PAA-YOLO: A Steel Surface Defect Detection Method Based on Progressive Axial Attention

Yunfei Xiong[1,2], Kaining Liu[3], Jingyue Sun[4], Chenglong Fu[1,2], Jian Yao[1,2], Chuang Wang[1,2], Pengjiang Qian[1,2*]
[1]School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
[2]The PRC Ministry of Education Engineering Research Center of Intelligent Technology for Healthcare, Wuxi 214122, China
[3]School of Computer Science and Technology, Huzhou University, Huzhou 330500, China
[4]School of Cyberspace Security, University of Science and Technology of China, Hefei 230000, China
qianpjiang@jiangnan.edu.cn

***Abstract*—To address the industrial demand for higher accuracy and efficiency in steel surface defect detection, this paper proposes PAA-YOLO, an improved model based on YOLOv11. To enhance the model's extraction of linear defects (e.g., scratches), we design the Progressive Axial Attention (PAA) module. Using the PAA module, we restructure the backbone's C3k2 and C2PSA modules into C3k2-PAA and C2PSPAA, boosting the backbone's feature extraction capability. Finally, we optimize the neck network by integrating the PAA module and hypergraph-based semantic collection and scattering framework (HGC-SCS), inserting a hypergraph computation intermediate layer between the backbone and neck to strengthen high-order semantic feature correlations and improve the neck's feature fusion performance. Evaluations on the public NEU-DET dataset show PAA-YOLO achieves an mAP50 of 81.0% (3.1% higher than the baseline), with precision and recall improved by 4.6% and 4.0% respectively. On GC10-DET, it gains a 3.3% mAP50 improvement over the baseline. Experimental results confirm PAA-YOLO effectively enhances steel surface defect detection accuracy. The source code is available at https://github.com/xyfmzmy/PAA**

***Index Terms*—Surface Defect Detection, Steel Strips, Progressive Axial Attention, YOLO.**

I. Introduction

In industrial manufacturing, steel surface quality is critical to the performance and reliability of end products. Studies indicate that surface-related anomalies account for 65% of finished steel product defects, severely restricting their application in high-value-added industries [1]. Therefore, advancing surface quality control is indispensable for the intelligent, quality-driven upgrading of modern steel production.

Traditional machine learning methods have demonstrated limited effectiveness in steel surface defect detection. Early research primarily adopted handcrafted features (e.g., gray-level co-occurrence matrices, Gabor filters, local binary patterns) combined with Support Vector Machine (SVM) classifiers. Such methods exhibit critical limitations: over-reliance on manual feature engineering, limited adaptability to complex defect morphologies, and subpar accuracy with poor generalization ability. These drawbacks render them incapable of addressing the dynamic challenges of industrial production environments where defect characteristics vary drastically.

With the rapid advancement of artificial intelligence and deep learning, the technology demonstrates superior generalization and performance, offering significant advantages for industrial quality inspection. Deep learning-based object detection algorithms are categorized into two types based on their detection pipelines: two-stage and single-stage [2]. Two-stage algorithms (e.g., R-CNN [3], Faster R-CNN [4]) first generate candidate regions before classification, offering high detection accuracy but lacking real-time capability for latency-sensitive scenarios. In contrast, single-stage algorithms directly predict object locations and categories without candidate region extraction, ensuring faster inference speeds; representative models include SSD [5], the YOLO series [6]–[13], and RT-DETR [14]. These techniques have advanced surface defect detection and expanded computer vision applications in industrial quality inspection. Ma et al. [15] incorporated linear attention into YOLOv8 and proposed a selective feature pyramid network to enhance cross-level feature fusion.

Despite these advances, practical steel surface defect detection still faces challenges: (1) Models often neglect spatial positional information, exhibit low sensitivity to linear defects (e.g., scratches), are vulnerable to ambient lighting interference, and lack adequate feature extraction capability, limiting detection accuracy. (2) Models ignore high-order feature correlations and lack the ability to fuse multi-scale, multi-location features, restricting generalization performance. To tackle these challenges, this paper proposes the PAA-YOLO steel surface defect detection network based on a progressive axial attention mechanism. The key contributions of this work are as follows:

1) Progressive Axial Attention: To boost the model's feature extraction capability and sensitivity to strip-like defects, we propose the Progressive Axial Attention (PAA) module. It effectively captures cross-spatial long-range dependencies, thus strengthening strip-like defect feature extraction.
2) C3k2-PAA: We enhance the C3k2 module of YOLOv11 via PAA module integration, yielding the C3k2-PAA module. This module effectively improves the backbone network's feature extraction performance.
3) C2PSPAA: We optimise YOLOv11's C2PSA module by integrating the PAA module, generating the C2PSPAA

module to further boost the backbone network's feature extraction capability.

4) Neck Network Optimisation: We optimise the neck network by integrating the PAA module and the Hypergraph-based Semantic Collection and Scattering Framework (HGC-SCS). An intermediate hypergraph computation layer is inserted between the original backbone and neck networks, thereby strengthening higher-order feature correlations and enhancing the neck network's feature fusion performance.

