Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 21, No. 1, 2026

A Real-Time PCB Defect Detection Framework on Raspberry Pi Based on YOLOv5n

Fan Huang, Wanrong Hui, Shuchang Wan, Xincan Wang , Haoxu Zhao and Lili Zhang

School of Information Engineering, Henan University of Science and Technology, Luoyang 471000, China

Abstract: PCB defect detection requires both high accuracy and real-time performance, while small defect sizes and complex backgrounds pose significant challenges for deployment on edge devices. This paper proposes a real-time PCB defect detection method based on the YOLOv5n model and constructs an edge-side detection system on Raspberry Pi 5. To enhance small defect detection capability, dataset integration, image slicing, and data augmentation strategies are employed. The trained model is deployed using the NCNN framework for lightweight inference. Experimental results demonstrate that the proposed method achieves 97.0% mAP50 on the test set and maintains a stable real-time detection speed of approximately 9 FPS on Raspberry Pi 5, showing strong performance on small-scale and sparse defects. The proposed approach provides a feasible solution for low-cost PCB defect detection on edge devices.

Keywords: PCB defect detection; Deep learning; Real-time detection; Raspberry Pi; YOLO; Edge computing; Micro-target detection.

1. Introduction

With the rapid development of Industry 4.0 and intelligent manufacturing, automated visual inspection technologies have been increasingly applied in electronic manufacturing. As a critical component of electronic products, the quality of printed circuit boards (PCBs) directly affects system reliability and safety. Therefore, achieving efficient and accurate PCB defect detection is of great engineering significance.

Traditional PCB inspection methods mainly rely on manual inspection or rule-based image processing techniques, which suffer from low efficiency and poor robustness. In recent years, deep learning-based object detection methods have significantly improved detection performance. Among them, the YOLO series has been widely used in industrial inspection scenarios due to its end-to-end architecture and high inference efficiency. However, PCB defects usually occupy only a very small portion of the image (approximately 1%–5%) and often exhibit complex shapes and low contrast. Meanwhile, edge devices such as Raspberry Pi have limited computational resources, making real-time industrial deployment particularly challenging.

To address these issues, this paper develops a real-time PCB defect detection system based on YOLOv5n and Raspberry Pi, providing a practical reference for industrial applications.

2. Related Works

In recent years, deep learning-based PCB defect detection methods have achieved significant progress. Early approaches mainly relied on traditional image processing techniques and handcrafted feature design, which often exhibited limited robustness under complex backgrounds and diverse defect patterns. With the advancement of object detection techniques, convolutional neural network-based methods have become the mainstream approach. Among them, the YOLO series has been widely adopted in industrial inspection tasks due to its end-to-end structure and efficient inference capability.

Previous studies have shown that introducing multi-scale feature fusion structures, such as Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), as well as improving backbone network design, can effectively enhance the detection performance for small defects.

However, existing methods still face two primary challenges: insufficient feature representation for small defects and the difficulty of balancing real-time performance and detection accuracy on resource-constrained devices. Therefore, it is necessary to develop a lightweight detection method suitable for edge deployment, which ensures high detection accuracy while meeting real-time requirements.

3. Methods

3.1. Dataset Construction and Preprocessing

This study utilizes two publicly available PCB defect datasets: the PKU PCB dataset and the DeepPCB-master dataset, comprising over 2,200 images in total. The datasets cover six common defect categories, including Missing_Hole, Mouse_Bite, Open_Circuit, Short, Spur, and Spurious_Copper, with significant variations in defect size, shape, and background texture. To construct a more comprehensive training set, the two datasets were merged and their annotation formats were unified to ensure training stability and reproducibility.

