PGS-DETR:A Lightweight Detector for Multi-Class Steel Surface Defects in Complex Backgrounds

Yongfeng Qiu[1,2], Shiyao Liu[1], Zhibing Wang[1,*], Bing Tang[1], Kaixi Luo[1], Lanlin Liu[1], Ning Liu[1]
[1]School of Computer Science and Artificial Intelligence, Hunan University of Technology, Zhuzhou 412007, China
[2]Hangzhou Huaxin Mechanical & Electrical Engineering Co., Ltd, Hangzhou 310030, China
Email: wzb@hut.edu.cn

***Abstract*—Steel surface defect detection is critical for online quality control, yet small-scale and low-contrast defects remain challenging under complex textures and reflections. To address the challenges that steel-surface defects often appear at small scales, exhibit low contrast, and are easily confused with the background in complex scenarios, we propose PGS-DETR, a lightweight end-to-end defect detection algorithm. In PGS-DETR, a C2f-PFD module is introduced into the backbone to enhance the extraction of fine-grained characteristics for tiny defects. Moreover, we design a polarity convolution gated attention module(AIFI-PCGA), to suppress disturbances caused by specular reflections and texture noise. Furthermore, we develop a lightweight cross-scale fusion structure(Slim-Neck), which improves cross-channel interaction efficiency and strengthens multi-scale information aggregation. Experimental results in the NEU-DET dataset show that PGS-DETR achieves an mAP50 of 76.5% and an mAP95 of 43.2%, outperforming the baseline by 5.1% and 2.6%, respectively, while reducing the number of parameters and computational cost by 40.3% and 35.2%. In addition, cross-dataset generalization on DeepPCB yields an mAP50 of 97.6%, further demonstrating the effectiveness and deployability of the proposed method in industrial visual inspection.**

***Index Terms*—defect detection, steel surface, frequency-domain convolution, RT-DETR, lightweight feature fusion**

I. Introduction

Steel is a key material in manufacturing and is widely used in machinery, automobiles, and construction, where surface quality directly affects safety and performance. During production, defects such as cracks, scratches, and oxide scales may arise from processing and environmental factors [1]. These defects degrade appearance and mechanical properties and can even cause equipment failures or safety hazards. As a result, steel surface defect detection has become a major topic in industrial inspection. However, traditional methods based on manual visual checks or simple optical techniques are often inefficient and inaccurate, limiting their use in modern high-throughput manufacturing. Therefore, improving detection accuracy and automation is essential for higher production efficiency and reliable product quality.

With the rapid development of computer vision and deep learning, image-based automated inspection has been widely adopted for industrial defect detection. Deep-learning-based detectors are commonly grouped into two-stage and one-stage paradigms. Two-stage methods (e.g., R-CNN [2], Faster R-CNN [3], Mask R-CNN [4]) generate proposals and then classify targets, but numerous proposals reduce efficiency and hinder production-line deployment. One-stage methods (e.g., SSD [5], YOLO [6]) directly predict classes and locations from feature maps, enabling faster inference, yet they still rely on thresholding and non-maximum suppression (NMS), which can limit real-time performance.

To overcome these issues, Transformers [7] have been increasingly introduced into object detection for their global context modeling. Carion et al. [8] proposed DETR, an end-to-end CNN-Transformer framework that removes proposal generation and NMS, but it suffers from high latency and slow convergence. Zhu et al. [9] addressed this with Deformable DETR by using deformable attention with sparse sampling and dynamic offsets to accelerate training and inference. For deployment on compute-constrained edge devices, Baidu further proposed RT-DETR [10], which adopts an efficient hybrid encoder and IoU-aware query selection to support real-time detection.

However, due to steel surface defect detection requirements, RT-DETR struggles to balance accuracy and real-time performance. It is limited in representing and localizing small or low-contrast defects, and less robust under complex backgrounds such as metallic reflections and rolling textures. Additionally, the accuracy-lightweight trade-off restricts industrial applicability. To address these issues, a lightweight model termed PGS-DETR is proposed. It improves weak-texture representation via a frequency-spatial collaborative backbone and enhances robustness through polarity attention and lightweight cross-scale fusion, achieving a better accuracy-efficiency balance for industrial applications. The main contributions are summarized as follows:

- We design a C2f-PFD backbone module that jointly models frequency-domain and spatial features, improving the representation of small-scale foreign objects, low-contrast targets, and occluded defects.
- A new AIFI-PCGA module is designed to boost defect-background contrast via PolaAttention polarity decomposition and uses CGLU gating to select informative channels, enhancing crack edges while suppressing reflection and rolling-texture noise.
- The Slim-Neck module is proposed to mitigate restricted semantic propagation caused by insufficient cross-channel interaction and to improve multi-scale fusion efficiency with lower computational overhead.

