iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.78746

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Lightweight PCB Defect Detection Using RepViT-Enhanced YOLOv8 with LAMP Pruning and Focaler-MPDIoU Loss

Imdadul Haque Enan, Zhang Dongliang, A.Haq, Hui Zhong
Faculty of Computer and Software Engineering, Huai'an University, Huai'an, Jiangsu 223001, China

Abstract: ***Printed circuit board (PCB) defect detection is critical for ensuring electronic product quality, yet it remains challenging due to small defect sizes, complex backgrounds, and the need for efficient deployment. This paper proposes a lightweight detection framework based on an improved YOLOv8 architecture to balance accuracy and efficiency. The model integrates RepViTm0.9 as the backbone to enhance small-target feature extraction and replaces the C2f module with RepNCSPELAN4 for more effective multi-scale feature fusion. Additionally, a Layer-Adaptive Magnitude-based Pruning (LAMP) strategy compresses the model by removing redundant channels, while a novel Focaler-MPDIoU loss function improves bounding box regression by focusing on hard samples and minimizing corner-point distances. Experimental results on a public PCB dataset demonstrate that the proposed model achieves 93.9% mAP@0.5 with only 1.27M parameters and 4.28G FLOPs, outperforming baseline YOLOv8n in accuracy while reducing computational cost by 48.9% and model size by 49.2%, offering an effective solution for real-world PCB inspection.***

Keywords: PCB defects; small targets; model pruning; lightweight network; loss function

I. INTRODUCTION

Printed circuit boards (PCBs) are fundamental components in modern electronic systems, providing the physical platform and electrical interconnections necessary for device functionality [1]. With the rapid advancement of electronics toward high-density integration and miniaturization, PCB structures have become increasingly complex, making defect detection a critical task for ensuring product reliability and manufacturing quality. Even minor defects such as open circuits, short circuits, spurious copper, burrs, or missing holes can significantly affect performance [2], leading to system failures and increased production costs. Traditionally, automated optical inspection (AOI) systems have been employed for PCB quality control [2]; however, early approaches based on image processing and handcrafted features exhibit significant limitations in real-world environments [3]. These methods are highly sensitive to noise, illumination variations, and alignment errors, and they lack the ability to generalize across diverse defect patterns and complex backgrounds [4], [5]. As a result, their effectiveness diminishes when dealing with subtle, small-scale, or low-contrast defects that commonly occur in modern PCB manufacturing.

The emergence of deep learning has transformed the landscape of PCB defect detection by enabling automatic feature extraction and hierarchical representation learning [6]. Convolutional neural networks (CNNs) and object detection frameworks such as Faster R-CNN [7] and the YOLO series have demonstrated significant improvements in detection accuracy and robustness compared to traditional approaches. Two-stage detectors, such as Faster R-CNN, provide strong localization capabilities [8] and have been successfully applied to PCB defect detection using feature pyramid networks to enhance multi-scale representation [9]. However, their relatively high computational cost limits their applicability in real-time industrial environments. In contrast, one-stage detectors such as YOLO have gained widespread adoption due to their fast inference speed and competitive accuracy, making them more suitable for deployment in automated inspection systems. Various YOLO-based methods, including YOLOv4-MN3 and subsequent improvements, have shown promising performance in detecting multiple PCB defect types simultaneously [10], [11]. Furthermore, recent studies have explored enhancements such as attention mechanisms, coordinate feature refinement, and context-aware learning to improve feature discrimination and detection performance [12], [13], [14].

Despite these advancements, PCB defect detection remains a challenging problem due to several inherent characteristics of defects [15]. PCB defects are typically small in size, densely distributed, and often exhibit low contrast against complex background textures, which makes them difficult to distinguish [16]. During the feature extraction process, repeated downsampling in deep networks can lead to the loss of critical spatial information, resulting in missed detections or inaccurate localization [17].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Additionally, the long-tail distribution of defect categories and limited availability of labeled data further complicate model training and generalization [18], [19]. To address these issues, researchers have proposed various strategies, including feature pyramid networks, multi-scale feature fusion, and attention-based modules to enhance the representation of small objects [20], [21], [22]. While these methods improve detection accuracy, they often introduce additional computational complexity, which can hinder real-time deployment in industrial settings.

In recent years, there has been a growing focus on lightweight models that aim to balance detection accuracy and computational efficiency. Lightweight YOLO-based frameworks, optimized backbone networks, and efficient feature fusion strategies have been proposed to reduce model size and inference cost while maintaining high performance. For example, LPCB-YOLO, YOLOv8-PCB, and other lightweight detection algorithms have demonstrated improved efficiency for PCB inspection tasks [23], [24], [25]. Similarly, recent works such as SCF-YOLO, PCB-YOLO, and SEPDNet have explored the integration of lightweight architectures with enhanced feature representation to achieve better performance under resource constraints [26], [27], [28]. Additionally, the development of large-scale datasets such as DsPCBSD and PCB-Defect has facilitated more robust evaluation and benchmarking of detection models, further advancing research in this field [29], [30]. Nevertheless, achieving an optimal trade-off between accuracy, robustness, and efficiency remains an open challenge, particularly for detecting small and complex defects in real-world industrial scenarios.

Overall, existing studies indicate that no single improvement strategy is sufficient to address all challenges simultaneously. While advanced feature extraction and attention mechanisms enhance detection accuracy, they may increase computational overhead. Conversely, lightweight models improve efficiency but may sacrifice performance, especially for small or low-contrast defects [31]. Furthermore, conventional bounding box regression methods based on IoU metrics may not adequately handle overlapping or densely distributed defects, leading to suboptimal localization performance [32], [33]. These limitations highlight the need for a comprehensive approach that integrates efficient feature extraction, enhanced multi-scale feature fusion, improved localization strategies, and model compression techniques within a unified framework.

Motivated by these challenges, the present study proposes a lightweight and high-performance PCB defect detection framework based on an improved YOLO architecture. The proposed method aims to enhance feature representation for small defects, improve multi-scale information fusion, optimize localization accuracy through an advanced loss function, and reduce computational complexity via model pruning. By addressing the key limitations identified in existing literature, this work seeks to provide a practical and scalable solution for real-world PCB inspection applications, achieving a better balance between detection accuracy and deployment efficiency.

II. METHODOLOGY

A. *Principles of the YOLOv8 Algorithm*

The YOLO family has shown clear strengths in object detection, particularly in terms of detection speed, accuracy, and versatility, which has led to its widespread adoption in industrial and everyday applications. Among these models, YOLOv8, released by Ultralytics in January 2023 as the successor to YOLOv5, introduces several architectural refinements that improve both efficiency and feature representation. As shown in Fig. 1, YOLOv8 replaces the C3 module used in YOLOv5 with the C2f module and integrates ideas from the Efficient Layer Aggregation Network (ELAN) structure proposed in YOLOv7. This modification reduces the use of standard convolutional layers while making fuller use of the Bottleneck module, thereby improving gradient flow without sacrificing the lightweight nature of the network [13]. Consequently, YOLOv8 is able to preserve computational efficiency while supporting richer information propagation during training.

