ADPA-PCB: Enhancing PCB Defect Detection Neural Networks with Adaptive Activate Conv and PCBSAdd

ZhiNan Huang[1,2][0009-0004-0667-2140] , GuoPeng Zhou[2][0009-0006-5696-4687] and JianQuan Zhang[3,*] [0000-0003-1847-0337]

[1] Engineering Research Center of Hubei Province for Clothing Information,
Wuhan 430200
[2] School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200
[2] School of Electronics and Electrical Engineering, Wuhan Textile University, Wuhan 430200
[3] School of Automation, Hubei University of Science and Technology, XianNing 437000
*Corresponding author:zhangjianquan@hbust.edu.cn

Abstract. This paper presents a novel method for detecting defects in high-quality printed circuit boards (PCBs) used in electronic products. The effective operation of electronic devices heavily relies on the utilization of high-quality PCBs, making defect detection crucial to minimize wastage and financial losses. However, existing methods struggle to strike a balance between accuracy and speed, limiting their practicality in real-world scenarios. To address this challenge, we propose augmenting publicly available PCB image datasets with a broader range of defect types, enabling better simulation of real-world scenarios. This augmentation enhances the model's ability to learn features associated with PCB defects. Additionally, we introduce the Adaptive Activate Conv module to further improve the model's capacity for accurate defect detection. Experimental results using an expanded PCB dataset showcase outstanding performance with a noteworthy 2.1% increase in detection accuracy. Importantly, the proposed method maintains real-time applicability, underscoring its practical significance in industrial settings. By offering an efficient and accurate solution, our approach contributes to minimizing waste and ensuring product reliability in PCB manufacturing.

Keywords: Printed Circuit Boards (PCBs), Defect Detection, Adaptive Activate Conv, PCBSAdd, Deep Learning.

1 Introduction

With the rapid development and widespread use of modern electronic devices, Printed Circuit Boards (PCBs) have become crucial components. PCBs are responsible for implementing circuit functions and facilitating efficient signal transmission by connecting electronic components. However, the PCB manufacturing process is prone to various defects that can hinder the proper functioning of the circuit board or even lead to equipment failure, causing irreversible losses in industrial production. Therefore, the detection of defects in printed circuit boards is of utmost importance to ensure optimal performance and reliability.

In the initial stages of PCB surface defect detection, two primary approaches were utilized: manual visual inspection and mechanical inspection[15].Manual visual inspection relies on human operators to visually examine PCB panels for defects; however, it is constrained by subjective factors and frequently yields low detection accuracy. On the other hand, mechanical inspection encompasses contact-based methods, yet it proves to be time-consuming, labor-intensive, and inefficient.

During the third industrial revolution, the integration of electronics and information technology led to significant automation in the industrial sector. This integration gave rise to automatic optical inspection (AOI) [1]technology for PCB inspection[2]. AOI utilizes machine vision and optical equipment to capture image information, replacing manual visual inspection and improving speed and accuracy. However, the high costs associated with AOI technology pose challenges for its adoption in small and medium-sized manufacturing industries, limiting its use to large-scale and highly automated sectors.

Deep learning and computer vision techniques have significantly advanced PCB defect detection. Convolutional neural networks (CNNs) are widely used in this field[16], with two main categories of detection algorithms: single-stage and two-stage.Two-stage algorithms, such as R-CNN[13], Fast R-CNN[14], and Faster R-CNN[12], generate candidate regions of interest and perform classification for accurate detection. In contrast, single-stage algorithms like SSD[4] and YOLO[3] directly predict category probabilities and position coordinates, prioritizing detection speed.While two-stage algorithms achieve higher accuracy, they are slower compared to single-stage algorithms. Balancing speed and accuracy remains a challenge in deep learning object detection.Overall, the application of CNNs and deep learning techniques has propelled PCB defect detection. Researchers continue to refine algorithms to improve both accuracy and speed, meeting the requirements of industrial applications. The contributions of this article are summarized as follows:

1. A novel approach called "Adaptive Activation Conv" is proposed, which enables the network to dynamically learn when to activate neurons, thereby enhancing its learning capability.
2. This study introduces PCBSAdd, a novel addition function that incorporates a learning parameter. PCBSAdd enables the conversion of non-trainable tensors into trainable parameters and seamlessly integrates them into the optimization process of the host model. By combining PCBSAdd with the Bifpn structure, the network's ability to classify PCB defect features is significantly improved.
3. To cater to the requirements of defect detection in real-life scenarios, the current publicly available printed circuit board dataset has been expanded in terms of both quantity and diversity of defect types. This augmentation enhances the dataset's capability to address a wider range of defect detection challenges encountered in practical applications.

