Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A PCB Surface Defect Detection Algorithm Integrating Lightweight Coordinate Channel Attention and Multi-Scale Receptive Field Enhancement

Liya Li

LiLiYa796@outlook.com

Taiyuan Institute of Technology

Fuguo Wu

Shanxi Aitplus Technology Co., Ltd.

Haijing Liu

Taiyuan Institute of Technology

Research Article

Keywords: PCB surface defect detection, lightweight network, coordinate channel attention, multi-scale receptive field, object detection

Posted Date: June 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9806864/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A PCB Surface Defect Detection Algorithm Integrating Lightweight Coordinate Channel Attention and Multi-Scale Receptive Field Enhancement

Liya Li[1] *, Fuguo Wu[2] , Haijing Liu[1]
([1] Department of Computer Engineering, Taiyuan Institute of Technology, Taiyuan 030008, China;
[2] Shanxi Aitplus Technology Co., Ltd., Taiyuan 030006, China)
Corresponding email: LiLiYa796@outlook.com

Abstract: Printed circuit board surface defect detection is a critical task in electronic manufacturing, where defects such as missing holes, open circuits, short circuits, mouse bites, spurs, and spurious copper are usually small, irregular, and easily confused with dense circuit textures. To improve detection accuracy while maintaining deployment efficiency, a PCB surface defect detection algorithm integrating lightweight coordinate channel attention and multi-scale receptive-field enhancement is proposed. The model adopts a lightweight backbone to reduce redundant computation, introduces the LCCA module to enhance coordinate-sensitive channel responses, and designs the MRFE module to strengthen multi-scale contextual representation for small and dense defects. A lightweight feature fusion structure is further used to balance semantic transmission and localization detail preservation. Experiments were conducted on a public PCB defect dataset containing 10,668 images, including 5,120 training images, 3,414 validation images, and 2,134 testing images. The proposed model achieved a peak F1-score of 0.958 and a P--R curve area of 0.9583. Compared with the baseline, the full model improved mAP from 95.25% to 97.68%, increased inference speed from 63.62 FPS to 66.40 FPS, and reduced model size from 153.0 MB to 101.3 MB. The results demonstrate that the proposed method provides a favorable balance between detection accuracy, lightweight design, and real-time performance.

Keyword: PCB surface defect detection; lightweight network; coordinate channel attention; multi-scale receptive field; object detection

1. Introduction

Printed circuit boards (PCBs) are the fundamental carriers of electronic components and electrical interconnections, and their surface quality directly affects the reliability of electronic products. During drilling, etching, soldering, transportation, and assembly, PCB surfaces may suffer from missing holes, open circuits, short circuits, mouse bites, spurs, and spurious copper. These defects are usually small in scale, irregular in shape, and easily confused with normal copper traces, pads, vias, and silk-screen regions. In practical industrial inspection, defect detection is further complicated by uneven illumination, dense circuit layouts, background texture interference, and the coexistence of multiple defect categories in a single image. Therefore, automatic PCB defect detection needs not only high recognition accuracy but also low computational cost and stable inference speed, especially when deployed on resource-limited industrial edge devices.

Traditional PCB defect inspection methods mainly rely on template matching, threshold segmentation, morphological operators, or handcrafted feature extraction. Although these methods are intuitive and computationally efficient, their robustness strongly depends on illumination conditions, image registration accuracy, and manually designed rules. With the development of

deep learning, object detection methods have become the dominant technical route for PCB surface defect detection. For example, Tang et al. proposed PCB-YOLO by improving YOLOv5 for PCB surface defect localization, showing that anchor optimization and small-target enhancement can improve PCB defect detection performance [1]. Du et al. further introduced YOLO-MBBi, in which enhanced convolutional blocks and feature optimization strategies were used to improve detection accuracy for surface defects with small spatial extent [2]. These studies indicate that one-stage detectors are more suitable for PCB inspection than conventional two-stage pipelines when real-time performance is required.

Recent studies have attempted to improve PCB defect detection from three major perspectives: feature enhancement, attention mechanism design, and lightweight network construction. Yuan et al. proposed YOLO-HMC, which strengthened hierarchical feature representation for PCB surface defect detection and demonstrated the importance of multi-level semantic modeling [3]. Zhao et al. investigated attention and multi-source feature fusion for fast PCB defect detection, showing that spatially adaptive feature integration can help suppress redundant background responses [4]. Zhang et al. designed a lightweight PCB surface defect detection algorithm, suggesting that reducing redundant convolutional operations is essential for practical deployment [5]. Gao et al. proposed YOLOv5_ES with a lightweight small-object detection head, emphasizing that small defect sensitivity and computational efficiency should be jointly considered [6]. These works collectively show that PCB defect detection has shifted from simply improving detection accuracy to balancing accuracy, speed, and model complexity.

In addition to YOLO-based improvements, recent PCB inspection studies have also explored semantic enhancement, global context modeling, and specialized feature aggregation. Yan et al. studied deep learning model enhancements for PCB surface defect detection and showed that feature-level optimization can improve the detector's ability to distinguish subtle local anomalies [7]. Wei et al. proposed PCB-YOLO with coordinate attention and multi-scale feature fusion, which further verified that directional position encoding is effective for defects distributed along circuit traces [8]. Bai et al. improved PCB defect detection through a refined detection scheme, highlighting the importance of adapting detection heads and feature fusion strategies to PCB-specific visual patterns [9]. Zhu and Zhao introduced separated global context attention to guide residual context aggregation, demonstrating that global semantic guidance can improve defect localization in cluttered PCB backgrounds [10]. These methods have improved the representation capability of PCB detectors, but many of them introduce additional branches or complex aggregation modules, which may increase inference latency and restrict edge deployment.

Another line of research focuses on lightweight and efficient PCB defect detection. Shen et al. developed a YOLOv5-based detection method for printed circuit board assembly defects and provided a useful reference for real industrial PCBA inspection scenarios [11]. Lang and Lv proposed SEPDNet as a simple and effective PCB surface defect detection method, showing that compact network design can still achieve competitive detection results [12]. Chen et al. designed an improved YOLOv10-based method with multiscale edge strengthening for small PCB defect detection, further confirming the value of edge and local detail enhancement in detecting tiny defects [13]. These studies show that lightweight PCB detection is not only a matter of reducing parameter size; the detector must also retain sufficient sensitivity to small-scale defects and complex local structures.

Meanwhile, general object detection research has provided important technical support for PCB defect detection. YOLOv7 improved the trade-off between accuracy and speed by optimizing network scaling and training strategies [14]. RT-DETR introduced an end-to-end real-time detection paradigm and showed that transformer-based detection can achieve competitive speed under proper architectural design [15]. YOLOv9 explored programmable gradient information to improve feature learning efficiency [16], while YOLOv10 further advanced real-time detection by reducing post-processing dependency and improving the efficiency of the detection pipeline [17]. These general detectors provide useful design inspiration, but they are not directly optimized for PCB surface defects. PCB defects are usually smaller, more densely distributed, and more easily confused with normal circuit structures than objects in natural scenes. Therefore, directly applying general-purpose detectors may lead to missed detection of small defects, false alarms in complex textures, or excessive computational cost.

Despite the progress above, several limitations remain. First, existing attention-based PCB detectors often use standard channel or spatial attention modules, which improve representation capability but may introduce redundant parameters and fail to fully exploit the directional characteristics of PCB traces. Second, many lightweight detectors reduce computational cost by simplifying the backbone, but excessive compression may weaken the extraction of fine-grained defect features. Third, multi-scale fusion structures are frequently adopted to improve small-object detection, yet their receptive field design is not always matched with PCB defect morphology. Defects such as open circuits and mouse bites require local detail preservation, while spurious copper and dense short-circuit defects require broader contextual perception. Fourth, some current methods focus on accuracy improvement but provide insufficient analysis of model size, inference speed, and robustness under illumination variation or dense-defect conditions. These limitations indicate that PCB defect detection still requires a balanced design that combines lightweight feature extraction, location-aware attention, and multi-scale receptive field enhancement.

To address these issues, a PCB surface defect detection algorithm integrating lightweight coordinate channel attention and multi-scale receptive field enhancement is proposed. The main idea is to construct a compact detection framework that preserves the local sensitivity required for tiny defects while improving the contextual discrimination ability needed for complex PCB backgrounds. A lightweight coordinate channel attention module is introduced into the backbone to encode long-range directional information with reduced channel transformation cost, so that defect-related responses along horizontal and vertical circuit structures can be emphasized without significantly increasing computational burden. A multi-scale receptive field enhancement module is further designed to aggregate local and contextual features through parallel receptive fields, thereby improving the detection of defects with different sizes and shapes. In addition, a lightweight feature fusion structure is adopted to balance semantic propagation and spatial detail preservation, supporting accurate multi-scale prediction with practical inference efficiency.

The main contributions of this study are summarized as follows. First, a lightweight PCB surface defect detection framework is constructed for industrial inspection scenarios, aiming to balance detection accuracy, model size, and inference speed. Second, a lightweight coordinate channel attention module is designed to enhance defect-sensitive directional features while limiting additional parameter overhead. Third, a multi-scale receptive field enhancement structure is introduced to strengthen the representation of small, dense, and morphologically diverse PCB defects. Fourth, comprehensive experiments are conducted from mainstream detector comparison,

ablation analysis, and robustness evaluation, thereby verifying the effectiveness of the proposed design under complex backgrounds, small-target defects, dense defects, and illumination variations.