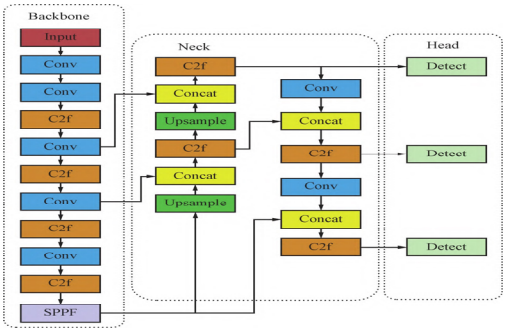


Fig. 1 Architecture of the YOLOv8 object detection model.

More specifically, YOLOv8 replaces the initial 6×6convolution in the backbone with a 3×3convolution and removes the 10th and 14th convolutional layers present in the YOLOv5 design. In the neck, the first 1×1convolution is also replaced with a 3×3convolution. In addition, YOLOv8 adopts a decoupled head that separates classification and detection tasks, which contributes to improved optimization and prediction quality. The final confidence score is obtained through a weighted combination of classification and regression outputs, and this score is then used to distinguish positive and negative samples more effectively. Based on different depth and width configurations, YOLOv8 is available in five variants, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, all of which exhibit strong performance. For PCB defect detection in particular, where conventional approaches often involve high computational cost and large model size, YOLOv8 offers a more efficient alternative by reducing the number of parameters and compressing the model while maintaining high detection accuracy.

B. Improved YOLOv8 Network

PCB surface defects often exhibit weak visual features and ambiguous boundaries, which makes them difficult to distinguish from the surrounding background and increases the likelihood of false positives and missed detections when conventional approaches are used. To address these challenges, the original YOLOv8 architecture is improved by redesigning both the backbone and neck components, as illustrated in Fig. 2. In the backbone, the conventional convolutional and C2f modules are replaced with the more efficient RepViT module, which provides stronger feature representation capability for small and subtle defect regions. In addition, downsampling is incorporated within the backbone to improve the extraction of discriminative features, thereby enhancing the network's ability to focus on defect-related information and reducing detection errors.

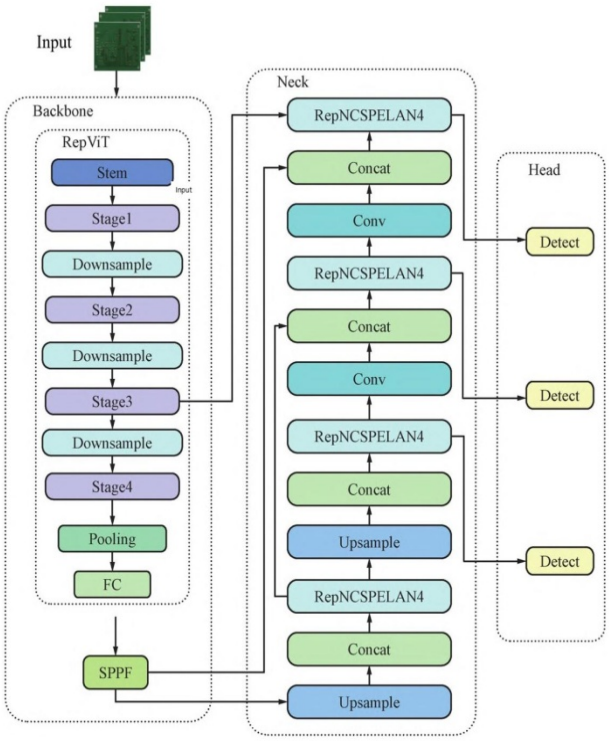


Fig. 2 Improved YOLOv8 architecture with RepViT backbone and RepNCSPELAN4 neck for PCB defect detection.

Further improvement is introduced in the neck network, where the original C2f module shows limited capacity for preserving fine-grained information associated with small-scale defects. This limitation not only weakens feature transmission but also contributes to higher computational complexity, parameter redundancy, and a larger model size, all of which hinder practical deployment. To overcome these issues, the RepNCSPELAN4 module is employed in place of C2f. This replacement enables more effective fusion of shallow and deep semantic features, reduces the loss of critical defect details, and strengthens the model's ability to detect small defects. Experimental results indicate that the modified architecture achieves improved accuracy and robustness in practical scenarios, thereby providing a more effective solution for PCB surface defect detection.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. RepViTm0.9

Due to the small scale and high density of PCB surface defects, the original YOLOv8 backbone shows limited effectiveness in detecting small targets, which can lead to frequent missed detections in complex scenes. To improve the discrimination between true defect regions and background noise, this study replaces the original backbone with RepViTm0.9. As shown in Fig. 3, RepViTm0.9 is a lightweight variant of the RepViT family, where the suffix "mX" denotes the extension scale and determines the number of repeated stacking layers in each stage. RepViT represents a lightweight architecture that combines the advantages of Vision Transformer design concepts with convolution-based network structures, making it suitable for efficient feature extraction in small-target detection tasks.

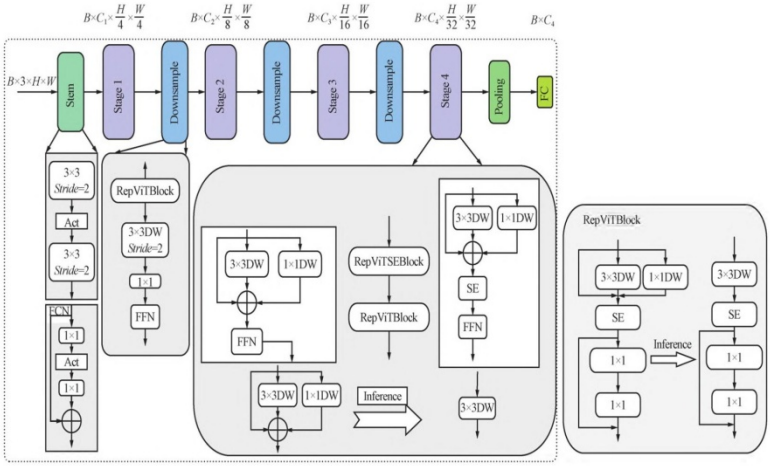


Fig. 3 Structural architecture of the RepViTm0.9 backbone network.

RepViT is mainly developed on the basis of MobileNetV3-L and adopts a fully convolutional design. Its structure begins with two 3×3convolutional layers with stride 2 as the main stem, followed by the application of the RepViT reparameterization strategy and the stacking of two 3×3depthwise convolutions with stride 1. The resulting feature maps are then upsampled by a factor of four, producing a series of pure convolutional models that can achieve performance comparable to that of lightweight Vision Transformer architectures. To further improve representational capability, Squeeze-and-Excitation (SE) modules are inserted into the odd-numbered layers of each stage, as illustrated in Fig. 4. This design strengthens channel-wise feature recalibration and helps compensate for the limited ability of standard convolution kernels to capture long-range dependencies.

