2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Defect Detection and Classification of PCB Based on RT-DETR

Shuai Hao, Xiaoqi He(✉), and Ya Li

Shanghai Jiao Tong University, Shanghai 200240, China
hexiaoqi@niii.com

Abstract. Printed circuit board (PCB) defect detection is crucial for ensuring PCB quality. However, small defect sizes on PCBs and the substantial parameter and computational requirements of deep learning models pose challenges in detection accuracy and deployment on resource-constrained devices. To address these issues, we introduce an optimized PCB defect detection method based on the RT-DETR model. Firstly, we propose the ContextAdown-D-LKAEfficientFormerV2 lightweight network to replace the ResNet18 backbone, reducing parameters and computations while enhancing small defect feature extraction. Secondly, we present the Bi-Slim-Neck lightweight structure to replace the CCFM component in the original model, achieving lightweight design and improved feature fusion capabilities to leverage effective features fully. Lastly, we propose the InnerShapeIoU loss function to replace the GIoU loss function, accounting for the influence of PCB defect bounding box shapes and scales on regression, and generating auxiliary bounding boxes suitable for PCB defect detection tasks and detectors. This enhances model generalization and detection accuracy. Experimental results show that the improved model achieves detection accuracies of 97% (mAP_{50}) and 55% ($mAP_{50:95}$), with a 23.8% reduction in parameters and a 44.9% decrease in computations compared to the original model. This indicates that the improved method significantly boosts parameter efficiency and reduces computational complexity while maintaining high detection accuracy.

Keywords: RT-DETR, PCB defect detection, Small defect.

1 Introduction

As the core component of electronic devices, printed circuit boards (PCBs) satisfy diverse application requirements through intricate circuit designs and component layouts. PCB defect detection, a crucial means of ensuring quality, has undergone continuous technological innovations, effectively identifying defects such as shorts, opens, and missing holes, ultimately contributing to a substantial enhancement in production quality and ensuring the reliable and consistent operation of electronic equipment in diverse operational environments. Initially, PCB defect detection relied on manual inspection. However, as PCB complexity increased, manual methods faced

issues such as operator fatigue, slow speed, and reduced accuracy, highlighting their increasing limitations. Consequently, detection techniques based on traditional image processing gained popularity, encompassing methods such as template matching, image subtraction (Pal [1]), K-means clustering (Melnyk [2]), and Gaussian Mixture Models (Cai [3]). Nevertheless, these methods are sensitive to translation and illumination changes and are primarily suited for detecting large-scale defects, lacking precision in identifying small-target PCB defects.

With the development of deep learning technology, its advantages of high accuracy and efficiency have gradually been applied in the field of PCB defect detection. Ding [4] proposed TDD-net based on Faster R-CNN, designing anchor boxes through K-means clustering, enhancing semantic interaction using FPN, and optimizing prediction results with OHEM. Hu [5] improved Faster R-CNN by using ResNet50 and FPN for feature extraction and fusion, introducing Shuffle V2 residual units and a generative adversarial region proposal network to enhance detection accuracy and efficiency. Wang et al. [6] designed the YOLO-Biformer model for challenges such as complex backgrounds, small defects, and irregular shapes in PCB defect images. Wu [7] proposed an improved YOLOv7-tiny network based on YOLOv7, employing the CARAFE upsampling method, global attention mechanism, and Focal-EIoU loss function to address training sample imbalance, improving target localization and regression accuracy, and achieving high-accuracy real-time detection. However, these deep learning models have limited feature extraction capabilities, unable to fully utilize defect features for object detection, and there is still considerable room for improvement in detection accuracy for small target defects.

The Transformer model, due to its global attention mechanism, effectively considers comprehensive global information and enhances the feature weights of small targets, thereby gradually gaining widespread application in small object detection tasks. Chen [8] introduced a Transformer-YOLO detection network, which utilizes Swin Transformer as the feature extraction network while incorporating an attention mechanism module. This approach successfully improved detection accuracy and featured a smaller model size compared to other mainstream object detection models. Li [9] proposed PCB-DETR, which successfully enhanced the detection accuracy of small target defects on printed circuit boards (PCBs) by improving Deformable-DETR. However, the aforementioned studies suffer from slow inference speeds due to the large amount of matrix calculations required by the global attention mechanism, making real-time detection unachievable, and the defect detection accuracy also needs further improvement. Therefore, this study proposes to improve the RT-DETR [10] with better real-time performance to achieve rapid and accurate PCB defect detection.

