Steel Strip Surface Defect Detection Method Based on Improved YOLOv8

Shihao Yuan, Yuanyuan Li, Zewei Gao

***Abstract*—To address the challenges of high computational cost, suboptimal detection accuracy, and ineffective feature exploitation in surface defect detection of strip steel, we propose RAC-YOLO, a lightweight detection model adapted from YOLOv8n. We design a lightweight feature extraction module, named RepNCSPELAN4 Context Anchor Attention Network (RepCAANet), to replace the default backbone of YOLOv8n, thereby improving gradient flow and significantly boosting detection performance. The Attention-based Intrascale Feature Interaction (AIFI) module is then integrated to enhance feature discriminability and classification accuracy by suppressing irrelevant contextual information. Additionally, we adopt a lightweight CARAFE (Content-Aware ReAssembly of FEatures) upsampling module to enrich the representation of defect-relevant features. Finally, the Minimum Pairwise Distance Intersection over Union (MPDIoU) loss is introduced, aimed at optimizing bounding box coordinate predictions and increasing localization accuracy. Experimental results show that the proposed RAC-YOLO achieves a mean Average Precision (mAP) of 80.2% on NEU-DET and 70.1% on GC10-DET, representing improvements of 2.7% and 4.3% over the baseline, respectively, as well as a 13.5% decrease in FLOPs. This demonstrates that RAC-YOLO effectively balances high detection accuracy with low computational overhead.**

***Index Terms*—RAC-YOLO, RepCAANet, AIFI, CARAFE, MPDIOU, Defect detection.**

I. Introduction

STEEL materials are indispensable in a wide range of industries, including automotive, defense, machinery, and chemical engineering. However, during the manufacturing process, steel strips are susceptible to various surface defects—such as crazing, inclusions, patches, pitting, rolled-in scale, and scratches—caused by environmental factors, equipment wear, or process variations. These imperfections can significantly degrade the intrinsic properties of steel, compromising its corrosion resistance, wear resistance, fatigue strength, and overall performance [1], thereby shortening service life and increasing the risk of safety-critical failures.

Automated detection of surface defects is critical to ensuring product quality in manufacturing. Traditional methods—manual visual inspection and classical machine vision—often suffer from limited precision, poor repeatability, and low scalability [2]. Over the past decade, deep learning has emerged as a transformative approach, offering fast inference, high accuracy, and strong robustness, thereby accelerating its adoption in surface defect detection [3].

Modern deep learning-driven object detection frameworks can be broadly categorized into two primary paradigms: two-stage and one-stage architectures. Two-stage approaches, such as R-CNN [4], Faster R-CNN [5], and Mask R-CNN [6], achieve superior precision by generating region proposals, but involve high computational costs, making them less suitable for real-time applications. In comparison, one-stage detectors like SSD [7] and YOLO [8] formulate object detection within a single forward pass, enabling fast inference and making them more applicable to industrial deployment. In recent years, vision Transformers such as DETR [9] have introduced a new paradigm in detection by exploiting global contextual understanding, exhibiting competitive results in complex visual recognition tasks [10].

Numerous studies have advanced deep learning for steel surface defect detection. Gou et al. [11] improved the YOLO architecture to enhance robustness against detection variations, whereas Zhao et al. [12] modified the backbone and proposed a dynamic feature response module to increase detection accuracy. Wu et al. [13] developed Hyper-YOLO by incorporating the Ghost module to enrich feature representation capabilities. Lv et al. [14] designed the EDDN network, an SSD-inspired model tailored for detecting defects across multiple scales. Zhang et al. [15] integrated the ELAN-C block and coordinate attention mechanism into YOLOv8 to address defects on aircraft surfaces. Sun et al. [16] presented an enhanced version of YOLOv10, embedding C3_Star_EMA and MobileOneBlock components to mitigate background noise and lower computational demands. Chai et al. [17] proposed DFP-YOLO, utilizing DWR_DRB and FPSConv blocks to improve multi-scale feature learning and capture long-range contextual dependencies. Gao et al. [18] designed CMS-YOLOv8s through the integration of the CBAM attention module to enhance performance in multi-scale defect localization.

The Transformer architecture, originally developed for natural language processing, has been increasingly adopted in computer vision [19]. Its self-attention mechanism enables global context modeling within a single layer, reducing reliance on deep convolutional stacks and preserving spatial information while lowering model depth and parameter count [20]. Several works have explored Transformer-enhanced detection models. For example, Ikchul Eum et al. [21] integrated multi-scale features and a global attention mechanism into YOLOv10 for high-accuracy detection in complex construction environments. VR Patheda et al. [22] combined YOLOv4 with a Vision Transformer (ViT) for breast cancer detection in CESM images, achieving effective

Manuscript received March 12, 2025; revised August 15, 2025.

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2024YFB3312000.

Shihao Yuan is a postgraduate student at the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China (e-mail: m325123252@sues.edu.cn).

Yuanyuan Li is a professor at the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China (corresponding author to provide phone: +86-021-67791121; e-mail: liyuanyuan@sues.edu.cn).

Zewei Gao is a postgraduate student at the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, 201620, China (e-mail: m320123318@sues.edu.cn).

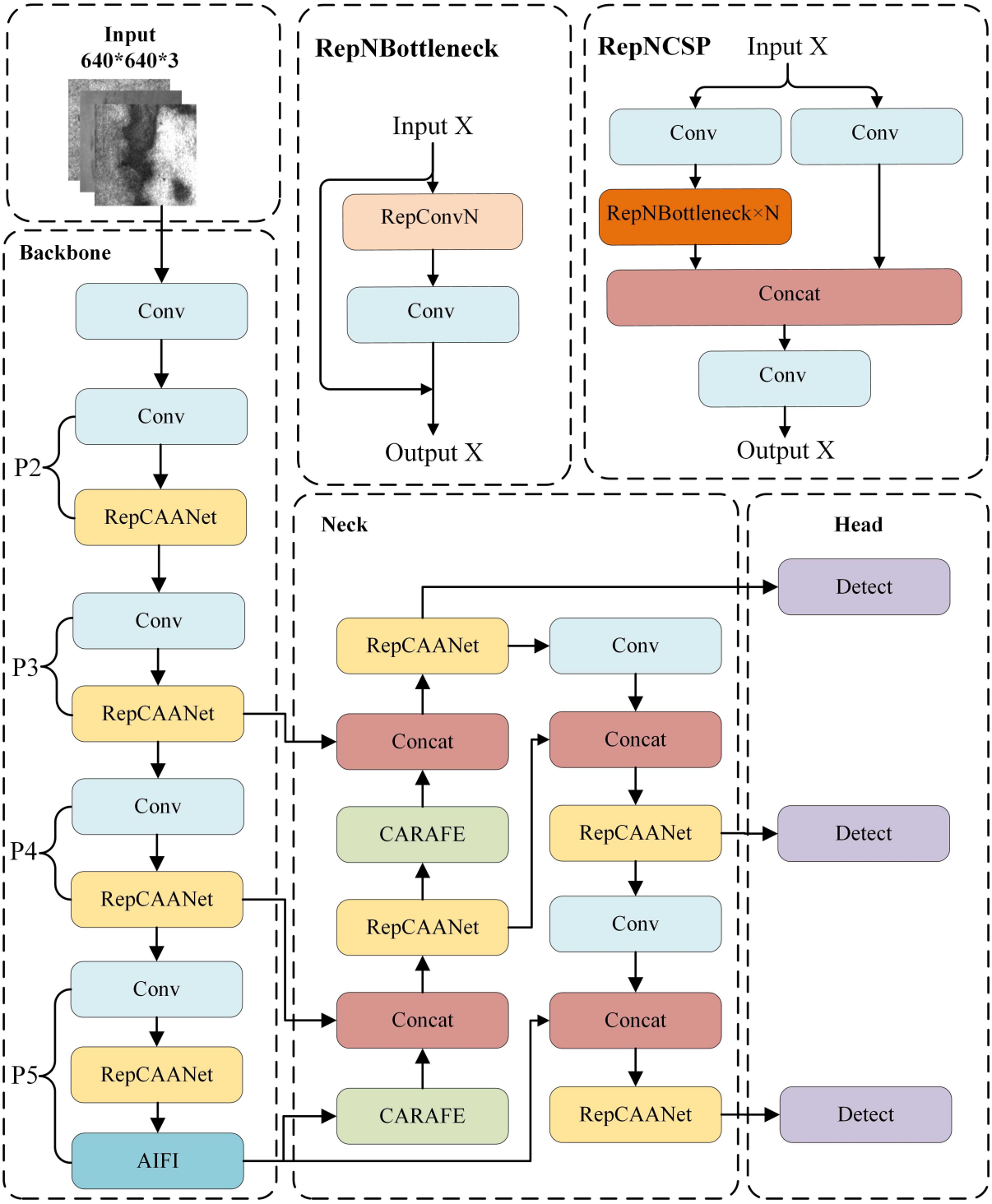

Fig. 1: RAC-YOLO Model Structure Diagram

classification through local-global feature fusion. Shao et al. [23] proposed MOD-YOLO, a multispectral detection model based on a Cross-Stage Partial Cross-Modality Fusion Transformer (CSP-CFT), which improves detection stability under low-visibility conditions. However, the self-attention mechanism suffers from quadratic computational complexity due to dot-product operations over query, key, and value vectors, further exacerbated by the inefficiency of the Softmax function, leading to high computational overhead in large-scale or real-time applications.

Despite recent advances, industrial steel surface defects remain challenging due to complex backgrounds, multi-scale variations, and uneven lighting, which contribute to high false alarm rates and unstable detection performance. Many existing methods improve detection accuracy at the expense of computational efficiency, thereby restricting their deployment in real-time industrial settings. To address these limitations, we propose RAC-YOLO, a lightweight defect detection framework based on YOLOv8n and specifically designed for industrial implementation. The main contributions of this work are outlined as follows:

(1) We propose RepCAANet, a lightweight feature fusion structure that alleviates gradient vanishing in deep networks and promotes efficient inter-layer feature transmission.