2. Proposed PCB Surface Defect Detection Algorithm

2.1 Overall Network Framework

PCB surface defects usually appear as small, sparse, and locally irregular regions on a highly repetitive circuit background. In such images, copper traces, solder joints, vias, pads, and silk-screen textures may produce strong non-defective responses, whereas true defects such as mouse bite, spur, open circuit, and spurious copper often occupy only a limited number of pixels[18-20]. A heavy generic detector can provide sufficient semantic representation, but its parameter redundancy and high computational cost are not consistent with real-time PCB inspection. Conversely, excessive backbone simplification may weaken the spatial response of small defects. Therefore, the proposed detector is designed under two constraints: maintaining small-defect feature sensitivity and reducing redundant computation.

The proposed architecture consists of a lightweight backbone, a lightweight coordinate channel attention module (LCCA), a multi-scale receptive-field enhancement module (MRFE), a bidirectional lightweight fusion neck, and a three-scale anchor-based decoupled detection head. The overall architecture is presented in Fig.1. The overall detection paradigm follows a one-stage YOLO-style framework [21], but the backbone, attention module, receptive-field enhancement module, and feature fusion structure are redesigned for lightweight PCB defect detection.

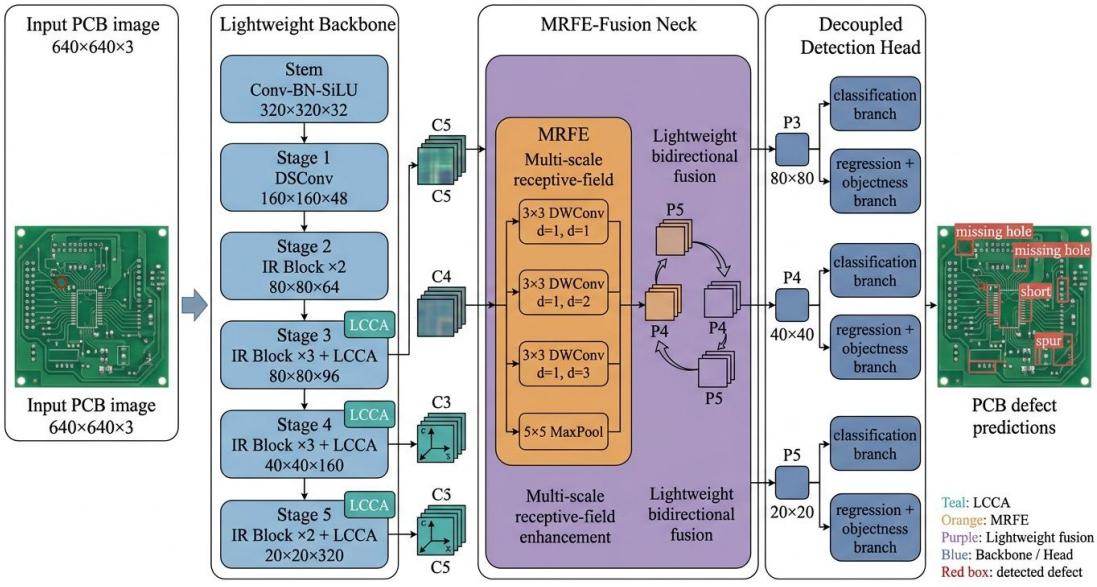


Fig. 1. Overall structure of the proposed PCB surface defect detection algorithm

To avoid functional overlap with Fig.1, Fig.2 focuses only on the feature transmission relationship among the backbone, neck, and detection head. The low-level feature map preserves edge and contour details, while the high-level feature map provides semantic discrimination. The neck first performs top-down semantic transmission and then conducts bottom-up localization reinforcement.

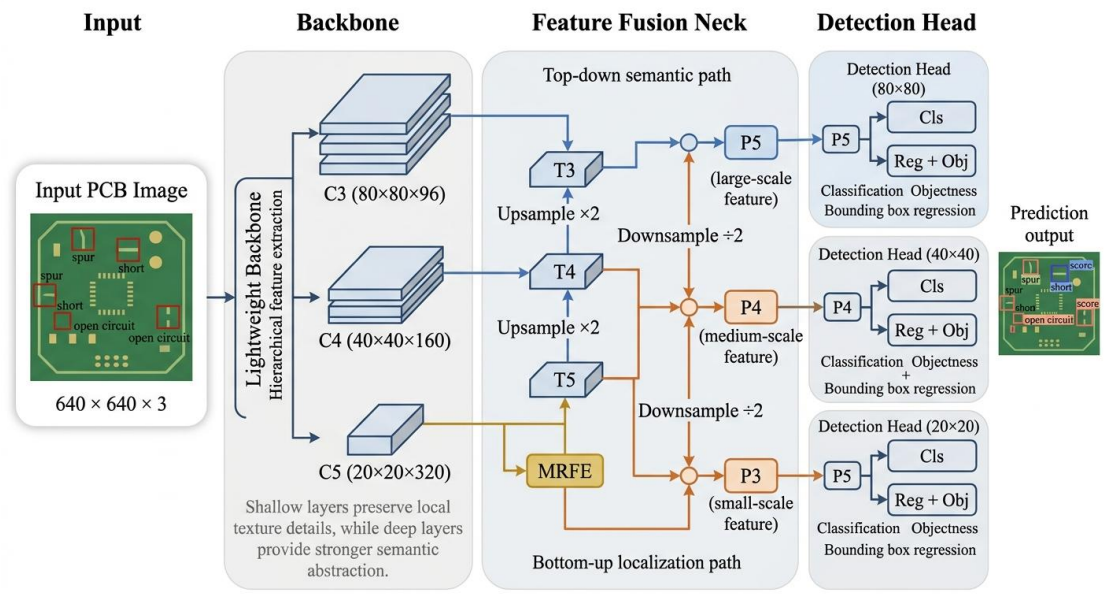


Fig. 2. Feature data flow among the backbone, feature fusion neck, and detection head

Let the input PCB image be denoted as $\mathbf{I}\in\mathbb{R}^{640\times640\times3}$, The lightweight backbone extracts three feature maps with strides of 8, 16, and 32:

$$\{C_3,C_4,C_5\}=\mathscr{B}(\mathbf{1};\Theta_b) \tag{1}$$

Where $\mathbf{C}_3\in\mathbb{R}^{80\times80\times96}$, $\mathbf{C}_4\in\mathbb{R}^{40\times40\times160}$, and $\mathbf{C}_5\in\mathbb{R}^{20\times20\times320}$. The neck generates fused feature maps:

$$\{\mathbf{P}_3,\mathbf{P}_4,\mathbf{P}_5\}=N(\mathbf{C}_3,\mathbf{C}_4,\mathbf{C}_5;\Theta_n) \tag{2}$$

The detection head predicts bounding-box regression, objectness, and class probability at the three scales:

$$\{\mathbf{Y}_3,\mathbf{Y}_4,\mathbf{Y}_5\}=\mathscr{H}(\mathbf{P}_3,\mathbf{P}_4,\mathbf{P}_5;\Theta_h) \tag{3}$$

The detailed stage configuration is listed in Table 1. LCCA is not inserted into all backbone stages. It is used only after Stage~3, Stage~4, and Stage~5, where feature maps have stronger semantic responses and lower spatial resolution. This insertion strategy reduces the attention overhead in high-resolution shallow layers and keeps the early texture extraction path simple.

Table 1 Stage-level configuration of the proposed lightweight detection model

Stage	Operator	Kernel	Channels	Repeat	Stride	Output size
Input	Image normalization	--	3	--	--	640×640×3
Stem	Conv-BN-SiLU	3×3	32	1	2	320×320×32
Stage 1	DSConv-BN-SiLU	3×3	48	1	2	160×160×48
Stage 2	Inverted residual block	3×3	64	2	2	80×80×64
Stage 3	Inverted residual block + LCCA	3×3	96	3	1	80×80×96
Stage 4	Inverted residual block + LCCA	5×5	160	3	2	40×40×160
Stage 5	Inverted residual block + LCCA	5×5	320	2	2	20×20×320
Neck-1	MRFE + top-down fusion	mixed	160 / 96	--	--	40×40,

Neck-2	MRFE + bottom-up fusion	mixed	160 / 320	--	--	80×80 40×40, 20×20
Head	Anchor-based decoupled head	1×1, 3×3	--	--	--	80^2, 40^2, 20^2

For the complete backbone, the number of parameters and FLOPs are calculated as

$$Params_{\mathcal{B}}=\sum_{l=1}^{L}\left(k_l^2 L_l^{in} g_l^{-1} C_l^{out}+C_l^{out}\right) \tag{4}$$

$$\mathrm{FLOPs}_{\mathcal{B}}=\sum_{l=1}^{L} H_l W_l\left(k_l^2 C_l^{in} g_l^{-1} C_l^{out}\right) \tag{5}$$

In the experimental section, the backbone-level and full-model complexity should be reported together with detection accuracy. The profiling items include Params_B, FLOPs_B, total parameters, total FLOPs, and FPS, so that the lightweight conclusion is supported by quantitative measurement rather than module-level complexity analysis alone.

