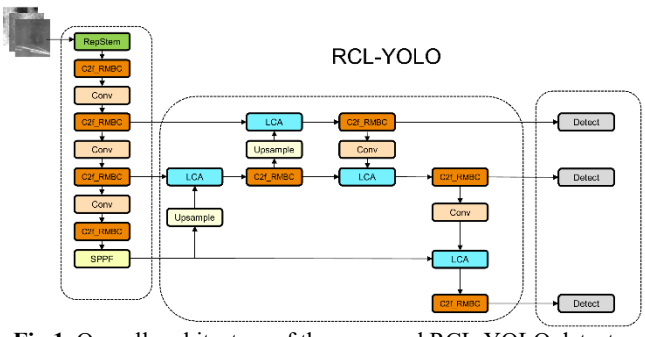


Fig 1. Overall architecture of the proposed RCL-YOLO detector

3.1. RepStem Downsampling Module

In YOLOv8n, early downsampling mainly relies on stride convolution. This operation is efficient,but when it blueuces spatial resolution, it can weaken fine edge responses and faint texture cues. The issue is clearer in strip steel surface defect detection, where many defects show low contrast,irregular boundaries,and subtle local patterns. To reduce the loss of defect-related information,t his paper introduces a RepStem-based downsampling module in the shallow feature extraction

stage[20].

As shown in Fig. 2, the proposed RepStem module enhances the representation ability of the downsampling process during training while maintaining a compact structure during inference. Specifically, the training-time structure introduces a reparameterizable convolutional formulation to strengthen local feature learning. For an input feature X, the RepStem operation can be expressed as:

$$Y = PWConv(\sigma(DWConvs{=}2(BN(X)) + Ps(X))) \quad (1)$$

where DWConvs=2 denotes stride-2 depth-wise convolution, BN denotes batch normalization, Ps(X) denotes the stride-2 projection branch, σ represents the nonlinear activation function, and PWConv denotes point-wise convolution. During inference, the linear operations and projection branch can be equivalently fused into a compact convolutional form:

$$Y = PWConv(\sigma(DWConvrep,s{=}2(X))) \quad (2)$$

where DWConvrep, s=2 denotes the equivalent reparameterized stride-2 depth-wise convolution.

Through this design, the proposed RepStem module improves the representation capacity of early downsampling without introducing additional inference complexity. As a result, more fine-grained defect edges and weak texture patterns can be preserved in the shallow feature maps, which provides a stronger basis for subsequent multi-scale defect representation.

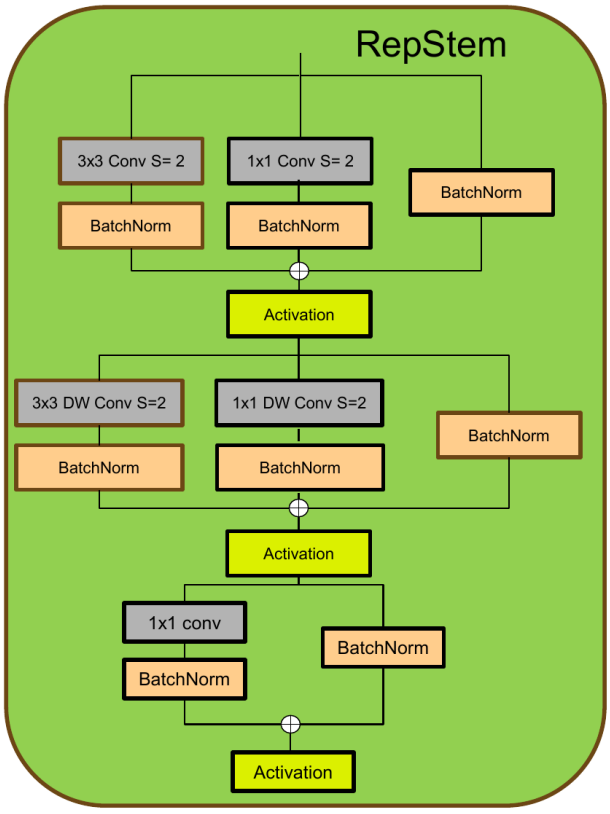


Fig 2. Structure of the proposed RepStem.