2 Proposed Method

2.1 Improved RT-DETR Model

Compared to other DETR-like models, RT-DETR stands out by significantly reducing computational costs and enhancing detection accuracy through the innovative integration of an efficient hybrid encoder and an IoU-aware mechanism. This model not only

achieves remarkable performance in terms of detection accuracy but also boasts a notable increase in inference speed. Despite achieving remarkable performance in detection accuracy and inference speed, the RT-DETR model encounters deployment challenges when utilized in PCB defect detection equipment due to its substantial parameter count and computational demands. Furthermore, its effectiveness in detecting small targets requires additional refinement. Based on the RT-DETR architecture, this study proposes an optimized network structure. Specifically, improvements are made to the original model by introducing ContextAdown-D-LKAEfficientFormerV2, Bi-Slim-Neck, and the InnerShapeIoU loss function. These enhancements collectively aim to lighten the model's load and leverage effective feature information to boost the detection capabilities for small PCB defects. The proposed model structure is illustrated in Fig. 1.

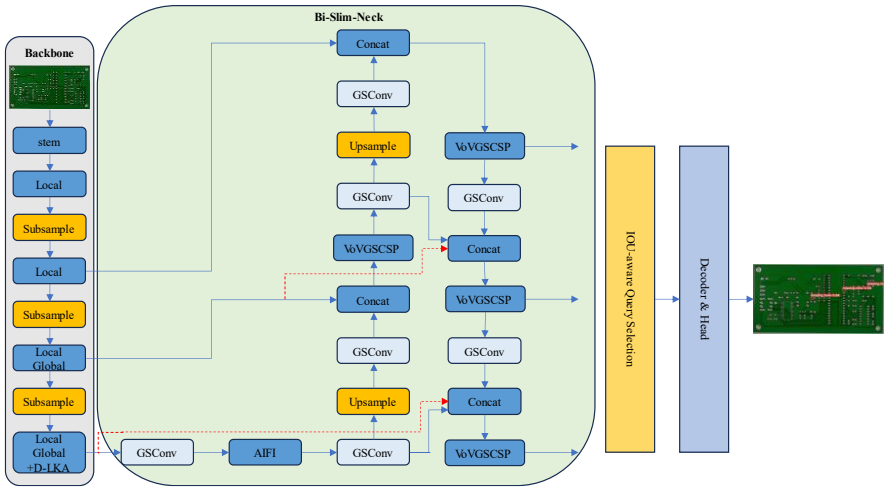


Fig. 1. Diagram of the improved RT-DETR network model structure for PCB defect detection.

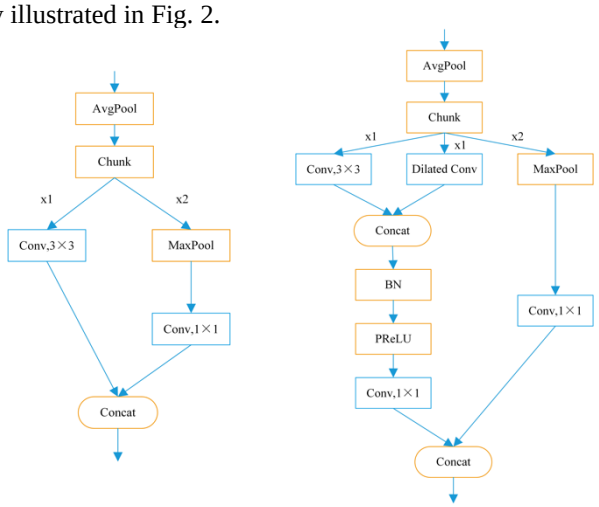
2.2 ContextAdown-D-LKAEfficientFormerV2

The improvement work is based on the RT-DETR model with ResNet18 as the backbone network. Although ResNet18 demonstrates strong feature extraction capabilities, its high parameter count and computational complexity limit its application in real-time detection tasks, and there are limitations in small target detection. Therefore, this study introduces the lightweight EfficientFormerV2 [11] network as the new backbone and proposes the ContextAdown downsampling module to replace some convolutional layers in EfficientFormerV2. Furthermore, the D-LKA (Deformable Large Kernel Attention) [12] is added to enhance the model's ability to extract multi-scale and complex shape features.

EfficientFormerV2 is a vision Transformer network specifically optimized for deployment on mobile devices. It achieves this by incorporating depthwise separable convolutions within the Feed-Forward Network (FFN) module. The network not only

enhances local feature extraction capabilities but also simplifies the model structure. Additionally, the network incorporates refined multi-head self-attention mechanism modules at multiple stages. These enhancements contribute significantly to enhancing the network's ability to detect small targets with higher accuracy.

Given the small size of PCB defect targets and the limited effective feature information available for subsequent detection, it is crucial to preserve effective small-target feature information while integrating surrounding contextual and global information during downsampling. The Adown module, a classical downsampling module, processes input features through two-dimensional average pooling, divides them equally along the channel dimension for local and global information extraction, and finally concatenates these parts via the Concat operation to obtain the final output. However, despite effectively fusing local and global information, the Adown module neglects the importance of surrounding contextual information, which is vital for supplementing detection information of small targets due to their minimal size and scarce feature information. To address this, this study introduces a surrounding contextual feature extraction branch into the Adown module, resulting in a new downsampling module named ContextAdown. This module simultaneously extracts additional information from the surrounding contextual feature extraction branch to aid in understanding complex scenes. Subsequently, local feature information and surrounding contextual feature information are concatenated and further processed using Batch Normalization (BN) and Parametric Rectified Linear Unit (PReLU) to enhance feature representation. Finally, a 1×1 convolution is applied for cross-channel information interaction, facilitating the fusion of local and surrounding contextual feature information. This improvement significantly boosts the representation capability of small targets with minimal computational cost, greatly enhancing the efficiency and accuracy of PCB defect detection and recognition. The structure diagrams for Adown and ContextAdown are comprehensively illustrated in Fig. 2.