The remainder of this paper is organized as follows. Section II provides the preliminaries for this study. In Section III, the details of this method are introduced. Then, the proposed method is evaluated in Section IV. Finally, a conclusion is made in Section V.

II. Preliminaries

A. HyperConv

Based on Graph Neural Networks (GNNs), Hypergraph Convolution (HyperConv) models higher-order data relationships beyond pairwise interactions. Traditional graphs only encode pairwise interactions, while hypergraphs connect arbitrary nodes via hyperedges, naturally capturing multi-body relationships. To enable efficient feature propagation on hypergraphs, various HyperConv mechanisms have been developed. Feng et al. [16] proposed Hypergraph Neural Networks (HGNNs) in 2019, establishing the first universal framework. It integrates hypergraph Laplacian operators with spectral approximation (e.g., Chebyshev polynomial expansion) to enable end-to-end learning of higher-order correlations. Bai et al. [17] later introduced a direct spatial hypergraph convolution operator; it defines trainable convolution and attention modules (embeddable in any GNN) via hyperedge-driven information aggregation and propagation, greatly enhancing model expressiveness. Yadati et al. [18] proposed HyperGCN, which leverages hypergraph structure for semi-supervised learning by converting hypergraphs into mediator graphs. These works laid the theoretical foundation for HyperConv and promoted its application in recommendation systems, action recognition, multimodal learning, etc. Building on this, Feng et al. [19] proposed the Hypergraph Computation-based Semantic Collection and Scattering (HGC-SCS) framework to map visual features to semantic spaces. It incorporates fine-grained hypergraph-modeled relationships to improve classification and localization accuracy, extending hypergraph learning to computer vision.

III. Methods

This paper proposes PAA-YOLO, a steel surface defect detection method with YOLOv11 as the baseline, which integrates the novel Progressive Axial Attention (PAA) module proposed herein, and its network architecture is illustrated in Figure 1. First, the PAA module is developed to enhance the model's perception of strip-like defects. Then, PAA-based C3k2-PAA and C2PSPAA modules are designed to replace the C3k2 and C2PSA modules in the backbone network, boosting its feature extraction capability. Furthermore, hypergraph computing is integrated as an intermediate layer based on the PAA module and HGC-SCS framework, which strengthens high-order feature correlations and improves the neck network's multi-scale feature fusion performance.

A. Progressive Axial Attention

To boost the model's steel surface defect feature extraction capability, perception of linear defects (e.g., scratches), and capture of long-range spatial dependencies, we propose Progressive Axial Attention (PAA), whose architecture is illustrated in Figure 2. First, strip-wise average pooling is applied to input features along horizontal and vertical directions. This captures long-range dependencies along one spatial dimension while retaining precise positional information along the other, generating horizontal and vertical spatial features, as formulated in (1) and (2).

$$\boldsymbol{z}_c^h(h) = \frac{1}{H} \sum_{0 \le i < H} \boldsymbol{x}_c(h, i) \tag{1}$$

$$\boldsymbol{z}_c^w(w) = \frac{1}{W} \sum_{0 \le j < W} \boldsymbol{x}_c(j, w) \tag{2}$$

Subsequently, the derived spatial features are split into two parallel streams. The first stream is fed into three multi-scale 1D depthwise convolutions (DWConv); their outputs are aggregated via summation to extract spatial features with distinct receptive fields, thereby enhancing the capture of strip-like defect characteristics, and this stream is denoted as F_1. The second stream is input to two fully connected layers that perform dimensionality expansion followed by reduction, enabling the modeling of complex feature correlations, and this stream is denoted as F_2. A total of four output feature representations are thus generated. Finally, 1D convolutions and group normalization (G_n) are applied to enhance horizontal and vertical positional information, yielding axis-specific spatial attention representations, as formulated in Equations (3) and (4).

$$\boldsymbol{y}_1^k = \sigma\left(G_n\left(F_k\left(F_1(\boldsymbol{z}_k)\right)\right)\right), \quad k \in \{h, w\} \tag{3}$$

$$\boldsymbol{y}_2^k = \sigma\left(G_n\left(F_k\left(F_2(\boldsymbol{z}_k)\right)\right)\right), \quad k \in \{h, w\} \tag{4}$$

On this basis, the two upper-layer outputs ($\boldsymbol{y}_1^h$ and $\boldsymbol{y}_1^w$) and two middle-layer outputs ($\boldsymbol{y}_2^h$ and $\boldsymbol{y}_2^w$) are element-wise multiplied by the original input features (denoted by $\odot$). This operation integrates spatial positional information and captures long-range feature dependencies, as formulated in (5) and (6), with the results denoted as $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ respectively. These outputs are then concatenated along the channel dimension. Finally, a CBS module (Convolution, Batch Normalisation, Sigmoid Linear Unit) is applied to enhance the extracted feature information, yielding the final output $\boldsymbol{y}_c$, as shown in (7).