2.2 Lightweight Backbone and Coordinate Channel Attention Module

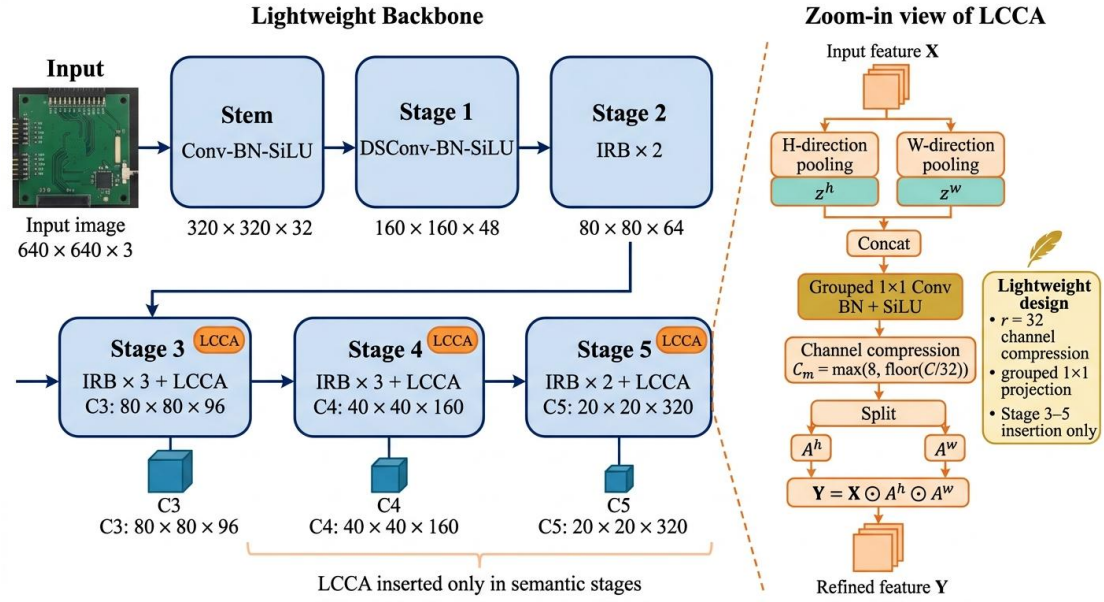


Fig. 3. Lightweight backbone and LCCA insertion positions

The lightweight backbone is constructed from depthwise separable convolution, inverted residual blocks, and pointwise convolution, following the design principle of mobile convolutional networks [22-23], as shown in Fig.3. A depthwise separable convolution decomposes standard convolution into depthwise spatial filtering and pointwise channel mixing. For a feature tensor $\mathbf{X}\in\mathbb{R}^{H\times W\times C}$ and C' output channels, the computational cost of standard convolution is

$$\Omega_{\mathrm{std}}=HWCC'k^2, \tag{6}$$

whereas depthwise separable convolution requires

$$\Omega_{\mathrm{ds}}=HWCk^2+HWCC' \tag{7}$$

The theoretical reduction ratio is therefore

$$r_{\Omega}=\frac{\Omega_{\mathrm{ds}}}{\Omega_{\mathrm{std}}}=\frac{1}{C'}+\frac{1}{k^2} \tag{8}$$

This reduction provides the computational margin for inserting attention and receptive-field enhancement modules.

However, a lightweight backbone alone may suppress weak defect features, especially after repeated down-sampling. LCCA is introduced to compensate for this loss, but its design is intentionally different from the conventional coordinate attention mechanism [24]. The lightweight property of LCCA is reflected in three aspects. First, the intermediate channel is compressed by a larger reduction ratio r=32 and its lower bound is constrained by $C_m=\max(8,\lfloor C/r \rfloor)$. Second, the transformation after coordinate pooling uses grouped 1×1 convolution rather than a full dense 1×1 projection when $C\geq 128$. Third, LCCA is inserted only into Stage~3--Stage~5 rather than all backbone stages, avoiding additional computation on high-resolution shallow feature maps. Compared with SE, CBAM, ECA, and standard coordinate attention, this design retains coordinate-sensitive channel recalibration while reducing the number of attention operations [25-27].

Given an input feature map $\mathbf{X}\in\mathbb{R}^{H\times W\times C}$, LCCA first performs two directional pooling operations:

$$\mathbf{z}_c^{\mathrm{h}}(\mathrm{h})=\frac{1}{W}\sum_{w=1}^{W}\mathbf{X}_c(\mathrm{h},w),\quad \mathbf{z}_c^{w}(w)=\frac{1}{H}\sum_{\mathrm{h}=1}^{H}\mathbf{X}_c(\mathrm{h},w) \tag{9}$$

The two coordinate descriptors are concatenated and compressed:

$$\mathbf{F}=\delta\left(\mathrm{BN}\left(\mathrm{GConv}_{1\times 1}\left(\left[\mathbf{z}^{\mathrm{h}},\mathbf{z}^{w}\right]\right)\right)\right),\quad \mathbf{F}\in\mathbb{R}^{(H+W)\times 1\times C_m} \tag{10}$$

After splitting $\mathbf{F}$ into horizontal and vertical components, two lightweight projections generate direction-aware attention maps:

$$\begin{aligned}\mathbf{A}^{w}&=\sigma\left(Conv_{1\times 1}^{w}\left(\mathbf{F}^{w}\right)\right)\\ \mathbf{A}^{\mathrm{h}}&=\sigma\left(Conv_{1\times 1}^{\mathrm{h}}\left(\mathbf{F}^{\mathrm{h}}\right)\right),\end{aligned} \tag{11}$$

The refined feature is obtained by

$$\mathbf{Y}_c(\mathrm{h},w)=\mathbf{X}_c(\mathrm{h},w)\cdot\mathbf{A}_c^{\mathrm{h}}(\mathrm{h})\cdot\mathbf{A}_c^{w}(w). \tag{12}$$

Compared with ordinary channel attention, LCCA preserves one-dimensional coordinate information during channel recalibration. Compared with standard coordinate attention, LCCA reduces the intermediate channel width, uses grouped projection in high-channel stages, and restricts the insertion positions. These design choices are intended to improve small-defect response under a limited computational budget. The complete computational procedure of LCCA is summarized in Algorithm 1.

Algorithm 1 Lightweight coordinate channel attention module

Require: Feature tensor $\mathbf{X}\in\mathbb{R}^{H\times W\times C}$, reduction ratio $r=32$
Ensure: Refined feature tensor $\mathbf{Y}$
1: Set compressed channel number $C_m=\max(8,\lfloor C/r\rfloor)$
2: Generate horizontal descriptor $\mathbf{z}^h$ by average pooling along width
3: Generate vertical descriptor $\mathbf{z}^w$ by average pooling along height
4: Concatenate $\mathbf{z}^h$ and $\mathbf{z}^w$ along the coordinate dimension
5: Apply grouped 1×1 convolution, batch normalization, and SiLU activation to obtain $\mathbf{F}$
6: Split $\mathbf{F}$ into $\mathbf{F}^h$ and $\mathbf{F}^w$
7: Generate $\mathbf{A}^h$ and $\mathbf{A}^w$ using two lightweight 1×1 projections
8: Reweight $\mathbf{X}$ by $\mathbf{A}^h$ and $\mathbf{A}^w$
9: **return** $\mathbf{Y}$

2.3 Multi-scale Receptive-field Enhancement and Feature Fusion Structure

Small PCB defects require local boundary information, but their semantic judgment often depends on a larger surrounding context. For example, a short circuit is not determined only by a bright local region; it must be identified according to its relation with adjacent copper traces. Similar

small-target perception problems have also been reported in recent PCB defect detection studies [28]. Simply increasing the convolution kernel size would increase parameters and FLOPs. Therefore, MRFE is designed as a four-branch lightweight module that combines local convolution, dilated depthwise convolution, and pooled context.

Given an input feature map $\mathbf{X}$, MRFE contains four parallel branches:

$$\mathbf{R}_1=\mathrm{PWConv}\left(\mathrm{DWConv}_{3\times3,d=1}(\mathbf{X})\right) \tag{13}$$

$$\mathbf{R}_2=\mathrm{PWConv}\left(\mathrm{DWConv}_{3\times3,d=2}(\mathbf{X})\right) \tag{14}$$

$$\mathbf{R}_3=\mathrm{PWConv}\left(\mathrm{DWConv}_{3\times3,d=3}(\mathbf{X})\right) \tag{15}$$

$$\mathbf{R}_4=\mathrm{PWConv}\left(\mathit{MaxPool}_{5\times5,s=1}(\mathbf{X})\right) \tag{16}$$

The first branch preserves local detail, the second and third branches enlarge the receptive field through dilation rates of 2 and 3, and the fourth branch introduces pooled contextual information without increasing convolution kernel size. The four branches are concatenated and compressed:

$$\mathbf{R}=\mathit{Conv}_{1\times1}(\mathit{Concat}[\mathbf{R}_1,\mathbf{R}_2,\mathbf{R}_3,\mathbf{R}_4]) \tag{17}$$

The output is generated through residual scaling:

$$\mathbf{X}'=\mathbf{X}+\gamma\mathbf{R} \tag{18}$$

Here, γ is initialized as 0.1 and optimized as a learnable scalar during training. A small initial value prevents the multi-scale branch from disturbing the original backbone feature at the beginning of training, while the learnable setting allows the network to adjust the contribution of contextual enhancement.