(2) We integrate an AIFI module to strengthen multi-scale feature integration, which is particularly beneficial for identifying small and low-contrast surface defects.

(3) We adopt the CARAFE mechanism for feature up-scaling, facilitating more accurate and semantically enhanced reconstruction of feature maps.

(4) We introduce the MPDIoU as a novel regression loss function, improving both bounding box localization accuracy and training stability.

(5) We demonstrate through extensive experiments on NEU-DET and GC10-DET that RAC-YOLO achieves competitive detection performance and strong generalization capability while maintaining a minimal computational load.

The remainder of this paper is organized as follows: Section II details the architecture and core components of RAC-YOLO. Section III presents the experimental setup and comparative evaluation. Section IV provides discussion and concluding remarks.

II. MODEL STRUCTURE AND PRINCIPLE

A. RAC-YOLO Algorithm Model

The proposed RAC-YOLO model, built upon YOLOv8n, is designed with the following architectural modifications. In the backbone network, the RepCAANet module substitutes the conventional C2f block, enhancing feature expressiveness and mitigating gradient degradation in deep layers. The AIFI module is integrated to supersede the standard SPPF component, promoting richer intra-scale feature fusion and boosting sensitivity to small and faint defects. Within the neck structure, the lightweight CARAFE upsampling module is employed in place of conventional upsampling methods. This modification enhances fine-grained detail preservation and feature reconstruction fidelity, yielding higher-precision feature maps. Finally, the MPDIoU loss function is implemented as a substitute for the CIoU loss employed in YOLOv8n. This offers a more precise regression target, leading to substantial gains in localization accuracy for detected objects. The overall architecture of the RAC-YOLO framework is illustrated in Figure 1.

B. Feature Extraction Networks Incorporating Attention Mechanisms

In object detection, ongoing innovation in network architectures and attention mechanisms remains crucial for enhancing detection performance. GELAN (General Efficient Layer Aggregation Network), recently proposed in YOLOv9, integrates advantages from both CSPNet and ELAN architectures. It is designed as a lightweight architecture guided by gradient path planning [24]. This design strikes a balance among lightweight structure, inference speed, and accuracy, and shows strong performance in joint detection and segmentation tasks [25]. At the core of GELAN lies the RepNCSPELAN4 module, which performs feature extraction and fusion by integrating elements from both CSP and ELAN structures. The RepNCSPELAN4 module mainly comprises a Conv module and a RepNCSP submodule. The latter contains another Conv module and a variable number of RepNBottleneck blocks. The number of RepNBottleneck blocks (denoted as N) is determined by the model's width factor. Each RepNBottleneck block is a residual unit composed of RepConvN and Conv layers.

The Context Anchor Attention (CAA) module serves as a key component for strengthening central feature representations and modeling long-range dependencies across spatially separated pixels [26]. Designed with structural efficiency in mind, the CAA module aims to amplify salient central features while recognizing contextual correlations over non-local regions. In its first stage, the module applies average pooling, succeeded by a 1×1 convolutional layer to capture localized feature patterns, as defined in Equation (1):

$$F_{l-1,n}^{pool} = Conv_{1\times1}(P_{avg}(X_{l-1,n}^{2})), n = 0, \ldots, N_{n-1} \quad (1)$$

P_{avg} is represents average pooling, for $n = 0$, there is $X_{l-1,n}^{2} = X_{l-1}^{2}$.

Subsequently, a pair of depthwise strip convolutions are utilized to approximate the receptive field of a conventional large-kernel depthwise convolution [27], as formulated in Equation(2)-(3):

$$F_{l-1,n}^{w} = DWConv_{1\times kb}(F_{l-1,n}^{pool}) \quad (2)$$

$$F_{l-1,n}^{h} = DWConv_{kb\times1}(F_{l-1,n}^{w}) \quad (3)$$

The use of depth-wise strip convolutions stems from two key motivations. On one hand, these operations enable a parameter-efficient design. When compared to standard large-kernel depth-wise convolutions, two 1D depth-wise kernels attain comparable effectiveness while reducing parameter count. On the other hand, they are well-suited for characterizing elongated defects, owing to their anisotropic receptive fields. The overall structure of the proposed CAA network is illustrated in Figure 2.

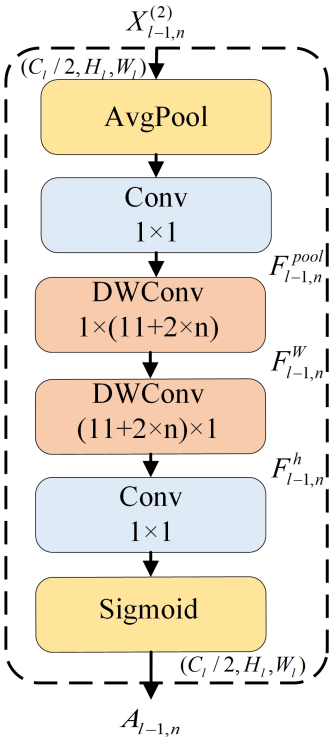

Fig. 2: CAAttention Model Structure Diagram

The proposed RAC-YOLO network integrates RepNCSPELAN4 with the CAA module to construct a novel backbone termed RepCAANet. This architecture is specifically

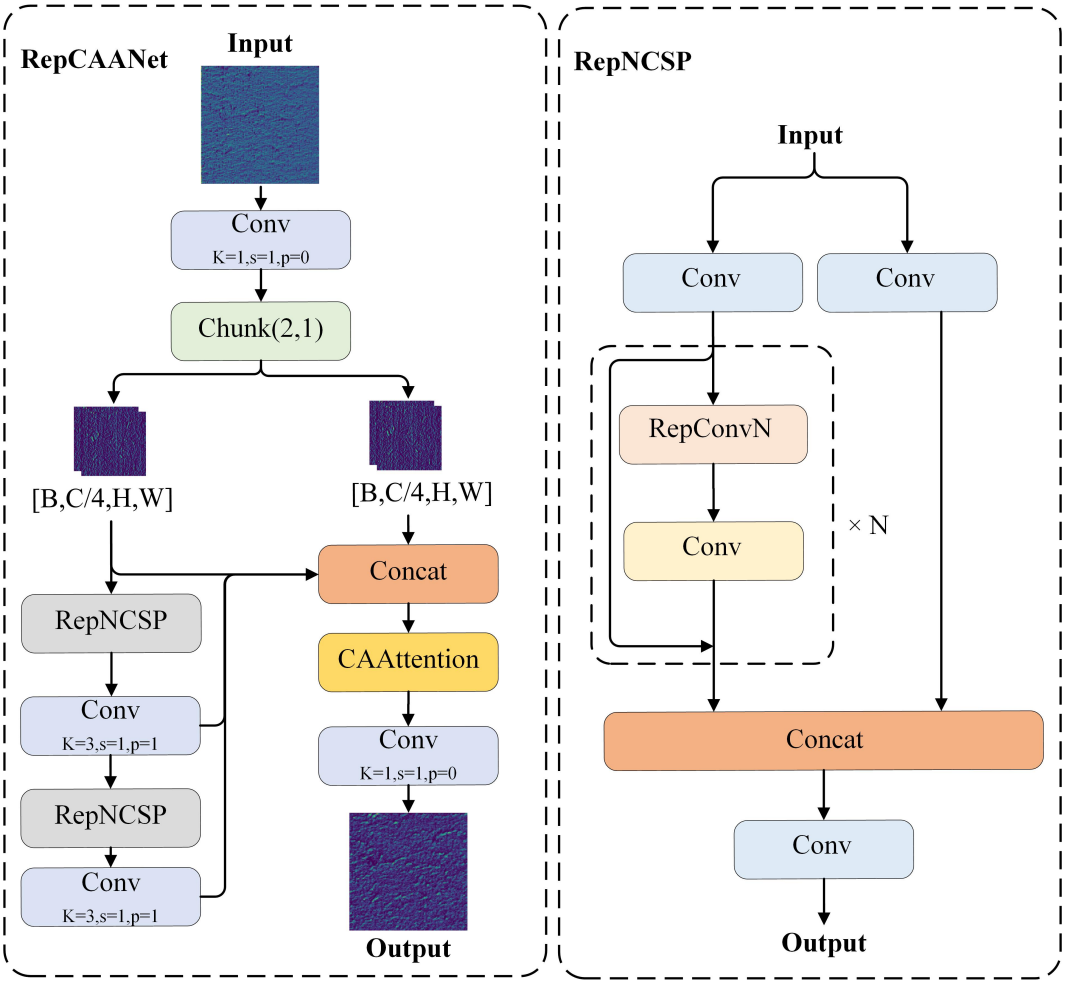

Fig. 3: RepCAANet and RepNCSP Model Structure Diagram

designed to retain a lightweight structure while improving gradient propagation, thereby enhancing robustness to scale variations and occlusions in input images. It effectively captures input characteristics and preserves critical spatial and semantic information, which is essential for accurate and efficient object detection. The detailed configurations of both RepCAANet and RepNCSP are illustrated in Figure 3.

C. Feature Pyramid Network Optimization

Surface defect detection of steel strips is often hindered by complex backgrounds and variable lighting conditions, which compromise detection accuracy and lead to frequent false positives and missed detections [28]. To address these challenges, we propose the AIFI module, which enhances object perception and classification while suppressing background interference.