$$\boldsymbol{X}_1 = \boldsymbol{x}_c \odot \boldsymbol{y}_1^h \odot \boldsymbol{y}_1^w \tag{5}$$

$$\boldsymbol{X}_2 = \boldsymbol{x}_c \odot \boldsymbol{y}_2^h \odot \boldsymbol{y}_2^w \tag{6}$$

$$\boldsymbol{y}_c = \mathrm{SiLU}\Big(\mathrm{BN}\big(\mathrm{Conv}(\mathrm{Concat}(\boldsymbol{X}_1, \boldsymbol{X}_2))\big)\Big) \tag{7}$$

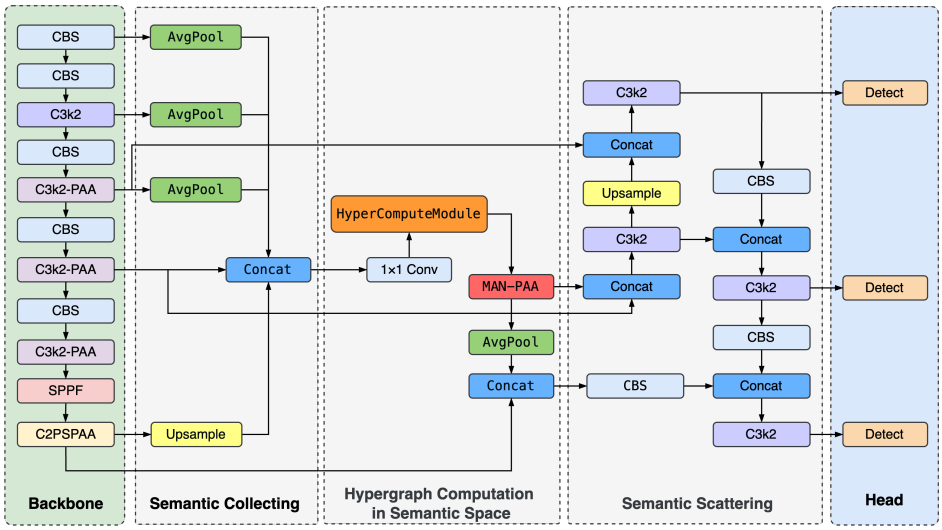

Fig. 1: PAA-YOLO Network Architecture.

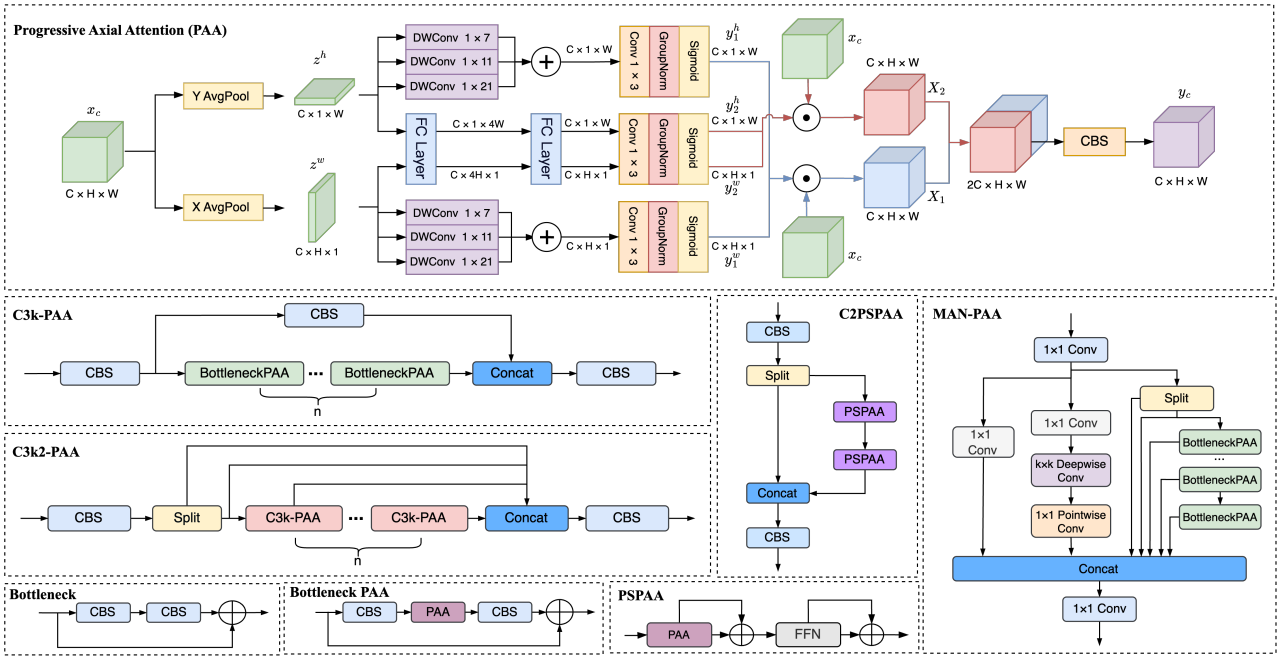

Fig. 2: Progressive Axial Attention and Its Derivative Modules.