The overall forward process of MRFE, including branch construction, feature concatenation, channel compression, and residual scaling, is summarized in Algorithm 2.

Algorithm 2 Multi-scale receptive-field enhancement module

Require: Feature tensor $\mathbf{X}$, learnable residual scale γ
Ensure: Enhanced feature tensor $\mathbf{X}'$

Generate $\mathbf{R}_1$ using 3×3 depthwise convolution with dilation rate 1
Generate $\mathbf{R}_2$ using 3×3 depthwise convolution with dilation rate 2
Generate $\mathbf{R}_3$ using 3×3 depthwise convolution with dilation rate 3
Generate $\mathbf{R}_4$ using 5×5 max-pooling with stride 1
Apply 1×1 pointwise convolution to each branch
Concatenate $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$, and $\mathbf{R}_4$
Compress the concatenated feature using 1×1 convolution
Compute $\mathbf{X}' = \mathbf{X} + \gamma\mathbf{R}$
return $\mathbf{X}'$

The feature fusion neck is divided into two explicit stages. The top-down path transmits high-level semantic information, following the basic semantic propagation mechanism of FPN [29]:

$$\mathbf{T}_5=\mathrm{MRFE}(\mathbf{C}_5) \tag{19}$$

$$\mathbf{T}_4=\psi_4^{td}(\mathit{Concat}[\mathrm{C}_4,\mathrm{Up}(\mathrm{T}_5)]) \tag{20}$$

$$\mathbf{T}_3=\psi_3^{id}(\mathit{Concat}[\mathrm{C}_3,\mathrm{Up}(\mathrm{T}_4)]) \tag{21}$$

The bottom-up path then reinforces localization information, which is related to the path aggregation idea used in PANet [30]:

$$\mathbf{P}_3=\psi_3^{bu}(\mathbf{T}_3), \tag{22}$$

$$\mathbf{P}_4=\psi_4^{bu}(\mathit{Concat}[\mathbf{T}_4,\mathit{Down}(\mathbf{P}_3)]). \tag{23}$$

$$\mathbf{P}_5=\psi_5^{bu}(\mathit{Concat}[\mathrm{T}_5,\mathit{Down}(\mathbf{P}_4)]) \tag{24}$$

This formulation removes the possible circular definition between $\mathbf{P}_{l+1}$ and $\mathbf{P}_{l-1}$. The refinement functions $\psi^{td}(\cdot)$ and $\psi^{bu}(\cdot)$ are implemented by 1×1 channel compression followed by 3×3 depthwise separable convolution. In contrast to heavier bidirectional fusion structures such as

BiFPN [31], the proposed neck uses 1×1 channel compression and 3×3 depthwise separable convolution to reduce fusion cost.

The detection head adopts an anchor-based decoupled design. For each scale, the classification branch and regression branch are separated after a shared lightweight convolution. The regression branch predicts (x,y,w,h) and objectness, while the classification branch predicts the probability of each defect category. Three anchors are assigned to each scale according to the defect size distribution in the training set. This design follows the multi-scale anchor-based prediction mechanism used in YOLO-style detectors. The prediction tensor at scale s is

$$\mathbf{Y}_s \in \mathbb{R}^{H_s \times W_s \times A(5+K)} \tag{25}$$

For the PCB dataset with six defect categories, K=6 and A=3, so each grid produces 3×(5+6) prediction channels.

2.4 Loss Function and Training Strategy

The training objective contains bounding-box regression, objectness estimation, and defect classification. Although recent IoU variants can be used for further improvement [32-33], the default version of the proposed method adopts CIoU as the bounding-box regression loss to keep the optimization target consistent with the anchor-based detector [34]:

$$\mathscr{L}_{\text{total}} = \lambda_{\text{box}} \mathscr{L}_{\text{CIoU}} + \lambda_{\text{obj}} \mathscr{L}_{\text{obj}} + \lambda_{\text{cls}} \mathscr{L}_{\text{cls}} \tag{26}$$

The CIoU loss is defined as

$$\mathscr{L}_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v, \tag{27}$$

where $\rho^2(\mathbf{b}, \mathbf{b}^{gt})$ denotes the squared Euclidean distance between the predicted box center and the ground-truth box center, c denotes the diagonal length of the smallest enclosing box, v measures the aspect-ratio consistency, and α is the trade-off coefficient \cite{zheng2020ciou}. The aspect-ratio term is calculated by

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{\text{h}^{gt}} - \arctan \frac{w}{\text{h}} \right)^2 \tag{28}$$

$$\alpha = \frac{v}{1 - \text{IoU} + v} \tag{29}$$

The objectness branch uses binary cross-entropy, which is commonly used in one-stage anchor-based detection frameworks :

$$\mathscr{L}_{\text{obj}} = -\sum_{i=1}^{N} \omega_i \left[q_i \log(\hat{q}_i) + (1 - q_i) \log(1 - \hat{q}_i) \right] \tag{30}$$

The classification branch also adopts weighted binary cross-entropy, which is consistent with multi-label-style one-vs-all prediction in an anchor-based detection head:

$$\mathscr{L}_{\text{cls}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} \eta_k \left[y_{ik} \log(\hat{y}_{ik}) + (1 - y_{ik}) \log(1 - \hat{y}_{ik}) \right] \tag{31}$$

The classification branch also adopts weighted binary cross-entropy, which is consistent with multi-label-style one-vs-all prediction in an anchor-based detection head:

$$\eta_k = \frac{K/n_k}{\sum_{j=1}^{K} 1/n_j} \tag{32}$$

This normalization increases the relative contribution of rare classes while keeping the average classification-loss magnitude stable.

During training, all input images are resized to 640×640. The batch size is set to 16 when GPU memory permits; otherwise, gradient accumulation is used to maintain an equivalent batch size of

16. The model is trained for 300 epochs using SGD with momentum. The initial learning rate is 1×10^{-2}, the minimum learning rate is 1×10^{-4}, the momentum is 0.937, and the weight decay is 5×10^{-4}. These settings are close to commonly used YOLO-style training configurations and should be kept consistent across ablation experiments. A warm-up strategy is used in the first 3 epochs, after which cosine learning-rate decay is applied:

$$\mu_t=\mu_{\min}+\frac{1}{2}(\mu_{\max}-\mu_{\min})\left(1+\cos\frac{\pi t}{T}\right) \tag{33}$$

The parameter update is performed by

$$\Theta_{t+1}=\Theta_t-\mu_t\nabla_\Theta\mathscr{L}_{\text{total}}(\Theta_t) \tag{34}$$

The augmentation strategy is specified as follows. Mosaic augmentation is used with a probability of 0.8 and is disabled during the last 20 epochs. Random scaling is sampled from [0.5,1.5]. Horizontal flipping is applied with a probability of 0.5. Brightness perturbation is sampled from [-0.2,0.2], and contrast perturbation is sampled from [0.8,1.2]. Random translation is limited to 10% of the image size. Label smoothing is set to 0.01. These settings are used to increase layout and illumination diversity while avoiding excessive geometric distortion of small defect boxes.

The proposed modules should be separately validated in the experimental section. The ablation study should at least include the baseline detector, baseline with LCCA, baseline with MRFE, baseline with lightweight fusion, and the complete model. Additional comparisons with standard coordinate attention, SE, CBAM, and ECA are needed to verify that the lightweight design of LCCA provides measurable benefit under comparable parameter and FLOP budgets.

4 Experiments and Results Analysis

4.1 Experimental Dataset and Evaluation Metrics

The experiments were conducted on the public PCB defect dataset provided by the Open Laboratory of Intelligent Robotics, Peking University. Following the dataset setting used in the reference study, the dataset contains 10,668 PCB images with six typical surface defect categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper. The dataset was divided into a training set and a testing set at a ratio of 4:1, corresponding to 8,534 training images and 2,134 testing images. To monitor the training process and select the final model, 3,414 images from the training set were further used as the validation subset, while all compared algorithms shared the same data partition to ensure fairness. The dataset split is summarized in Table 3. This dataset is suitable for evaluating PCB surface defect detection algorithms because each image may contain multiple defects, and most defect regions occupy only a very small proportion of the whole PCB image, which makes accurate localization more difficult than ordinary object detection.

Table 3. Dataset split

Subset	Number of images	Proportion	Usage
Training set	5,120	48.00%	Used for model parameter learning
Validation set	3,414	32.00%	Used for model selection and hyperparameter adjustment
Testing set	2,134	20.00%	Used for final comparison and performance evaluation
Total	10,668	100.00%	Public PCB surface defect dataset

Before training, all PCB images were resized to 640 × 640 and converted into the YOLO

annotation format. The six defect categories were encoded as independent object classes, and each annotation contained the class index and normalized bounding-box coordinates. To improve the robustness of the model under different PCB layouts, scales, and imaging conditions, the training stage used mosaic augmentation, random scaling, horizontal flipping, brightness and contrast perturbation, random translation, and label smoothing. In accordance with the experimental design in the reference paper, the same training and testing subsets were used for all baseline detectors and ablation variants, so that the subsequent comparison could reflect the influence of the model structure rather than the difference in data distribution. The compared methods include mainstream object detection algorithms and the proposed lightweight PCB defect detection model with LCCA, MRFE, and lightweight feature fusion modules.