In the backbone, YOLOv8n generates a multi-level feature hierarchy comprising low-level, mid-level, and high-level resolution maps. Among these, high-level features capture abstract semantic representations, essential for object discrimination in cluttered scenes. Nevertheless, the standard SPPF module carries out parallel multi-scale pooling without explicit intra-scale communication [29]. To enhance the representational power of high-level semantics, we propose the AIFI module, which facilitates contextual exchange within the high-level feature map. By limiting fusion with shallow features that contain limited semantic content, AIFI mitigates

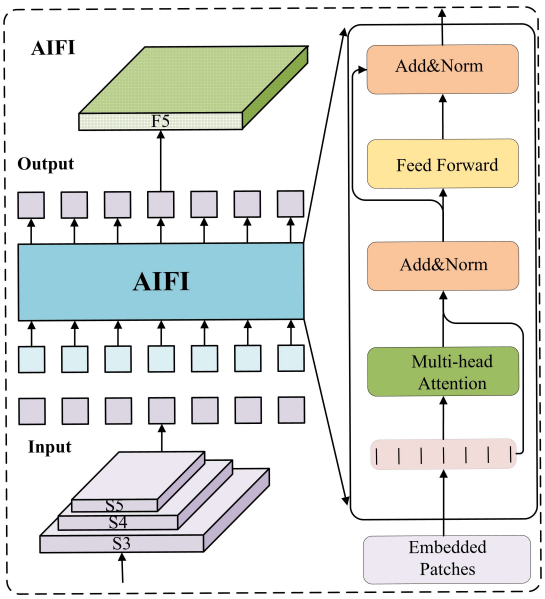

Fig. 4: AIFI Model Structure Diagram

unnecessary information integration and alleviates ambiguity in semantic interpretation. This design promotes more efficient use of high-level context, leading to a reduction in both false alarms and undetected instances. The structure of the AIFI module is illustrated in Figure 4.

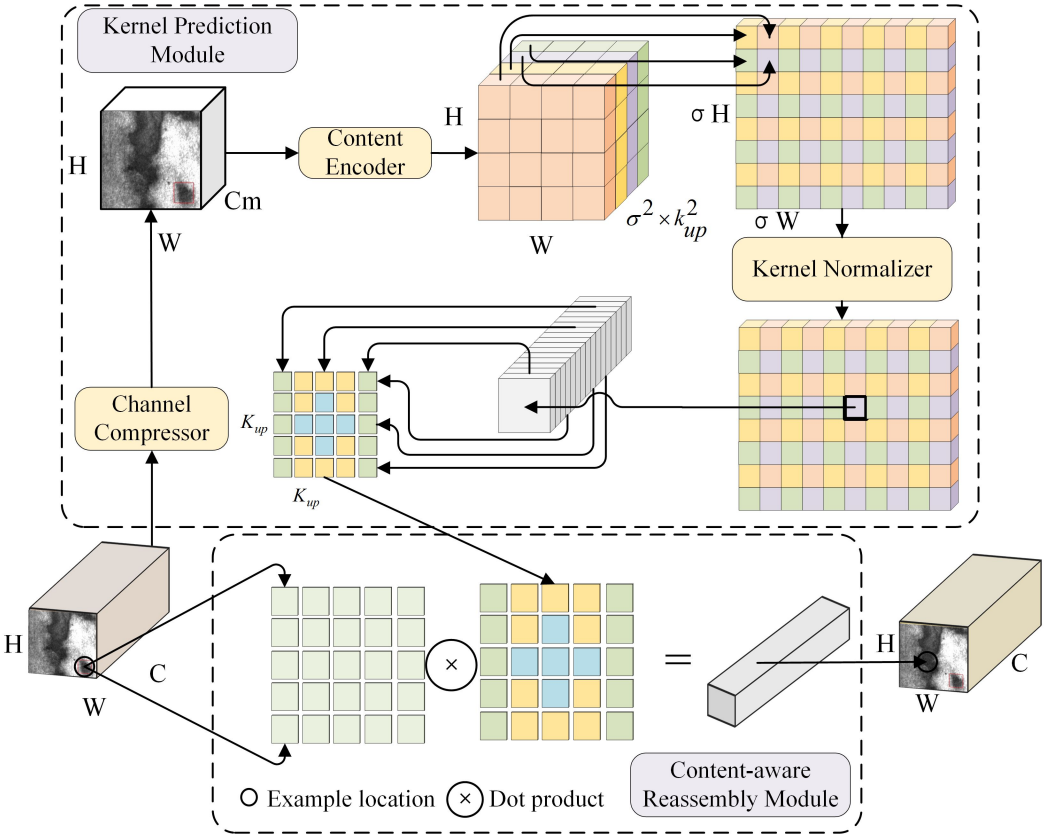

Fig. 5: CARAFE Model Structure Diagram

The AIFI module employs a self-attention mechanism to enhance contextual consistency among high-level semantic representations at identical spatial resolutions. In this process, the S_5 feature map is reshaped into a sequential vector format, followed by positional encoding. A multi-head self-attention operation is then performed on these encoded features, which enables precise capture of long-range semantic relationships. Subsequently, the resulting features are refined through fully connected layers with normalization, facilitating smooth integration into subsequent network stages before being restructured into 2D spatial feature maps [30]. These transformations are mathematically formulated in Equations (4) and (5).

$$Q = K = V = Flatten(S_5) \tag{4}$$

$$F_5 = Reshape(AIFI(Q, K, V)) \tag{5}$$

AIFI facilitates interactions among high-level features across different scales, thereby enhancing the network's capacity to discern relationships between conceptual entities within an image. This improvement in relational understanding contributes to the network's proficiency in processing intricate semantic information, ultimately leading to an increase in detection accuracy.

D. Upsampling Optimization

Upsampling is a fundamental operation in image processing and computer vision, primarily employed to enlarge the spatial dimensions and enhance the resolution of images or feature representations [31]. In dense prediction tasks like steel strip defect detection, traditional upsampling approaches typically depend on local pixel neighborhoods, potentially resulting in information degradation and compromising detection accuracy and robustness. To mitigate these limitations, the lightweight CARAFE module is incorporated into the YOLOv8n architecture to improve feature upscaling quality. The CARAFE module captures rich contextual cues and produces content-aware adaptive kernels, enabling accurate feature reassembly and replacing conventional bilinear interpolation. This strategy aims to refine semantic feature representation and boost detection accuracy for surface defects in steel strips [32]. The architectural layout of the CARAFE module is depicted in Figure 5.

The CARAFE module reconstructs feature maps in two phases: Initially, the upsampling kernel prediction module ψ performs convolution with dimensions $k_{encoder} \times C_m \times C_{up}$ using local features X_1 to create dynamic reassembly kernels $W_{1'}$. Next, the feature reassembly module ϕ utilizes these kernels for content-aware reassembly of the feature maps, determining output feature points within the $k_{up} \times k_{up}$ area by taking the dot product of $W_{1'}$ and $X_{1'}$, thereby improving the feature representation. The detailed process is outlined in equations (6)-(7):

$$W_{1'} = \psi(N(X_1, k_{encoder})) \tag{6}$$

$$X_{1'} = \phi(N(X_1, k_{up}), W_{1'}) \tag{7}$$

This approach allows for the meaningful reorganization and enhancement of feature maps according to their unique content traits, greatly boosting their representational power. The CARAFE operator is efficient and quick, adding only a small amount of extra computational load, which facilitates

its incorporation into contemporary network designs. Consequently, this paper incorporates the CARAFE upsampling operator into the YOLOv8 framework to obtain more detailed feature information.

E. Loss Function Optimization

The goal of bounding box regression is to improve the accuracy of predicted bounding boxes, ensuring they closely match the ground truth [33]. Since its introduction, Intersection over Union (IoU) has remained the predominant metric for evaluating localization loss in object detection tasks. As articulated in Equation (8):

$$IoU(pbb, gbb) = \frac{Area(pbb \cap gbb)}{Area(pbb \cup gbb)} \tag{8}$$

here, the predicted bounding box is denoted as pbb, and the ground truth bounding box is represented as gbb.

Nevertheless, IoU has several well-known limitations. First, it may assign identical scores to distinct predicted boxes, which can hinder convergence and compromise regression accuracy. Second, because IoU considers only the overlapping area, it ignores object size, leading to inconsistent performance across varying object scales. Third, its computational complexity increases significantly in dense scenes or with overlapping targets.

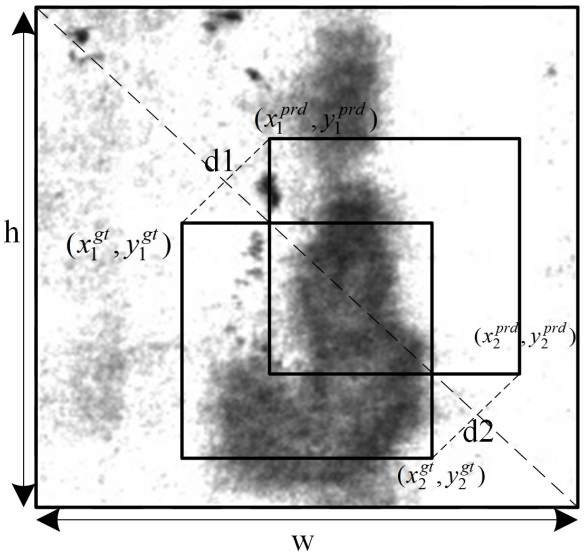

Fig. 6: MPDIOU Loss Function Diagram.

To address these limitations, we introduce the MPDIoU loss function. Rather than calculating intersection and union areas, MPDIoU lowers computational cost by measuring the pairwise distances between the centroids of bounding boxes. This approach proves particularly effective in densely populated scenes [34]. By emphasizing center-point distances, MPDIoU more accurately models spatial relationships, especially for closely located yet non-overlapping boxes, thereby improving detection precision. The formulation of MPDIoU is detailed in Equations (9)-(11):

$$d_1^2 = (x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2 \tag{9}$$

$$d_2^2 = (x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2 \tag{10}$$

$$MPDIoU = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{11}$$

Consider two arbitrary convex regions, labeled A and B, within an input image of width w and height h. The top-left and bottom-right vertices of shape A are given by (x_1^A, y_1^A) and (x_2^A, y_2^A), respectively. The quantities d_1^2 and d_2^2 are defined as the squared Euclidean distances from the top-left corner of A to a reference point within A, and from that point to the right boundary of shape B, respectively. The geometric configuration of the Modified Partial Distance Intersection over Union (MPDIOU) is depicted in Figure 6.