(a) (b)

Fig. 2. Diagram of the (a) Adown and (b) ContextAdown module structures.

Given the diverse scales and complex shapes of PCB defects, this study introduces the D-LKA mechanism to enhance feature extraction and reduce irrelevant information. D-LKA integrates the broad receptive field of large convolutional kernels with the flexibility of deformable kernel. The large convolutional kernels in D-LKA are implemented using depthwise separable and dilated convolutions, enabling the network to learn features within a large receptive field while maintaining low computational and parametric costs, achieving a performance akin to global self attention. The deformable kernels are used to learn PCB defect features of various shapes and sizes by adjusting convolution sampling positions adaptively through an additional offset calculation network. The final improved backbone network structure is depicted in Fig. 3.

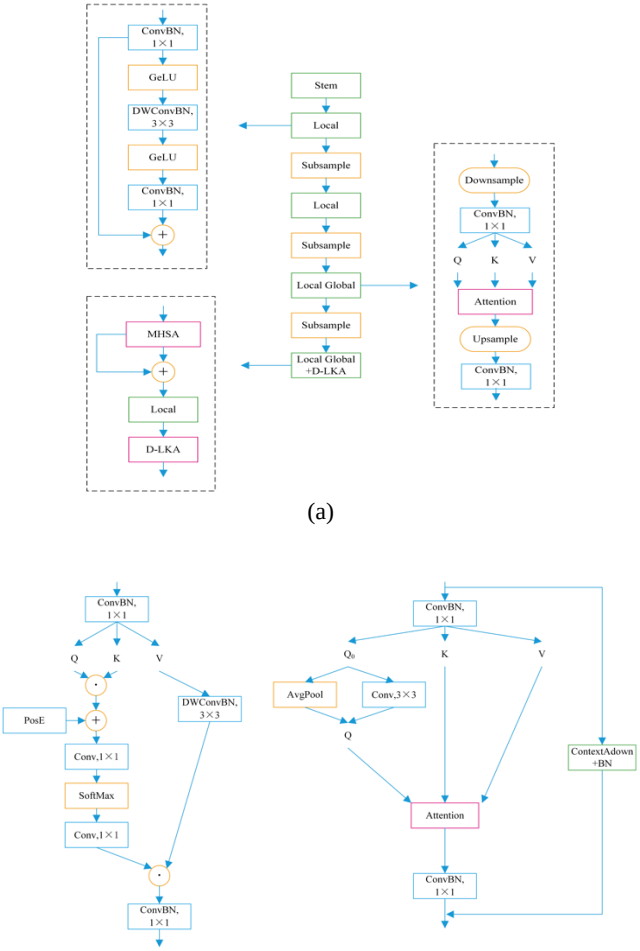


Fig. 3. (a) The architectural diagram of the improved EfficientFormerV2. (b) The architectural diagram of the Subsample. (c) The architectural diagram of the MHSA.

2.3 Bi-Slim-Neck

The CCFM method fuses S3, S4, and S5 feature maps extracted by the backbone network through bottom-up and top-down paths, utilizing upsampling/downsampling modules and fusion modules to achieve multi-scale feature interaction and fusion. However, the high parameter count and computational complexity of CCFM limit its application in model deployment.

This study introduces a lightweight neck network structure, the Slim-Neck network, as an alternative to CCFM. The Slim-Neck [13] network introduces the GSConv (Group-wise Spatial Convolution) structure, which, compared to traditional convolutions, reduces computational costs by half while enhancing implicit inter-channel connections, ensuring sufficient detection accuracy. Based on this, GS Bottleneck and VoV-GSCSP (Variable-size Group Convolution with Spatial Pyramid) are proposed. The GS Bottleneck module enhances feature learning ability by extracting features through multiple GSConvs and enriches feature information by adding the extracted features to the results of regular convolutions. The VoV-GSCSP module, combined with convolution and GS Bottleneck, leverages different path schemes to enhance information exchange among multi-scale features, further improving feature representation capability. These modules enable the Slim-Neck network to maintain powerful feature processing capabilities while reducing computational complexity, thereby effectively fusing multi-scale feature information and providing strong support for subsequent detection tasks.