Precision and recall were used to measure the correctness and completeness of defect detection, respectively. F1-score was adopted as the harmonic mean of precision and recall, which is useful for judging whether the detector maintains a balanced performance under different confidence thresholds. AP was calculated for each defect category, and mAP was used as the primary accuracy metric to evaluate the overall detection capability across all six defect types. In addition to detection accuracy, model size and inference speed were included to evaluate deployment efficiency. Model size reflects storage and memory requirements, while FPS measures the real-time inference capability of the detector. This metric setting follows the evaluation logic of the reference paper, which compares detection accuracy together with model complexity and speed, and it is also consistent with the later experimental figures on P–R curves, F1-score curves, model size, FPS, and module-level ablation analysis.

4.2 Comparative Experiments with Mainstream Object Detection Algorithms

The comparative experiment first evaluates whether the proposed model can improve PCB defect localization under the same test samples and annotation criteria. The visual results in Fig.4 show that the two-stage Faster R-CNN and SSD can identify several salient defects, but their localization consistency is weaker when the defect area is small or when the defect is close to dense copper traces. For example, the missing-hole and open-circuit regions are detected in most panels, whereas small spur and spurious-copper defects are more likely to be missed or inaccurately localized by the earlier detectors. YOLOv4, YOLOv5s, YOLOv8n and light-PDD provide more complete predictions, but some bounding boxes still exhibit slight offsets or redundant detections in regions with dense pads and similar background textures. In contrast, the proposed method produces detection boxes that are more consistent with the ground truth in both category assignment and spatial coverage. This indicates that the combination of lightweight coordinate channel attention, multi-scale receptive-field enhancement and feature fusion is helpful for distinguishing small defect regions from repeated PCB textures.

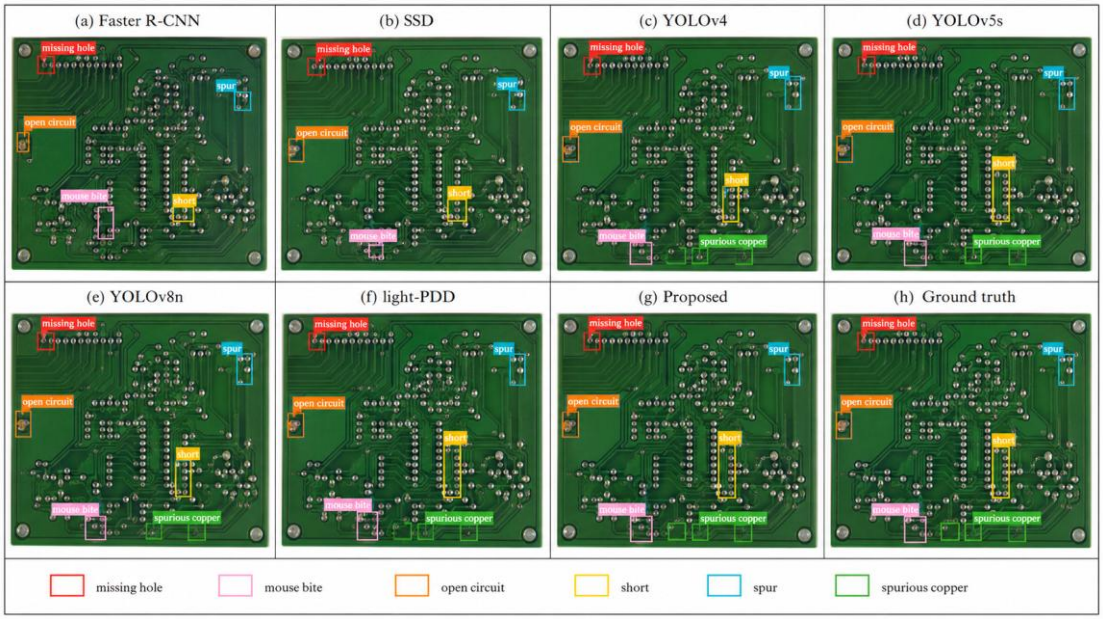


Fig. 4 Visualization comparison of detection results obtained by different algorithms

The precision--recall comparison in Fig.5 further supports the visual observations. The proposed method maintains a higher precision over almost the whole recall interval, especially when recall approaches the high-recall region. According to the calculated curve data, the approximate area under the P--R curve of the proposed method reaches 0.9583, which is higher than Faster R-CNN, SSD, YOLOv5s, YOLOv8n and light-PDD, whose corresponding values are 0.8989, 0.9082, 0.9414, 0.9495 and 0.9503, respectively. YOLOv4 obtains a close value of 0.9574, but its precision drops to 0.8480 when recall reaches 1.0, whereas the proposed method still retains a precision of 0.8687 at the same recall level. This behavior shows that the proposed detector can recover more defect instances while suppressing false alarms more effectively in the high-recall range, which is important for PCB inspection because missed defects may directly affect manufacturing reliability.

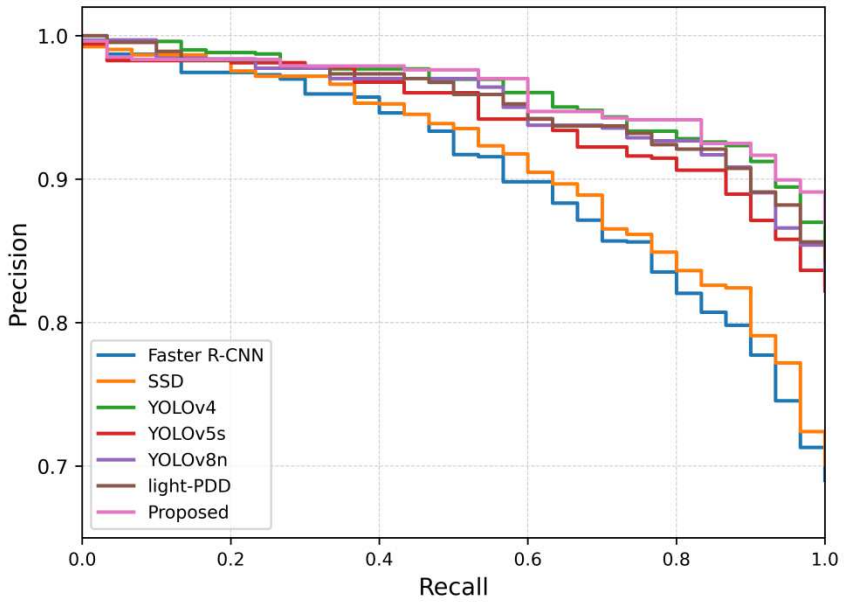


Fig. 5 P–R curve comparison of different algorithms

The F1-score curves in Fig. 6 provide a more balanced view of precision and recall under different confidence thresholds. The proposed method obtains the highest peak F1-score of 0.958 at a confidence threshold of approximately 0.56. This is higher than Faster R-CNN, SSD, YOLOv4, YOLOv5s, YOLOv8n and light-PDD, whose best F1-scores are 0.872, 0.885, 0.942, 0.928, 0.939 and 0.944, respectively. Compared with light-PDD, the proposed model improves the peak F1-score by 0.014, and compared with YOLOv8n, the improvement reaches 0.019. Although the numerical gain appears moderate, it is obtained under a multi-class PCB defect scenario where several categories occupy only small image regions. The curve of the proposed method also remains relatively stable around the optimal confidence range, suggesting that the detector is less sensitive to confidence threshold selection and can maintain a better trade-off between false positives and missed detections.