Regarding annotation consistency, the two datasets originally adopt XML and TXT formats, respectively, both based on bounding boxes defined by top-left and bottom-right coordinates. To adapt to the YOLO format, which represents bounding boxes using normalized center coordinates and width–height parameters, the annotations were converted using Eqs. (1)–(4). Here, (xmin , ymin) and (xmax , ymax) denote the top-left and bottom-right pixel coordinates of the bounding box, while W and H represent the image width and height. (cx , cy) denote the normalized center coordinates, and w, h denote the normalized width and height.

$$cx = (xmin + xmax) / 2W. \quad (1)$$

$$cy = (ymin + ymax) / 2H. \quad (2)$$

$$w = (xmax - xmin) / W. \quad (3)$$

$$h = (ymax - ymin) / H. \quad (4)$$

After conversion, a self-developed visual annotation tool was used for manual inspection and evaluation to ensure the accuracy and completeness of defect annotations (Fig. 1), thereby avoiding error accumulation caused by annotation inconsistencies.

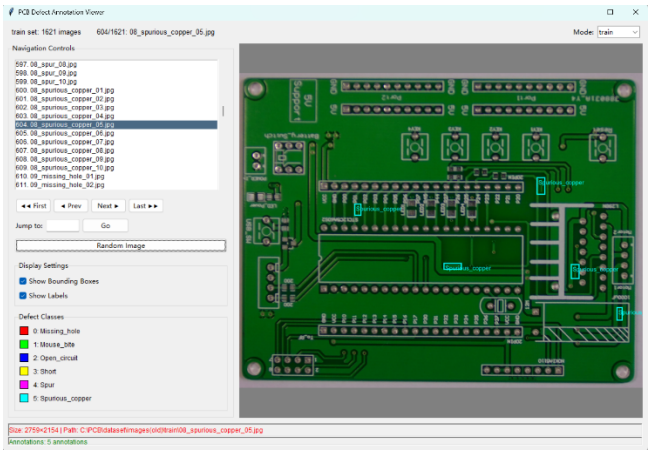

Fig. 1 Self-developed visualization tool for YOLO annotation inspection

To further expand the dataset and adapt to the input requirements of lightweight YOLO models, all images were resized to 640×640. For high-resolution or defect-dense PCB images, a self-developed slicing strategy was applied to generate multiple local patches (Fig. 2), ensuring that each defect is fully captured. The slicing process focuses on defect-centered regions with an adaptive overlap ratio of 30%–50% to prevent boundary information loss. Additionally, redundant samples were removed, and class distribution was balanced to mitigate the impact of class imbalance.

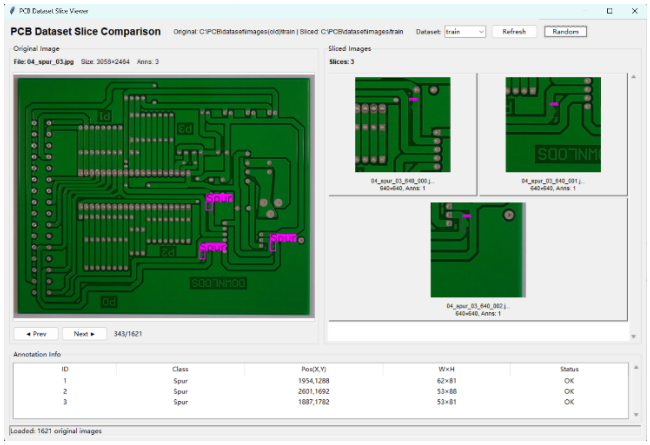

Fig. 2 Self-developed visualization tool of PCB image slicing process

3.2. YOLOv5n-Based Detection Framework

In this study, to achieve real-time PCB defect detection under constrained computational resources, both the hardware limitations of Raspberry Pi 5 and the deployment requirements of the NCNN inference framework were considered. The selected model must not only provide high detection accuracy but also ensure structural stability, strong operator compatibility, and low inference overhead. Compared with newer models that introduce complex components such as anchor-free decoupled heads, attention mechanisms, or end-to-end optimization strategies, YOLOv5n demonstrates superior maturity, engineering stability, and cross-platform compatibility. Therefore, YOLOv5n is selected as the baseline detection model for this study, and further optimizations are built upon it.