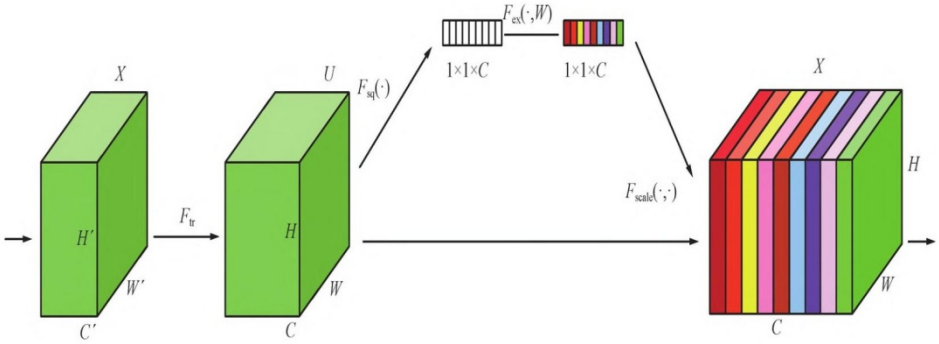


Fig. 4 Squeeze-and-Excitation (SE) module structure used in RepViT.

Overall, the adoption of RepViTm0.9 provides a more effective backbone for PCB defect detection by enhancing the extraction of small and densely distributed target features while maintaining a lightweight architecture. The integration of reparameterization and depthwise convolution improves computational efficiency, whereas the inclusion of SE modules enhances the network's sensitivity to informative defect features. As a result, the modified backbone is better suited to suppress background interference, reduce missed detections, and improve the overall robustness and accuracy of the detection framework in practical PCB inspection scenarios.

D. RepNCSPELAN4

PCB surface defects are typically characterized by weak feature representations and blurred boundaries, which makes them easily obscured by the surrounding background [16]. As a result, effectively preserving critical feature information remains a major challenge in PCB defect detection. In the YOLOv8 architecture, the C2f module in the head network primarily relies on conventional convolution operations for feature extraction. This design introduces redundant structural information, increases computational complexity, and often leads to false positives and missed detections, thereby degrading the mean Average Precision (mAP). To address these limitations, the RepNCSPELAN4 module is introduced as a replacement for the C2f module, as illustrated in Fig. 5.

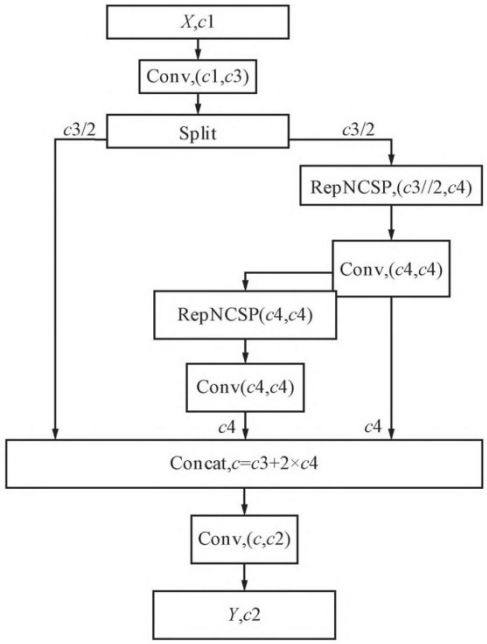


Fig. 5 Structural diagram of the RepNCSPELAN4 module for multi-scale feature fusion and enhanced feature representation.

Furthermore, as input data propagate through multiple layers of feature extraction and spatial transformation, the original information is progressively lost, resulting in biased gradient flows during model training. According to the information bottleneck principle, this phenomenon can be expressed as:

$$I(X,X) \geq I(X, f_\theta(X)) \geq I(X, g_\phi(f_\theta(X))) \quad (1)$$

where I denotes mutual information, f and g represent transformation functions, θ and ϕ are their corresponding parameters, and $f_\theta(X)$ and $g_\phi(f_\theta(X))$ correspond to two consecutive layers in a deep neural network. As indicated by Eq. (1), increasing network depth leads to greater information loss, which in turn affects gradient reliability and convergence during training. This incomplete information may cause the model to establish incorrect associations between inputs and targets, ultimately reducing prediction accuracy.

The RepNCSPELAN4 block [17] mitigates these issues by enhancing feature representation through splitting and merging operations on feature maps, which reduces redundant computation while improving feature extraction via layer aggregation. It incorporates convolutional layers to learn discriminative features and effectively preserve both spatial and semantic information. In addition, the module employs upsampling operations at different network stages to improve the spatial resolution of feature maps. By concatenating upsampled feature maps with those from earlier layers, multi-scale feature fusion is achieved, allowing the model to retain fine-grained details and spatial relationships. Consequently, the detection head can generate more accurate bounding boxes and class probabilities, leading to improved localization and recognition performance for PCB surface defects.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. LAMP-Based Pruning Algorithm

Model pruning is one of the most effective strategies for lightweighting deep neural networks, as it reduces model complexity by removing redundant components that consume substantial computational resources while contributing little to detection performance. Although the improved YOLOv8 network enhances the accuracy of PCB defect detection, it also introduces higher computational cost and a larger model size. To address this issue, the Layer-Adaptive Magnitude-based Pruning (LAMP) method [18] is employed to perform channel pruning on the improved detection model. This approach is straightforward to implement and requires only the adjustment of an acceleration ratio to control the pruning strength, as illustrated in Fig. 6.

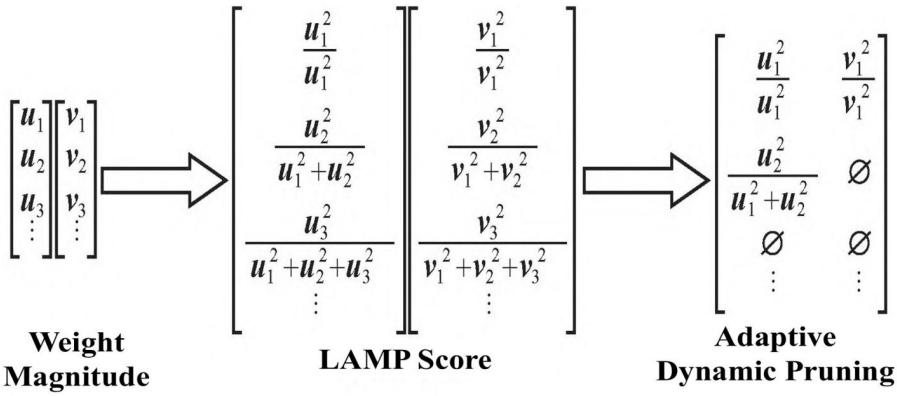


Fig. 6 Overall framework of the LAMP-based pruning process.