2 Related Work

Deep learning neural networks have shown promise in PCB defect detection. Researchers have proposed enhanced algorithms such as an improved Faster R-CNN with a multi-scale feature pyramid network [5], a lightweight deep neural network designed for efficient component detection [6], and a denoising convolutional autoencoder [7]. These advancements highlight the potential of deep learning in improving accuracy in PCB defect detection. Additionally, the YOLO series, known for its fast and accurate performance in object detection, has been extended for PCB defect detection with models like GSC YOLOv5, which incorporates attention mechanisms for precise defect detection [8]. Ongoing research is focused on further enhancing defect detection systems using sophisticated neural network architectures.

Niu et al. [9] and Tang et al. [10] proposed improved YOLOv5-based algorithms for PCB defect detection, refining anchor box matching, introducing novel loss functions, and incorporating attention mechanisms. However, current methods have limitations in capturing a comprehensive range of defects. To address this, we propose ADPA-PCB(Figure 1), a novel small object detection algorithm that integrates Adaptive Activate Conv, a new Add function, and utilizes new PCB datasets. ADPA-PCB aims to enhance the accuracy of PCB defect detection by overcoming existing limitations.

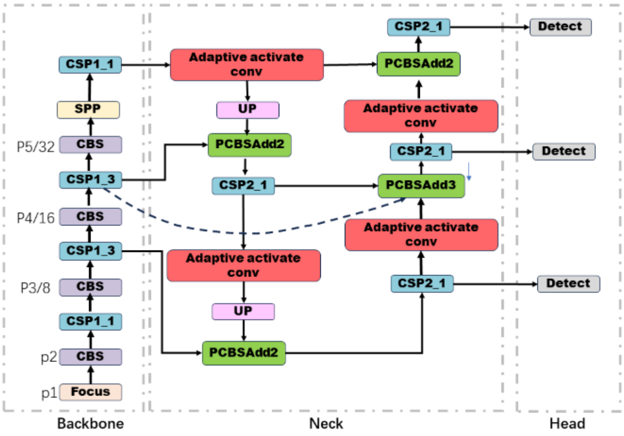


Fig. 1. ADPA-PCB. The proposed architecture comprises three main compo- nents: a feature map extraction layer serving as the backbone, a feature fusion layer acting as the neck, and a detection layer functioning as the head.

3 Proposed Method

The subsequent sections of this chapter are organized as follows: Section 3.1 offers an overview of the implementation principles behind AdaptiveThe remaining sections of this chapter are structured as follows: Section 3.1 provides an overview of the implementation principles of Adaptive Activate Conv. Section 3.2 describes the working principles of PCBSAdd. Lastly, Section 3.3 presents our supplementary contributions to existing datasets on defect types.

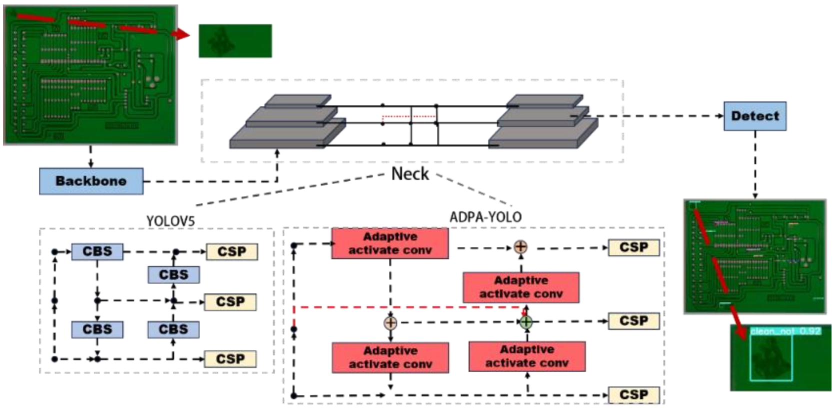


Fig. 2. Comparison of YOLOv5 and ADPA-PCB. During the feature extraction process, the CBS (Convolutional Block Attention Module) module was substituted with the Adaptive Activate Conv module, while introducing a novel feature weighting approach.