In neck networks, feature maps typically undergo multiple convolution operations for feature extraction and fusion to obtain the final output features. However, in tasks such as PCB defect detection, where targets are often small and feature-poor, the downsampling process can lead to the loss of some small target features, thereby affecting the final detection results. To address this issue, this study adds two new paths to the Slim-Neck network, forming the Bi-Slim-Neck network. Specifically, the input nodes of the S4 and S5 feature maps are connected to the corresponding Concat modules in the bottom-up path, allowing rich feature information from the original feature maps to be directly provided to subsequent paths. This supplements potentially lost small target information, enhances the network's feature fusion capability, and thus improves detection accuracy. The final structural diagram of Bi-Slim-Neck is illustrated in Fig. 4.

2.4 InnerShapeIoU

Due to the diverse shapes and scales of PCB defect bounding boxes, their shape and size can impact bounding box regression outcomes. Therefore, this study introduces the Shape-IoU [14] loss function, which adjusts the loss function according to the shape and size of the bounding box. This method incorporates a scale factor and weight coefficients related to the bounding box shape to adjust the bounding box regression loss, enabling the loss function to focus on the shape and scale of the bounding box itself.

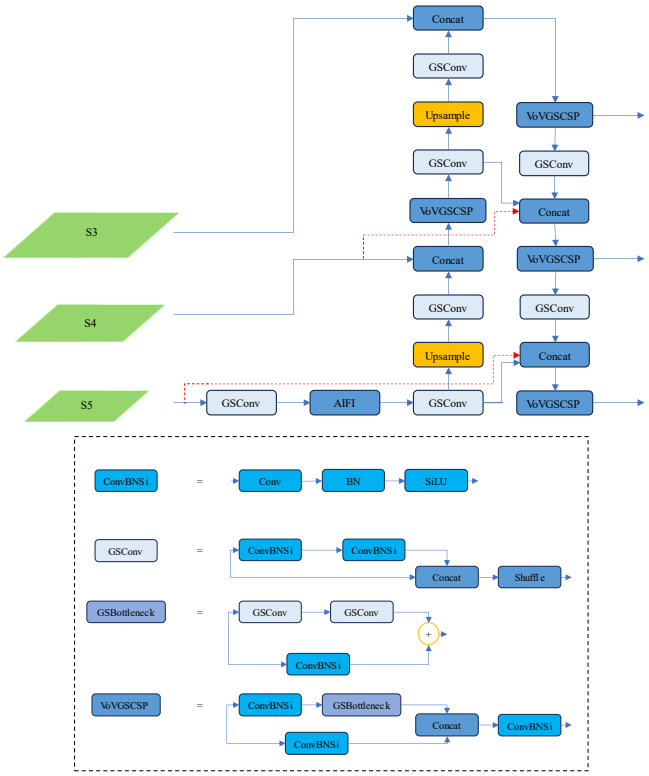


Fig. 4. Diagram of the Bi-Slim-Neck network model structure.

Addressing the generalization weakness and slow convergence of existing IoU losses across various detection tasks, as well as the inability of original bounding box regression to adapt to PCB defect detection tasks and detectors, this study introduces an improvement to the Shape-IoU loss function by incorporating Inner-IoU [15] loss, forming an auxiliary bounding box-based bounding box regression loss. This loss function selects an appropriately scaled auxiliary bounding box through a ratio factor for loss calculation and accelerated convergence. For different detection tasks and detectors, an appropriate auxiliary bounding box scale can be generated by selecting different ratio values, typically ranging from 0.5 to 1.5. For high-IoU samples, setting ratio < 1 selects a smaller-scale auxiliary bounding box with a narrower regression effective range than the actual bounding box but a larger absolute regression gradient than the original IoU loss, enhancing convergence speed. For low-IoU samples, setting ratio > 1 selects a larger-scale auxiliary bounding box, expanding the regression effective range and facilitating the regression of low-IoU samples.

All formulas for InnerShapeIoU are as follows:

$$b_l = x_c - \frac{w \times ratio}{2}, b_r = x_c + \frac{w \times ratio}{2} \tag{1}$$

$$b_t = y_c - \frac{h \times ratio}{2}, b_b = y_c + \frac{h \times ratio}{2} \tag{2}$$

$$b_l^{gt} = x_c^{gt} - \frac{h^{gt} \times ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} \times ratio}{2} \tag{3}$$

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} \times ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} \times ratio}{2} \tag{4}$$

$$ww = \frac{2 \times \left(w^{gt}\right)^{scale}}{\left(w^{gt}\right)^{scale} + \left(h^{gt}\right)^{scale}} \tag{5}$$

$$hh = \frac{2 \times \left(h^{gt}\right)^{scale}}{\left(w^{gt}\right)^{scale} + \left(h^{gt}\right)^{scale}} \tag{6}$$

$$c^2 = \left(\max\left(b_r, b_r^{gt}\right) - \min\left(b_l, b_l^{gt}\right)\right)^2 + \left(\max\left(b_b, b_b^{gt}\right) - \min\left(b_t, b_t^{gt}\right)\right)^2 \tag{7}$$