As shown in Fig. 6, the LAMP pruning strategy evaluates the importance of network connections according to their relative weight magnitudes within each layer. By assigning higher importance scores to more significant channels, the method is able to preserve critical feature extraction capability while removing less informative channels. This makes LAMP particularly suitable for reducing the computational burden of the improved YOLOv8 model without severely compromising detection accuracy.

The LAMP score is defined as:

$$\text{score}(u; W) := \frac{W[u]^2}{\sum_{v \geq u} W[v]^2} \tag{2}$$

and satisfies the relationship

$$W[u]^2 > W[v]^2 \Rightarrow \text{score}(u; W) > \text{score}(v; W) \tag{3}$$

where W denotes the weights of a given layer. In Eq. (2), the numerator $W[u]^2$ represents the squared magnitude of the target connection weight, whereas the denominator $\sum_{v \geq u} W[v]^2$ denotes the sum of squared magnitudes of the remaining connections within the same layer. The weights are sorted in ascending order according to a predefined index mapping. From this formulation, it can be observed that larger-magnitude weights correspond to higher LAMP scores and are therefore more likely to be retained during pruning.

The main procedure of LAMP pruning consists of four steps. First, the network weights are initialized using the pretrained base model. Second, the squared magnitudes of the connection weights are calculated and normalized to obtain the LAMP scores for each layer. Third, according to the LAMP scores and a predefined acceleration ratio, a corresponding number of connections are pruned. To prevent model collapse and ensure that the network retains its detection capability, at least one channel is preserved in each layer. Finally, the pruned model is fine-tuned to recover potential performance degradation caused by the pruning process. Through this procedure, the model achieves effective compression while maintaining most of its detection accuracy.

F. Focaler-MPDIoU Loss Function

The original YOLOv8 model employs the Complete Intersection over Union (CIoU) loss function. However, when applied to the dataset used in this study, CIoU leads to slower convergence and results in a significant number of missed detections and false positives. In PCB defect detection, defects are typically located near the center of bounding boxes, while background regions are distributed around them. The CIoU loss function fails to account for the influence of the distribution of hard and easy samples in bounding box regression, thereby limiting regression performance.

To address this issue, this study combines the ideas of Focaler-IoU [19] and Minimum Point Distance IoU (MPDIoU) [20] to propose a novel loss function, termed Focaler-MPDIoU. Both Focaler-IoU and MPDIoU improve bounding box regression by adaptively focusing on hard and easy samples, enabling more accurate characterization of the relative positional relationship between predicted boxes and ground-truth boxes. By integrating the advantages of these two loss functions, Focaler-MPDIoU becomes more sensitive to the positional information of PCB defects. Compared with CIoU, it significantly improves bounding box prediction accuracy and effectively reduces missed and false detections.

The Focaler-MPDIoU loss is defined as:

$$L_{\text{Focaler-MPDIoU}} = L_{\text{MPDIoU}} + \text{IoU} - \text{IoU}_{\text{Focaler}} \qquad (4)$$

where $\text{IoU}_{\text{Focaler}}$ denotes the reconstructed Focaler-IoU, and L_{MPDIoU} represents the MPDIoU loss.

The reconstructed Focaler-IoU is defined as:

$$\text{IoU}_{\text{Focaler}} = \begin{cases} 0, & \text{IoU} < d \\ \dfrac{\text{IoU} - d}{u - d}, & d \ll \text{IoU} \ll u \\ 1, & \text{IoU} > u \end{cases} \qquad (5)$$

where IoU represents the original Intersection over Union value, and $[d, u] \in [0,1]$. By adjusting the values of d and u, Focaler-IoU can focus on different regression samples. The corresponding loss function is given by:

$$L_{\text{Focaler-IoU}} = 1 - \text{IoU}_{\text{Focaler}} \qquad (6)$$

MPDIoU directly models the distances between the predicted bounding box and the ground-truth box at both the top-left and bottom-right corners. It is defined as:

$$\text{MPDIoU} = \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \qquad (7)$$

$$d_1^2 = (x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2 \qquad (8)$$

$$d_2^2 = (x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2 \qquad (9)$$

where A and B denote two arbitrary bounding boxes; w and h represent the width and height of the input image; (x_1^A, y_1^A) and (x_2^A, y_2^A) denote the coordinates of the top-left and bottom-right corners of box A, respectively; (x_1^B, y_1^B) and (x_2^B, y_2^B) denote those of box B; and d_1 and d_2 represent the Euclidean distances between the corresponding top-left and bottom-right corners of the two boxes. By adopting the Focaler-MPDIoU loss function, the model can effectively alleviate the distortion of bounding boxes in cases of overlapping defects, thereby reducing both missed detections and false positives in PCB defect detection.

III. EXPERIMENTS AND ANALYSIS

A. *Experimental Environment*

All experiments were conducted on an Ubuntu 22.04 operating system, using an NVIDIA RTX 3090 GPU with 24 GB of memory. The implementation was based on the PyTorch 1.13.1 deep learning framework with CUDA 11.6, and Python 3.9.18. The input image size was set to 640 × 640 pixels. The initial learning rate was 0.01, and the stochastic gradient descent (SGD) optimizer was employed. The model was trained for 300 epochs with a batch size of 33. The dataset was divided into training and validation sets with a ratio of 9:2.

B. *Dataset*

The dataset used in this study is the publicly available PCB defect dataset released by Peking University [21]. It contains 693 images suitable for defect detection tasks, covering six types of PCB defects: missing holes, mouse bites, open circuits, short circuits, spurs, and spurious copper.

Due to the limited number of samples, data augmentation techniques including cropping, brightness adjustment, and rotation were applied to expand the dataset to 5,766 images.

C. Evaluation Metrics

The evaluation metrics used in this study include computational complexity, number of parameters, model size, mean Average Precision (mAP), F1-score, and Precision (P). Among these, mAP, F1-score, and Precision are standard evaluation metrics in the COCO benchmark.

Let True Positive (TP) and False Positive (FP) denote correctly and incorrectly predicted positive samples, respectively, and False Negative (FN) denote missed positive samples. True Negative (TN) represents correctly predicted negative samples. False detection refers to incorrectly identifying a region as defective, while missed detection refers to failing to identify actual defects.

The evaluation metrics are defined as follows:

Precision (P):

$$P = \frac{TP}{TP + FP} \qquad (10)$$

Recall (R):

$$R = \frac{TP}{TP + FN} \qquad (11)$$

Mean Average Precision (mAP):

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \qquad (12)$$

F1-score:

$$F1 = \frac{2PR}{P + R} \qquad (13)$$

D. Experimental Results and Analysis

1) Pruning Experiment Comparison

To investigate the impact of pruning on the performance of the improved model, comparative experiments were conducted under different acceleration ratios. The acceleration ratio is defined as the ratio of the computational cost of the improved model to that of the pruned model. An acceleration ratio of 1.0 indicates that no pruning has been applied.