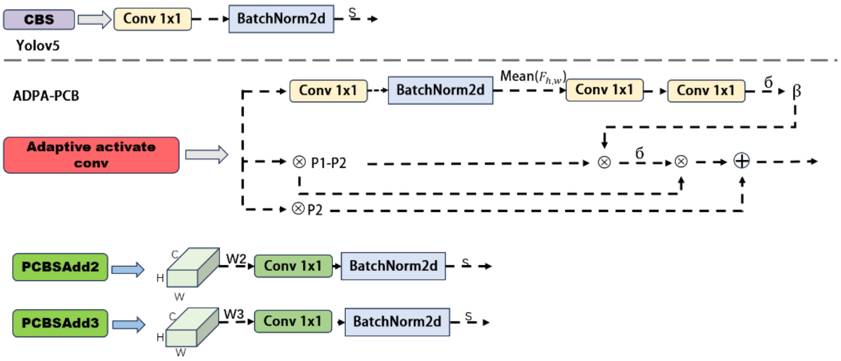


Fig. 3. The PCBSAdd and comparison of CBS and Adaptive activate conv. In the given context, the symbol "s" corresponds to the SiLU activation function. The notation "Mean($F_{h,w}$)" indicates the operation of averaging feature maps across the channels h and w. The symbol "б" denotes the sigmoid activation function. The variables "p1" and "p2" refer to two learnable parameters. The symbol "⊗" represents element-wise multiplication, while "⊕" signifies element-wise addition. Lastly, the parameter "β" is utilized to regulate the activation of neurons.

3.1 Adaptive Activate Conv

In this study, we propose a novel approach that enhances the CBS module by introducing additional operations and learnable parameters. Our approach incorporates feature map addition and multiplication, allowing for the extraction of more valuable information. We introduce learnable parameters initialized with random tensors, which are continuously adjusted and learned during training. Inspired by the ACON[11] activation function, we integrate its concept into our research and present a new convolution module called the Adaptive Activation Convolution module. This module combines convolution, batch normalization, and carefully designed activation functions to comprehensively enhance the performance of convolutional neural networks. The architectural design of the proposed module is illustrated in Figure 4.

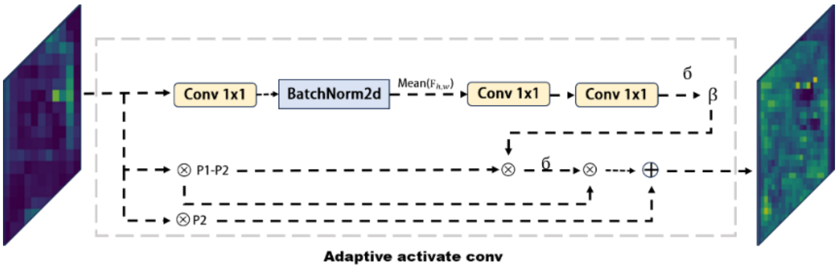


Fig. 4. Adaptive activate conv

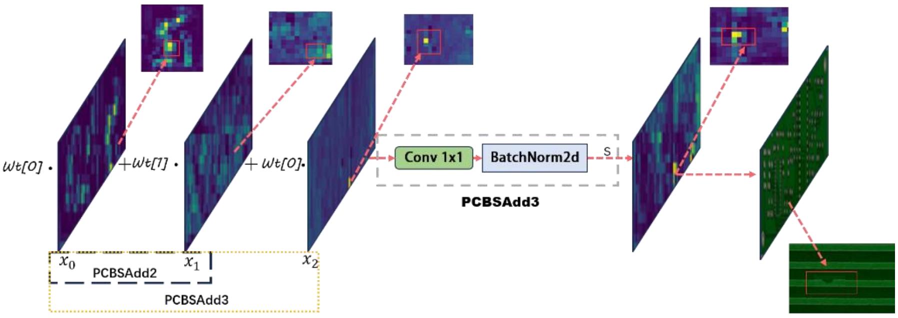


Fig. 5. PCBSAdd. where "wt[i]" refers to the i-th element of the computed weight vector, while "x_i" represents a feature map. The module incorporates the SiLU activation function denoted by "s" and signifies the weighted operation conducted on two feature maps.