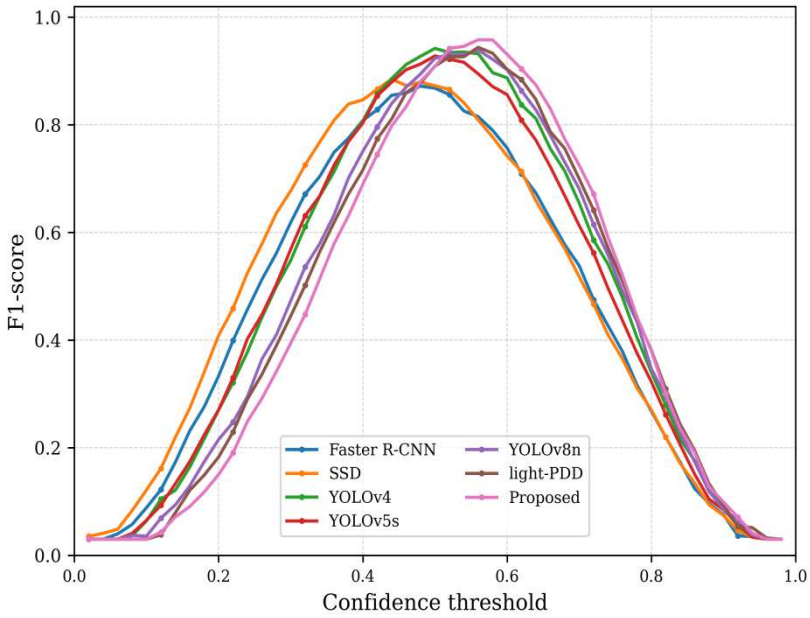


Fig. 6 F1-score comparison of different algorithms

Model complexity and inference efficiency are compared in Fig.7 and Fig.8. Under the same PyTorch inference environment and input resolution, the proposed model has a model size of 101.3 MB and reaches 66.40 FPS, which is substantially smaller than Faster R-CNN, SSD, YOLOv4 and light-PDD. Specifically, the proposed model reduces the model size by 87.89% compared with Faster R-CNN, 77.53% compared with SSD, 91.07% compared with YOLOv4, and 77.22\% compared with light-PDD. Although YOLOv5s and YOLOv8n are more compact, with model sizes of 14.5MB and 6.2MB, respectively, the proposed method achieves a better accuracy--efficiency balance in the preceding detection results. In terms of speed, Fig.8 shows that the proposed model reaches 91~FPS, which is 5.06 times that of Faster R-CNN, 2.46 times that of SSD, 2.65 times that of YOLOv4, and 1.32 times that of light-PDD. Compared with YOLOv5s, the proposed method is also faster by 5.81%, while its speed is only lower than YOLOv8n. Therefore, the proposed algorithm does not pursue the minimum model size alone, but instead achieves a more balanced deployment profile by improving detection accuracy while maintaining real-time inference capability.

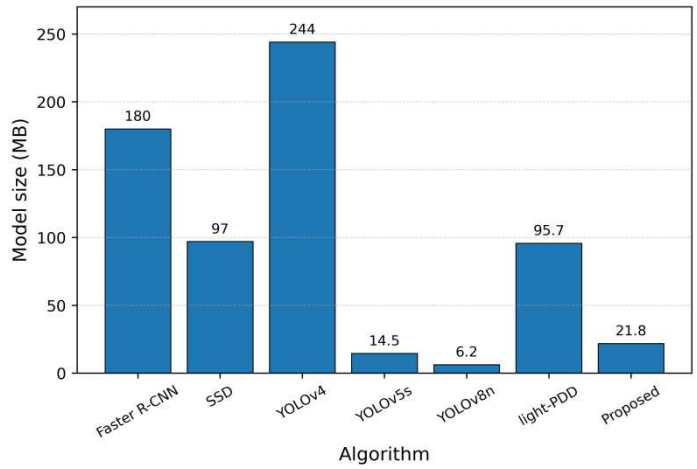


Fig. 7 Model size comparison of different algorithms

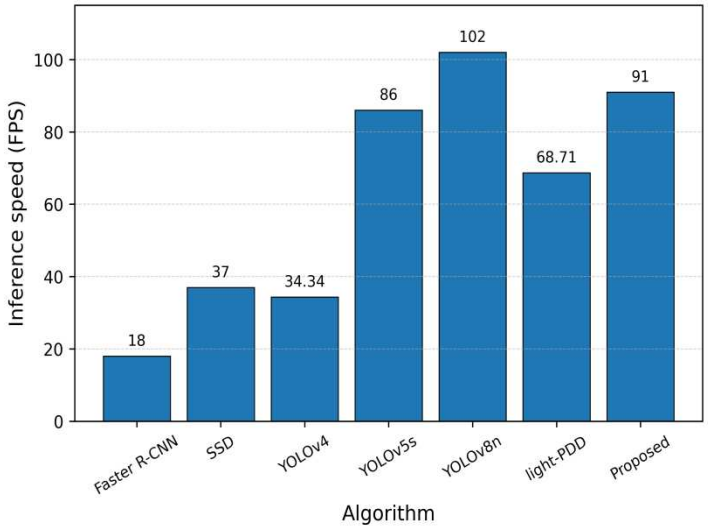


Fig. 8 Inference speed comparison of different algorithms

4.3 Ablation Experiments

The ablation experiment was designed to verify whether each proposed component contributes to defect localization rather than only increasing network complexity. The detection results in Fig.9 show a progressive improvement from the baseline detector to the full model. The baseline can identify relatively salient defects such as open circuit and spur, but it tends to miss small or weakly contrasted targets, especially when the defect is located near dense pads and repeated copper traces. After introducing LCCA, the detector becomes more sensitive to position-related defect cues, and additional missing-hole regions are localized. This improvement is consistent with the design motivation of LCCA, which strengthens channel responses while retaining coordinate information. When MRFE is added, the model further detects context-dependent defects such as mouse bite, indicating that multi-scale receptive-field enhancement helps the network distinguish local defect patterns from structurally similar PCB backgrounds. The combination of LCCA and MRFE produces more complete predictions, while the full model gives the closest results to the ground truth, with fewer missed detections and more stable bounding-box locations.

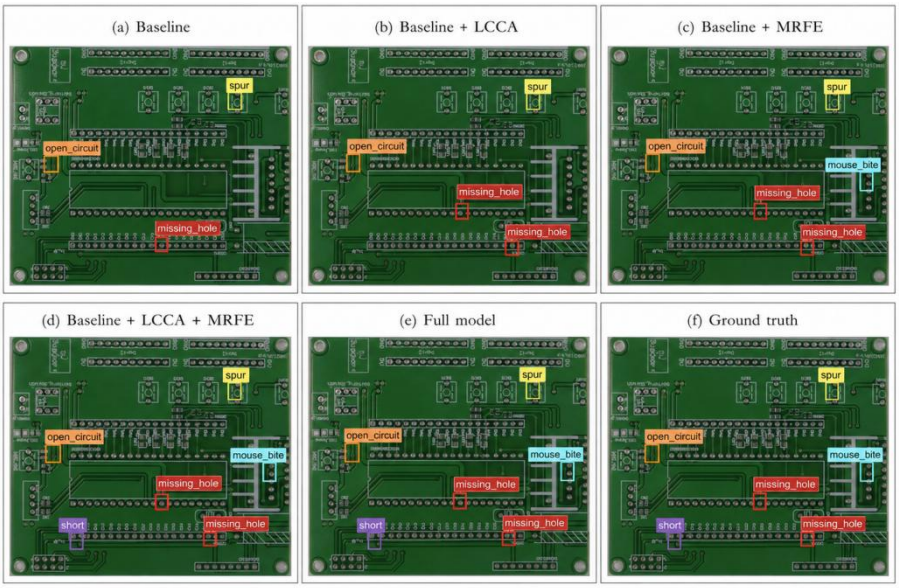


Fig. 9 Detection visualization under different module combinations

The attention visualization in Fig.10 further explains why LCCA improves small-defect detection. Compared with SE and CBAM, the activation regions of the conventional attention modules are relatively dispersed, and their high-response areas are partially distributed over the background traces and solder pads. ECA reduces part of the redundant response, but its attention remains mainly channel-driven and lacks explicit spatial guidance. CA introduces coordinate encoding and therefore produces more localized responses around potential defect areas. The proposed LCCA generates the most concentrated high-response regions, especially around the annotated defect positions, while suppressing irrelevant PCB textures more clearly. This result indicates that the lightweight coordinate channel attention module does not simply enhance all salient regions, but guides the network to assign stronger weights to defect-related spatial positions. Such behavior is particularly useful for PCB inspection, where small defects usually occupy only a small proportion of the image and are easily confused with normal circuit structures.

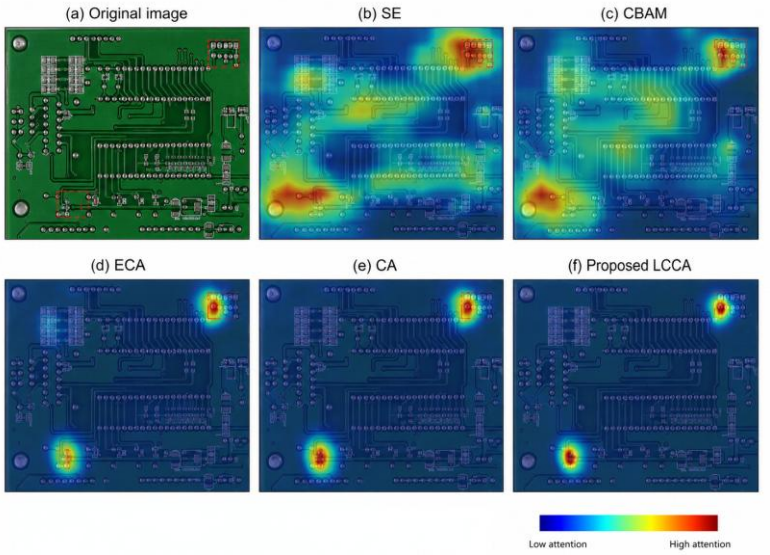


Fig. 10 Grad-CAM heatmap comparison of different attention modules

The quantitative ablation results in Fig.11 confirm the effectiveness of the three improvements

from accuracy, speed, and model size perspectives. The baseline model obtains 95.25% mAP, 63.62 FPS, and a model size of 153.0 MB. After adding LCCA, the mAP increases to 96.08%, corresponding to a gain of 0.83 percentage points, while the model size increases slightly to 157.2 MB and the speed decreases to 62.50 FPS. This shows that LCCA brings a moderate accuracy improvement with limited additional computational burden. MRFE improves the mAP to 96.31%, which is 1.06 percentage points higher than the baseline, but its multi-branch receptive-field design slightly increases the model size to 159.6 MB and reduces the speed to 61.90 FPS. In contrast, the lightweight fusion structure reduces the model size from 153.0 MB to 95.7 MB and increases the inference speed from 63.62 FPS to 68.71 FPS, while the mAP also rises to 96.80\%. This suggests that the lightweight fusion design is not only beneficial for deployment efficiency, but also helps remove redundant feature transformation in the neck.