III. EXPERIMENTAL

A. Experimental Platform and Parameters

The experimental environment configuration and parameter settings are shown in Table I.

TABLE I: Experimental environment and parameter settings.

Name	Parameters
GPU	NVIDIA A10 24G
System	Ubuntu 22.04.5
Deep Learning Framework	Pytorch2.1
Epochs	300
Batch size	16
Learning rate	0.01
optimizer	AdamW

B. Evaluation Indicators

The performance of the network is mainly assessed using mean Average Precision (mAP) derived from training and the model's capability on the validation set. Core evaluation indicators—including Precision (P), Recall (R), and mAP—are employed to measure detection accuracy in a quantitative manner [35]. The formal definitions of Precision and Recall are delineated in Equations (12)-(15):

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$AP = \int_a^b P(R)d(R) \tag{14}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{15}$$

Here, True Positives (TP) denote instances correctly classified as positive, False Negatives (FN) correspond to actual positives overlooked by the model, and False Positives (FP) refer to negative samples falsely predicted as positive. Average Precision (AP) is calculated as the integral over the precision–recall curve, whereas mean Average Precision (mAP) represents the averaged AP values across all object classes, often weighted by class frequency. Additionally, the evaluation encompasses computational cost (in GFLOPs), total parameter count (Params), and inference speed, expressed in frames per second (FPS).

C. Experimental Datasets

The generalization ability of the proposed model is evaluated on two publicly available datasets for steel surface defect detection: NEU-DET [36] and GC10-DET [37].

The NEU-DET dataset contains six categories of steel surface defects: crazing (cr), inclusion (in), patches (pa), pitted surface (ps), rolled-in scale (rs), and scratches (sc). Each class includes 300 grayscale images of size 200 × 200 pixels, resulting in a total of 1,800 samples. The dataset is randomly split into training and testing sets following a 7:3 ratio, yielding 1,260 training and 540 test images.

The GC10-DET dataset comprises ten categories of steel surface defects: punching hole (ph & 1_chongkong), welding line (wl & 2_hanfeng), crescent gap (cg & 3_yueyawan), water spot (ws & 4_shuiban), oil spot (os & 5_youban), silk spot (ss & 6_siban), inclusion (in & 7_yiwu), rolled pit (rp & 8_yahen), crease (cr & 9_zhehen), and waist folding (wf & 10_yaozhe). It comprises 2,294 grayscale images, each with dimensions of 2048 × 1000 pixels. The data is divided into training and test subsets according to an 8:2 split, producing 1,848 training and 446 test samples.

Utilizing grayscale imagery enhances contrast, suppresses noise, and improves detail visibility, thereby reducing the influence of surface characteristics—such as reflectivity and surface roughness—on detection accuracy.

D. Ablation Study

To evaluate the effectiveness of each proposed module, we conduct an ablation study to quantify the contribution of individual components within the enhanced architecture. We particularly assess the impact of the redesigned backbone and neck structures on feature representation and detection accuracy. In addition, to identify the most effective loss function, we compare several commonly used variants under the proposed framework on the NEU-DET and GC10-DET datasets. The following abbreviations denote distinct configurations of the YOLOv8n-based model:

- YOLOv8n with the RepCAANet block is denoted as R-YOLO.
- YOLOv8n with both the RepCAANet block and AIFI is denoted as RA-YOLO.
- YOLOv8n with the RepCAANet block and CARAFE is denoted as RC-YOLO.
- YOLOv8n with the RepCAANet block, AIFI, and CARAFE is denoted as RAC-YOLO.

1) Ablation on NEU-DET: The results of the ablation experiments are summarized in Table II.

In the R-YOLO method, the backbone network is replaced by the RepCAANet module, reducing the number of parameters from 3.007M to 2.595M, a 14% decrease. Additionally, the computational load decreases from 8.1G to 6.9G, marking a 15% reduction compared to the baseline. Furthermore, R-YOLO achieves higher detection accuracy than YOLOv8n in the cr, pa, rs, and sc categories. The mean Average Precision (mAP) also increases by 0.6%, indicating that the RepCAANet module not only reduces model parameters but also improves detection speed while maintaining a lightweight architecture.

In the RA-YOLO method, the backbone network is again replaced by the RepCAANet module, this time combined with the AIFI module. Although this integration slightly increases the number of parameters, it reduces the computational cost from 8.1G to 7.2G, an 11% decrease. The mAP improves by 1%, indicating that the AIFI module effectively captures high-level semantic information and strengthens intra-scale feature interactions, thereby enhancing detection accuracy.

In the RC-YOLO method, the backbone network is replaced by the RepCAANet module and integrated with the CARAFE upsampling module. This modification reduces the parameter count from 3.007M to 2.735M, a 9% decrease. The computational load also drops from 8.1G to 7.2G, a total reduction of 1%. The mAP increases by 1.2%, highlighting that the CARAFE module improves semantic feature representation by effectively aggregating feature information, thereby enhancing the detection of small defects.

The RAC-YOLO method incorporates all three modules, slightly increasing the parameter count but reducing computational cost by 13.5%. The mAP increases significantly by 2.7%, marking the highest improvement among all methods compared to the baseline.

As shown in Figure 7, RAC-YOLO demonstrates the most balanced performance, achieving the highest mAP of 80.2% and leading in four out of six subcategory APs (cr: 48.1%, pa: 88.2%, rs: 82.7%, sc: 97.0%). The polygon area is 9.7% larger than that of the baseline YOLOv8n, reflecting a significant improvement in cross-scene generalization.

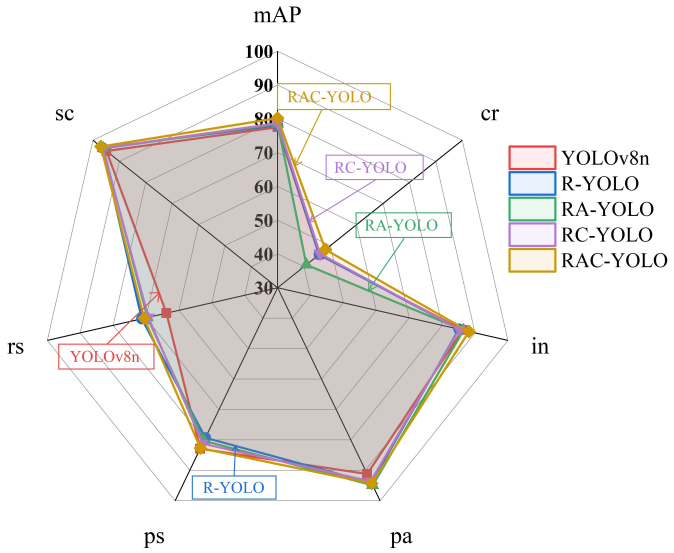

Fig. 7: Radar plot of ablation experiments on NUE-DET

These results confirm that the synergy of the three modules significantly enhances model accuracy, resulting in the best overall performance. Specifically, RAC-YOLO achieves optimal performance in the cr, in, pa, rs, and sc categories, further validating that the proposed enhancements effectively support the identification of various defect types.

2) Ablationon on GC10-DET: The ablation experiment results are summarized in Table III.

In the R-YOLO method, the backbone network is replaced by the RepCAANet module, leading to improved detection accuracy in the cr and wf categories compared to YOLOv8n, along with a 0.7% increase in mean Average Precision (mAP). RA-YOLO, which integrates the AIFI module with

TABLE II: Ablation on NEU-DET

Method	mAP50/%	AP/%						mAP50-95/%	Flops/G	Params/M
		cr	in	pa	ps	rs	sc			
YOLOv8n	77.5	45.7	86.3	91.2	**82.9**	63.8	94.9	44.9	8.1	3.007
R-YOLO	78.1	45.8	85.1	94.1	79.3	**71.3**	96.5	45.8	**6.9**	**2.595**
RA-YOLO	78.5	40.9	86.5	**94.9**	80.3	69.6	96.1	46.3	7.2	3.220
RC-YOLO	78.7	46.1	85.3	93.6	81.2	69.3	96.3	45.3	7.2	2.735
RAC-YOLO	**80.2**	**48.1**	**88.2**	94.3	82.7	70.6	**97.0**	**46.8**	7.0	3.165

TABLE III: Ablation on GC10-DET

Method	mAP50/%	AP/%										mAP50-90/%	Flops/G	Params/M
		ph	wl	cg	ws	os	ss	in	rp	cr	wf			
YOLOv8n	65.8	98.2	91.3	**97.7**	87.7	**70.1**	52.0	32.7	13.2	54.3	60.9	33.8	8.1	3.007
R-YOLO	66.5	98.0	91.6	97.6	**88.7**	65.4	52.2	**35.9**	12.2	55.3	67.5	33.6	**6.9**	**2.595**
RA-YOLO	67.8	97.3	91.5	96.4	86.4	66.7	53.6	34.1	24.9	56.8	69.9	34.8	7.2	3.220
RC-YOLO	68.4	**98.3**	**92.5**	97.2	86.2	69.0	**55.5**	32.0	25.7	59.8	67.8	34.5	7.2	2.735
RAC-YOLO	**70.1**	98.2	92.2	95.3	85.6	66.7	50.4	35.8	**32.1**	**72.4**	**72.8**	**35.3**	7.0	3.165

the RepCAANet backbone, achieves a 2% improvement in mAP. Similarly, RC-YOLO, which integrates the CARAFE upsampling module with the RepCAANet backbone, yields a 2.6% increase in mAP. The RAC-YOLO model, integrating all three modules, shows the most substantial improvement, with a 4.3% increase in mAP—the highest gain compared to the baseline.