*Zhibing Wang is the corresponding author

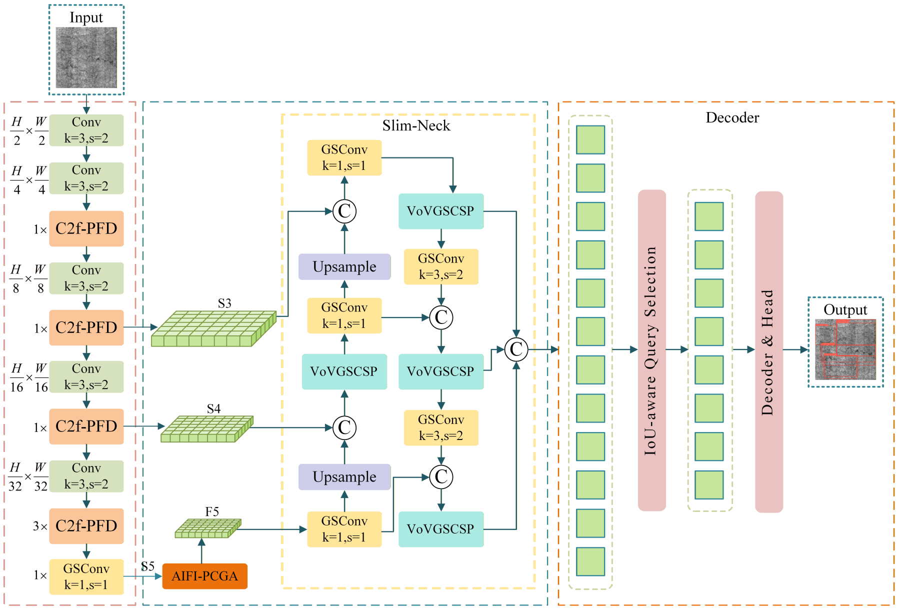


Fig. 1: Overall architecture of PGS-DETR. It consists of a C2f-PFD backbone, an AIFI-PCGA enhancement module, a Slim-Neck for multi-scale feature fusion, and an RT-DETR decoder with IoU-aware query selection.

II. Related Work

A. Industrial Surface Defect Detection Methods

Industrial surface defect detection targets accurate localization and recognition of diverse defects on production lines under complex textures, illumination variations, and noise. Unlike general object detection, defects are often small and irregular with strong background clutter, which frequently leads to missed detections and false alarms. Recent work therefore improves robustness mainly through stronger feature extraction, multi-scale fusion, and better label handling.

Existing methods mainly improve industrial defect detection along three directions. First, they refine training strategies and label assignment to stabilize optimization under class imbalance and noisy samples. For example, Ye et al. [11] improve Faster R-CNN with enhanced feature extraction and fusion and dynamic label assignment, while Chen et al. [12] use decoupled head training to alleviate long-tail effects and strengthen cross-scale compensation. Second, they enhance representation learning by emphasizing defect semantics and boundary cues. Cui et al. [13] propose a dual-branch design to progressively strengthen defect features and edge details. Third, they improve robustness through multi-scale modeling and background-aware aggregation. Zhang et al. [14] introduce CTM-FPN for better small-defect representation, and Xin et al. [15] further improve performance under complex backgrounds. Building on prior designs, we focus on improving small-defect representation and multi-scale fusion while keeping the backbone and neck lightweight for real-time deployment.

B. Lightweight Detection Methods

In industrial online inspection and edge deployment, detectors must run reliably under limited compute and strict latency constraints, making lightweight design essential for practical deployment. Lightweight modeling aims not only to reduce parameters but also to lower FLOPs, memory footprint, and inference latency while maintaining accuracy. Accordingly, existing studies optimize backbones, necks, detection heads, and efficient operators, leading to diverse lightweight solutions.