3.2. C2f-RMBC Module

The C2f module is an important feature extraction component in YOLOv8n. It improves feature reuse and gradient propagation through feature splitting, bottleneck transformation, concatenation, and fusion. However, the standard bottleneck structure mainly relies on conventional convolutional transformation, which may be insufficient for capturing weak, irregular, and visually similar defect patterns. To enhance local structural modeling while preserving inference efficiency, this paper constructs a RMBC-enhanced C2f module[23].

As shown in Fig. 3, the core idea of RMBC is to use a richer training-time transformation to improve feature learning and convert it into an equivalent compact convolution during inference. Given an input feature X, the training-time RMBC operation is simplified as:

$$C(X) = Frep(X) + X \quad (3)$$

where Frep(·) denotes the reparameterizable convolutional transformation, and the residual term preserves the original feature information. During deployment, the convolutional branches and identity mapping are fused into an equivalent convolution:

$$C(X) = X * Krep + brep \quad (4)$$

where Krep and brep denote the reparameterized convolution kernel and bias, respectively.

Based on this operation, the proposed C2f-RMBC module replaces the standard bottleneck in C2f with RMBC blocks. The input feature is first divided into two branches. One branch is directly propagated to retain shallow information, while the other branch is processed by stacked RMBC blocks to extract stronger local structural features. The output of the module is formulated as:

$$Y = Convout(Concat(X1, H0, H1, ..., Hn)) \quad (5)$$

where X1 denotes the directly propagated branch, H0 is the initial transformed feature, Hi denotes the output of the i-th RMBC block, and Convout denotes the final fusion convolution.

Compared with the original C2f module, the RMBC-enhanced C2f module improves the local representation ability of the backbone. It is more suitable for modeling small-scale defects, weak textures, and irregular surface patterns, while maintaining a compact inference structure after reparameterization.

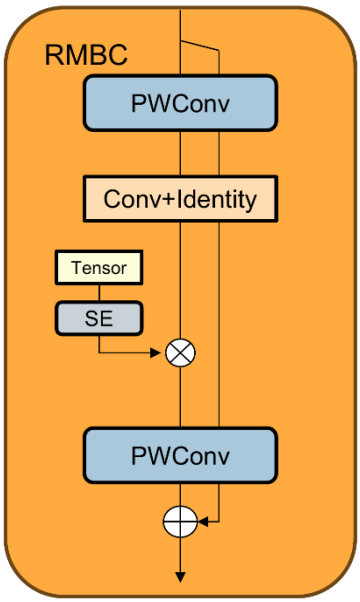


Fig 3. Structure of the proposed RMBC.

3.3. Lighten Cross-Attention Neck

The neck structure of YOLOv8n aggregates multi-scale

features through top-down and bottom-up paths. However, conventional feature fusion mainly depends on feature concatenation and convolution; this method cannot clearly model the complementary relationship between high-resolution detail features and low-resolution semantic features. In strip steel surface defect detection, shallow features usually contain important edge and texture information, while deep features provide stronger semantic information and category discrimination. When these two types of features are directly concatenated, the model may not fully focus on the real defect regions, and background texture responses may still interfere with the detection results.

To solve this problem, this paper proposes a Lighten Cross-Attention Neck to improve the interaction between features at different scales [24]. Let (F_d) represent the high-resolution detail feature, and let (F_s) represent the low-resolution semantic feature after channel alignment and spatial resizing; the cross-attention operation can be expressed as:

$$\hat{F}d = WO(Vs \cdot Softmax(QdKTs / \alpha H) + Fd) \quad (6)$$

where Qd = WQ(Fd), Ks = WK(Fs), and Vs = WV(Fs). WQ, WK, WV, and WO denote lightweight feature embedding convolutions, αH is the scaling factor, and Softmax(·) is used to generate the cross-scale attention weights.

In this formulation, the detail feature provides the query, while the semantic feature provides the key and value. Therefore, high-level semantic information can guide shallow detail features to focus on more discriminative defect regions. After cross-attention enhancement, the refined detail feature and the corresponding semantic feature are concatenated and fused:

$$Fout = Conv(Concat(\hat{F}d, Fs)) \quad (7)$$

where Conv denotes the final fusion convolution.

By introducing explicit cross-scale interaction, the proposed LCA-Neck improves the fusion quality of detail and semantic information. It enables the network to better distinguish weak defects from complex background textures, especially for low-contrast, small-scale, and irregular strip steel surface defects.