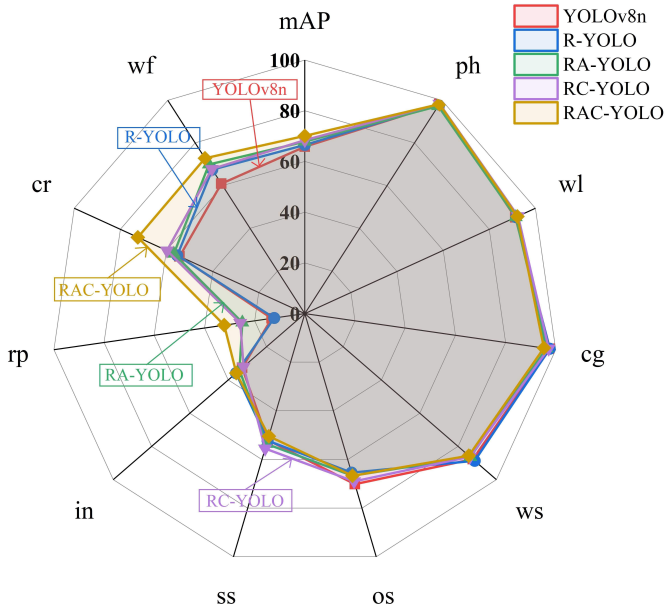

Fig. 8: Radar plot of ablation experiments on GC10-DET

Figure 8 shows a radar plot illustrating the ablation experiment results on the GC10-DET dataset. The proposed RAC-YOLO model demonstrates superior overall performance, reaching an average accuracy of 70.1%, a 4.3% improvement over the baseline YOLOv8n. It is particularly notable for its robustness in classification and feature representation, while also maintaining competitive performance in positional accuracy and contextual understanding. However, challenges remain in relocalization accuracy and occlusion invariance, highlighting domain-specific limitations common to all models. The progressive improvements from R-YOLO to RAC-YOLO confirm the cumulative effectiveness of architectural enhancements, especially in balancing detection accuracy and operational stability.

Moreover, the model demonstrates favorable efficiency in computational cost (FLOPs) and parameter optimization on the NEU-DET dataset, indicating that the proposed method achieves a more efficient solution for real-time steel surface defect detection. Additionally, ablation studies conducted on the GC10-DET dataset validate the contribution of the three key components, which simplify the architecture and improve generalization capability.

E. Comparative Experiments of Different Loss Functions

Owing to the suboptimal generalization and slow convergence of CIoU in defect detection scenarios, we propose the MPDIoU as an auxiliary bounding box regression loss, designed to enhance localization accuracy and model robustness. A comparative evaluation of the proposed loss is summarized in Table IV.

TABLE IV: The loss function compares the experimental results.

Loss Function	mAP50/%	Precision/%	Recall/%
EIoU	79.8	74.3	**75.1**
GIoU	78.3	74.4	72.2
CIoU	79.0	75.3	74.4
SIoU	78.5	70.5	74.5
MPDIoU	**80.2**	**77.1**	74.7

MPDIoU outperformed CIoU by 1.2% in the original model and showed improvements of 1.9%, 1.7%, and 0.4% over Generalized IoU (GIoU), Squared IoU (SIoU), and Enhanced IoU (EIoU), respectively. While EIoU achieved the highest recall rate, MPDIoU attained the best detection precision, reaching 80.2%. Overall, the proposed loss function demonstrates notable advantages across multiple evaluation metrics.

F. Comparative Experiments of Different Models

To assess the performance benefits of the proposed model, RAC-YOLO was systematically compared with several contemporary state-of-the-art methodologies. Specifically, the EfficientDet model was referenced from Ref. [38], DFP-YOLO from Ref. [17], CABF-FCOS from Ref. [39], Improved-YOLOX from Ref. [40], YOLO-DBL from Ref. [41], MD-YOLO from Ref. [42], FMR-YOLO from Ref. [43], Sun's method from Ref. [44], YOLOv5s-Improved from Ref. [45], LIDD-YOLO from Ref. [46], DCC-CenterNet from Ref. [47], TridentNet from Ref. [48], EDDN from Ref. [49], FPDNet from Ref. [50], and Ade-YOLO from Ref. [51]. The data pertaining to the other methodologies were acquired through our experimental investigations.

1) Comparison of Experimental Results from Different Models on the NEU-DET Dataset: Results in Table V demonstrate that the proposed model achieves superior performance over existing approaches with respect to mean Average Precision (mAP), with notable advantages in identifying defects such as patches (pa) and scratches (sc). RAC-YOLO attains an mAP of 80.2%, indicating improvements of 2.7% over YOLOv8n and 4.1% over YOLOv10n. It achieves a relative gain of 10.1% compared to EfficientDet. Despite DFP-YOLO achieving the best performance on patch detection, its overall mAP remains 5.2% behind RAC-YOLO. Even though CABF-FCOS excels in detecting crazing (cr) and pitted surface (ps) defects, its aggregate mAP is still 3.5% lower. RAC-YOLO outperforms Improved-YOLOX, YOLO-DBL, MD-YOLO, and FMR-YOLO by margins of 3.2%, 2.2%, 2.0%, and 1.6%, respectively, even though FMR-YOLO operates at a reduced frame rate. Compared to Sun's method, which exhibits limited effectiveness for ps defect recognition, RAC-YOLO delivers a net gain of 1.6%. Furthermore, it maintains advantages over YOLOv5s-Improved and LIDD-YOLO, with improvements of 1.4% and 0.7%, respectively.

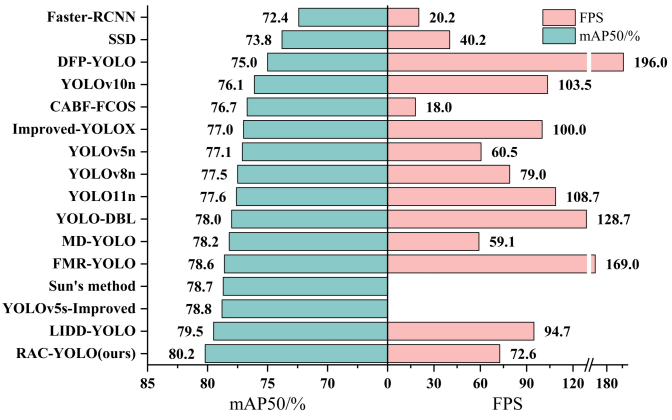

Fig. 9: Histogram comparison of different models on NEU-DET

Figure 9 reveals a fundamental compromise between detection accuracy and inference efficiency across the evaluated models. RAC-YOLO delivers the best detection performance with an mAP of 80.2%, operating within real-time constraints at 72.6 FPS. While capable of processing 196.0 FPS, DFP-YOLO falls short of RAC-YOLO by 5.2% in mAP. By comparison, conventional approaches like Faster R-CNN exhibit prohibitively high computational latency, making them less suitable for real-time deployment.

Overall, the results highlight the proposed model's superior balance of accuracy and efficiency in steel surface defect detection.

2) Comparison of Experimental Results from Different Models on the GC10-DET Datasets: To assess the generalization performance of RAC-YOLO in steel surface defect detection, we perform comprehensive evaluations using the GC10-DET benchmark, as presented in Table VI.

As illustrated in Figure 10, RAC-YOLO delivers notable gains over existing approaches, achieving a mean Average Precision (mAP) of 70.1% and an inference speed of 74.8 frames per second (FPS). On the GC10-DET benchmark, the model exhibits performance gains of 8.2%, 6.9%, 5.0%, 4.8%, 4.3%, 3.8%, 3.6%, 3.5%, 3.3%, 2.6%, 1.9%, and 1.9% in mAP relative to DCC-CenterNet, TridentNet, EDDN, YOLOv11n, YOLOv8n, SSD, YOLOv7-tiny, FMR-YOLO, FPDNet, Faster R-CNN, YOLOv5n, and Ade-YOLO, respectively. Collectively, these outcomes indicate that RAC-YOLO offers an improved trade-off among detection accuracy, model compactness, and computational efficiency. The success of the model stems from its efficient backbone for feature extraction, an enhanced feature fusion pyramid, and a lightweight upsampling design, jointly contributing to stronger generalization in practical defect detection scenarios.

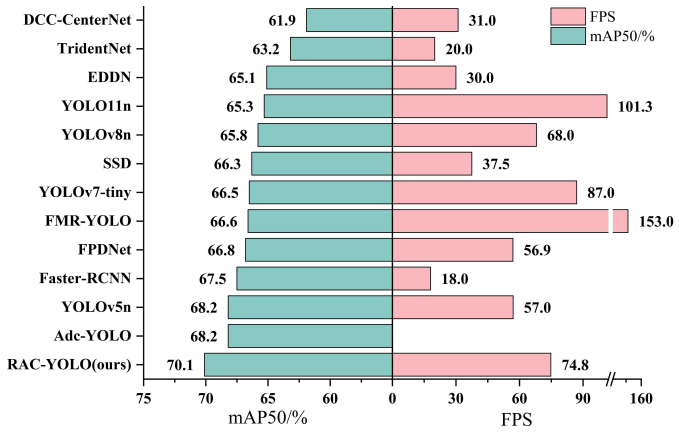

Fig. 10: Histogram comparison of different models on GC10-DET

G. Visual Comparison of Experimental Results

In the visualization analysis of experimental results, a multi-dimensional comparative approach is adopted to evaluate the effectiveness of the proposed method. By applying parallel detection comparison visualization, differences in coverage and localization accuracy between the original images and the detection results are illustrated for both the NEU-DET and GC10-DET datasets. This effectively highlights the model's adaptability to complex scenes. In addition, accuracy curves for YOLOv8n and RAC-YOLO are plotted to enable a systematic comparison of two key metrics: mAP50 and mAP50:95. The slope of these curves reflects the differences in detection stability between the algorithms. This two-fold analysis combines qualitative instance-level