Recent lightweight detectors reduce cost via efficient designs across the backbone, neck, and head, aiming to maintain accuracy under strict latency and resource constraints in edge deployment. Song et al. [16] and Li et al. [17] propose compact architectures and efficient fusion (e.g., LW-SARDet and C3k2-DyGhost) to cut parameters and GFLOPs. Mao et al. [18] targets lower memory and latency with a lightweight encoder, while He et al. [19] and Ren et al. [20] improve small-object perception and multi-scale fusion through efficient downsampling and neck redesign. In line with these efforts, our method improves mAP50 by 5.1% while reducing parameters and GFLOPs by 40.3% and 35.2%, respectively.

III. Methodology

This section presents the methods and design principles of C2f-PFD, AIFI-PCGA, and Slim-Neck. The overall network architecture of PGS-DETR is illustrated in Fig. 1.

A. C2f-PFD Module Design

ResNet-18 is widely used for its concise design, but it is less effective for complex steel-surface defects since standard convolution ignores frequency differences and residual addition lacks scale adaptivity. To address this, we adopt the YOLOv8 [21] backbone style and introduce C2f-PFD by replacing the C2f Bottleneck with our PFDConvBlock in Fig. 2. PFDConvBlock integrates PConv from FasterNet-Block [22] and FDConv [23] to promote frequency-spatial collaboration and improve representations for small and low-contrast defects.

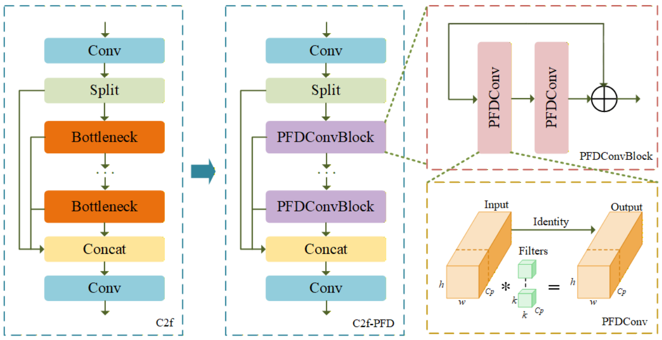


Fig. 2: Architecture comparison of the C2f and the proposed C2f-PFD module, with the detailed structure of PFDConvBlock and PFDConv.

The key innovation of C2f-PFD is the proposed PFDConvBlock, which combines PConv and FDConv from FasterNetBlock to model frequency cues in a lightweight manner. Although FasterNetBlock reduces computation by convolving only part of the channels, it lacks explicit frequency modeling for low-frequency context and high-frequency edges. We therefore introduce FDConv, a frequency-adaptive convolution that improves weight frequency diversity and feature selectivity via FDW, KSM, and FBM. The architecture is shown in Fig. 3.

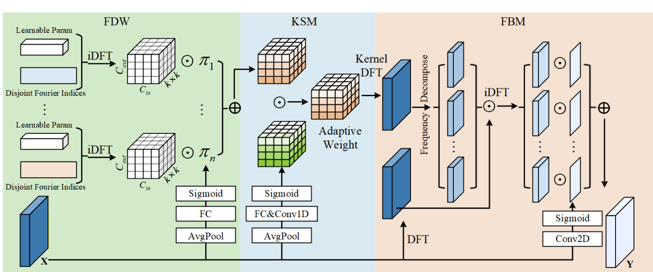


Fig. 3: The structure of FDConv. FDW learns Fourier-domain grouped weights, KSM performs kernel-space modulation, and FBM conducts frequency-band modulation.

FDW parameterizes convolution weights as learnable Fourier-domain coefficients and reshapes them into $P \in \mathbb{R}^{KC_{\text{in}} \times KC_{\text{out}}}$. Based on the ℓ_2 norm of the Fourier indices, P is divided into n disjoint frequency groups $P^{i}{}_{i=1}^{n}$, each mapped by iDFT and reassembled into spatial weights W^i, which are then fused by dynamic convolution to emphasize different bands. To reduce coarse mixing, KSM combines lightweight Conv1D with global average grouping and a fully connected layer, their element-wise product forms α to modulate W through the Hadamard product. FBM applies DFT to W (zero-padded to match X) to obtain $\mathcal{F}(W)$, uses $\mathcal{M}_b$ for band separation and applies the inverse transform to derive W^b. In parallel, convolution in the frequency-domain in X yields Y_b, while convolution with Sigmoid produces A_b. The band-wise features are finally modulated and fused to output Y, as summarized below.