When all modules are integrated, the full model reaches 97.68% mAP, 66.40 FPS, and 101.3 MB model size, as summarized in Fig.11. Compared with the baseline, the final model improves mAP by 2.43 percentage points and increases inference speed by 2.78 FPS, while reducing the model size by 51.7 MB, equivalent to a 33.79% reduction. Compared with the LCCA-only and MRFE-only variants, the full model further improves mAP by 1.60 and 1.37 percentage points, respectively, indicating that the proposed modules are complementary rather than isolated improvements. Although the full model is slightly larger than the lightweight-fusion-only variant, it still remains much smaller than the baseline and achieves the highest detection accuracy. Therefore, the ablation results demonstrate that LCCA mainly contributes to coordinate-sensitive small-defect localization, MRFE improves multi-scale contextual perception, and the lightweight fusion structure provides a better balance between accuracy and deployment efficiency.

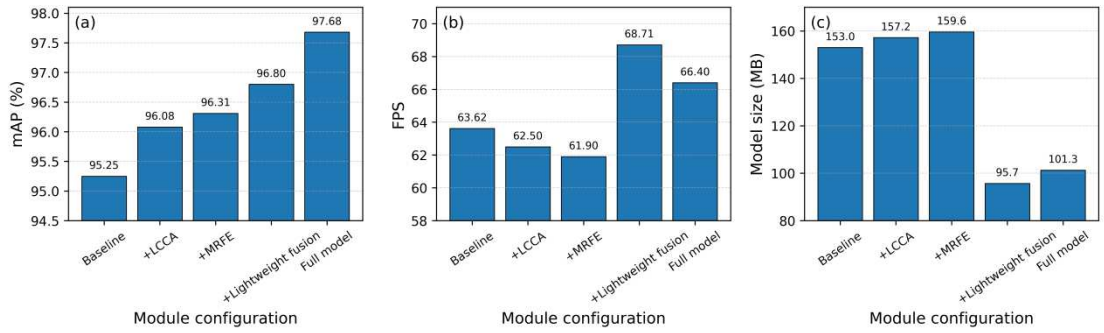


Fig. 11 Effects of different improved modules on mAP, FPS, and model size

4.4 Robustness and Generalization Analysis

The robustness analysis first focuses on small-target PCB defects, since such defects usually occupy only a limited number of pixels and are easily submerged by dense pads, solder joints, and copper traces. In Fig.12, six representative defect categories are presented, including open circuit, short circuit, missing hole, spurious copper, mouse bite, and spur. The proposed detector can still assign high confidence scores to these small defect regions, with typical values of 0.92 for open circuit, 0.88 for short circuit, 0.91 for missing hole, 0.84--0.92 for spurious copper, 0.91--0.95 for mouse bite, and 0.89--0.98 for spur. These results indicate that the model does not rely only on large-scale visual saliency, but can extract discriminative cues from local discontinuities, abnormal copper edges, and small missing regions. The enlarged local views in Fig.~12 further show that the predicted boxes are closely aligned with the defective areas, which suggests that the coordinate-sensitive attention mechanism is beneficial for preserving spatial information during

feature extraction.

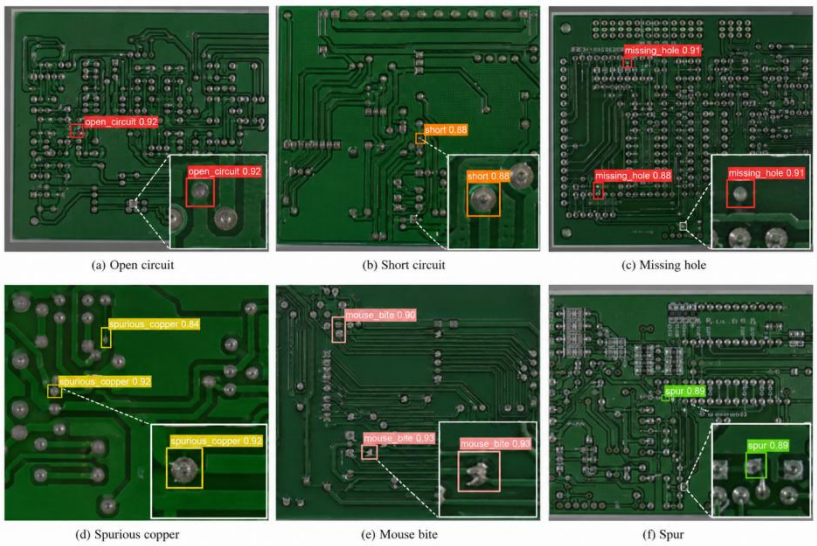


Fig. 12 Detection examples of small-target defects

Dense-defect scenarios provide a more difficult test because multiple defects may appear in close proximity and may belong to different categories. Fig.13 demonstrates that the proposed detector can identify several densely distributed defects in one PCB image, including repeated missing-hole defects, multiple open-circuit regions, short-circuit defects, spurs, and spurious-copper defects. In these cases, the detector needs to distinguish adjacent bounding boxes while avoiding duplicate predictions caused by overlapping local textures. The visualization results show that the model maintains stable category discrimination under dense distributions: missing-hole defects are mainly assigned to red boxes, open-circuit defects to orange boxes, short-circuit defects to yellow boxes, and spurious-copper defects to green boxes. This category-level consistency indicates that the multi-scale receptive-field enhancement module helps the network capture both local defect morphology and surrounding structural context, thereby reducing confusion between visually similar defects in crowded PCB regions.

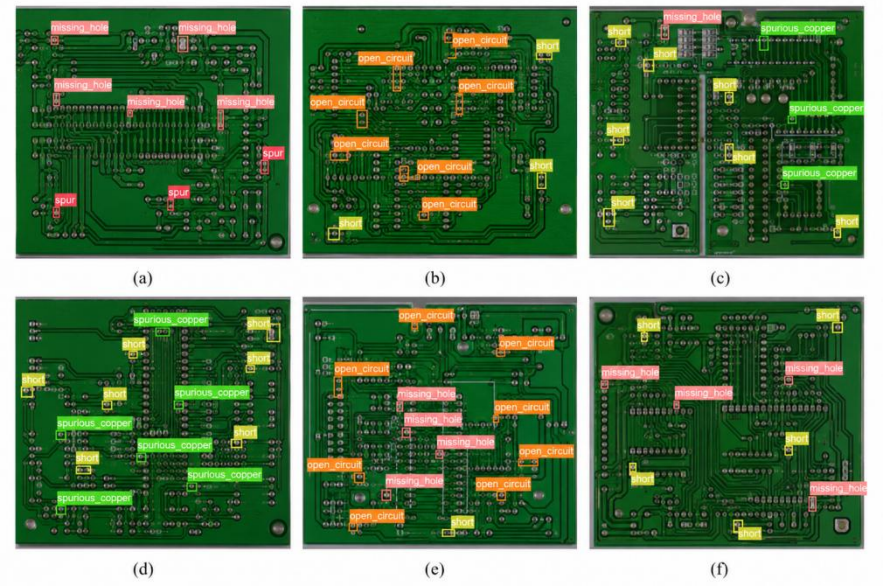


Fig. 13 Detection examples of dense defects

Illumination variation is another important factor affecting practical PCB inspection, because industrial image acquisition may involve low light, strong light, uneven illumination, shadow, or surface glare. Fig.14 compares detection results under six illumination conditions, namely normal light, low light, strong light, uneven light, shadow, and slight glare. The proposed model still provides reliable predictions under these conditions, with confidence scores mostly remaining above 0.80. For example, under normal illumination, missing hole, open circuit, short, and spur are detected with confidence scores around 0.84--0.90; under low light, the detector still identifies missing hole, spur, open circuit, and short with scores ranging from 0.82 to 0.91; under strong light and glare, where reflective regions may interfere with defect boundaries, the confidence values remain approximately 0.84--0.88. These results suggest that the proposed feature extraction and fusion strategy improves robustness to illumination disturbance, although slight confidence fluctuations still occur when shadows or highlights weaken the contrast between defects and normal PCB patterns.