YOLOv5n adopts a CSP (Cross Stage Partial)-based backbone network combined with FPN and PAN for multi-scale feature fusion (Fig. 3), achieving stable feature representation under low computational complexity. By compressing the network depth and width (depth=0.33, width=0.25), the model significantly reduces parameter count and computational cost, making it suitable for deployment on resource-constrained devices such as Raspberry Pi 5.

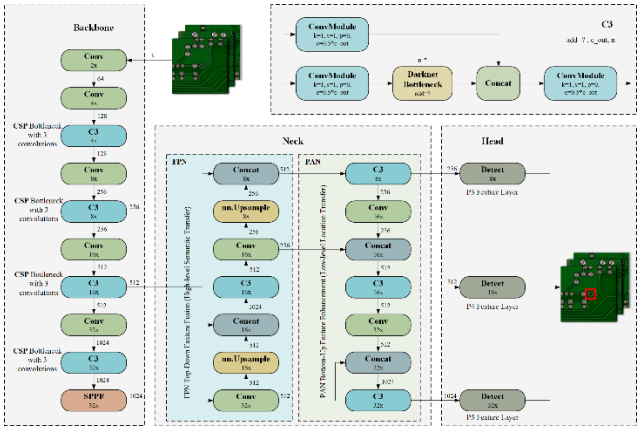

Fig. 3 YOLOv5 model structure

In terms of detection architecture, YOLOv5n employs an anchor-based multi-scale detection mechanism using P3–P5 feature layers. The P3 layer is particularly sensitive to small objects, enabling effective detection of fine-grained PCB defects. Additionally, its mature training and inference pipeline reduces engineering complexity and enhances system reliability. Overall, YOLOv5n achieves a favorable balance among lightweight design, detection performance, and deployment feasibility, making it well-suited for real-time PCB defect detection on edge devices.

3.3. Model Training Strategy

During training, the YOLOv5n model was optimized using the PyTorch framework. Considering the characteristics of PCB defects—small size, sparse distribution, and complex background—multiple data augmentation strategies were applied to improve detection performance and generalization capability. Mosaic augmentation increases the frequency of small objects by combining multiple images into a single input. MixUp and random geometric transformations (rotation, translation, and scaling) enhance robustness to diverse scenarios. HSV color augmentation simulates variations in industrial lighting conditions.

The AdamW optimizer was adopted with an initial learning rate of 0.001, combined with a cosine annealing schedule for stable convergence. The batch size and number of training epochs were set to 16 and 100, respectively. To prevent overfitting in early training stages, partial layer freezing was applied. Additionally, Focal Loss and CIoU Loss were used to address class imbalance and improve localization accuracy.

Model performance was evaluated after each epoch using Precision, Recall, and mAP metrics. Experimental reproducibility and stability were ensured by fixing random seeds and maintaining consistent data augmentation settings.

3.4. Edge Device Deployment

To enable real-time PCB defect detection on edge devices, the trained model was deployed on a Raspberry Pi 5 (4GB RAM) platform using the NCNN inference framework.

During deployment, the trained PyTorch model was converted into NCNN format to reduce memory consumption and improve inference efficiency. In the data acquisition stage, PCB images were captured in real time using a USB industrial camera. The images were resized to 640×640 and processed using OpenCV to ensure consistency with model input requirements.

In the real-time detection pipeline, input images were processed by the YOLO model to output defect categories and spatial coordinates. The detection results were visualized by overlaying bounding boxes and class labels on the original images. Meanwhile, results were recorded and organized into structured reports for subsequent analysis. Under continuous inspection conditions, the system is capable of processing multiple PCB image streams stably, meeting industrial requirements for real-time performance and reliability.

4. Results

4.1. Quantitative Results and Model Comparison

To evaluate the overall performance of the proposed method in terms of accuracy and real-time capability, quantitative experiments were conducted on the test set.

The results (Table 1) show that YOLOv5n achieves 97.0% mAP50, with Precision and Recall reaching 95.8% and 93.2%, respectively, indicating strong performance in both defect classification and detection. The mAP50-95 value is 67.2%, demonstrating that the model maintains relatively good localization performance under stricter evaluation criteria.