B. Optimizing Backbone Networks Using PAA

To enhance the backbone network's steel surface defect feature extraction capability and improve its perception of linear features (e.g., scratches), we redesigned the C3k2 and C2PSA modules in the YOLOv11 backbone by integrating the proposed Progressive Axial Attention (PAA) mechanism, yielding the C3k2-PAA and C2PSPAA modules.

The architecture of the C3k2-PAA module is illustrated in Figure 2. Specifically, we reconstruct the Bottleneck unit of the C3k2 module by inserting the proposed PAA module between its two Convolutional Batch Normalisation (CBS) units to form the Bottleneck-PAA unit. This unit then replaces the C3k component in C3k2 to complete the C3k2-PAA construction. Compared with the original C3k2 module, the C3k2-PAA integrates the PAA mechanism to enable efficient feature representation and multi-scale information fusion, thereby boosting spatial positional information capture. Replacing the C3k2 module at the YOLOv11 backbone-neck junction with C3k2-PAA effectively enhances the model's long-range spatial dependency capture capability, elevating sensitivity to linear defects (e.g., scratches and cracks) while strengthening the backbone's feature extraction performance.

The architecture of the C2PSPAA module is illustrated in Figure 2. The YOLOv11 C2PSA module enhances feature representation by dynamically adjusting the importance of features at different positions via position-sensitive attention (PSA). In this paper, we replace the self-attention component in PSA with the proposed PAA to construct the position-sensitive progressive axial attention (PSPAA) module, thus deriving C2PSPAA. This module further improves the model's feature representation and spatial positional information capture capabilities, and enhances sensitivity to strip-like defects.

C. Optimizing the Neck Network Based on PAA and HGC-SCS

Existing detection models often overlook higher-order feature correlations and lack robust multi-scale, multi-location feature fusion capacity, leading to limited generalization and a tendency to miss subtle strip-like defects (e.g., scratches). To mitigate this, this paper enhances the YOLOv11 neck network by integrating the hypergraph computation-based semantic scattering collection framework (HGC-SCS) [19] with the proposed Progressive Axial Attention (PAA). An intermediate hypergraph computation layer is inserted between the backbone and neck networks, alongside the Progressive Axial Mixed Path Aggregation Network (MAN-PAA) module, to strengthen higher-order feature correlations and multi-scale fusion performance, as illustrated in Figure 1. HGC-SCS comprises three stages: semantic collection, hypergraph computation, and semantic scattering.

In the semantic collection stage, the backbone network is divided into five stages denoted as $\boldsymbol{X}_1$ to $\boldsymbol{X}_5$, which are channel-concatenated to form cross-layer visual features $\boldsymbol{X}_{mixed}$, as formulated in (8). For PAA-YOLO, the outputs of the 1st, 3rd, 5th, 7th and final backbone layers are designated as $\boldsymbol{X}_1$ to $\boldsymbol{X}_5$ respectively.

$$\boldsymbol{X}_{mixed} = \boldsymbol{X}_1 \parallel \boldsymbol{X}_2 \parallel \boldsymbol{X}_3 \parallel \boldsymbol{X}_4 \parallel \boldsymbol{X}_5 \tag{8}$$

In the hypergraph computation stage, a hypergraph is first constructed, followed by hypergraph convolution to capture complex higher-order relationships between base features. The hypergraph $\boldsymbol{G} = \{V, E\}$ is defined by the vertex set V and hyperedge set $\boldsymbol{E}$. In this paper, the feature points of the feature map form the hypergraph vertex set $\boldsymbol{V}$. The hyperedge set E is defined as $\boldsymbol{E} = \{\mathrm{ball}(v, \varepsilon) \mid v \in \boldsymbol{V}\}$, where $ball(v, \varepsilon) = \{\|x_u - x_v\|_d < \varepsilon \mid u \in \boldsymbol{V}\}$ denotes the neighborhood vertex set of vertex v, and $\|\mathrm{x_u} - \mathrm{x_v}\|_\mathrm{d}$ represents the distance function. A hypergraph $\boldsymbol{G}$ is typically represented by its adjacency matrix $\boldsymbol{H}$. For a hypergraph with n vertices $\boldsymbol{V} = \{v_1, v_2, \ldots, v_n\}$ and m hyperedges $\boldsymbol{E} = \{e_1, e_2, \ldots, e_m\}$, the n×m adjacency matrix $\boldsymbol{H}$ is defined as shown in (9).

$$\boldsymbol{H}_{i,j} = \begin{cases} 1, & \text{if } \mathrm{v}_i \in \mathrm{e}_j \\ 0, & \text{if } \mathrm{v}_i \notin \mathrm{e}_j \end{cases} \tag{9}$$

HyperConv performs high-order message passing and feature updates on hypergraphs. It essentially integrates high-order association information between vertices into feature representations via hypergraph topology, while boosting feature learning capability through residual connections. The HyperConv computation process is formulated in (10), where $\boldsymbol{X}$ denotes the original input feature matrix, H is the hypergraph adjacency matrix, $\boldsymbol{D}_v$ is the n×n vertex diagonal matrix with $\boldsymbol{D}_v(i,i) = \sum_{j=1}^{m} \boldsymbol{H}_{i,j}$, $\boldsymbol{D}_e$ is the m×m hyperedge diagonal matrix with $\boldsymbol{D}_e(j,j) = \sum_{i=1}^{n} \boldsymbol{H}_{i,j}$, and $\ominus$ represents the trainable parameter matrix.