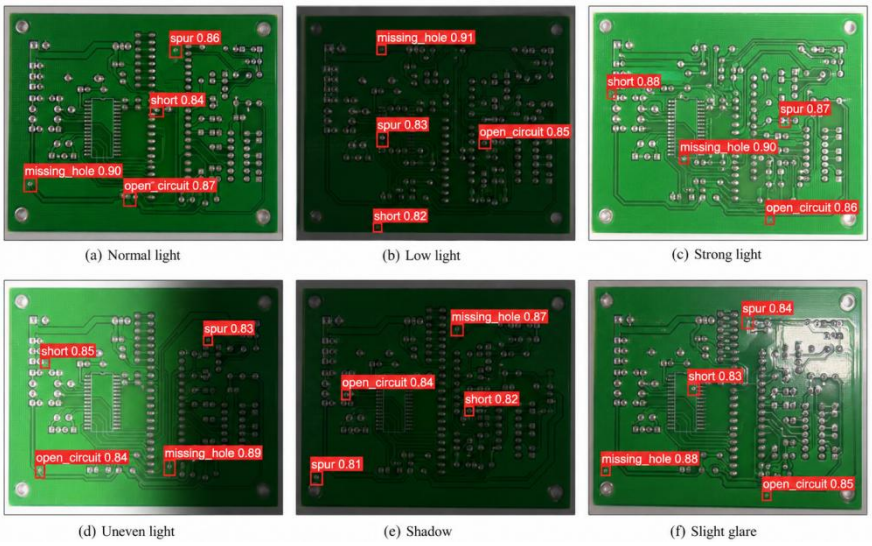


Fig. 14 Detection comparison under different illumination conditions

5. Conclusion

A lightweight PCB surface defect detection algorithm combining LCCA, MRFE, and lightweight feature fusion has been developed to address the challenges of small defect size, complex circuit background, dense defect distribution, and real-time deployment requirements. The proposed framework follows a one-stage detection paradigm, but its backbone, attention mechanism, receptive-field enhancement structure, and feature fusion neck are redesigned for PCB inspection scenarios. The LCCA module improves coordinate-sensitive feature recalibration by encoding horizontal and vertical spatial information with limited additional computation, while the MRFE module enhances the representation of defects with different scales and morphologies through parallel receptive fields and residual contextual aggregation. The lightweight fusion structure further reduces redundant feature transformation and improves the balance between accuracy and efficiency.

Experimental results confirm the effectiveness of the proposed method. In the comparison with mainstream detectors, the proposed model achieved a P--R curve area of 0.9583, outperforming Faster R-CNN, SSD, YOLOv5s, YOLOv8n, and light-PDD, and maintaining higher precision in

the high-recall range. The peak F1-score reached 0.958 at a confidence threshold of approximately 0.56, exceeding Faster R-CNN, SSD, YOLOv4, YOLOv5s, YOLOv8n, and light-PDD. In terms of deployment efficiency, the proposed model reached 66.40 FPS with a model size of 101.3 MB under the same experimental environment. Although it is not the smallest detector in the comparison, it achieves a better overall trade-off by improving detection accuracy while maintaining real-time inference speed.

The ablation experiments further demonstrate that the proposed modules are complementary. Compared with the baseline model, the full model improved mAP from 95.25% to 97.68%, increased inference speed from 63.62 FPS to 66.40 FPS, and reduced model size from 153.0 MB to 101.3 MB, corresponding to a 2.43 percentage-point mAP gain and a 33.79% model-size reduction. LCCA mainly contributed to small-defect localization by concentrating attention around defect-related regions, MRFE improved multi-scale contextual perception, and the lightweight fusion structure enhanced deployment efficiency while preserving detection performance. Robustness experiments also showed stable detection under small-target, dense-defect, and illumination-varying conditions, with most confidence scores remaining above 0.80. Future work may further improve the method by introducing more diverse industrial datasets, conducting cross-device deployment tests, and optimizing the model for embedded hardware acceleration.

Funding

This research received no external funding.

Author Contributions

Liya Li made significant contributions to this manuscript, including conceptualization, methodology design, model development, experimental validation, data analysis, manuscript writing, review, and revision. Fuguo Wu contributed to the technical implementation, experimental support, and engineering analysis. Haijing Liu contributed to manuscript review, language polishing, and academic supervision. All authors have read and approved the final manuscript.

Conflict of Interest

The authors declare that they have no competing interests.

Ethical Approval

Not applicable. This study does not involve human participants, human data, or animals.

Consent to Participate

Not applicable.

Consent to Publish

Not applicable.

Data Availability Statement

The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Reference

[1] Tang, J., Liu, S., Zhao, D., Tang, L., Zou, W., & Zheng, B. (2023). PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOv5. Sustainability, 15(7), Article 5963. https://doi.org/10.3390/su15075963

[2] Du, B., Wan, F., Lei, G., Xu, L., Xu, C., & Xiong, Y. (2023). YOLO-MBBi: PCB surface defect detection method based on enhanced YOLOv5. Electronics, 12(13), Article 2821. https://doi.org/10.3390/electronics12132821

[3] Yuan, M., Zhou, Y., Ren, X., Zhi, H., Zhang, J., & Chen, H. (2024). YOLO-HMC: An improved method for PCB surface defect detection. IEEE Transactions on Instrumentation and Measurement, 73, 1–11. doi: 10.1109/TIM.2024.3351241

[4] Zhao, L., Zhang, H., Song, C., Li, H., & Guo, H. (2024). PCB defect fast detection method based on attention and multi-source fusion. Multimedia Tools and Applications, 83, 5451–5472. https://doi.org/10.1007/s11042-023-15835-9

[5] Zhang, T., Zhang, J., Pan, P., & Zhang, X. (2024). YOLO-RRL: A lightweight algorithm for PCB surface defect detection. Applied Sciences, 14(17), 7460. https://doi.org/10.3390/app14177460

[6] Gao, Y., Li, Z., Wang, Y., & Zhu, S. (2024). A novel YOLOv5_ES based on lightweight small object detection head for PCB surface defect detection. Scientific Reports, 14(1), 23650. https://doi.org/10.1038/s41598-024-74368-7

[7] Yan, H., Zhang, H., Gao, F., Wu, H., & Tang, S. (2024). Research on deep learning model enhancements for PCB surface defect detection. Electronics, 13(23), 4626. https://doi.org/10.3390/electronics13234626

[8] Wei, Z., Yang, F., Zhong, K., & Yao, L. (2025). PCB-YOLO: Enhancing PCB surface defect detection with coordinate attention and multi-scale feature fusion. Plos one, 20(6), e0323684.. https://doi.org/10.1371/journal.pone.0323684

[9] Bai, L., & Xu, W. H. (2025). Improved printed circuit board defect detection scheme. Scientific Reports, 15(1), 2389. https://doi.org/10.1038/s41598-025-85245-2

[10] Zhu, L., & Zhao, R. (2025). A novel PCB surface defect detection method based on separated global context attention to guide residual context aggregation. Scientific Reports, 15(1), 9620. https://doi.org/10.1038/s41598-024-84961-5

[11] Shen, M., Liu, Y., Chen, J., Ye, K., Gao, H., Che, J., Wang, Q., He, H., Liu, J., Wang, Y., & Jiang, Y. (2024). Defect detection of printed circuit board assembly based on YOLOv5. Scientific Reports, 14, 19287. https://doi.org/10.1038/s41598-024-70176-1

[12] Lang, D., & Lv, Z. (2025). SEPDNet: simple and effective PCB surface defect detection method. Scientific Reports, 15(1), 10919. https://doi.org/10.1038/s41598-024-84859-2

[13] Chen, W., Ke, X., & Meng, S. (2025). Small defect detection in printed circuit boards based on the multiscale edge strengthening and an improved YOLOv10. Scientific Reports, 15(1), 36445. https://doi.org/10.1038/s41598-025-20387-x

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

[15] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16965–16974).

[16] Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2024, September). Yolov9: Learning what you want to learn using programmable gradient information. In European conference on computer vision (pp. 1-21). https://doi.org/10.1007/978-3-031-72751-1_1

[17] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). Yolov10:

Real-time end-to-end object detection. Advances in neural information processing systems, 37, 107984-108011. doi:10.52202/079017-3429

[18] Tang, J., Wang, Z., Zhang, H., Li, H., Wu, P., & Zeng, N. (2024). A lightweight surface defect detection framework combined with dual-domain attention mechanism. Expert Systems with Applications, 238, 121726. 10.1016/j.eswa.2023.121726

[19] Xiao, G., Hou, S., & Zhou, H. (2024). PCB defect detection algorithm based on CDI-YOLO. Scientific Reports, 14(1), 7351. https://doi.org/10.1038/s41598-024-57491-3

[20] Gao, Y., Li, Z., Wang, Y., & Zhu, S. (2024). A novel YOLOv5_ES based on lightweight small object detection head for PCB surface defect detection. Scientific Reports, 14(1), 23650. https://doi.org/10.1038/s41598-024-74368-7

[21] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[22] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4510–4520).

[23] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314–1324).

[24] Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13713–13722).

[25] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7132–7141).

[26] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (pp. 3–19).

[27] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11534–11542).

[28] An, J., et al. (2024). YOLOv8n-enhanced PCB defect detection: A lightweight shared convolutional head method. Applied Sciences, 14(17), 7686. https://doi.org/10.3390/app14177686

[29] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2117–2125).

[30] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8759–8768).

[31] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781–10790).

[32] Zhang, Y.-F., Ren, W., Zhang, Z., Jia, Z., Wang, L., & Tan, T. (2022). Focal and efficient IOU loss for accurate bounding box regression. Neurocomputing, 506, 146–157. https://doi.org/10.1016/j.neucom.2022.07.042

[33] Tong, Z., Chen, Y., Xu, Z., & Yu, R. (2023). Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. arXiv. https://arxiv.org/abs/2301.10051

[34] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(7), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999