Table 1. Performance Comparison of YOLOv5n

Class	Precision	Recall	mAP50	mAP50-95
All	95.8%	93.2%	97.0%	67.2%
Missing Hole	95.3%	97.8%	98.6%	73.6%
Mouse Bite	98.6%	91.7%	98.2%	66.1%
Open Circuit	97.4%	94.3%	97.9%	62.2%
Short	88.4%	88.7%	92.8%	56.5%
Spur	97.3%	90.5%	96.1%	65.7%
Spurious Copper	98.0%	96.0%	98.7%	79.2%

From the per-class analysis, it can be observed that defects with clear structural characteristics, such as Missing_Hole and Spurious_Copper, achieve higher detection accuracy (mAP50 close to or exceeding 98%). In contrast, defects such as Short and Open_Circuit, which exhibit complex shapes or low contrast, show relatively lower performance under stricter localization metrics.

In terms of training dynamics (Fig. 4), the model converges rapidly during the early training stages. Precision and Recall stabilize after approximately 20 epochs, and mAP50 approaches 0.97 without significant overfitting, indicating the effectiveness of the adopted data augmentation and slicing strategies.

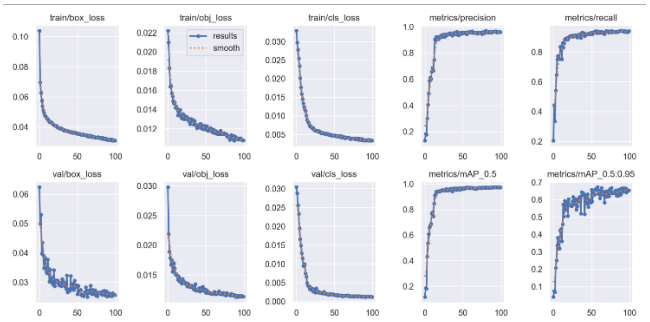

Fig. 4 Training curves of YOLOv5n (loss, precision, recall and mAP metrics)

On the Raspberry Pi 5 platform, the model achieves a stable inference speed of approximately 9 FPS, satisfying basic real-time detection requirements. Overall, YOLOv5n demonstrates a favorable balance between detection accuracy and computational efficiency.

4.2. Qualitative Results and System Validation

To visually evaluate the detection performance, real-time inference results (Fig. 5) and representative PCB defect images (Fig. 6) are presented. The results show that YOLOv5n can accurately detect all six defect categories with high spatial consistency between predicted bounding boxes and actual defect regions.