$$W_b = \mathcal{F}^{-1}(\mathcal{M}_b \odot \mathcal{F}(W)) \tag{1}$$

$$\mathcal{M}_b(u,v) = \begin{cases} 1, & \text{if } \psi_b \leq \max(|u|,|v|) < \psi_{b+1} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$Y_b = \mathcal{F}^{-1}((\mathcal{M}_b \odot \mathcal{F}(W)) \odot \mathcal{F}(X)) \tag{3}$$

$$Y = \sum_{b=0}^{B-1} (A_b \odot Y_b) \tag{4}$$

$$Y = \sum_{b=0}^{B-1} (A_b \odot (W_b * X)) \tag{5}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the DFT and inverse DFT, respectively, and $\mathcal{M}_b$ is a binary mask used to isolate a specific frequency range. The thresholds ψ_b and ψ_{b+1} are selected from a predefined frequency set $\{0, \psi_1, \ldots, \psi_{B-1}, \frac{1}{2}\}$.

B. AIFI-PCGA Module Design

Steel surface defects are often tiny and irregular, and are easily confused with backgrounds dominated by specular reflections and rolling textures, resulting in low defect-background contrast. Meanwhile, conventional CNNs with single-scale convolutions mainly capture local cues and struggle with multi-scale, low-contrast defects. To address this, we propose AIFI-PCGA to replace AIFI by integrating PolaAttention [24] and CGLU [25] in Fig. 4. PolaAttention uses polarity decomposition to separate defect foreground from background disturbances and enhance feature-level contrast, while CGLU gates informative channels to emphasize key cues and suppress noise, producing more discriminative features.

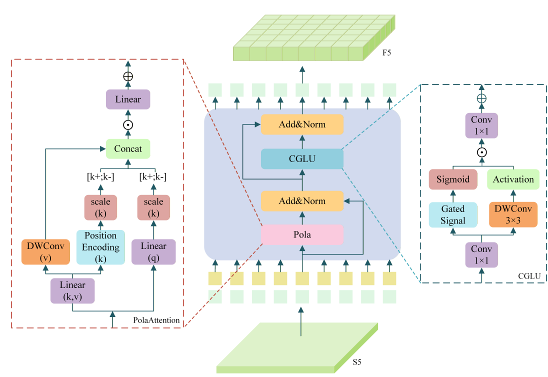


Fig. 4: The structure of AIFI-PCGA. PolaAttention performs polarity-aware feature interaction, and CGLU applies gated channel selection for defect enhancement and noise suppression.

To enhance the discriminability between defects and background, the AIFI-PCGA module first incorporates the core polarity-decomposition capability of the PolaAttention mechanism. Specifically, a linear projection is employed to map the input feature map into a query vector q and a key vector k, followed by the standard polarity decomposition in PolaAttention:

$$q = q^{+} - q^{-}, \quad k = k^{+} - k^{-}, \tag{6}$$

This operation maps the intensity difference between defects and background into polarity feature components, enabling explicit separation of the two. Built on the positive or negative branches, PolaAttention performs feature interaction and low-frequency enhancement to model both defect–background contrast (hetero-sign branch) and background self-correlation (homo-sign branch), thereby strengthening responses to tiny defects and reducing background dominance.

On top of the polarity-enhanced features, CGLU further mitigates defect-background confusion by dynamically reweighting features with a fine-grained gating mechanism, achieving defect enhancement and interference suppression. The computation is formulated as:

$$O_{\mathrm{CGLU}} = (W * O_{\mathrm{Pola}} + B) \odot \mathrm{GELU}\Big(W' * \mathrm{DepthwiseConv}_{3\times 3}(O_{\mathrm{Pola}}) + B' \Big) \tag{7}$$

where O_{Pola} is the polarity feature map output by PolaAttention, and GELU is the Gaussian error linear unit.