4. Experiments and Analysis

4.1. Dataset and Experimental Settings

The experiments were conducted on the NEU-DET dataset, which is commonly used for evaluating strip steel surface defect detection methods. The dataset contains 1,800 grayscale images, and the resolution of each image is 200 × 200 pixels. It includes six representative surface defect categories appearing on hot-rolled strip steel, namely crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc); each defect category contains 300 images. Following the experimental setting commonly used on this dataset, all images were randomly divided into training, validation, and test sets at a ratio of 8:1:1.

Before the images were input into the network, their sizes were adjusted to 640 × 640 pixels. Although all original images in NEU-DET have the same image size, the letterbox preprocessing method provided by YOLOv8 was still used to keep the preprocessing process consistent. In this process, the image is first resized while its original aspect ratio is maintained, and symmetric padding is then added to obtain the required input size; the corresponding bounding-box annotations are transformed using the same scaling ratio and offset values, which avoids additional geometric deformation of the target regions.

All experiments were carried out on a workstation running Ubuntu 22.04.2 LTS, and the workstation was equipped with an NVIDIA GeForce RTX 4090 GPU with 24 GB of video memory. The models were implemented using PyTorch 2.3.0 and CUDA 12.1, while YOLOv8n from the Ultralytics framework was selected as the baseline detector. Stochastic gradient descent was used to optimize the model; the initial learning rate was set to 0.01, the momentum was set to 0.937, and the weight decay was set to 5×10^{-4}. Each model was trained for 300 epochs, with a batch size of 64 and eight workers used for data loading.

The performance of the models was evaluated using precision (P), recall (R), average precision (AP), and mean average precision at an intersection-over-union threshold of 0.5 (mAP@0.5). The model complexity and computational efficiency were also evaluated by using the number of parameters, GFLOPs, and frames per second (FPS); unless otherwise stated, the reported FPS refers to the end-to-end inference speed, so that the efficiency of different detection methods can be compared under relatively consistent conditions.

4.2. Comparative Experiments

To evaluate the performance of the proposed RCL-YOLO framework more fully, two groups of comparative experiments were carried out on the NEU-DET dataset. The first group compares RCL-YOLO with several representative detectors from the YOLO series, including YOLOv8n, YOLOv9t [25], YOLOv10n [26], YOLOv11n [27], and YOLOv12n [28]; these experiments are mainly used to compare the balance between detection accuracy and computational efficiency within the YOLO family.

The second group compares RCL-YOLO with several classical and relatively recent surface defect detection methods, including SGS-YOLO [29], SDD-YOLO [30], Faster R-CNN, YOLO-SDS [31], and RT-DETR [32]. Through these comparisons, the detection accuracy, model complexity, and inference performance of RCL-YOLO can be further examined, and its competitiveness in industrial surface defect inspection can also be evaluated.

4.2.1. Comparison with YOLO-Series Detectors

Table 1 reports the comparison results among different YOLO-series detectors. Compared with the YOLOv8n baseline, RCL-YOLO improves the mAP@0.5 from 75.66% to 78.20%, yielding an absolute gain of 2.54 percentage points. Meanwhile, RCL-YOLO achieves 71.57% precision and 71.53% recall with 2.84M parameters and 7.6 GFLOPs. In terms of inference efficiency, RCL-YOLO reaches 129.05 FPS in end-to-end inference, which still satisfies the real-time requirement of industrial defect inspection. These results indicate that the proposed RepStem, C2f-RMBC, and LCA-Neck modules can improve defect feature representation while maintaining efficient industrial deployment performance.

Table 1. Comparison of detection results among YOLO-series detectors on the NEU-DET dataset.

Model	Params (M)	GFLOPs	P (%)	R (%)	mAP@0.5 (%)	FPS
YOLOv8n	3.01	8.1	71.10	70.98	75.66	134.23
YOLOv9t	1.97	7.6	68.91	68.45	75.03	153.16
YOLOv10n	2.27	6.5	72.51	66.03	73.52	165.73
YOLOv11n	2.58	6.3	73.84	68.40	75.64	130.49
YOLOv12n	2.56	6.3	65.46	68.48	74.50	146.22
RCL-YOLO	2.84	7.6	71.57	71.53	78.20	129.05

As shown in Table 1, RCL-YOLO achieves the highest mAP@0.5 among the compared YOLO-series detectors. Compared with YOLOv8n, RCL-YOLO improves the mAP@0.5 from 75.66% to 78.20%, while reducing the number of parameters from 3.01M to 2.84M and the computational cost from 8.1 GFLOPs to 7.6 GFLOPs. In terms of inference speed, RCL-YOLO reaches 129.05 FPS. Although its FPS is slightly lower than those of several YOLO-series baselines, it still satisfies the real-time requirement of industrial defect inspection. These results show that RCL-YOLO mainly improves the balance between detection accuracy and model compactness, rather than simply pursuing the highest inference speed.