$$distance^{shape} = hh \times \frac{\left(b_l + b_r - b_l^{gt} - b_r^{gt}\right)^2}{4 \times c^2} + ww \times \frac{\left(b_t + b_b - b_t^{gt} - b_b^{gt}\right)^2}{4 \times c^2} \tag{8}$$

$$\Omega^{shape} = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta}, \theta = 4 \tag{9}$$

$$\begin{cases} \omega_w = hh \times \dfrac{\left|w - w^{gt}\right|}{\max\left(w, w^{gt}\right)} \\ \omega_h = ww \times \dfrac{\left|h - h^{gt}\right|}{\max\left(h, h^{gt}\right)} \end{cases} \tag{10}$$

$$inter = \left(\min\left(b_r^{gt}, b_r\right) - \max\left(b_l^{gt}, b_l\right)\right) \times \left(\min\left(b_b^{gt}, b_b\right) - \max\left(b_t^{gt}, b_t\right)\right) \tag{11}$$

$$union = \left(w^{gt} \times h^{gt}\right) \times \left(ratio\right)^2 + \left(w \times h\right) \times \left(ratio\right)^2 - inter \tag{12}$$

$$IoU^{inner} = \frac{inter}{union} \tag{13}$$

$$L_{Shape-IoU} = 1 - IoU^{inner} + distance^{shape} + 0.5 \times \Omega^{shape} \tag{14}$$

where (x_c, y_c) and (x_c^{gt}, y_c^{gt}) represent the center point coordinates of the anchor box and the ground truth box, respectively; h and w represent the height and width of the anchor box; h^{gt} and w^{gt} represent the height and width of the ground truth box; and ratio

denotes the scaling factor controlling the size of the auxiliary bounding box, typically ranging from 0.5 to 1.5.

3 Introduction

3.1 Dataset

This study utilizes the PKU-Market-PCB [16] publicly released by the Intelligent Robotics Open Lab of Peking University as the experimental subject. The dataset comprises 693 high-quality PCB defect images and their corresponding label files, covering six defect categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper, with a relatively balanced distribution among the defect types. However, due to the limited number of images and the homogeneous image environment in the original dataset, which may not fully reflect the complexity of actual production environments, the risk of overfitting during model training is increased, potentially affecting the model's performance in real-world detection tasks. Therefore, this study proposes to employ data augmentation strategies to expand the dataset. Specifically, based on the actual production environment, two of the five enhancement techniques—rotation, translation, brightness transformation, contrast transformation, and Gaussian noise addition—are randomly applied to each image twice, resulting in a new dataset named EX-PCB dataset, which contains 2079 PCB defect images. Subsequently, the newly generated images are annotated using the labelimg tool to generate corresponding label files, and the dataset is split into training, validation, and test sets in an 8:1:1 ratio for subsequent experiments.

3.2 Implementation Details

The experimental environment for this study is configured with Windows 11 as the operating system, Python 3.9.7 as the programming language, and PyTorch 1.12.1 as the deep learning framework. The hardware includes a 12th-generation Intel(R) Core(TM) i9-12900H processor with a base frequency of 2.50 GHz and an NVIDIA GeForce RTX 3090 GPU with CUDA version 11.7. In terms of experimental parameters, the number of training epochs is set to 400, patience to 50, batch size to 4, and image input size to 640×640. The AdamW optimizer is employed with an initial learning rate of 0.0001, a final learning rate of 1, and a momentum parameter of 0.9.

3.3 Evaluation Metrics

To comprehensively assess model performance, this study adopt mAP_{50}, $mAP_{50:95}$, number of parameters (Params), and computational complexity (GFLOPS) as evaluation metrics. Params and GFLOPS are used as lightweight indicators, where Params represent the total number of parameters to be trained in the network model, correspond to spatial complexity and GFLOPS, indicating the capability to perform one billion floating-point operations per second, correspond to temporal complexity. In this paper,

mAP_{50}, and $mAP_{50:95}$ are used to measure the detection capability of the model. Samples are classified as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Precision (P) is defined as the proportion of true positive samples among all samples classified as positive, while recall (R) is the proportion of positive samples correctly classified as positive. AP represents the area enclosed by the PR curve for each class in the dataset and the coordinate axes, and mAP is the average of AP across all classes. In this study, mAP_{50} denotes the mAP value at an IoU threshold of 0.5, while $mAP_{50:95}$ represents the average mAP value across IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05. The relevant calculation formulas are as follows:

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$AP = \int_0^1 P(R) dR \tag{17}$$

$$mAP = \frac{1}{n} \sum_{j=1}^{n} AP_j \tag{18}$$

3.4 Experimental Results and Analysis

The InnerShapeIoU loss function includes two parameters: scale and ratio. The scale is closely related to the target scale of the dataset, while the ratio is used to select auxiliary bounding box scales suitable for specific detection tasks and detectors. To determine the optimal values of these two parameters, comparative experiments were conducted on our dataset, with results presented in the following Table 1. Ultimately, scale=0.0 and ratio=0.75, which yielded the highest accuracy, were selected for this task.