C. Slim-Neck Module Design

In industrial real-time detection, DSC is efficient but weak in cross-channel coupling, particularly for neck-level multi-scale fusion of small and low-contrast defects. We therefore adopt GSConv [26] at key fusion stages, by mixing standard and depthwise features with channel shuffle, it brings DSC outputs closer to standard convolution (SC) under strict compute budgets. Given $X \in \mathbb{R}^{H\times W\times C_1}$, GSConv is formulated as:

$$Y_1 = \mathrm{Conv}(X), \quad Y_2 = \mathrm{DWConv}(Y_1), \tag{8}$$

$$Y = \mathrm{Shuffle}(\mathrm{Concat}(Y_1, Y_2)). \tag{9}$$

Applying GSConv to all stages may hinder feature propagation in deeper layers and increase latency. Thus, we use it selectively in computation-intensive local and multi-scale fusion stages. Moreover, we introduce a VoVGSCSP module within the Slim-Neck design to restructure the neck, improving feature reuse and computational efficiency. The detailed architecture is shown in Fig. 5.

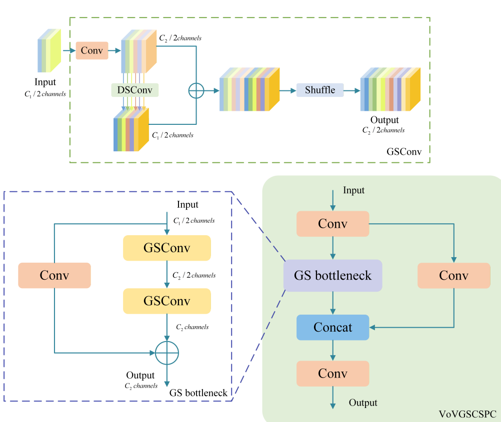


Fig. 5: The structure of GSConv and its derived blocks, including the GS bottleneck and VoVGSCSPC.

VoVGSCSP adopts cross-stage partial connections by splitting the input into two channel-wise branches: one applies channel compression and a GS bottleneck, while the other preserves shortcut information via lightweight mapping, the outputs are then concatenated for efficient fusion. The GS bottleneck stacks GSConv to enhance local features and cross-channel mixing through combined standard and depthwise convolution with channel reordering, mitigating the interaction loss of DSC. Overall, VoVGSCSP reduces parameters and computation with minimal accuracy loss, improving the efficiency-performance balance for steel surface defect detection.

IV. Experiment

A. Dataset and Implementation

Datasets. To evaluate the lightweight design and real-time inference capability of the proposed model while ensuring detection accuracy, and to verify its deployability in practical industrial inspection scenarios, we conduct experiments on the NEU-DET [27] and DeepPCB [28] datasets. The NEU-DET dataset contains 1800 images covering six common defect categories: Crazing, Inclusion, Patches, Scratches, Rolled-in_scale, and Pitted_surface, with 300 images per category. The dataset is randomly split into the training, validation, and test sets with a ratio of 8:1:1, resulting in 1440, 180, and 180 images, respectively. The DeepPCB dataset consists of 1500 images with six typical PCB defect categories: copper, mousebite, open, pin-hole, short, and spur. It is divided into the training, validation, and test sets with a ratio of 6:2:2, corresponding to 900, 300, and 300 images, respectively.

Implementation Details. All experiments are implemented in Python 3.10 with PyTorch 2.1.2 and CUDA 11.8. The hardware platform includes an Intel(R) Xeon(R) Platinum 8358P CPU and an NVIDIA RTX 4090 GPU with 24GB memory. The training settings are as follows: the number of epochs is set to 300, the batch size is 4, the AdamW optimizer is adopted, the initial learning rate $lr0$ is 0.0001, and the input image resolution is fixed to 640×640. These configurations are used for all comparative and ablation experiments to ensure fairness and reproducibility.

TABLE I: Comparison of Different Models on the NEU-DET Dataset

Models	P	R	mAP50	mAP95	GFLOPs	Params(M)
Faster R-CNN [3]	65.7	61.8	68.7	35.9	209.2	41.8
YOLOv8m [21]	78.9	71.2	75.4	**43.6**	78.4	25.9
YOLOv10m [29]	79.7	71.8	72.9	42.3	59.1	15.4
YOLOv11m [30]	80.2	71.5	74.1	41.5	68.0	20.1
RT-DETR-R18 [10]	71.6	69.9	71.4	40.6	58.3	20.1
RT-DETR-R34 [10]	74.8	70.6	73.9	41.9	92.0	31.5
RT-DETR-L [10]	76.2	71.3	74.2	42.6	110.7	32.0
Deformable DETR [9]	79.2	68.8	70.3	38.6	169.2	40.3
DDQ [31]	75.8	70.3	71.1	42.9	241.0	48.7
PGS-DETR	**81.6**	**72.4**	**76.5**	43.2	**37.8**	**12.0**

B. Comparative Experiments

In the comparative experiments, PGS-DETR was evaluated against two-stage detectors (e.g., Faster R-CNN), one-stage detectors (e.g., YOLOv8m, YOLOv10m and YOLOv11m), and Transformer-based methods (e.g., RT-DETR-r18, RT-DETR-r34, RT-DETR-L, Deformable DETR, and DDQ). The results are summarized in Table I.