TABLE V: The experimental results are compared with different models on NEU-DET

Method	mAP50/%	AP/%						FPS
		cr	in	pa	ps	rs	sc	
EfficientDet [38]	70.1	45.9	62.0	83.5	85.6	70.7	73.1	11.9
Faster-RCNN	72.4	43.2	68.5	84.9	78.3	69.8	89.9	20.2
SSD	73.8	45.8	75.2	84.1	85.3	68.6	83.5	40.2
DFP-YOLO [17]	75.0	38.1	82.2	**97.9**	80.2	60.9	90.8	**196.0**
YOLOv10n	76.1	43.7	84.0	92.6	76.7	65.3	94.3	103.5
CABF-FCOS [39]	76.7	55.4	75.0	93.5	88.9	62.9	84.4	18.0
Improved-YOLOX [40]	77.0	55.1	83.0	93.6	86.1	59.7	84.2	100.0
YOLOv5n	77.1	41.3	88.8	92.3	81.3	63.6	95.0	60.5
YOLOv8n	77.5	45.7	86.3	91.2	82.9	63.8	94.9	79.0
YOLO11n	77.6	44.2	87.6	92.6	80.4	64.4	96.6	108.7
YOLO-DBL [41]	78.0	47.2	**93.4**	82.0	64.2	**91.5**	89.9	128.7
MD-YOLO [42]	78.2	46.7	81.4	91.3	85.1	72.6	92.0	59.1
FMR-YOLO [43]	78.6	46.5	86.1	95.4	83.5	66.6	93.6	169.0
Sun's method [44]	78.7	41.6	84.6	93.7	**95.9**	66.1	90.5	-
YOLOv5s-Improved [45]	78.8	**58.5**	80.3	81.3	84.2	74.3	83.4	-
LIDD-YOLO [46]	79.5	51.0	86.9	96.6	85.8	67.2	89.8	94.7
RAC-YOLO(ours)	**80.2**	48.1	88.2	94.3	82.7	70.6	**97.0**	72.6

TABLE VI: The experimental results are compared with different models on GC10-DET

Method	mAP50/%	AP/%										FPS
		ph	wl	cg	ws	os	ss	in	rp	cr	wf	
DCC-CenterNet [47]	61.9	84.4	85.5	96.2	77.3	50.9	**84.8**	30.2	13.9	49.9	76.6	31.0
TridentNet [48]	63.2	96.6	43.5	95.8	76.9	72.9	67.0	24.0	40.2	28.4	79.4	20.0
EDDN [49]	65.1	90.0	88.5	84.8	55.8	62.2	65.0	25.6	36.4	52.1	91.9	30.0
YOLO11n	65.3	98.2	92.8	96.3	**88.6**	65.2	58.3	36.3	2.8	44.2	70.1	101.3
YOLOv8n	65.8	98.2	91.3	97.7	87.7	70.1	52.0	32.7	13.2	54.3	60.9	68.0
SSD	66.3	90.8	91.5	95.0	87.0	61.6	50.3	31.9	**98.0**	24.2	**99.1**	37.5
YOLOv7-tiny	66.5	97.9	91.5	95.3	84.2	65.4	58.6	32.6	12.3	60.8	66.5	87.0
FMR-YOLO [43]	66.6	95.7	67.1	52.7	70.7	85.4	31.4	**58.1**	39.8	**95.7**	69.8	**153.0**
FPDNet [50]	66.8	97.1	**94.7**	94.2	72.5	63.9	40.2	35.3	45.0	40.9	84.1	56.9
Faster-RCNN	67.5	96.5	29.6	**98.3**	73.1	**80.4**	83.8	52.4	43.1	45.9	68.9	18.0
YOLOv5n	68.2	**98.6**	92.0	94.7	86.8	69.8	54.7	34.5	15.9	64.7	70.7	57.0
Ade-YOLO [51]	68.2	95.8	91.7	94.9	70.0	76.9	67.8	35.4	22.0	41.7	85.6	-
RAC-YOLO(ours)	**70.1**	98.2	92.2	95.3	85.6	66.7	50.4	35.8	32.1	72.4	72.8	74.8

comparisons with quantitative statistical evaluations, offering an essential visual foundation for further model optimization.

1) Comparative Analysis on NEU-DET: As shown in Figure 11, the detection results of the proposed algorithm and the YOLOv8n algorithm are visually presented. The first, second, and third rows represent the original image, the detection result of YOLOv8n, and that of the enhanced RAC-YOLO algorithm on the NEU-DET dataset, respectively.

Comparative analysis indicates that YOLOv8n tends to miss small defect targets (e.g. rs) and exhibits localization errors when detecting larger ones (e.g. ps). In contrast, the improved RAC-YOLO demonstrates significantly better performance in detecting large defects and shows increased sensitivity to hard-to-detect targets, effectively reducing missed detections. Moreover, it adopts a more refined detection strategy for adjacent or overlapping defects, which substantially alleviates the detection errors observed in the baseline model.

2) Comparative Analysis on GC10-DET: The GC10-DET dataset does not provide explicit labels for the defect categories in each image. To assess detection performance, a diverse set of images containing various defect types was selected for comparative analysis. As shown in the heatmaps, the YOLOv8n model failed to detect several defects. For example, sub-image (1) missed the 6_siban defect, sub-image (2) did not detect the 1_chongkong defect, sub-image (3) overlooked the 5_youban defect, and sub-image (6) again failed to identify the 6_siban defect.The comparative detection outcomes are visualized in Figure 12.

Relative to baseline models, the proposed RAC-YOLO substantially lowers the false negative rate, leading to improved overall detection accuracy. The performance gain can be largely explained by the integration of several key components: the RepCAANet backbone, which enables efficient feature learning; the AIFI module, designed to strengthen