4.2.2. Comparison with Representative Defect Detection Methods

To further examine the competitiveness of RCL-YOLO, the proposed model is compared with several representative surface defect detection methods on the NEU-DET dataset. As shown in Table 2, the comparison includes the overall mAP@0.5 and the AP results of six defect categories, namely crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scale (Rs), and scratches (Sc). In addition, the number of parameters and GFLOPs are also reported, so that the detection accuracy and computational complexity of different methods can be compared more completely.

Table 2. Performance comparison of different methods on the NEU-DET dataset.

Metrics	SGS-YOLO	SDD-YOLO	Faster R-CNN	YOLO-SDS	RT-DETR	RCL-YOLO
Params (M)	1.5	3.4	111.4	1.97	19.8	2.84
GFLOPs	4.7	6.4	60.5	5.3	57.0	7.6
mAP@0.5 (%)	78.5	76.1	75.1	77.7	62.4	78.2
Cr (%)	58.5	58.1	38.9	47.1	28.6	48.3
In (%)	85.5	88.1	77.9	79.8	77.9	87.0
Pa (%)	94.9	94.9	85.3	95.4	85.8	96.7
Ps (%)	82.4	83.4	89.9	82.6	63.4	82.0
Rs (%)	66.4	61.7	67.5	66.9	51.8	72.2
Sc (%)	83.4	91.3	90.9	94.6	66.8	83.1

As shown in Table 2, RCL-YOLO achieves an overall mAP@0.5 of 78.2%, and this result is competitive with several representative defect detection methods. Compared with SDD-YOLO, Faster R-CNN, YOLO-SDS, and RT-DETR, RCL-YOLO obtains better overall detection accuracy. Although its mAP@0.5 is slightly lower than that of SGS-YOLO, the difference is only 0.3 percentage points. At the same time, RCL-YOLO has only 2.84M parameters and requires 7.6 GFLOPs; therefore, its model size and computational cost are much lower than those of heavier detectors such as Faster R-CNN and RT-DETR.

From the results of different defect categories, RCL-YOLO achieves the highest AP values for patches and rolled-in scale, reaching 96.7% and 72.2%, respectively. These results show that RepStem and C2f-RMBC can improve the representation of local texture information, while LCA-Neck can strengthen the fusion between shallow detail features and deep semantic features. Such improvements are helpful for detecting defects that have relatively clear regional characteristics, as well as surface defects that are easily confused with the steel background.

For inclusion and pitted surface, RCL-YOLO achieves AP values of 87.0% and 82.0%, respectively. The results indicate that the model can maintain relatively stable detection performance for defect categories with different sizes and appearance characteristics. For scratches, the AP reaches 83.1%, which is close to the result of SGS-YOLO, but it is still lower than the results of SDD-YOLO, Faster R-CNN, and YOLO-SDS. One possible reason is that scratches usually appear as narrow, long, and irregular structures; meanwhile, their visual characteristics are easily confused with rolling marks and normal background textures. Therefore, even a small localization deviation may cause an obvious decrease in AP. For crazing, RCL-YOLO obtains an AP of 48.3%, which shows that fragmented crack structures with weak contrast are still difficult for the model to recognize accurately.

Considering the overall experimental results, RCL-YOLO provides a relatively balanced performance in detection accuracy, model size, and inference efficiency. Although its mAP@0.5 is 0.3 percentage points lower than that of SGS-YOLO, it performs better than most of the other compared methods. In addition, the model contains only 2.84M parameters and achieves an end-to-end inference speed of 129.05 FPS; this combination of a compact model structure and relatively high inference speed makes RCL-YOLO suitable for real-time strip steel surface defect inspection, especially for industrial applications that require fast image processing and stable defect detection.

4.3. Ablation Study

Ablation experiments were conducted on the NEU-DET dataset to analyze the contribution of each module in RCL-YOLO, and the original YOLOv8n was used as the baseline model. As shown in Table 3, three modules were evaluated, including RepStem, C2f-RMBC, and LCA-Neck. Besides the baseline model and the complete model, each module was also added to YOLOv8n separately; in this way, the effects of different modules on detection accuracy, model parameters, and inference speed could be analyzed independently.