Within the Adaptive Activate Conv structure, the input feature map (referred to as "input") undergoes a convolution operation to extract relevant features. Following this, the convolved feature map undergoes batch normalization, ensuring that the feature values conform to a normal distribution. This process can be formulated as follows:

$$F_{h,w} = BN\big(C1(input)\big) \tag{1}$$

C1 and C2 denote the convolution operations, while BN represents batch normalization operations. The parameter β regulates the activation of neurons. $\overline{F_{h,w}}$ signifies the average of the feature map across the dimensions h and w. Additionally, p1 and p2 are two learnable parameters. The visual representation of this module can be seen in Fig. 4. Consequently, we can summarize the reformulated formulation of the Adaptive Activate Conv module as follows:

$$\beta = \frac{C1C2(\overline{F_{h,w}})}{1+e^{-x}}$$

$$f_{Adapt\_act\_conv}(x) = (\mathrm{p1}\text{-}\mathrm{p2}) \cdot \mathrm{x} \cdot \frac{(\beta \cdot (p1-p2) \cdot x)}{1+e^{-x}} + \mathrm{p2} \cdot \mathrm{x} \tag{2}$$

3.2 PCBSAdd

We propose a novel fusion method called PCBSAdd as an improvement over existing techniques. The PCBSAdd method introduces learnable parameters called tensors, which actively participate in the backpropagation process and are trained through gradient updates. It combines convolution, batch normalization, and activation function operations to extract feature information, normalize feature distributions, and enhance non-linear transformation capability. Inspired by the Weighted Bi-directional Feature Pyramid Network, it achieves efficient multi-scale feature fusion, improving the network's ability to detect and recognize PCB defects. The proposed module structure is depicted in Figure 3.

We approach utilizes two weighting schemes for feature map fusion. We introduce a learnable parameter "w" to represent the weights used in the fusion process. During forward propagation, "w" undergoes normalization to ensure its suitability for weighting. The feature maps are then weighted based on the normalized "w," enhancing the informative content without increasing the dimensionality of the descriptors. Overall, our method involves the introduction of "w," its normalization, and the subsequent weighting of feature maps for effective fusion.

$$w = (w[0], w[1], w[2], \dots w[i-1]) \tag{3}$$

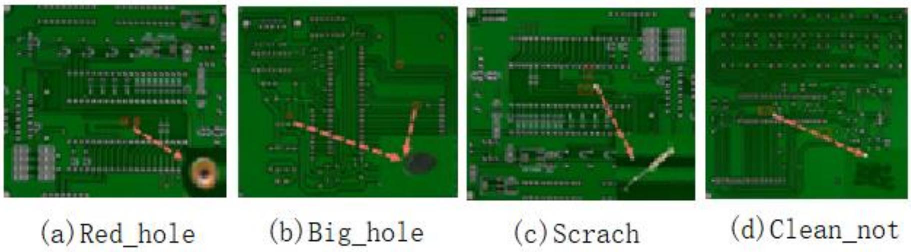
Where, "w[i]" denotes the i-th element of the vector "w," while "wt[i]" represents the i-th element of the calculated weight vector. The term "sum" refers to the summation of elements, and "a" represents a non-zero small positive value. The detailed implementation of PCBSAdd can be observed in Figure 5. The two weighting methods are as follows:

$$W[2] = wt[0] \cdot x[0] + wt[1] \cdot x[1]$$

$$W[3] = wt[0] \cdot x[0] + wt[1] \cdot x[1] + wt[2] \cdot x[2] \tag{4}$$

3.3 Expansion of public dataset defect types

The PCB Basic Defect Data Set, provided by Peking University, focuses on PCB defects and includes six defect types: missing_hole, mouse_bite, open_circuit, short, spur, and spurious_copper. However, existing circuit board defect classifications have limitations in precise detection and identification. To address this, the study collected real-world PCB defect types and enhanced the dataset using PS and Python technologies. Data annotation was performed with the Labelimg tool. New defect types, such as clean_not, big_hole, scratch, and red_hole(Figure 7), were introduced to improve the dataset's comprehensiveness. These additions enable more accurate detection and identification of PCB defects in real-world production scenarios.



(a)Red_hole (b)Big_hole (c)Scrach (d)Clean_not

Fig. 6. New PCB defect types.

4 Experiments and Analysis

4.1 Dataset and Environment

This article the PCB defects in the dataset are classified into 10 distinct categories, which include missing_hole, mouse_bite, open_circuit, short, spur, spurious_copper, clean_not, big_hole, scratch, and red_hole. The detailed statistics regarding the datasets and the corresponding defect types are presented in Figure. 8.