Before pruning, certain network structures that cannot be pruned are bypassed through layer-skipping operations. Since pruning may lead to varying degrees of accuracy degradation, fine-tuning is performed after each pruning step to recover lost performance. The experimental results are presented in Table 1.

TABLE I
EXPERIMENTAL COMPARISON UNDER DIFFERENT ACCELERATION RATIOS.

Acceleration Ratio	FLOPs (G)	Params (M)	Model Size (MB)	mAP@0.5 (%)	Precision (%)	F1 (%)
1.0	17.6	6.24	13.0	92.8	89.2	88.0
1.5	11.6	4.22	9.4	96.1	86.8	96.0
2.0	8.7	3.24	7.2	96.4	82.7	94.0
2.5	6.9	2.61	5.4	95.9	83.2	95.0
3.0	5.7	2.17	5.9	94.8	84.1	95.0
3.5	4.8	1.89	4.3	94.4	87.8	94.0
4.0	4.2	1.14	3.3	93.7	90.9	91.0
4.5	3.7	1.52	3.6	92.3	85.8	90.0
5.0	3.3	0.95	2.8	92.2	82.3	92.0
5.5	3.0	0.88	2.9	91.6	91.6	91.0
6.0	2.8	0.84	2.2	91.4	77.3	91.0

When the acceleration ratio is less than 4.0, the fine-tuned model achieves higher mAP@0.5 compared to the unpruned model, and the decline in mAP@0.5 shows a positive correlation with the increase in pruning ratio.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

When the acceleration ratio exceeds 2.5, all evaluation metrics outperform those of the original YOLOv8 model. At an acceleration ratio of 4.0, the mAP@0.5 remains higher than that of the improved (non-pruned) model, while the computational cost is reduced by 13.2 G, the number of parameters is reduced by 5.11 M, and the model size is reduced by 9.2 MB. Therefore, the pruning configuration with an acceleration ratio of 4.0 is selected as the final setting.

2) Ablation Study

To evaluate the contribution of each proposed module, ablation experiments were conducted by incrementally incorporating different components. Computational cost, parameter count, model size, and mAP@0.5 were selected as evaluation metrics.

Model1 represents the original YOLOv8 model. Model2 replaces the backbone with RepViT. Model3 replaces the C2f module in the head network with RepNCSPELAN4. Model4 replaces the CIoU loss with Focaler-MPDIoU. Model5 combines the modifications of Model3 and Model4. Model6 integrates all three improvements. The final model ("Ours") further applies LAMP pruning with an acceleration ratio of 4.0 based on Model5.

The results are shown in Table 2. After introducing RepViT (Model2), the computational cost decreases by 0.9 G and the parameter count is reduced by 0.4 M, while mAP@0.5 decreases slightly by 0.18%. After replacing the loss function, mAP@0.5 decreases by 1.58%, with no significant change in computational cost or parameter count.

TABLE II
ABLATION EXPERIMENT RESULTS.

Model	RepViT_m0.9	RepNCSPELAN4	Focaler-MPDIoU	FLOPs (G)	Params (M)	Size (MB)	mAP@0.5 (%)	Precision (%)	F1 (%)
Model1	—	—	—	8.2	3.01	6.3	90.08	89.0	89.0
Model2	√	—	—	7.3	2.61	5.6	89.90	81.5	87.0
Model3	—	√	—	20.3	7.51	15.7	90.14	84.2	89.0
Model4	—	—	√	8.1	3.01	6.3	88.50	86.1	89.5
Model5	—	√	√	19.1	6.70	15.7	91.40	88.2	90.0
Model6	√	√	√	17.6	6.28	13.3	92.40	85.7	89.0
Ours (Pruned)	√	√	√	4.2	1.17	3.1	93.90	90.6	93.0

When RepNCSPELAN4 is introduced (Model3), mAP@0.5 increases by 0.06%, but computational cost, parameter count, and model size all increase. Model4 shows that replacing the loss function alone leads to a 1.58% drop in mAP@0.5 with negligible changes in other metrics. Model5, which combines RepNCSPELAN4 and Focaler-MPDIoU, improves mAP@0.5 by 1.32% due to better bounding box regression, but does not reduce model complexity.

Model6, which integrates all improvements, achieves a 2.42% increase in mAP@0.5; however, its computational and structural complexity remains higher than the original YOLOv8. After applying LAMP pruning, the final model achieves a 3.87% improvement in mAP@0.5, while significantly reducing computational cost, parameter count, and model size.

3) Comparative Experiments

To demonstrate the effectiveness of the proposed method, comparative experiments were conducted against mainstream YOLO-based object detection models. The results are presented in Table 3.

The baseline YOLOv8n model has relatively low computational cost, parameter count, and model size, but its detection accuracy is limited. Although YOLOv8s improves mAP@0.5 by 2.62% compared to YOLOv8n, its computational cost, parameter count, and model size are approximately three times larger, making it less suitable for deployment. YOLOv7 exhibits the lowest mAP@0.5 and the highest computational and parameter overhead, indicating inferior detection performance.

TABLE III
COMPARATIVE EXPERIMENTAL RESULTS.

Method	FLOPs (G)	Params (M)	Model Size (MB)	mAP@0.5 (%)	F1 (%)	Precision (%)	Latency (ms)
YOLOv8n	8.25	3.21	6.1	90.18	89.0	89.0	565.8
YOLOv8s	28.67	11.20	22.65	92.64	90.0	87.7	228.5
YOLOv7	105.24	37.23	74.93	89.26	89.0	86.8	186.6

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Ref. [7]	—	—	63.91	96.78	—	—	—
Ref. [8]	—	—	61.5	93.27	—	—	—
Ref. [16]	—	6.17	23.3	99.28	—	—	142.4
YOLOv8 (Improved)	17.63	6.38	13.34	92.40	91.0	86.7	175.6
Ours (Pruned)	4.28	1.27	3.17	93.90	92.0	91.6	312.8

The method in [8] achieves higher mAP@0.5 than the baseline but remains inferior to the proposed method and has a larger model size. Methods in [7,16] achieve higher mAP@0.5 than both the baseline and the proposed model; however, their model sizes are significantly larger and do not satisfy lightweight requirements.

The proposed pruned YOLOv8 model does not achieve the highest mAP@0.5 overall, but it improves mAP@0.5 by 3.89% compared to the original YOLOv8n, while achieving the smallest computational cost, parameter count, and model size. Specifically, compared with YOLOv8n, the reductions reach 48.87%, 61.21%, and 49.19%, respectively. It is worth noting that the improved model before pruning introduces additional RepNCSPELAN4 layers, resulting in increased computational cost and redundancy, and thus higher latency compared to the original YOLOv8. After pruning removes redundant structures, latency is significantly reduced, although it remains slightly higher than the baseline. Overall, the proposed model satisfies the requirements of lightweight deployment.