The baseline YOLOv8n model achieves an mAP@0.5 of 75.66%, with 3.01M parameters and an inference speed of 134 FPS. After the original shallow downsampling structure is replaced by RepStem, the mAP@0.5 increases to 76.74%, which is 1.08 percentage points higher than that of the baseline model. At the same time, the number of parameters slightly decreases to 3.00M, while the inference speed changes from 134 FPS to 130 FPS. These results indicate that RepStem mainly improves detection accuracy, but it does not bring an obvious increase in inference speed. This improvement may come from its better preservation of edge information and weak texture features during shallow feature extraction; these features are important for detecting surface defects with low contrast or unclear boundaries.

When only C2f-RMBC is introduced, the model obtains an mAP@0.5 of 76.83%, which is 1.17 percentage points higher than the baseline result. The number of parameters remains at approximately 3.01M, while the inference speed increases to 156 FPS. Among the three single-module configurations, C2f-RMBC provides the largest improvement in detection accuracy. This result shows that C2f-RMBC can improve the representation of local structural information without clearly increasing the model size; meanwhile, its reparameterized structure also has a positive effect on inference efficiency.

When LCA-Neck is added separately, the mAP@0.5 increases from 75.66% to 75.92%, with an improvement of 0.26 percentage points. However, this configuration contains 3.67M parameters, and its inference speed decreases to 115 FPS. Compared with RepStem and C2f-RMBC, the accuracy improvement brought by LCA-Neck alone is relatively small; at the same time, the additional feature interaction and fusion operations increase the computational cost and model size. A possible reason is that cross-scale feature fusion depends on the quality of the input features. When the shallow and backbone features have not been sufficiently improved, the effect of LCA-Neck may not be fully reflected. Therefore, LCA-Neck may have a more obvious effect when it is combined with improved shallow and backbone feature extraction modules.

After RepStem, C2f-RMBC, and LCA-Neck are integrated into the same model, the complete RCL-YOLO achieves the highest mAP@0.5 of 78.20%. Compared with the YOLOv8n baseline, the detection accuracy increases by 2.54 percentage points; meanwhile, the number of parameters decreases from 3.01M to 2.84M. The inference speed is reduced from 134 FPS to 129 FPS, but it still meets the requirements of real-time strip steel surface defect detection. The complete model performs better than all three single-module models, which indicates that the three modules do not only produce independent improvements. RepStem is mainly used to preserve shallow edge and texture details, C2f-RMBC improves the representation of local defect structures, while LCA-Neck strengthens feature interaction between different scales; therefore, the combination of the three modules provides a more balanced improvement in detection accuracy and model compactness.

Table 3. Ablation study of the proposed modules

RepStem	C2f-RMBC	LCA-Neck	Params/M	mAP@0.5/%	FPS
-	-	-	3.01	75.66	134
√	-	-	3.00	76.74	130
-	√	-	3.01	76.83	156
-	-	√	3.67	75.92	115
√	√	√	2.84	78.20	129