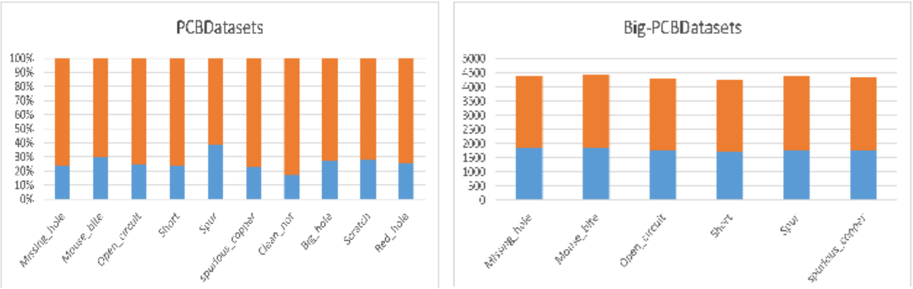


Fig. 7. Datasets and Defect Types. The figures present blue bars representing the relative proportions of PCB images associated with each specific defect type, while the orange bars indicate the distribution of defect labels across the various defect types.

To address variations in resolution and limited original image datasets, this study introduces four new defect types and employs rotation, noise, and scaling as image processing techniques for dataset augmentation. The dataset is expanded from 693 to 2,444 images and divided into training, validation, and testing sets in a 9:1:1 ratio. The learning rate is initialized to 0.01 and reduced using cosine annealing. A batch size of 16 is used, and the training process consists of 300 epochs. The experiments were conducted on a Linux system with an Intel quad-core i7-7700HQ CPU, 16GB of memory, and an NVIDIA GeForce GTX2080Ti graphics card with 11GB of memory.

4.2 Evaluation Method

The model's performance is evaluated through five metrics: Precision (P), Recall (R), Mean Average Precision (mAP), Detection Speed (FPS), and mAP. Precision (P) and Recall (R) are determined by considering the counts of False Positives (FP), True Positives (TP), False Negatives (FN), and True Negatives (TN) according to the following formulas:

$$P = \frac{T_P}{T_P + F_P} \quad R = \frac{T_P}{T_P + T_N} \tag{5}$$

Following that, precision-recall (P-R) curves were generated utilizing the obtained recall and precision values. The area under the curve was subsequently computed to quantify the average precision.

$$AP = \int_0^1 P(r)dr \quad mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{6}$$

4.3 Experimental Results

This study validated the model using two datasets: the augmented PCB dataset and BigPCB-Datasets,. The augmented PCB dataset consists of real circuit boards, while BigPCB-Datasets contains cropped images from complete circuit boards (10,668 images). While BigPCB-Datasets are more suited for academic research, the augmented PCB dataset proposed in this study better meets the defect detection requirements in real production scenarios.

In our experiments, we used the augmented PCB dataset and the BigPCB-Datasets dataset to train the YOLOv5 and ADPA-PCB models. The performance results on the augmented PCB test set are presented in Table 2. ADPA-PCB showed significant improvements over the original YOLOv5 model, achieving an average enhancement of 1.7% in precision (P), 2.8% in recall (R), and 2.6% in mean average precision (mAP). On the BigPCB-Datasets test set, as shown in Table 3, ADPA-PCB demonstrated a 0.4% improvement in precision (P), comparable recall rate (R) to YOLOv5, and an average increase of 0.3% in average precision (mAP). This highlights the superiority of ADPA-PCB in terms of precision (P) and average precision (mAP) over YOLOv5. Visualized detection results in Figure 10 demonstrate that ADPA-PCB outperforms YOLOv5 in identifying various defects, showing higher detection efficiency.

Table 1. The results of the ablation experiment, which involve the integration of two modules, namely Adaptive Activate Conv (AD) and PCBSAdd, into the YOLOv5 model, are presented.