4) Visualization of Experimental Results

To visually evaluate the effectiveness of the proposed method for PCB surface defect detection, a qualitative comparison between the original YOLOv8n model and the improved model was conducted, as illustrated in Fig. 7. It can be observed that the baseline YOLOv8n model suffers from issues such as missed detections, false detections, and relatively lower detection accuracy across all six defect categories. In contrast, the proposed model demonstrates more accurate and consistent detection results, effectively identifying defect regions while reducing both false positives and missed detections.

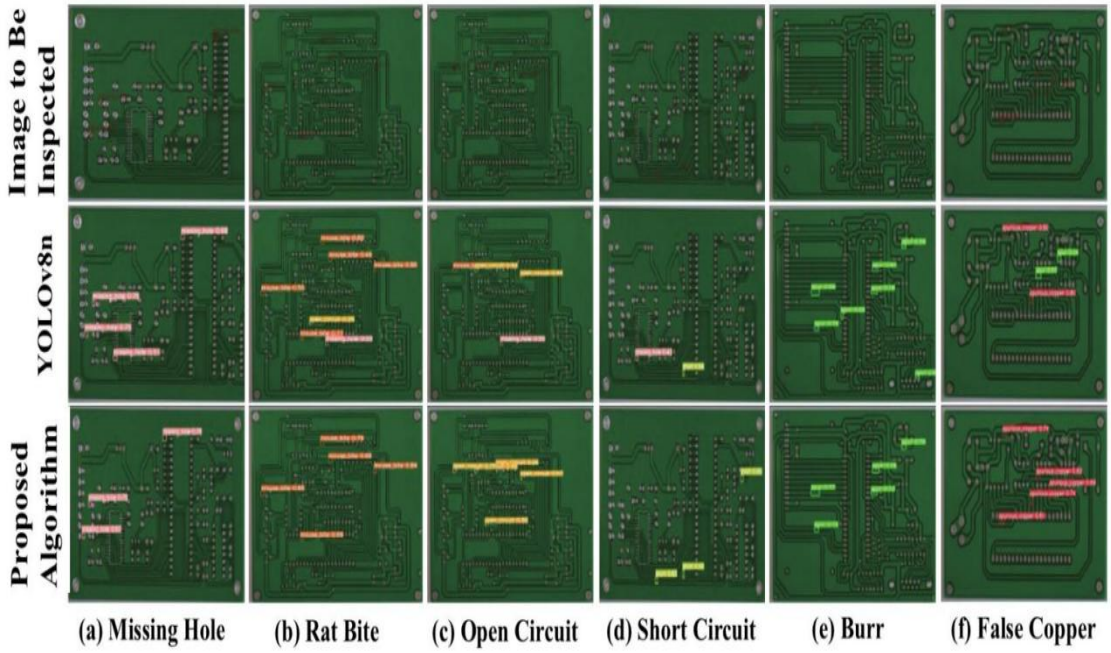


Fig. 7 Qualitative comparison of PCB defect detection results between YOLOv8n and the proposed model.

To further analyze the model performance from a feature representation perspective, heatmaps generated by the YOLOv8n model and the improved model are compared, as shown in Fig. 8. The visualization indicates that the improved model exhibits stronger activation in regions surrounding the defects, highlighting its enhanced ability to focus on relevant features. At the same time, it suppresses background interference more effectively than the baseline model, leading to clearer and more discriminative feature responses.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

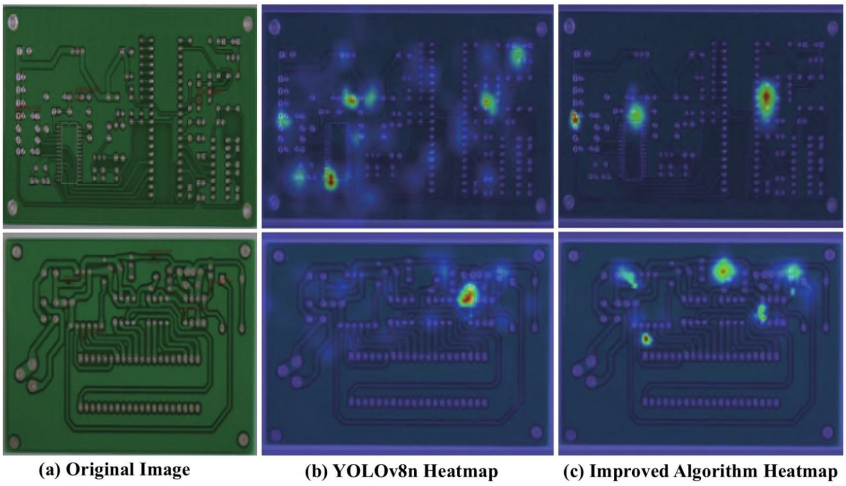


Fig. 8 Heatmap comparison between YOLOv8n and the proposed model for PCB defect detection.

By combining the qualitative comparison and heatmap analysis with the quantitative results presented earlier, it can be concluded that the proposed method significantly improves detection performance, particularly in small-target and complex-background scenarios. The improved model not only reduces the occurrence of missed and false detections but also achieves better localization and classification accuracy. Moreover, despite these performance gains, the model maintains a lightweight structure with fewer parameters and lower computational cost, making it more suitable for practical PCB inspection applications.

IV. CONCLUSION

In this study, a lightweight PCB defect detection framework was developed by enhancing the YOLOv8 architecture with a series of targeted improvements. The proposed model integrates RepViTm0.9 as the backbone to strengthen small-target feature extraction, replaces the original C2f module with RepNCSPELAN4 in the neck to enable more effective multi-scale feature fusion, and introduces the Focaler-MPDIoU loss function to refine bounding box regression by focusing on hard samples and minimizing corner-point distances. Additionally, a Layer-Adaptive Magnitude-based Pruning (LAMP) strategy is applied to remove redundant channels, significantly reducing computational complexity and model size while preserving detection accuracy. Experimental results on a public PCB defect dataset demonstrate that the proposed framework achieves an optimal balance between accuracy and efficiency, making it well-suited for real-world industrial deployment.

1) The final pruned model attains a mean Average Precision (mAP@0.5) of 93.9%, which is 3.89% higher than the baseline YOLOv8n.
2) Compared to YOLOv8n, the model reduces FLOPs by 48.9%, parameters by 61.2%, and model size by 49.2%, satisfying strict lightweight deployment requirements.
3) The integration of RepViTm0.9 and RepNCSPELAN4 effectively preserves fine-grained spatial information, leading to improved detection of small, densely distributed defects.
4) The Focaler-MPDIoU loss reduces missed and false detections by adaptively handling hard regression samples and modeling corner-point distances.
5) LAMP pruning removes redundant connections without significantly compromising performance, enabling a compact and efficient model suitable for resource-constrained environments.
6) Comparative experiments confirm that the proposed method outperforms several state-of-the-art approaches in terms of both detection accuracy and model compactness.