Among all methods, Faster R-CNN yields lower accuracy with higher cost. YOLOv8m achieves 75.4% mAP50 and 43.6% mAP95. DDQ reaches 42.9% mAP95 but has the highest computational burden. Overall, PGS-DETR offers the best balance, achieving 76.5% mAP50 and 43.2% mAP95 with only 37.8 GFLOPs and 12.0M parameters, making it more suitable for real-time industrial deployment under limited compute budgets. As shown in Fig. 6, PGS-DETR produces tighter boxes and more focused heatmaps, with fewer false positives and misses than RT-DETR-R18, indicating stronger feature representation and more reliable localization in challenging surface textures.

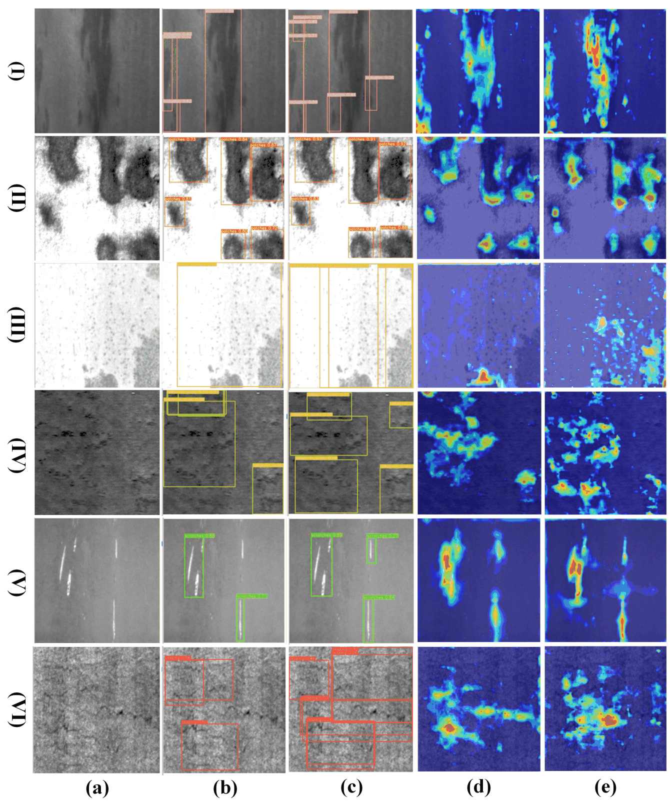


Fig. 6: The visualization and heatmap comparisons of RT-DETR-R18 and PGS-DETR on six defect types. From top to bottom, the rows correspond to In, Pa, PS, RS, Sc, and Cr. From left to right, the columns show the original images, RT-DETR-R18 detections, PGS-DETR detections, RT-DETR-R18 heatmaps, and PGS-DETR heatmaps, respectively.

C. Ablation Experiments

To thoroughly evaluate the impact of each proposed module on overall performance, we conducted ablation studies on the NEU-DET dataset using the original RT-DETR-R18 as the baseline under identical training and hardware settings, so as to quantify the contributions of C2f-PFD, AIFI-PCGA, and Slim-Neck. The results in Table II show that introducing C2f-PFD improves mAP50 by 2.0% while substantially reducing the number of parameters. With AIFI-PCGA, mAP50 and mAP95 further increase by 1.1% and 0.4%, respectively. Finally, when all three modules are jointly applied, the model achieves the best performance: compared with the baseline, mAP50 and mAP95 improve by 5.1% and 2.6%, respectively, while precision and recall also increase by 10.0% and 2.5%. Meanwhile, GFLOPs and parameters are reduced by 35.2% and 40.3%, demonstrating a significant overall improvement in the accuracy-efficiency trade-off.