$$HyperConv(\boldsymbol{X}, \boldsymbol{H}) = \boldsymbol{X} + \boldsymbol{D}_v^{-1}\boldsymbol{H}\boldsymbol{D}_e^{-1}\boldsymbol{H}^T\boldsymbol{X}\ominus \tag{10}$$

The hypergraph convolution output is fed into a Mixed-Path Aggregation Network (MAN), which fuses features via the synergistic operation of 1×1 bypass convolutions, depth-separable convolutions, and C2f modules. However, the C2f module tends to lose fine-grained spatial information, hindering full utilisation of hypergraph convolution features and thus compromising the detection of tiny strip-like defects. To address this issue, we retain the original two convolutional structures of MAN while replacing its Bottleneck module with the Bottleneck PAA module, constructing the Progressive Axial Mixed-Path Aggregation Network (MAN-PAA), as illustrated in Figure 2. By leveraging PAA's strength in capturing spatial long-range dependencies, the HGC-SCS framework's capability for strip-like feature extraction and fusion is enhanced.

In the semantic scattering stage, the original feature fusion mechanism is overly simplistic, suffering from low feature dimension alignment accuracy and potential spatial information loss. As shown in Figure 1, we perform channel concatenation between the hypergraph-computed high-order features and the backbone network outputs, before feeding the fused features into the PAN [20]+FPN [21] architecture. This enables precise feature fusion, thereby improving the model's generalisation capability.

IV. Experiments

A. Experimental Preparation

a) Dataset: The NEU-DET dataset, released by Northeastern University for metal surface defect detection, contains precise bounding box annotations. It consists of 1,800 200×200-pixel images covering six typical defect categories (e.g., cracks and scratches), with 300 samples per category. For experiments, the dataset was randomly split into a training set (1,440 images) and a validation set (360 images) at an 8:2 ratio.

The GC10-DET dataset is tailored for metal surface defect detection, comprising 2,292 2048×1000-pixel images across ten defect categories (e.g., stamping marks and weld seams). Its samples feature complex scenarios with multiple coexisting defects, enabling the verification of model robustness against multi-scale targets and intricate backgrounds. The dataset was also partitioned into training (1,833 images) and validation (459 images) sets following an 8:2 split.

b) Parameter Details: The model was trained and evaluated based on the PyTorch deep learning framework, with the experimental environment configured as follows: PyTorch 2.0.0, CUDA 11.8, Python 3.8, Xeon Platinum 8474C CPU, Nvidia GeForce RTX 4090D GPU (24GB VRAM), and Ubuntu 20.04 OS. For training, the SGD optimizer was adopted with 300 epochs, an initial learning rate of 0.01, a final learning rate scaling factor of 0.01, a batch size of 16, 8 worker threads, a weight decay of 0.0005, a momentum of 0.973, and a cosine annealing learning rate decay strategy. The input image resolution was set to 640×640. To fully validate the performance and generalization capability of the proposed method, training and testing were conducted on two public datasets: NEU-DET [22] and GC10-DET [23].

c) Experimental Indicators: To evaluate the performance of our model, we adopted Precision, Recall, and Mean Average Precision (mAP) as primary evaluation metrics. Additionally, we adopt parameters (Params) and frames per second (FPS) to evaluate model detection efficiency.

B. Experimental Results and Analysis

a) Comparison with Other Methods on NEU-DET: To fully assess the performance of the proposed method, we systematically compared PAA-YOLO against YOLO series algorithms, the classic two-stage detector Faster R-CNN, the single-stage baseline SSD, and state-of-the-art high-performance methods Hyper-YOLO and RT-DETR.

All methods were trained and tested under identical parameter configurations in the experiments. The detection results of PAA-YOLO and YOLO series algorithms on the NEU-DET dataset are presented in Figure 3. These visualizations include model-detected information such as prediction boxes, confidence scores, and defect categories. As shown in the figure, PAA-YOLO achieves the highest confidence scores for In and Sc-type elongated defects; it also delivers accurate localization for In defects despite complex background interference. This verifies that the Progressive Axial Attention (PAA) mechanism effectively boosts the model's sensitivity to elongated defects. Moreover, PAA-YOLO identifies other defect categories with high confidence and precision.