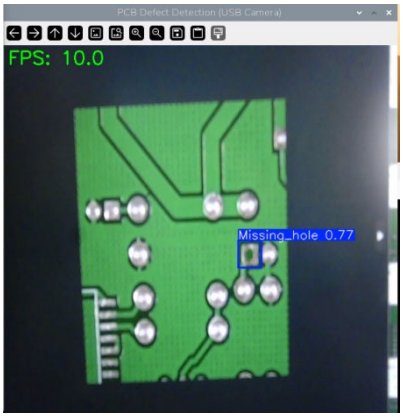

Fig. 5 Visual inference results of PCB defects on Raspberry Pi 5

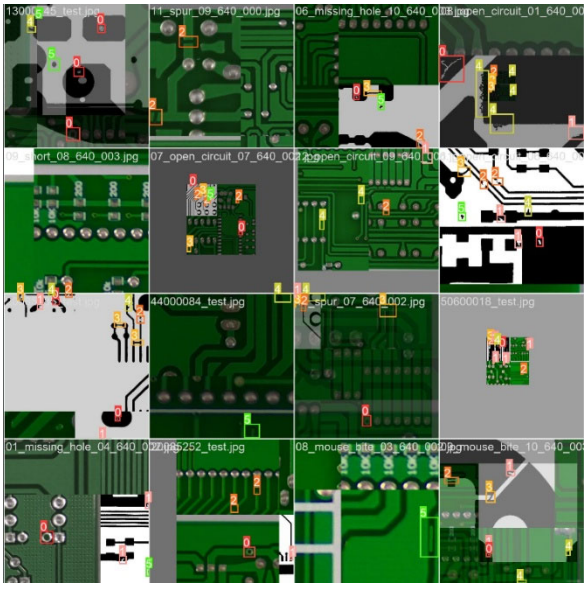

Fig. 6 Typical detection results of typical PCB defects by YOLOv5n

Under complex background conditions, the model effectively distinguishes defects from circuit textures, demonstrating strong robustness, especially for small and sparsely distributed defects. Different defect categories are visualized using distinct colors, providing intuitive representation of multi-class detection capability.

From a system perspective, a complete real-time detection pipeline was implemented on Raspberry Pi 5, including data acquisition, preprocessing, model inference, and result visualization. Experimental results show that the system maintains stable operation at approximately 9 FPS under continuous detection scenarios, without noticeable latency or frame loss.

5. Discussion

This study develops a lightweight PCB defect detection system based on YOLOv5n and validates its feasibility on Raspberry Pi 5. The results indicate that the proposed method achieves a good balance between accuracy and real-time performance, enabling stable operation on resource-constrained edge devices.

However, small-scale and low-contrast defects (such as Short and Open_Circuit) remain challenging due to weak texture features and interference from complex backgrounds. Data augmentation and image slicing strategies play a critical role in improving detection performance by increasing the visibility of small defects and mitigating class imbalance. Nevertheless, excessive augmentation may introduce redundancy and slightly affect training stability.

From a deployment perspective, YOLOv5n combined with NCNN achieves stable real-time inference on Raspberry Pi 5. However, higher-resolution inputs or more complex industrial environments may still be limited by hardware resources. Future work may explore model optimization techniques such as lightweight attention mechanisms, inference acceleration (e.g., INT8 quantization), and hardware upgrades.

6. Summary

This paper proposes a lightweight real-time PCB defect detection method based on YOLOv5n and constructs an edge-side detection system on Raspberry Pi 5, covering the complete pipeline from data processing to deployment.

Through dataset integration, image slicing, and data augmentation strategies, the proposed method effectively improves detection performance for small and sparse defects. Under unified experimental settings, the model achieves 97.0% mAP50 and maintains stable real-time inference at approximately 9 FPS on edge devices.

The main contributions of this work are summarized as follows:

(1) A unified PCB dataset integration and annotation pipeline is developed to ensure data consistency;

(2) A lightweight YOLOv5n-based detection framework is proposed for edge deployment;

(3) A real-time PCB defect detection system is implemented on Raspberry Pi 5, demonstrating a low-cost and practical industrial solution.

Future work will focus on improving lightweight feature representation and exploring deployment on more powerful edge platforms to enhance robustness and real-time performance in complex industrial environments.

Acknowledgements

The authors would like to express their sincere gratitude to their supervisor for valuable guidance and support in research direction, methodology selection, and experimental design. They also thank the laboratory members for their assistance in data collection, model training, and system deployment. In addition, the authors acknowledge the open-source community for providing YOLO models, and datasets, which offered essential technical support for this work.

References

[1] Wang Z, Wu J, Wang J. Research on the PCB defect detection algorithm based on YOLOv5. Journal of Project Management. 2024, Vol. 5 (No. 1), p. 235-237.

[2] Liu L, Zhang Y, Karimi HR. Defect detection of printed circuit board surface based on animproved YOLOv8 with FasterNet backbone algorithms. Signal, Image and Video Processing. 2025, Vol. 19 (No. 1), p. 89.

[3] Qin C, Zhou Z. YOLO-FGD: a fast lightweight PCB defect method based on FasterNet and the Gather-and-Distribute mechanism. Journal of Real-Time Image Processing. 2024, Vol. 21, p. 122.

[4] Zhuang J, Luo S, Hou C, et al. Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. Computers and Electronics in Agriculture. 2018, Vol. 152, p. 64-73.

[5] Information on: https://github.com/ultralytics/yolov5