Table 1. Performance Comparison Across Different Values of Scale and Ratio.

ratio	scale	mAP_{50}	$mAP_{50:95}$
0.75	0.0	97.0%	55.0%
	0.6	95.6%	53.8%
	1.2	96.2%	53.3%
1.0	0.0	95.5%	54,0%
	0.6	96.4%	54.2%
	1.2	96.0%	54.3%
1.25	0.0	96.6%	54.0%
	0.6	95.9%	54.0%
	1.2	96.1%	54.0%

The results presented in Table 2 indicate the effectiveness of the modifications to the backbone network. Model A utilizes ResNet18 as the backbone in our model, whereas Model B replaces it with the lightweight EfficientFormerV2, resulting in a slight decrease of 0.7% and 1.1% in mAP_{50} and $mAP_{50:95}$, respectively, but significantly reducing the number of parameters and computational load. Model C further introduces the Context-Aware Downsampling module onto Model B, which integrates rich feature information while simplifying the model, leading to another notable reduction in parameters and computational cost with only a 0.5% drop in both mAP_{50} and $mAP_{50:95}$. Additionally, Model D incorporates the D-LKA attention mechanism at the end of the backbone. Leveraging the broad receptive field of large convolution kernels and the flexibility of deformable convolution kernels, the network becomes more effective in focusing on complex and diverse feature information, ultimately improving model accuracy by 1% in mAP_{50} and 0.3% in $mAP_{50:95}$. In summary, these modifications not only achieve model lightweighting but also enhance detection accuracy, validating their effectiveness.

Table 2. Performance Comparison of Improved Backbone Networks.

Variant	mAP_{50}	$mAP_{50:95}$	Params(M)	GFLOPS
A	97.2%	56.3%	19.5	54.0
B	96.5%	55.2%	13.9	30.6
C	96.0%	54.7%	13.7	30.4
D	97.0%	55.0%	15.1	31.5

The results presented in Table 3 indicate that the modifications to the neck network are equally effective. Model A employs CCFM as the feature fusion network in our model, whereas Model B utilizes Slim-Neck. It is observed that, although Model B sacrifices detection accuracy to a certain extent, it achieves significant lightweighting of the neck network. Additionally, Model C connects the input nodes of the S4 and S5 feature maps to the corresponding concat modules in the bottom-up path to supplement potentially lost small object information. The results show improvements of 0.7% and 1.5% in mAP_{50} and $mAP_{50:95}$, respectively, validating the effectiveness of the modifications.

Table 3. Performance Comparison of Improved Neck Networks.

Variant	mAP_{50}	$mAP_{50:95}$	Params(M)	GFLOPS
A	96.6%	54.6%	15.6	34.7
B	96.3%	53.5%	15.1	31.4
C	97.0%	55.0%	15.1	31.5

The results in Table 4 indicate that the improvement to the loss function is effective. A denotes the utilization of GIoU as the loss function for our model. B represents the substitution of the GIoU loss function with the Shape-IoU loss function, which is rationally adjusted by focusing on the intricate shapes and scales of PCB defect features, ultimately leading to significant improvements in model accuracy, as measured by mAP_{50} and $mAP_{50:95}$. C introduces the concept of Inner-IoU into the Shape-IoU loss function, aiming to adaptively select appropriate auxiliary bounding box scales based on specific PCB defect detection tasks. Experimental results demonstrate additional enhancements in detection accuracy, as indicated by increased mAP_{50} and $mAP_{50:95}$ values, further strengthening the model's generalization capability.

Table 4. Performance Comparison of Improved Loss Functions.

Variant	mAP_{50}	$mAP_{50:95}$
A	96.2%	54.0%
B	96.3%	54.5%
C	97.0%	55.0%

To intuitively compare the performance of the proposed improved model with other mainstream object detection models, we conducted comparative experiments on the self-constructed EX-PCB dataset with our algorithm model, alongside YOLOv5m, YOLOv8m, YOLOv9m, YOLOv10m, and RT-DETR. The experimental results, presented in the following Table 5.

Table 5. Performance Comparison of Real-Time Models for PCB Defect Detection.

Method	mAP_{50}	$mAP_{50:95}$	Params(M)	GFLOPS
YOLOv5m	94.6%	54.9%	22.1	52.5
YOLOv8m	95.1%	55.2%	23.2	67.4
YOLOv9m[17]	95.7%	55.0%	16.6	60.0
YOLOv10m[18]	94.5%	54.2%	16.5	63.4
YOLOv11m	95.5%	55.1%	20.0	67.7
RT-DETR	96.9%	55.3%	19.9	57.2
ours	97.0%	55.0%	15.1	31.5

The experimental results indicate that, compared to the original RT-DETR model, the proposed model in this study achieves significant reductions of 23.8% and 44.9% in the number of parameters and computational cost, respectively, effectively mitigating parameter and computational redundancy. In terms of detection accuracy, although mAP_{50} increased by only 0.1% and $mAP_{50:95}$ decreased by 0.3%, the overall accuracy remained stable, demonstrating that the model lightweighting process

successfully retained the high detection accuracy performance of RT-DETR. Furthermore, compared to other real-time detection models with excellent performance, our model achieved significant improvements in the mAP_{50} metric, and the gap with the optimal model, Yolov8m, in the $mAP_{50:95}$ metric was only 0.2%. This fully demonstrates the advantages of our model in detecting small PCB defects. Simultaneously, the significant reductions in the number of parameters and computational cost indicate that our model exhibits higher parameter and computational efficiency.