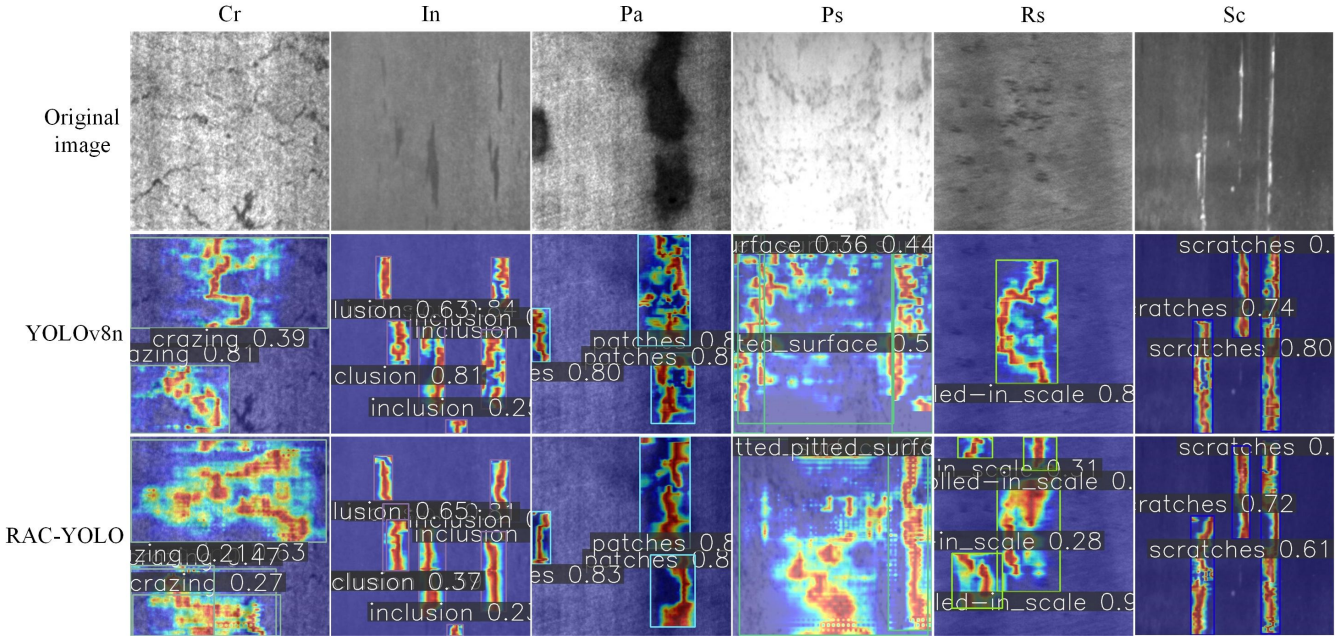

Fig. 11: Comparison of heat maps of detection results on NEU-DET

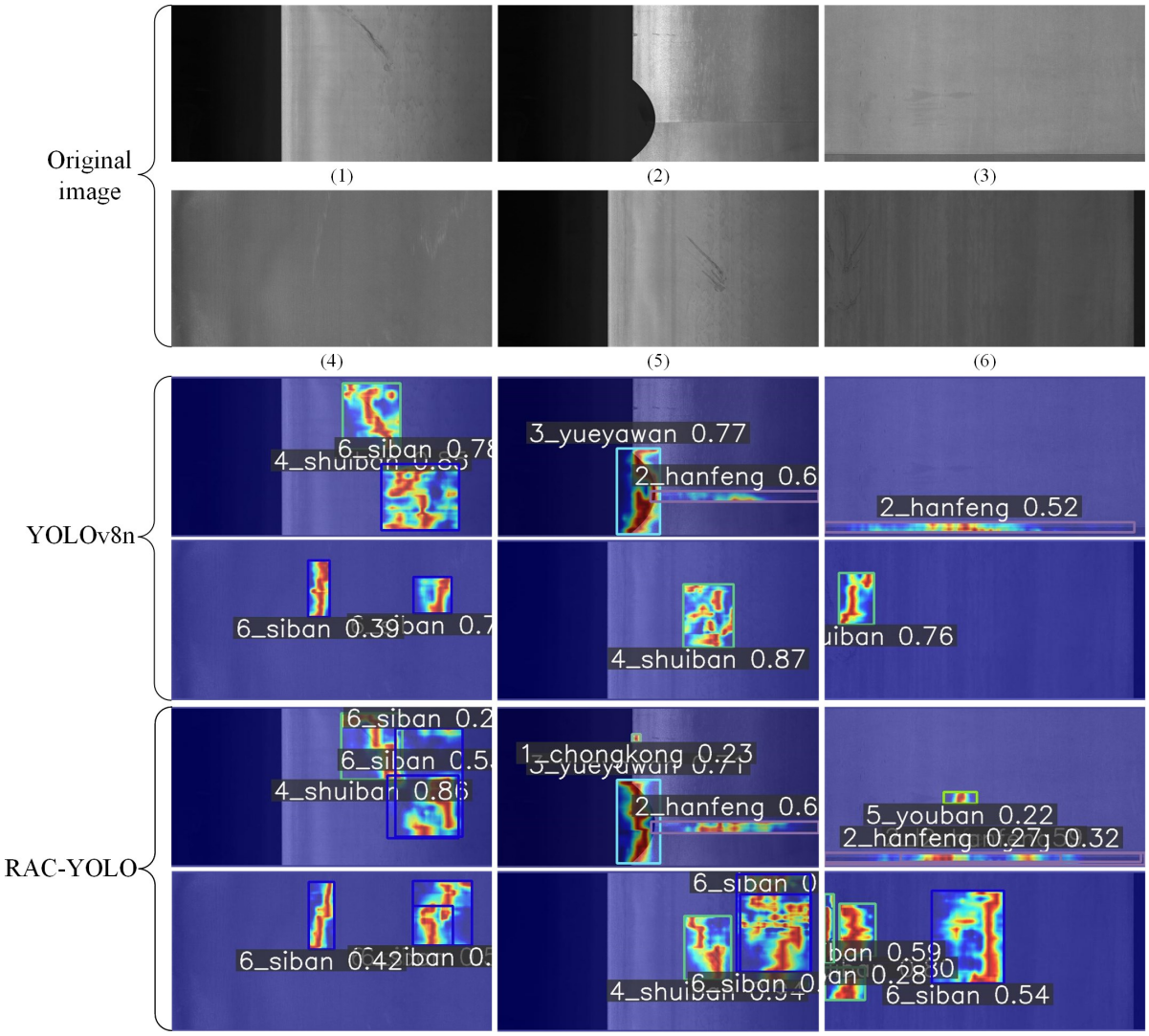

Fig. 12: Comparison of heat maps of detection results on GC10-DET

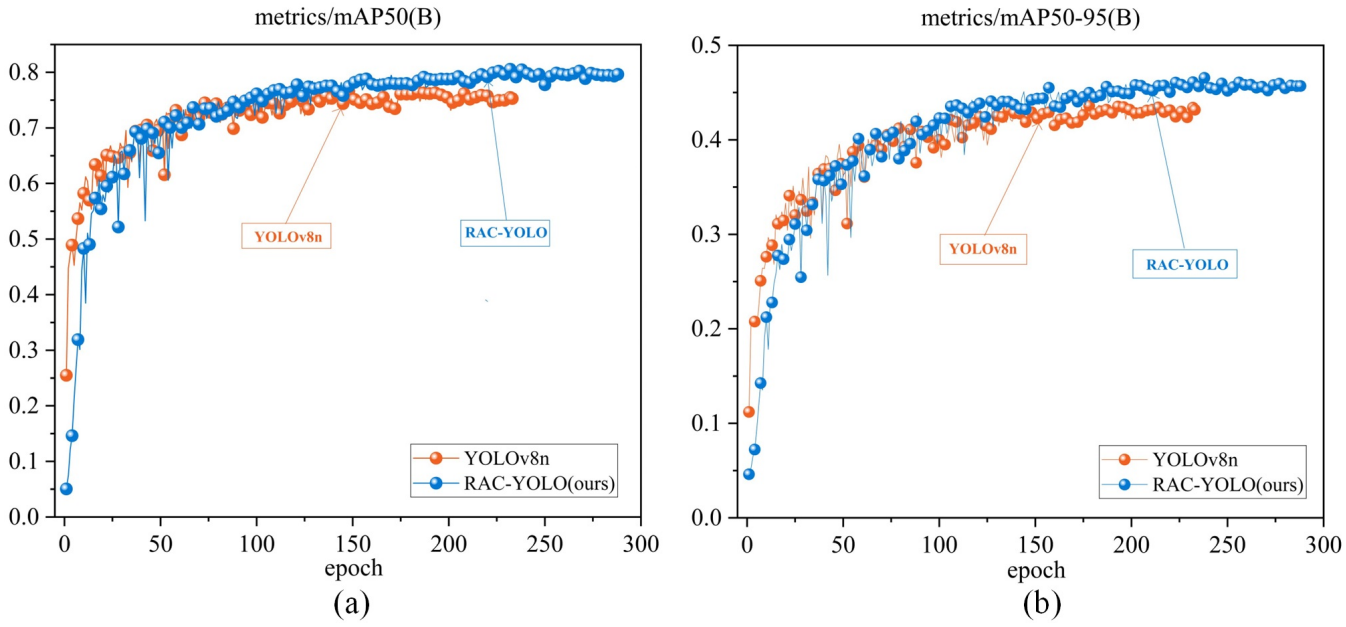

Fig. 13: Comparison line plot of two mAPs of different models on NEU-DET: (a) Comparison of mAP50/% line-charts; (b) Comparison of mAP50-95/% line-charts

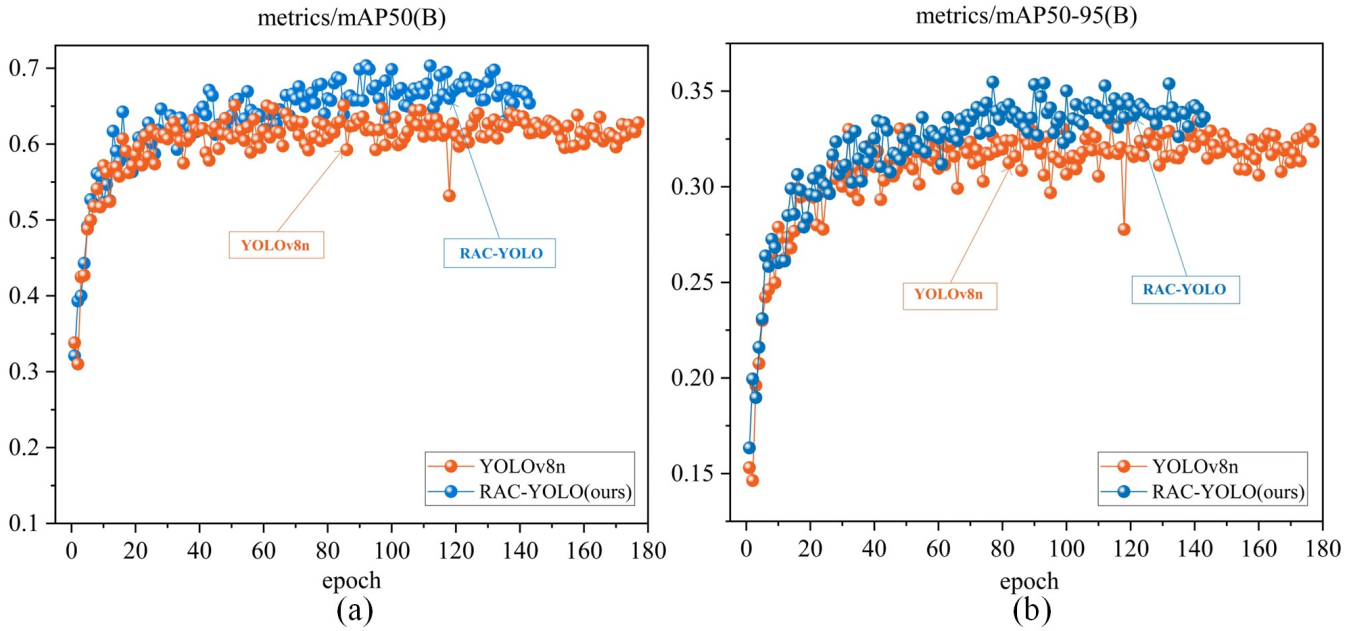

Fig. 14: Comparison line plot of two mAPs of different models on GC10-DET: (a) Comparison of mAP50/% line-charts; (b) Comparison of mAP50-95/% line-charts

long-range object awareness; CARAFE, facilitating high-fidelity feature upsampling; and the MPDIOU loss, optimized for precise bounding box refinement. Together, these components enable RAC-YOLO to capture subtle defect patterns, contributing to superior localization and classification performance.

3) Comparison with the Baseline Model: To evaluate the improvements of the proposed RAC-YOLO model over the baseline, both RAC-YOLO and the original YOLOv8n were trained for 300 epochs under identical hyperparameter settings. Performance was assessed using the validation set.

Figures 13 and 14 illustrate the evolution of key performance metrics throughout training on two datasets. In these plots, the blue and yellow curves correspond to the RAC-YOLO and YOLOv8n models, respectively. An early stopping strategy was employed, terminating training if the mean Average Precision (mAP) showed no notable improvement over 50 consecutive epochs. This strategy was adopted to address three critical aspects. First, it mitigates overfitting by preventing excessive adaptation to dataset-specific noise, which is crucial for assessing generalization across diverse datasets. Second, it optimizes computational efficiency by dynamically monitoring both mAP50 and mAP50–95, enabling training to stop once performance stabilizes. Third, the 50-epoch threshold offers a practical trade-off between responsiveness to performance fluctuations and sufficient

observation time to confirm training stagnation.