For future research, we intend to explore adaptive pruning techniques that dynamically adjust compression ratios based on layer sensitivity to further enhance efficiency while maintaining accuracy. Extending the proposed framework to other industrial inspection tasks—such as wafer defect detection, surface quality assessment for electronic components, or automated optical inspection in manufacturing lines—will be a valuable direction. Additionally, we plan to deploy the model on edge devices (e.g., Jetson or FPGA platforms) to evaluate real-time performance under practical constraints. Further improvements may also be pursued through the integration of knowledge distillation and post-training quantization to achieve even faster inference speeds while preserving detection precision.

REFERENCES

[1] A. Canal Marques, J.-M. Cabrera, and C. De Fraga Malfatti, 'Printed circuit boards: A review on the perspective of sustainability', *J. Environ. Manage.*, vol. 131, pp. 298–306, Dec. 2013, doi: 10.1016/j.jenvman.2013.10.003.

[2] M. T. Tran and S. H. Dau, 'The optical system design to support defect detection for the automatic inspection system on printed circuit boards on the production line', *Int. J. Interact. Des. Manuf. IJIDeM*, Feb. 2026, doi: 10.1007/s12008-026-02528-2.

[3] M. Ronchi, A. Regattieri, M. Gamberi, and C. Cafarella, 'Multi-criteria selection of image acquisition technologies for automatic optical inspection in assembly lines', *Int. J. Adv. Manuf. Technol.*, vol. 141, no. 12, pp. 5997–6016, Dec. 2025, doi: 10.1007/s00170-025-17021-5.

[4] Q. Ling and N. A. M. Isa, 'Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey', *IEEE Access*, vol. 11, pp. 15921–15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[5] Y. Zhou, M. Yuan, J. Zhang, G. Ding, and S. Qin, 'Review of vision-based defect detection research and its perspectives for printed circuit board', *J. Manuf. Syst.*, vol. 70, pp. 557–578, Oct. 2023, doi: 10.1016/j.jmsy.2023.08.019.

[6] J. Li, E. Hu, W. Wei, and F. Shi, 'From Wafers to Electrodes: Transferring Automatic Optical Inspection (AOI) for Multiscale Characterization of Smart Battery Manufacturing', *Adv. Funct. Mater.*, p. e28142, Mar. 2026, doi: 10.1002/adfm.202528142.

[7] A. Magdy, M. S. Moustafa, H. M. Ebied, and M. F. Tolba, 'Lightweight faster R-CNN for object detection in optical remote sensing images', *Sci. Rep.*, vol. 15, no. 1, p. 16163, May 2025, doi: 10.1038/s41598-025-99242-y.

[8] M. Li, Y. Hu, H. Chen, and X. Li, 'Feature-Enhanced Faster R-CNN for Small Object Detection in Complex Scenes', *IEEE Access*, vol. 13, pp. 174297–174315, 2025, doi: 10.1109/ACCESS.2025.3618019.

[9] B. Hu and J. Wang, 'Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network', *IEEE Access*, vol. 8, pp. 108335–108345, 2020, doi: 10.1109/ACCESS.2020.3001349.

[10] X. Liao, S. Lv, D. Li, Y. Luo, Z. Zhu, and C. Jiang, 'YOLOv4-MN3 for PCB Surface Defect Detection', *Appl. Sci.*, vol. 11, no. 24, p. 11701, Dec. 2021, doi: 10.3390/app112411701.

[11] V. A. Adibhatla *et al.*, 'Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once', *Math. Biosci. Eng.*, vol. 18, no. 4, pp. 4411–4428, 2021, doi: 10.3934/mbe.2021223.

[12] . Yang, Z. Liu, W. Du, and S. Zhang, 'A PCB Defect Detector Based on Coordinate Feature Refinement', *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023, doi: 10.1109/TIM.2023.3322483.

[13] J. Lim, J. Lim, V. M. Baskaran, and X. Wang, 'A deep context learning based PCB defect detection model with anomalous trend alarming system', *Results Eng.*, vol. 17, p. 100968, Mar. 2023, doi: 10.1016/j.rineng.2023.100968.

[14] B. Du, F. Wan, G. Lei, L. Xu, C. Xu, and Y. Xiong, 'YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5', *Electronics*, vol. 12, no. 13, p. 2821, Jun. 2023, doi: 10.3390/electronics12132821.

[15] S. Alawandi, K. Mallibhat, U. Kudachi, and A. Beedanal, 'Correction: PCB Defects: a Unified Survey of Trends, Detection Techniques, and Limitations Through Systematic Literature Review', *J. Electron. Test.*, vol. 41, no. 5–6, pp. 789–789, Dec. 2025, doi: 10.1007/s10836-025-06207-0.

[16] L. Zhu, S. Wang, M. Chen, Y. Zhu, K. Xing, and A. Shen, 'Subtle Defect Detection Network: More accurately detect subtle defects on the Printed Circuit Board surface', *Eng. Appl. Artif. Intell.*, vol. 152, p. 110765, Jul. 2025, doi: 10.1016/j.engappai.2025.110765.

[17] S. Luo *et al.*, 'YOLO-DAFS: A Composite-Enhanced Underwater Object Detection Algorithm', *J. Mar. Sci. Eng.*, vol. 13, no. 5, p. 947, May 2025, doi: 10.3390/jmse13050947.

[18] D. Bai *et al.*, 'Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s', *Eng. Appl. Artif. Intell.*, vol. 130, p. 107697, Apr. 2024, doi: 10.1016/j.engappai.2023.107697.

[19] M. Nikouei *et al.*, 'Small object detection: A comprehensive survey on challenges, techniques and real-world applications', *Intell. Syst. Appl.*, vol. 27, p. 200561, Sep. 2025, doi: 10.1016/j.iswa.2025.200561.

[20] K. C. Fung, K.-W. Xue, C.-M. Lai, K.-H. Lin, and K.-M. Lam, 'Improving PCB defect detection using selective feature attention and pixel shuffle pyramid', *Results Eng.*, vol. 21, p. 101992, Mar. 2024, doi: 10.1016/j.rineng.2024.101992.

[21] M. Yuan, Y. Zhou, X. Ren, H. Zhi, J. Zhang, and H. Chen, 'YOLO-HMC: An Improved Method for PCB Surface Defect Detection', *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024, doi: 10.1109/TIM.2024.3351241.

[22] T. Zhang, J. Zhang, P. Pan, and X. Zhang, 'YOLO-RRL: A Lightweight Algorithm for PCB Surface Defect Detection', *Appl. Sci.*, vol. 14, no. 17, p. 7460, Aug. 2024, doi: 10.3390/app14177460.