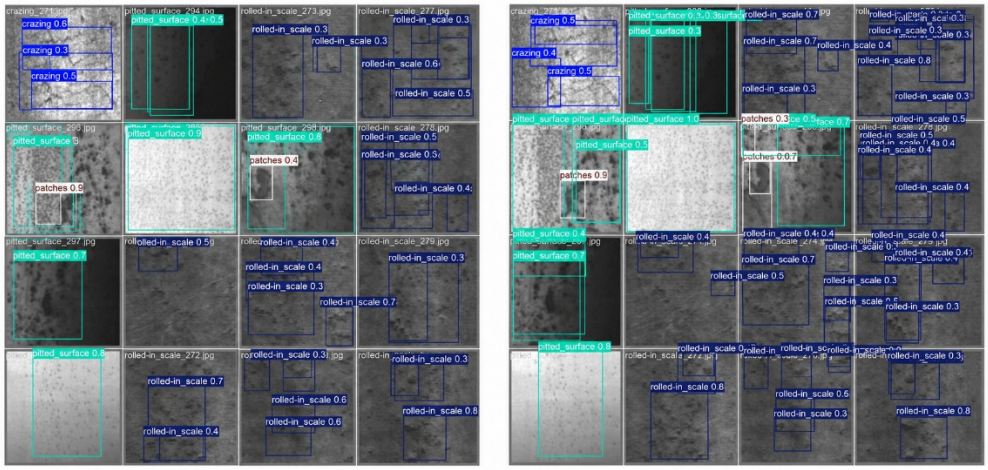
4.4. Visualization Analysis

Figure 4 presents representative detection results produced by YOLOv8n and RCL-YOLO on the NEU-DET dataset. In the displayed examples, RCL-YOLO generally produces bounding boxes that cover the target regions more completely and remain better aligned with the visible defect areas. The difference is particularly noticeable in images containing weak defect boundaries, low-contrast textures, or background patterns resembling the defects. YOLOv8n can identify most visually prominent regions, but some predictions cover only part of the target or are assigned relatively low confidence when the distinction between the defect and the surrounding steel surface is unclear.

For crazing, the crack-like patterns are often discontinuous and distributed over narrow regions. RCL-YOLO follows these fragmented structures more closely and provides more complete localization than the baseline. In the pitted-surface examples, where numerous small depressions are densely distributed and have limited contrast with the background, the proposed model identifies a greater proportion of the visible defective regions and produces more confident predictions. The difference is also evident for rolled-in scale. Because this defect often appears as an irregular texture rather than a clearly separated object, it can be confused with normal variations in the steel surface. RCL-YOLO produces more concentrated feature responses in the relevant regions and generates fewer uncertain predictions. For patch defects, the predicted bounding boxes are also more consistent with the actual spatial range of the targets.

The visual comparison shows that RCL-YOLO can better preserve weak defect information and distinguish irregular defect textures from background interference. This result is generally consistent with the design purposes of the three proposed modules. RepStem is mainly used to reduce the loss of fine details during the early downsampling stage, C2f-RMBC strengthens the representation of local structural information, while LCA-Neck improves the information interaction between features at different scales. Although the qualitative results cannot independently show the contribution of each module, the observed localization results are consistent with the improvements obtained in the

quantitative comparison and ablation experiments. Overall, these results further show that RCL-YOLO is effective for detecting strip steel surface defects with weak contrast and complex texture characteristics.



(a) YOLOv8n (b) RCL-YOLO

Fig 4. Qualitative detection comparison between the proposed RCL-YOLO and YOLOv8n on the NEU-DET dataset

5. Summary

This paper proposes RCL-YOLO, a lightweight improved YOLOv8n detector for strip steel surface defect detection. To address the challenges of weak texture, low contrast, irregular morphology, small defect regions, and complex background interference, three key modules are introduced into the baseline network: RepStem, C2f-RMBC, and LCA-Neck. RepStem enhances shallow feature extraction during early downsampling and preserves defect-sensitive edge and texture cues. C2f-RMBC strengthens local structural representation through reparameterized mobile convolutional blocks, improving the modeling of weak and irregular defect patterns. LCA-Neck introduces Lighten Cross-Attention into the neck to enhance cross-scale interaction between high-resolution detail features and low-resolution semantic features.

Experimental results on the NEU-DET dataset demonstrate that RCL-YOLO achieves 78.20% mAP@0.5, outperforming the YOLOv8n baseline by 2.54 percentage points. Meanwhile, the model requires only 2.84M parameters and 7.6 GFLOPs, and reaches 129.05 FPS in end-to-end inference. These results show that the proposed method achieves a favorable balance between detection accuracy, model complexity, and real-time performance.

Ablation experiments verify the effectiveness and complementarity of the proposed modules. C2f-RMBC improves local defect representation, RepStem provides more reliable shallow features, and LCA-Neck further enhances cross-scale feature fusion when supported by stronger shallow and backbone representations. The full RCL-YOLO model achieves the best overall performance, confirming the synergistic contribution of the three modules. Visualization results further show that RCL-YOLO produces more accurate and stable detection results than YOLOv8n, especially for low-contrast, irregular, and background-confusable defects.

Overall, RCL-YOLO provides an effective lightweight detection framework for real-time strip steel surface defect inspection. Future work will further investigate its generalization ability on more industrial defect datasets and explore its practical deployment in online inspection systems.

Acknowledgments

Research Project on the Reform of Graduate Education and Teaching in Sichuan University of Science and Engineering (No. JG202440).