During training, the RAC-YOLO model demonstrated sustained improvements over the baseline in both mAP and detection rate, which is visually illustrated by the blue curves consistently exceeding the yellow counterparts. These observations indicate the enhanced detection accuracy and resilience to variations achieved by the proposed model.

IV. CONCLUSION

This work introduces RAC-YOLO, a novel framework for steel surface defect detection designed to achieve an optimal trade-off between model efficiency and detection accuracy. The backbone architecture is restructured to minimize computational burden without compromising detection precision. The AIFI module is incorporated to strengthen feature discrimination and class separation, while suppressing contextual distractions from the background. Additionally, the CARAFE module is adopted to enhance feature map reconstruction, with particular benefits for identifying small-scale defects. Furthermore, the MPDIoU loss is utilized to improve bounding box coordinate prediction, leading to more precise object localization.

Experimental evaluations demonstrate that RAC-YOLO attains an mAP of 80.2% on NEU-DET and 70.1% on GC10-DET, confirming its competitive performance relative to current state-of-the-art approaches in steel surface defect detection. Ablation experiments further corroborate the efficacy of each proposed component, underscoring their individual roles in achieving performance improvements.

However, the model remains challenged in localizing very small defects, including crazing and rolled-in scale. This limitation indicates a need for future work to concentrate on improving small-region defect detection and accelerating inference speed to satisfy practical requirements in industrial deployment.

REFERENCES

[1] R. Usamentiaga, D. G. Lema, O. D. Pedrayes et al., "Automated surface defect detection in metals: A comparative review of object detection and semantic segmentation using deep learning," *IEEE Transactions on Industry Applications*, vol. 58, no. 3, pp. 4203-4213, 2022.

[2] X. Cao, W. Xie, S. M. Ahmed et al., "Defect detection method for rail surface based on line-structured light," *Measurement*, vol. 159, p. 107771, 2020.

[3] S. Dong, P. Wang, K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[4] X. Xie, G. Cheng, J. Wang et al., "Oriented R-CNN for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3520-3529.

[5] L. Qiao, Y. Zhao, Z. Li et al., "Defrcn: Decoupled faster R-CNN for few-shot object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8681-8690.

[6] X. Xu, M. Zhao, P. Shi et al., "Crack detection and comparison study based on faster R-CNN and mask R-CNN," *Sensors*, vol. 22, no. 3, p. 1215, 2022.

[7] V. Sehwag, M. Chiang, P. Mittal, "SSD: A unified framework for self-supervised outlier detection," *arXiv preprint arXiv:2103.12051*, 2021.

[8] T. Diwan, G. Anirudh, J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243-9275, 2023.

[9] X. Zhu, W. Su, L. Lu et al., "Deformable DETR: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[10] Y. Cai, H. Bian, J. Lin et al., "Retinexformer: One-stage retinex-based transformer for low-light image enhancement," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 12504-12513.

[11] Z. Guo, C. Wang, G. Yang et al., "MSFT-YOLO: Improved YOLOv5 based on transformer for detecting defects of steel surface," *Sensors*, vol. 22, no. 9, p. 3467, 2022.

[12] C. Zhao, Y. Fan, J. Tan et al., "FCS-YOLO: an efficient algorithm for detecting steel surface defects," *Measurement Science and Technology*, vol. 35, no. 8, p. 086004, 2024.

[13] Guinan Wu, Qinghong Wu, "Enhancing Steel Surface Defect Detection: A Hyper-YOLO Approach with Ghost Modules and Hyper FPN," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp. 1321-1330, 2024.

[14] X. Lv, F. Duan, J. Jiang et al., "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, 2020.

[15] W. Zhang, J. Liu, Z. Yan et al., "FC-YOLO: an aircraft skin defect detection algorithm based on multi-scale collaborative feature fusion," *Measurement Science and Technology*, vol. 35, no. 11, p. 115405, 2024.

[16] Luyu Sun, Yujun zhang, "Lightweight Steel Surface Defect Detection Based on YOLOv10," *Engineering Letters*, vol. 33, no. 7, pp. 2673-2685, 2025.

[17] Jiawei Chai, ziwei Zhou, "DFP-YOLO: An Efficient Algorithm for Detecting Steel Surface Defects," *IAENG International Journal of Computer Science*, vol. 52, no. 6, pp. 1754-1763, 2025.

[18] ShiQuan Gao, Ying Tian, "Research on Steel surface Defects Detection Algorithms by YOLOv8 Based on Attention Mechanism," *IAENG International Journal of Computer Science*, vol. 51, no. 9, pp. 1309-1315, 2024.

[19] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] M. Raghu, T. Unterthiner, S. Kornblith et al., "Do vision transformers see like convolutional neural networks?," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12116-12128, 2021.

[21] I. Eum, J. Kim, S. Wang et al., "Heavy Equipment Detection on Construction Sites Using You Only Look Once (YOLO-Version 10) with Transformer Architectures," *Applied Sciences*, vol. 15, no. 5, pp. 2320, 2025.

[22] -V. R. Patheda, G. Laxmisai, B. Gokulnath et al., "A Robust Hybrid CNN+ ViT Framework For Breast Cancer Classification Using Mammogram Images," *IEEE Access*, 2025.

[23] Y. Shao, Q. Huang et al., "MOD-YOLO: Multispectral object detection based on transformer dual-stream YOLO," *Pattern Recognition Letters*, vol. 183, pp. 26-34, 2024.

[24] C. Y. Wang, I. H. Yeh, H. Y. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision*, Cham, Switzerland: Springer Nature Switzerland, 2024, pp. 1-21.

[25] C. Y. Wang, H. Y. M. Liao, Y. H. Wu et al., "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*, 2020, pp. 390-391.

[26] X. Cai, Q. Lai, Y. Wang et al., "Poly kernel inception network for remote sensing detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 27706-27716.

[27] K. W. Lau, L. M. Po, Y. A. U. Rehman, "Large separable kernel attention: Rethinking the large kernel attention design in CNN," *Expert Systems with Applications*, vol. 236, p. 121352, 2024.

[28] X. Fang, Q. Luo, B. Zhou et al., "Research progress of automated visual surface defect detection for industrial metal planar materials," *Sensors*, vol. 20, no. 18, p. 5136, 2020.

[29] X. Wang, H. Gao, Z. Jia et al., "BL-YOLOv8: An improved road defect detection model based on YOLOv8," *Sensors*, vol. 23, no. 20, p. 8361, 2023.

[30] Y. Zhao, W. Lv, S. Xu et al., "Detrs beat YOLOs on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16965-16974.

[31] K. Luo, C. Wang, S. Liu et al., "Upflow: Upsampling pyramid for unsupervised optical flow learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1045-1054.

[32] J. Wang, K. Chen, R. Xu et al., "Carafe: Content-aware reassembly of features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3007-3016.

[33] Y. Shen, F. Zhang, D. Liu et al., "Manhattan-distance IOU loss for fast and accurate bounding box regression and object detection," *Neurocomputing*, vol. 500, pp. 99-114, 2022.

[34] S. Ma, Y. Xu, "MPDIoU: A loss for efficient and accurate bounding box regression," *arXiv preprint arXiv:2307.07662*, 2023.

[35] J. Miao, W. Zhu, "Precision–recall curve (PRC) classification trees," *Evolutionary Intelligence*, vol. 15, no. 3, pp. 1545-1569, 2022.

[36] Y. He, K. Song, Q. Meng et al., "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493-1504, 2019.

[37] X. Lv, F. Duan, J. Jiang et al., "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, 2020.

[38] M. Tan, R. Pang, Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10781-10790.

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

[40] C. Li, A. Xu, Q. Zhang et al., "Steel surface defect detection method based on improved YOLOX," *IEEE Access*, 2024.

[41] K. Xu, D. Zhu, C. Shi et al., "YOLO-DBL: a multi-dimensional optimized model for detecting surface defects in steel," *Journal of Membrane Computing*, p. 1-11, 2025.

[42] H. Zheng, X. Chen, H. Cheng et al., "MD-YOLO: Surface defect detector for industrial complex environments," *Optics and Lasers in Engineering*, vol. 178, p. 108170, 2024.

[43] Y. Ni, Q. Wu, X. Zhang, "FMR-YOLO: An improved YOLOv8 algorithm for steel surface defect detection," *IET Image Processing*, vol. 19, no. 1, p. e70009, 2025.

[44] Luyu Sun, Yujun Zhang, "Steel Surface Defect Detection Based on YOLOv10," *Engineering Letters*, vol. 33, no. 5, pp. 1220-1231, 2025.

[45] H. Li, M. Liu, Y. Yin et al., "Steel surface defect detection based on multi-layer fusion networks," *Scientific Reports*, vol. 15, no. 1, p. 10371, 2025.

[46] S. Luo, Y. Xu, C. Zhang et al., "LIDD-YOLO: a lightweight industrial defect detection network," *Measurement Science and Technology*, vol. 36, no. 1, p. 0161b5, 2024.

[47] R. Tian, M. Jia, "DCC-CenterNet: A rapid detection method for steel surface defects," *Measurement*, vol. 187, p. 110211, 2022.

[48] Y. Li, Y. Chen, N. Wang, Z. X. Zhang, "Scale-aware trident networks for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6053-6062.

[49] X. Lv, F. Duan, J. Jiang et al., "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, 2020.

[50] F. Jiang, K. Lu, W. Wang et al., "FPDNet: A fast and high-precision detection network for hot-rolled strip surface defects," *Measurement*, vol. 253, p. 117309, 2025.

[51] Y. Wei, R. Wang, M. Zhang et al., "ADE-YOLO: real-time steel surface flaw recognition through enhanced adaptive attention and dilated convolution fusion," *Signal, Image and Video Processing*, vol. 19, no. 6, p. 457, 2025.