[23] H. Zhang, Y. Li, D. M. Sharid Kayes, Z. Song, and Y. Wang, 'Research on PCB defect detection algorithm based on LPCB-YOLO', *Front. Phys.*, vol. 12, p. 1472584, Jan. 2025, doi: 10.3389/fphy.2024.1472584.

[24] J. Wang, X. Xie, G. Liu, and L. Wu, 'A Lightweight PCB Defect Detection Algorithm Based on Improved YOLOv8-PCB', *Symmetry*, vol. 17, no. 2, p. 309, Feb. 2025, doi: 10.3390/sym17020309.

[25] S. Yu, F. Pan, X. Zhang, L. Zhou, L. Zhang, and J. Wang, 'A lightweight detection algorithm of PCB surface defects based on YOLO', *PLOS One*, vol. 20, no. 4, p. e0320344, Apr. 2025, doi: 10.1371/journal.pone.0320344.

[26] [26] Y. Li *et al.*, 'Lightweight PCB defect detection method based on SCF-YOLO', *PLOS ONE*, vol. 20, no. 4, p. e0318033, Apr. 2025, doi: 10.1371/journal.pone.0318033.

[27] Z. Wei, F. Yang, K. Zhong, and L. Yao, 'PCB-YOLO: Enhancing PCB surface defect detection with coordinate attention and multi-scale feature fusion', *PLOS One*, vol. 20, no. 6, p. e0323684, Jun. 2025, doi: 10.1371/journal.pone.0323684.

[28] D. Lang and Z. Lv, 'SEPDNet: simple and effective PCB surface defect detection method', *Sci. Rep.*, vol. 15, no. 1, p. 10919, Mar. 2025, doi: 10.1038/s41598-024-84859-2.

[29] S. Lv *et al.*, 'A dataset for deep learning based detection of printed circuit board surface defect', *Sci. Data*, vol. 11, no. 1, p. 811, Jul. 2024, doi: 10.1038/s41597-024-03656-8.

[30] A. J. Rashid, M. A. Ullah, A. Isfara, N. Ahmed, Md. M. Mian, and Md. M. Shalehin, 'PCB-defect: An annotated dataset for surface defect detection in printed circuit boards', *Data Brief*, vol. 64, p. 112296, Feb. 2026, doi: 10.1016/j.dib.2025.112296.

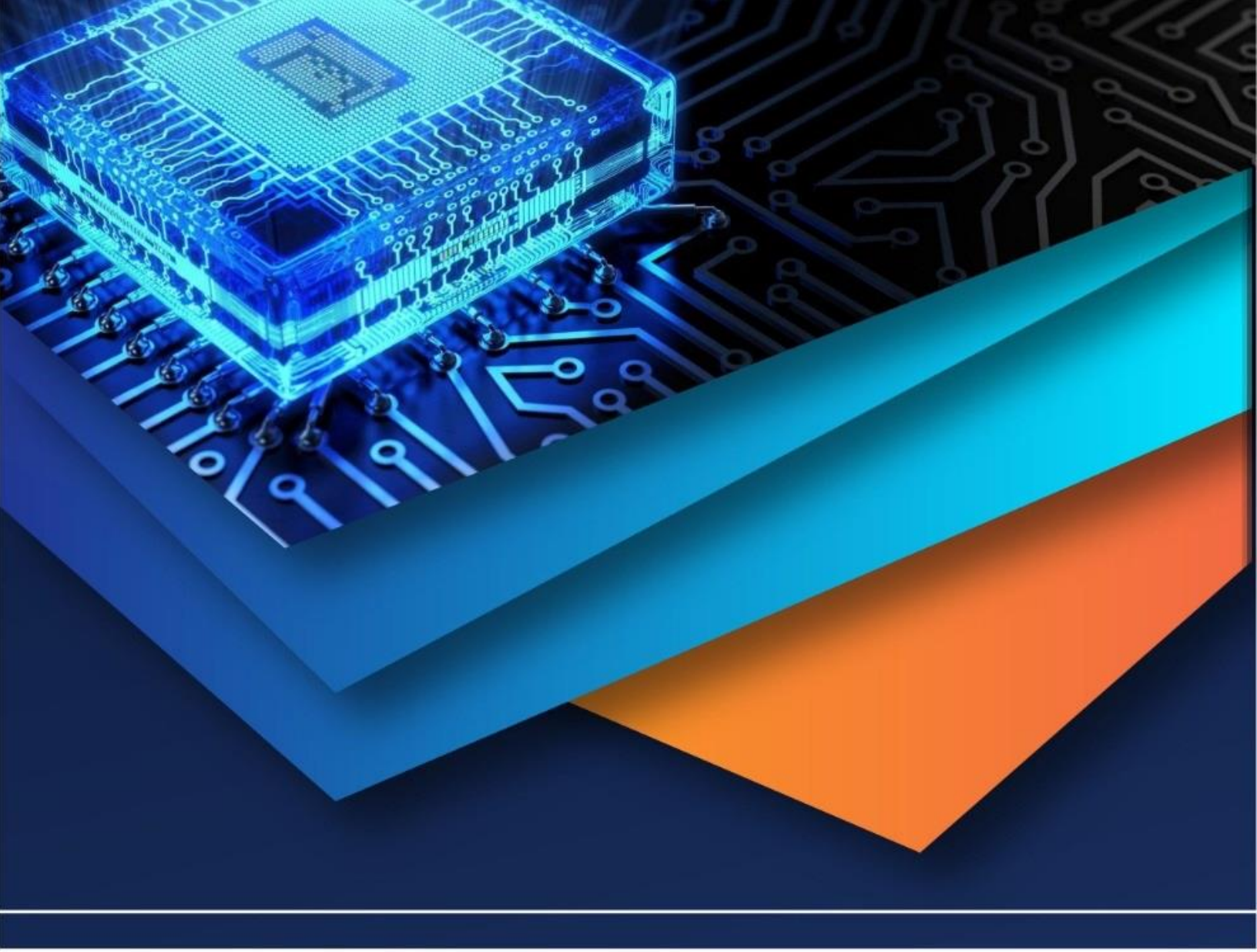
©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

[31] F.-C. Lin, H.-A. Wu, and J.-Y. Lin, 'A two-stage lightweight network for real-time IC surface defect detection', *Int. J. Adv. Manuf. Technol.*, vol. 142, no. 7–8, pp. 3933–3946, Feb. 2026, doi: 10.1007/s00170-025-17341-6.

[32] M. Xiao, B. Yang, S. Wang, F. Mo, Y. He, and Y. Gao, 'GRA-Net: Global receptive attention network for surface defect detection', *Knowl.-Based Syst.*, vol. 280, p. 111066, Nov. 2023, doi: 10.1016/j.knosys.2023.111066.

[33] N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu, and X. Liu, 'A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach With Application to Defect Detection', *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022, doi: 10.1109/TIM.2022.3153997.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)