3.5 Ablation Experiments

To thoroughly validate the improvement effects of the proposed network and loss function on the original RT-DETR model, this study conducted a series of ablation experiments on the EX-PCB dataset.The experimental results are presented in Table 6.

Table 6. Results of Ablation Experiments on the Self-Constructed EX-PCB Dataset.

Group	A	B	C	mAP_{50}	$mAP_{50:95}$	Params(M)	GFLOPS
1				96.9%	55.3%	19.9	57.2
2	√			96.2%	53.9%	15.6	34.7
3		√		96.5%	55.8%	19.5	54.0
4			√	96.9%	55.4%	19.9	57.2
5	√	√	√	97.0%	55.0%	15.1	31.5

Group 1: The baseline model, RT-DETR, achieved 96.9% mAP_{50} and 55.3% $mAP_{50:95}$ on a self-constructed dataset, with specific parameter and computational counts recorded.

Group 2: Introducing the ContextAdown-D-LKAEfficientFormerV2 backbone reduced mAP_{50} to 96.2% and $mAP_{50:95}$ to 53.9%, but significantly decreased parameter and computational costs.

Group 3: Applying the Bi-Slim-Neck feature fusion network on the baseline model resulted in 96.5% mAP_{50} and 55.8% $mAP_{50:95}$, with a slight drop in mAP_{50} but a notable increase in $mAP_{50:95}$, accompanied by reduced parameter and computational loads.

Group 4: Implementing the InnerShapeIoU loss function on the baseline model maintained mAP_{50} but elevated $mAP_{50:95}$ to 55.4%, representing a slight improvement over the original model.

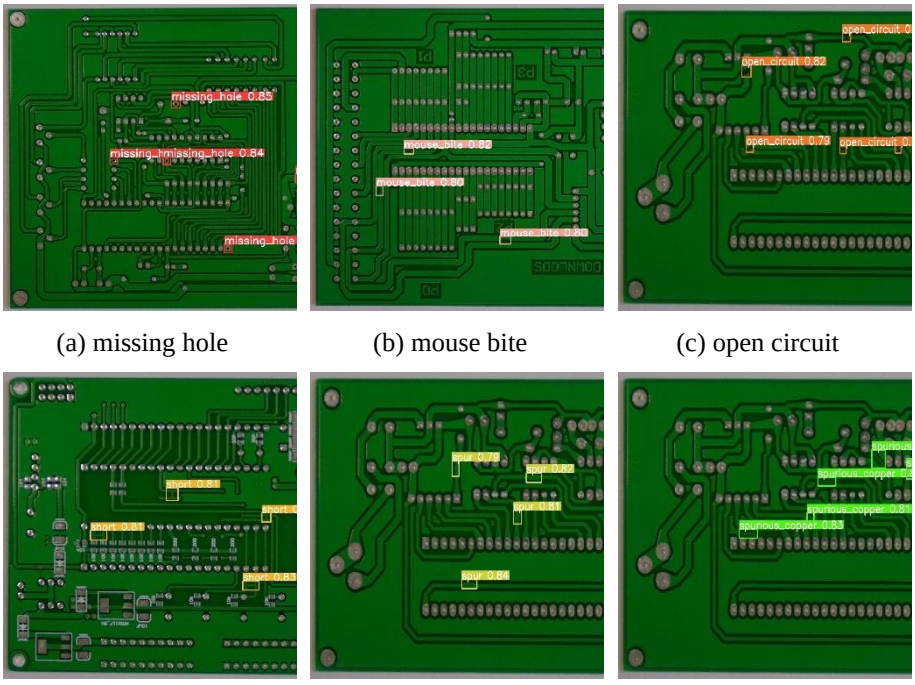
Group 5: Simultaneously introducing ContextAdown-Efficientformerv2, Bi-slim-neck, and InnerShapeIoU achieved 97% mAP_{50} and 55% $mAP_{50:95}$, maintaining high detection accuracy while significantly reducing parameter and computational costs compared to the original model. enhancing its practicality and efficiency in PCB defect detection.

The results indicate that the proposed modifications not only significantly streamlined model parameters and reduced computational load, but also maintained high

small-object detection accuracy. By addressing the challenges posed by small targets and the high parameter and computational costs associated with PCB defect detection, the integration of ContextAdown-Efficientformerv2, Bi-slim-neck, and InnerShapeIoU successfully lightweighted the RT-DETR model. This lightweighting was achieved without compromising detection accuracy, thereby effectively enhancing the model's overall performance.