References

[1] Wang, J., Chen, T., Xu, X., Zhao, L., Yuan, D., Du, Y., Guo, X., & Chen, N. (2025). An improved YOLOv8 model for strip steel surface defect detection. Applied Sciences, 15(1), 52. https: //doi.org/10.3390/app15010052.

[2] Kou, X., Liu, S., Cheng, K., & Qian, Y. (2021). Development of a YOLO-V3-based model for detecting defects on steel strip surface. Measurement, 182, 109454. https://doi. org/ 10. 1016/ j. measurement.2021.109454.

[3] Zheng, X., Zheng, S., Kong, Y., & Chen, J. (2021). Recent advances in surface defect inspection of industrial products using deep learning techniques. The International Journal of Advanced Manufacturing Technology, 113, 35–58. https://doi. org/ 10.1007/s00170-021-06582-2.

[4] Wen, X., Shan, J., He, Y., & Song, K. (2023). Steel surface defect recognition: A survey. Coatings, 13(1), 17. https:// doi. org/ 10.3390/coatings13010017.

[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580–587). https:// doi. org/10.1109/CVPR.2014.81.

[6] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 1440–1448). https://doi.org/10.1109/ICCV.2015.167.

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi. org/10. 1109/ TPAMI. 2016.2577403.

[8] Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6154–6162). https://doi.org/ 10.1109/ CVPR. 2018.00644.

[9] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot MultiBox detector. In Proceedings of the European conference on computer vision (pp. 21–37). https://doi.org/10.1007/978-3-319-46448-0_2.

[10] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781–10790). https://doi.org/10. 1109/CVPR42600. 2020. 01080.

[11] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779–788). https://doi.org/ 10.1109/ CVPR. 2016.90.

[12] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

[13] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7464–7475). https://doi.org/10.1109/CVPR52729.2023.00717.

[14] Jocher, G., Chaurasia, J., Qiu, A., & Stoken, K. (2023). Ultralytics YOLOv8 [Computer software]. GitHub. https:// github. com/ultralytics/ultralytics.

[15] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117–2125). https:// doi. org/ 10.1109/CVPR.2017.374.

[16] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8759–8768). https://doi.org/10. 1109/ CVPR. 2018. 00913.

[17] Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Applied Surface Science, 285, 858–864. https:// doi. org/ 10.1016/j.apsusc.2013.08.132.

[18] Vasu, P. K. A., Gabriel, J., Zhu, J., Tuzel, O., & Ranjan, A. (2023). FastViT: A fast hybrid vision transformer using structural reparameterization. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 5785–5795). https://doi.org/10.1109/ICCV51070.2023.00528.

[19] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style ConvNets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13733–13742). https://doi.org/10. 1109/ CVPR46437.2021.01350.

[20] Wang, Y., Li, Y., Wang, G., & Liu, X. (2024). PlainUSR: Chasing faster ConvNet for efficient super-resolution. arXiv preprint arXiv:2409.13435.

[21] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., et al. (2019). Searching for MobileNetV3. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 1314–1324). https://doi.org/10.1109/ICCV.2019.01404.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[23] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In Proceedings of the European conference on computer vision (pp. 3–19). https:// doi. org/10.1007/978-3-030-01234-2_1.

[24] Yan, Q., Feng, Y., Zhang, C., Pang, G., Shi, K., Wu, P., et al. (2025). HVI: A new color space for low-light image enhancement. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

[25] Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). YOLOv9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616.

[26] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al. (2024). YOLOv10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458.

[27] Jocher, G., Chaurasia, J., Qiu, A., & Stoken, K. (2024). Ultralytics YOLOv11 [Computer software]. GitHub. https:// github. com/ultralytics/ultralytics.

[28] Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOv12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524.

[29] Jiang, J., Su, Y., & Song, J. (2026). A YOLOv8-based network for surface defect detection in strip steel. Journal of Computer and Communications, 14, 106–129. https://doi.org/ 10. 4236/ jcc. 2026.141009.

[30] Wu, Y., Chen, R., Li, Z., Ye, M., & Dai, M. (2024). SDD-YOLO: A lightweight, high-generalization methodology for real-time detection of strip surface defects. Metals, 14(6), 650. https://doi.org/10.3390/met14060650.

[31] Chu, Y., Yu, X., & Rong, X. (2024). A lightweight strip steel surface defect detection network based on improved YOLOv8. Sensors, 24(19), 6495. https://doi.org/10.3390/s24196495.

[32] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 16965–16974). https://doi.org/10.1109/CVPR52729.2024.01617.