As shown in Table I, our method attains the highest mAP50 (81.0%) and joint-highest mAP50-90 (47.1%), while also ranking first in mAP50 for the In, Rs, and Sc categories. Versus the YOLOv11 baseline, it achieves a 3.1% mAP50 improvement across all categories. Despite a slight parameter increase, its detection FPS remains comparable. Compared with the recent SOTA method Hyper-YOLO, our approach gains an additional 0.8% in mAP50 with fewer parameters and higher FPS. Hence, it strikes a favorable balance between detection accuracy and efficiency, satisfying the requirements of practical industrial scenarios.

b) Comparison with Other Methods on GC10-DET: Practical industrial scenarios often involve images with multi-scale, multi-category defects. The GC10-DET dataset encompasses both single-category cases and complex multi-category defect coexistence scenarios, enabling rigorous validation of the proposed method's efficacy in real industrial settings. The detection results of our method on GC10-DET are illustrated in Figure 4, demonstrating its performance in single-category scenarios and robustness to multi-scale, multi-category defects. As observed from the figure, for single-category defects, the method accurately identifies minute point-like defects (e.g., In, Ss) while effectively suppressing complex background interference, thus enabling reliable detection of subtle strip-like defects (e.g., Os, Wl). In multi-scale, multi-category defect scenarios, it exhibits excellent generalization ability, precisely localizing defect targets of varying scales and categories, and reliably detecting complex mixed defects.

As shown in Table II, our method attains the highest mAP50 (65.0%) while also ranking first in mAP50 for the Ss, In, Rp, and Cr categories. Versus the YOLOv11 baseline, it boosts mAP50 by 3.3%. Relative to the recent SOTA method Hyper-YOLO, our approach delivers a higher overall mAP50 along with superior per-category mAP50 for most classes.

c) Ablation Experiment: To validate the efficacy of the C3k2-PAA, C2PSPAA modules, and HGC-SCS framework in PAA-YOLO, ablation experiments were performed on NEU-DET and GC10-DET datasets, with results presented in Tables 3 and 4. Tables III illustrate that each module enhances overall model performance. Specifically, integrating C3k2-PAA and C2PSPAA into the backbone markedly improves accuracy: on NEU-DET, C3k2-PAA boosts accuracy by 3.9% and mAP50 by 1.6%, while C2PSPAA achieves 3.7% and 1.4% gains respectively; on GC10-DET, C3k2-PAA increases accuracy by 1.6% and mAP50 by 2.0%, with C2PSPAA improving them by 1.4% and 1.0%. This confirms the Progressive Axial Attention (PAA) mechanism effectively captures spatial positional information and strengthens backbone feature extraction. Furthermore, the HGC-SCS framework fuses multi-scale features and enhances high-order feature correlations. Its integration into the neck network notably improves recall: on NEU-DET, recall and mAP50 rise by 1.5% and 1.4%. On GC10-DET, by 2.4% and 1.6%. Combining the C3k2-PAA and C2PSPAA modules further boosts the model's accuracy, while the integration of these two modules with the HGC-SCS module improves both the precision and recall of the model against the baseline.

Integrating all three modules delivers substantial gains: on NEU-DET, precision, recall, and mAP50 improve by 4.6%, 4.0%, and 3.1%. On GC10-DET, by 8.0%, 2.9%, and 3.3% respectively. This validates their synergistic effect in enhancing model robustness and generalization for detecting tiny, dense, and morphologically diverse steel surface defects in complex industrial environments.

V. Conclusion

This paper proposes PAA-YOLO, a YOLOv11-based model for industrial steel surface defect detection, with key innovations to boost performance. First, to enhance spatial feature extraction and sensitivity to strip-like defects (e.g., scratches), we propose the Progressive Axial Attention (PAA) module for capturing cross-spatial long-range dependencies. Second, using PAA, we reconstruct the backbone's C3k2 and