TABLE II: Ablation Study on the NEU-DET Dataset

C2f-PFD	AIFI-PCGA	Slim-Neck	P	R	mAP50	mAP95	GFLOPs	Params(M)
			71.6	69.9	71.4	40.6	58.3	20.1
✓			80.2	69.8	73.4	42.1	42.6	12.9
	✓		**81.7**	69.3	73.1	42.2	58.9	20.4
		✓	76.6	68.4	72.2	41.1	53.6	19.3
✓	✓		80.1	71.2	74.5	42.5	42.7	12.8
✓		✓	76.8	71.1	73.7	42.7	**37.3**	**11.3**
	✓	✓	80.8	70.2	73.8	42.2	53.7	19.2
✓	✓	✓	81.6	**72.4**	**76.5**	**43.2**	37.8	12.0

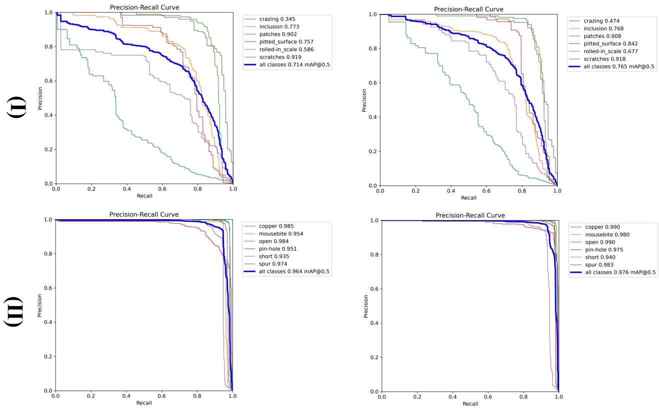
D. Generalization Experiments

To further validate the effectiveness of PGS-DETR, we conduct generalization experiments on the DeepPCB dataset. As shown in Table III, PGS-DETR consistently outperforms RT-DETR-R18, improving Precision and Recall by 1.6% and 2.3%, and increasing mAP50 and mAP95 by 1.5% and 2.7%, respectively. Meanwhile, it reduces computation from 57.0 to 38.2 GFLOPs and parameters from 19.9M to 12.3M, achieving higher accuracy with a lighter model. These results demonstrate the effectiveness and generalization capability of the proposed method across datasets.

TABLE III: Comparison on the DeepPCB Dataset

Dataset	Model	P	R	mAP50	mAP95	GFLOPs	Params(M)
DeepPCB	RT-DETR-R18 [10]	97.4	95.5	96.4	81.9	57.0	19.9
	PGS-DETR	**99.0**	**97.8**	**97.6**	**84.6**	**38.2**	**12.3**

Fig. 7 compares the PR curves of RT-DETR-R18 and PGS-DETR on the NEU-DET and DeepPCB datasets. On NEU-DET, PGS-DETR shows clearer gains on weak-texture and low-contrast defects such as Crazing and Rolled-in_scale. These improvements mainly come from the frequency-spatial collaboration in C2f-PFD for finer feature representation, together with the polarity decomposition and gating in AIFI-PCGA that suppress rolling-texture and specular-reflection interference more effectively in challenging scenes. On DeepPCB, both methods perform strongly, but PGS-DETR is more stable on small defects (e.g., open, short, and pin-hole), largely due to Slim-Neck, which strengthens cross-scale fusion and improves small-object localization robustness.



(a) **(b)**

Fig. 7: Precision-Recall (PR) curves of RT-DETR-R18 and PGS-DETR on two datasets. From top to bottom, the rows correspond to NEU-DET and DeepPCB, respectively. From left to right, the columns show the results of RT-DETR-R18 and PGS-DETR, respectively.

V. Conclusion

This paper proposes a lightweight defect detection model, PGS-DETR, for steel surface inspection under complex industrial conditions. By introducing C2f-PFD into RT-DETR-R18, it enhances fine-grained defect representation. In addition, AIFI-PCGA is employed to strengthen the discrimination between defects and background. A Slim-Neck is further designed to achieve efficient cross-scale feature fusion, so that strong feature expression is maintained with lower computational cost. Experimental results show that PGS-DETR achieves higher mAP50 and mAP95 than RT-DETR-R18 on NEU-DET, while significantly reducing the number of parameters and computational complexity. Ablation studies also confirm that the combination of the three improvements brings the most comprehensive gains. Cross-dataset evaluation on DeepPCB demonstrates consistent accuracy improvements together with further reductions in GFLOPs and parameters, indicating good generalization ability and practical potential for deployment.