3.6 Detection Effect Display

By leveraging an enhanced detection model to conduct inference on PCB defect images within the designated test dataset, six distinct types of PCB defect can be successfully and precisely detected. Representative images showcasing the ultimate detection outcomes are presented in the subsequent Fig. 5. These images provide a clear visualization of the identified defects, underscoring the model's capacity to precisely pinpoint and discern even minute imperfections on the PCB surface. These findings not only validate the model's efficacy in detecting a wide range of defect types but also demonstrate its reliability and accuracy in practical applications.



(a) missing hole (b) mouse bite (c) open circuit

(d) short (e) spur (f) spurious copper

Fig. 5. Sample results of detection for six types of PCB defect.

4 Conclusion

This study improves the RT-DETR model by incorporating the ContextAdown-D-LKAEfficientFormerV2 backbone, Bi-Slim-Neck feature fusion network, and Inner-ShapeIoU loss function. These enhancements enable effective detection of small PCB defects with a lightweight model. The backbone lightweights the model while enhancing small target feature extraction. The feature fusion network enhances fusion capabilities and reduces model parameters and computation. The loss function further improves detection accuracy and generalization. This study addresses challenges in PCB defect detection, including small target features and high model complexity, demonstrating the potential of deep learning in this field and providing a useful reference for future research.

References

1. R.A., Melnyk, R.B., Tushnytskyy: Detection of defects in printed circuit boards by clustering the etalon and defected samples. In: 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), pp. 961-964. IEEE (2020)
2. Cai, N., Ye, Q., Liu, G., Wang, H., Yang, Z. N.: IC solder joint inspection based on the Gaussian mixture model. In: Soldering & Surface Mount Technology, vol. 28, no. 4, pp. 207-214 (2016)
3. Wu, H., Zhang, X., Xie, H., Kuang, Y., Ouyang, G.: Classification of solder joint using feature selection based on Bayes and support vector machine. In: IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 3, no. 3, pp. 516-522. IEEE (2013)
4. Ding, R., Dai, L., Li, G., Liu, H.: TDD-Net: A tiny defect detection network for printed circuit boards. In: CAAI Transactions on Intelligence Technology. Intell. Technol., vol. 4, no. 2, pp. 110–116 (2019)
5. Hu, B., Wang, J.: Detection of PCB surface defects with improved faster-RCNN and feature pyramid network. In: IEEE Access, vol. 8, pp. 108335–108345. IEEE (2020)
6. Wang, X., Zhang, H., Liu, Q., Gong, W., Bai, S., You, H.: You-Only-Look-Once Multiple-Strategy Printed Circuit Board Defect Detection Model. In: IEEE MultiMedia, vol. 31, no. 1, pp. 76-87. IEEE (2024)
7. Wu, H., Tang, M.: PCB surface defect detection based on improved YOLOv7-tiny. In: 5th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pp. 334-337. IEEE (2023)
8. Chen, W., Huang, Z., Mu, Q., Sun, Y.: PCB Defect Detection Method Based on Transformer-YOLO. In: IEEE Access, vol. 10, pp. 129480-129489. IEEE (2022)
9. Li, Q., Wu, L., Xiao, H., Huang, C.: PCB-DETR: A Detection Network of PCB Surface Defect With Spatial Attention Offset Module. In: IEEE Access, vol. 12, pp. 158436-158445. IEEE (2024)
10. Zhao, Y., et al.: DETRs beat YOLOs on real-time object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16965-16974.IEEE (2024)
11. Li, Y. et al.: Rethinking vision transformers for mobilenet size and speed. In: IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16843-16854. IEEE (2023)

12. Azad, R., et al.: Beyond self-attention: Deformable large kernel attention for medical image segmentation. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1276-1286. IEEE (2024)
13. Li, H., Li, J., Wei, H., Liu Z., Zhan Z., Ren, Q.: Slim-neck by GSConv: A lightweight-design for real-time detector architectures. In: J. Real-Time Image Process., vol. 21, no. 3, art. no. 62 (2024)
14. Zhang, H., Zhang, S.: Shape-IoU: More accurate metric considering bounding box shape and scale. arXiv preprint arXiv: 2312.17663. (2023)
15. Zhang, H., Xu, C., Zhang, S.: Inner-IoU: More effective intersection over union loss with auxiliary bounding box. arXiv preprint arXiv: 2311.02877. (2023)
16. Huang, W., Wei, P.: A PCB dataset for defects detection and classification. arXiv preprint arXiv:1901.08204. (2019)
17. Wang, CY., Yeh, IH., Mark Liao, HY.: YOLOv9: Learning what you want to learn using programmable gradient information. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) ECCV 2024. LNCS, vol. 15089. Springer, Cham(2024)
18. Wang, A., et al.: YOLOv10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458. (2024)