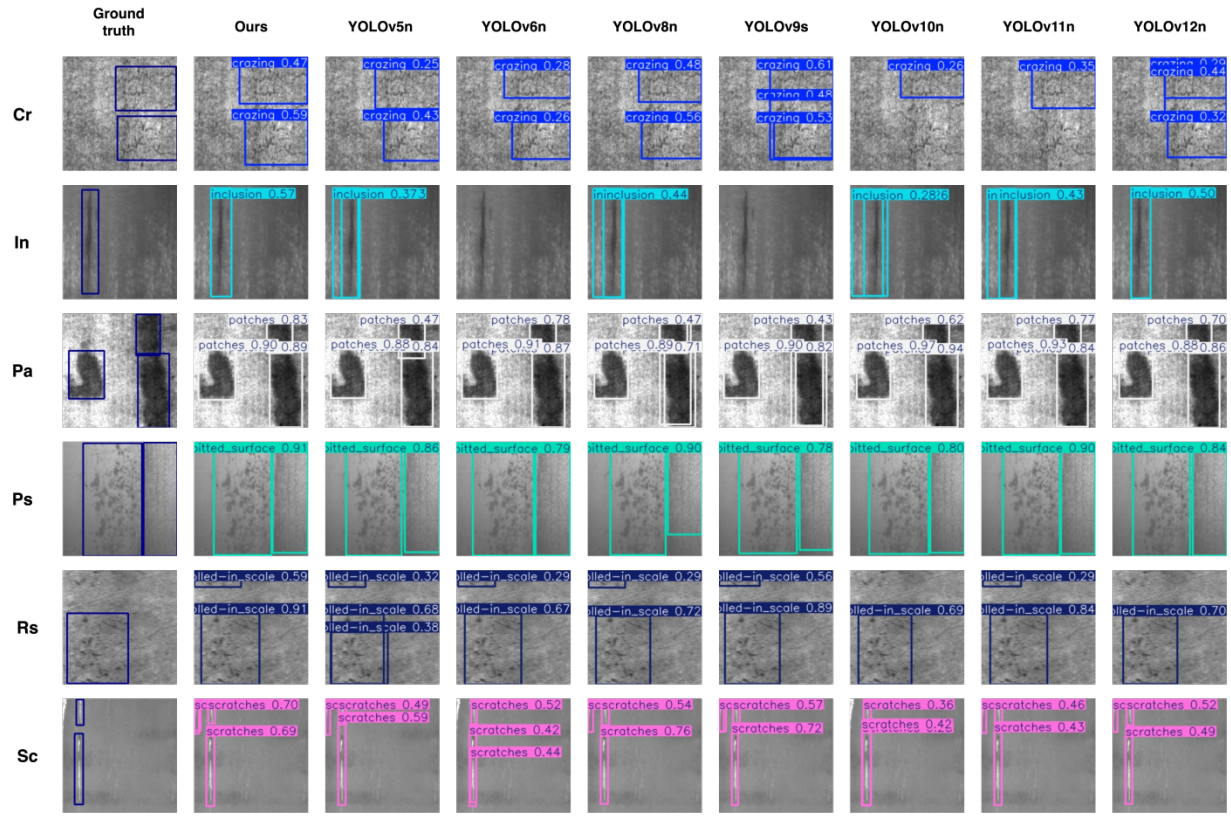

Fig. 3: Comparison of Detection Results on the NEU-DET Dataset.

TABLE I: Comparative Experiments on The NEU-DET Dataset

Methods	mAP50(%) ↑						mAP50(%)	mAP50-95(%)	Params(M) ↓	FPS ↑
	Cr	In	Pa	Ps	Rs	Sc				
YOLOv5n [7]	39.0	87.0	94.5	91.9	59.0	94.3	77.6	45.7	2.504	454.5
YOLOv6n [8]	43.1	85.5	94.4	**94.1**	58.5	93.0	78.1	45.0	4.234	526.3
YOLOv8n [9]	44.0	86.7	93.9	91.2	57.2	95.2	78.1	46.0	3.006	512.8
YOLOv9s [10]	42.0	85.2	94.7	89.6	65.6	94.8	79.0	46.9	7.169	322.5
YOLOv10n [11]	39.4	81.5	91.6	89.8	54.7	88.8	74.3	43.6	2.696	217.4
YOLOv11n [12]	42.2	86.2	94.3	91.0	61.8	91.0	77.9	46.0	2.583	476.2
YOLOv12n [13]	42.7	85.1	95.2	89.8	58.5	94.2	77.6	45.8	2.557	448.4
Faster R-CNN [4]	47.2	81.9	94.1	91.8	63.0	95.3	79.1	46.5	137.099	11.6
SSD [5]	47.7	83.9	**95.2**	91.3	66.8	76.6	76.9	44.2	2.285	62.4
RT-DETR-l [14]	28.6	79.2	84.3	72.0	50.6	90.8	67.6	39.4	31.996	232.6
Hyper-YOLO [19]	**47.8**	87.5	94.7	92.2	64.3	94.4	80.2	47.1	3.943	344.8
Ours	45.4	**89.2**	94.5	92.8	**68.5**	**95.4**	**81.0**	**47.1**	3.714	434.7

TABLE II: Comparative Experiments on The GC10-DET Dataset

Methods	mAP50(%) ↑										mAP50(%)
	Pu	Wl	Cg	Ws	Os	Ss	In	Rp	Cr	Wf	
YOLOv5n [7]	86.7	33.3	54.9	69.2	80.5	28.9	52.1	10.1	93.0	78.6	58.7
YOLOv6n [8]	84.6	**53.4**	**57.1**	72.3	86.7	34.4	50.8	6.4	92.8	87.4	62.6
YOLOv8n [9]	90.5	26.4	54.8	68.8	84.0	33.9	52.2	11.1	95.1	84.5	60.1
YOLOv9s [10]	92.2	41.5	53.1	**73.9**	85.9	37.3	55.9	18.4	93.6	89.0	63.6
YOLOv10n [11]	87.4	32.6	54.1	68.3	84.3	34.4	51.9	9.3	94.4	81.1	59.8
YOLOv11n [12]	**92.7**	37.2	54.9	71.6	87.4	27.3	54.0	15.1	94.6	81.3	61.7
YOLOv12n [13]	87.7	30.8	50.5	69.2	82.5	22.4	48.2	15.2	92.5	75.8	57.5
Faster R-CNN [4]	82.5	14.5	43.3	66.9	43.2	8.9	39.5	14.2	68.3	48.3	43.0
SSD [5]	88.7	20.9	48.2	63.1	77.8	8.1	38.5	5.7	73.3	91.7	51.6
RT-DETR-l [14]	81.7	33.1	45.2	64.9	**93.5**	28.2	50.1	10.9	89.8	49.4	54.7
Hyper-YOLO [19]	87.9	39.5	54.7	73.0	87.7	36.0	54.6	20.7	93.9	**90.7**	63.9
Ours	89.1	43.3	55.8	73.4	83.4	**37.5**	**59.0**	**24.7**	**95.8**	88.9	**65.0**