References

[1] B. Zhao, Y. Chen, X. Jia, and T. Ma, “Steel surface defect detection algorithm in complex background scenarios,” *Measurement*, vol. 237, Sep. 2024, Art. no. 115189.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6000–6010.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[9] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” *arXiv preprint* arXiv:2010.04159, 2020.

[10] Y. Zhao, W. Lv, S. Xu, *et al.*, “DETRs beat YOLOs on real-time object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 16965–16974.

[11] Q. Ye, Y. Dong, X. Zhang, *et al.*, “Robustness defect detection: Improving the performance of surface defect detection in interference environment,” *Optics and Lasers in Engineering*, vol. 175, Art. no. 108035, 2024.

[12] M. Chen, J. Tian, X. Cao, *et al.*, “DM-YOLO for MLCCs’ automatic defect detection,” *Optics & Laser Technology*, vol. 192, Art. no. 113977, 2025.

[13] W. Cui, K. Song, X. Jia, *et al.*, “An efficient targeted design for real-time defect detection of surface defects,” *Optics and Lasers in Engineering*, vol. 178, Art. no. 108174, 2024.

[14] H. Zhang, J. Liu, D. Pan, *et al.*, “A novel context-adaptive multi-scale defect detection method for object surface defects,” *Neurocomputing*, Art. no. 130927, 2025.

[15] H. Xin and J. Song, “YOLOv5-ACCOF steel surface defect detection algorithm,” *IEEE Access*, 2024.

[16] Y. Song, L. Tang, and G. He, “LW-SARDet: A lightweight SAR ship detector via decomposed convolution,” *IEEE Geoscience and Remote Sensing Letters*, 2024.

[17] F. Li, X. Lv, M. Zhao, *et al.*, “IFD-YOLO: A lightweight infrared sensor-based detector for small UAV targets,” *Sensors*, vol. 25, no. 24, Art. no. 7449, 2025.

[18] Q. Mao, Q. Li, B. Wang, *et al.*, “SpirDet: Towards efficient, accurate and lightweight infrared small target detector,” *IEEE Trans. Geosci. Remote Sens.*, 2024.

[19] Q. He, Z. Li, and W. Yang, “LMFE-RDD: A road damage detector with a lightweight multi-feature extraction network,” *Multimedia Systems*, vol. 30, no. 4, Art. no. 176, 2024.

[20] Q. Ren, S. Dong, W. Qian, *et al.*, “A lightweight pipeline weld radiographs defect detection detector based on inverted bottleneck,” *Journal of Pipeline Science and Engineering*, Art. no. 100322, 2025.

[21] G. Jocher, “YOLOv8,” 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[22] Y. Chai, X. Yao, M. Chen, *et al.*, “FPFS-YOLO: An insulator defect detection model integrating FasterNet and an attention mechanism,” *Sensors*, vol. 25, no. 13, Art. no. 4165, 2025.

[23] L. Chen, L. Gu, L. Li, *et al.*, “Frequency dynamic convolution for dense image prediction,” *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 30178–30188, 2025.

[24] W. Meng, Y. Luo, X. Li, *et al.*, “PolaFormer: Polarity-aware linear attention for vision transformers,” *arXiv preprint* arXiv:2501.15061, 2025.

[25] S. Dai, “TransNeXt: Robust foveal visual perception for vision transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 17773–17783.

[26] H. Li, J. Li, H. Wei, *et al.*, “Slim-neck by GSConv: A lightweight design for real-time detector architectures,” *Journal of Real-Time Image Processing*, vol. 21, no. 3, Art. no. 62, 2024.

[27] Y. He, K. Song, Q. Meng, *et al.*, “An end-to-end steel surface defect detection approach via fusing multiple hierarchical features,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, 2019.

[28] S. Tang, F. He, X. Huang, *et al.*, “Online PCB defect detector on a new PCB defect dataset,” *arXiv preprint* arXiv:1902.06197, 2019. [Online]. Available: https://arxiv.org/abs/1902.06197

[29] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, “YOLOv10: Real-time end-to-end object detection,” *arXiv preprint* arXiv:2405.14458, 2024.

[30] R. Khanam and M. Hussain, “YOLOv11: An overview of the key architectural enhancements,” *arXiv preprint* arXiv:2410.17725, 2024.

[31] S. Zhang, X. Wang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo, and K. Chen, “Dense distinct query for end-to-end object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 7329–7338.