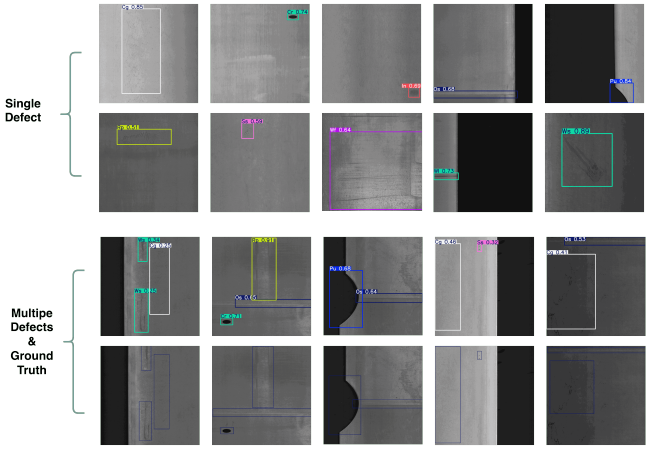

Fig. 4: Detection Results on the GC10-DET Dataset.

TABLE III: Ablation Experiments on NEU-DET and GC10-DET Datasets

C3k2	C2PSP	HGC-SCS	P(%)	R(%)	mAP50(%)
NEU-DET Dataset					
			71.5	73.6	77.9
✓			75.4	74.3	79.5
	✓		75.2	72.4	79.3
		✓	73.9	75.1	79.3
✓	✓		75.8	73.9	80.1
	✓	✓	74.8	73.7	79.4
✓		✓	74.7	77.4	80.5
✓	✓	✓	76.1	77.6	81.0
GC10-DET Dataset					
			62.4	61.3	61.7
✓			64.0	61.8	63.7
	✓		63.8	61.9	62.7
		✓	58.5	63.7	63.3
✓	✓		67.6	64.9	64.2
	✓	✓	67.6	63.2	64.5
✓		✓	63.4	67.5	64.8
✓	✓	✓	70.4	64.2	65.0

C2PSA modules into C3k2-PAA and C2PSPAA, strengthening the backbone's feature extraction and object sensitivity, and improving linear defect detection. Finally, we optimize the neck network via HGC-SCS and PAA, inserting a hypergraph computation intermediate layer between backbone and neck to strengthen high-order feature correlations, enhance multi-scale/multi-location feature fusion, and improve generalization.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62572218, and in part by the Basic Research Program of Jiangsu Province under Grant BK20252080.

References

[1] J. Wang, G. Yi, S. Wan, and J. Jiang, "Recent advances in the mechanism of oxide scale structure evolution and application-oriented control technologies on steel surfaces," *Journal of Chinese Society for Corrosion and Protection*, vol. 43, no. 5, pp. 948–956, 2023.

[2] X. Tao, W. Hou, and D. Xu, "A review on deep learning-based surface defect detection methods," *Acta Automatica Sinica*, vol. 47, no. 5, p. 18, 2021.

[3] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[7] Ultralytics, "Yolov5: State-of-the-art object detection," https://github.com/ultralytics/yolov5, 2020.

[8] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[9] Ultralytics, "Yolov8: State-of-the-art object detection, segmentation, and pose estimation," https://github.com/ultralytics/ultralytics, 2023.

[10] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European conference on computer vision*. Springer, 2024, pp. 1–21.

[11] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han *et al.*, "Yolov10: Real-time end-to-end object detection," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 984–108 011, 2024.

[12] Ultralytics, "Yolov11: State-of-the-art object detection, segmentation, pose estimation, and tracking," https://github.com/ultralytics/ultralytics, 2024.

[13] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," *arXiv preprint arXiv:2502.12524*, 2025.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[15] R. Ma, J. Chen, Y. Feng, Z. Zhou, and J. Xie, "Ela-yolo: An efficient method with linear attention for steel surface defect detection during manufacturing," *Advanced Engineering Informatics*, vol. 65, p. 103377, 2025.

[16] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.

[17] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognition*, vol. 110, p. 107637, 2021.

[18] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," *Advances in neural information processing systems*, vol. 32, 2019.

[19] Y. Feng, J. Huang, S. Du, S. Ying, J.-H. Yong, Y. Li, G. Ding, R. Ji, and Y. Gao, "Hyper-yolo: When visual object detection meets hypergraph computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2388–2401, 2024.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[22] Z. Li, Y. Li, J. Zhang, and H. Liu, "Neu-det: A real-world dataset for object detection in unconstrained environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1–8.

[23] Y. Wang, L. Chen, and Q. Zhang, "Gc10-det: A large-scale dataset for object detection in complex urban scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 1234–1243.