Model	AD(head and backbone)	PCBSAdd	mAP@.5%	Parameters	GFLOPS
Baseline			92.0	28.46	17.4
AD-PCB	✓		90.1(-1.9)	31.01(+2.55)	17.4
PA-PCB		✓	93(+1.0)	28.30(-0.16)	17.5(+0.1)
ADPA-PCB	✓	✓	90.3(-1.7)	30.01(+1.55)	17.4
Model	AD(backbone)	PCBSAdd	mAP@0.5%	Parameters(M)	GFLOPS
Baseline			92.0	28.46	17.4

AD-YOLO	✓		91.7(-0.3)	28.61(+0.15)	17.4
PA-YOLO		✓	93(+1.0)	28.30(-0.16)	17.5(+0.1)
ADPA-PCB	✓	✓	90.7(-1.3)	28.50(+0.04)	17.5(+0.1)
Model	AD(head)	PCBSAdd	mAP@.5%	Parameters	GFLOPS
Baseline			92.0	28.46	17.4
AD-PCB	✓		93.2(+1.2)	28.55(+0.09)	17.4
PA-PCB		✓	93(+1.0)	28.30(-0.16)	17.5(+0.1)
ADPA-PCB	✓	✓	**94.1(+2.1)**	**28.43(-0.03)**	**17.5(+0.1)**

Table 2. The performance evaluation of YOLOv5 and ADPA-PCB is conducted using the PCB-Datasets. The abbreviations MH, MB, OC, SH, SP, SC, CN, BH, SCR, and RH correspond to the following defect types: missing_hole, mouse_bite, open_circuit, short, spur, spurious_ copper, clean_not, big_hole, scratch, and red_hole, respectively.

Model	yolov5	yolov5	yolov5	ADPA-PCB	ADPA-PCB	ADPA-PCB
	P	R	mAP	P	R	mAP
MH	0.878	0.985	0.967	0.89	0.986	0.98
MB	0.959	0.987	0.98	0.945	0.985	0.984
OC	0.772	0.97	0.872	0.764	0.972	0.879
SH	0.768	0.808	0.811	0.822	0.902	0.879
SP	0.775	0.885	0.864	0.779	0.943	0.899
SC	0.807	0.935	0.927	0.805	0.939	0.942
CN	0.923	0.964	0.992	0.945	1	0.992
BH	0.886	0.895	0.887	0.924	0.936	0.943
SCR	0.875	0.879	0.898	0.936	0.939	0.955
RH	1	0.983	0.996	1	0.966	0.996

Table 3. Performance of YOLOv5 and ADPA-PCB on Big_PCBDatasets.The abbreviations MH, MB, OC, SH, SP,and SC correspond to the following defect types: missing_hole, mouse_bite, open_circuit, short, spur, spurious_ copper respectively.

Model	class	MH	MB	OC	SH	SP	SC
Yolov5	P	0.977	0.974	0.973	0.986	0.977	0.982
Yolov5	R	0.997	0.997	0.996	0.982	0.994	0.995
Yolov5	mAP@0.5	0.99	0.991	0.99	0.984	0.99	0.992
ADPA-PCB	P	0.982	0.978	0.976	0.986	0.979	0.991
ADPA-PCB	R	0.997	0.996	0.996	0.982	0.99	0.999
ADPA-PCB	mAP@0.5	0.992	0.989	0.998	0.99	0.993	0.993

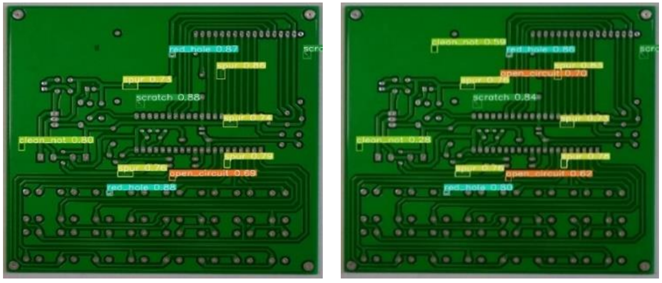


Fig. 8. Detection results on PCB datasets.Left:YOLOv5. Right:ADPA-PCB.

This study analyzed the effectiveness of two modules, Adaptive Activate Conv and PCBSAdd, in the network and conducted ablation experiments to evaluate their impact. The results, shown in Table 1, indicate the performance and parameter changes caused by these modules. Three sets of ablation experiments were performed, applying the Adaptive Activate Conv to different components of the model. Experiment 1 applied it to both the Backbone and Neck, Experiment 2 to the Backbone only, and Experiment 3 to the Neck only. Applying the Adaptive Activate Conv in the Backbone led to a small decrease of 0.3% in Map@0.5(%) performance and an increase of 0.15M parameters. To mitigate this, the PCBSAdd module was introduced in the Neck to reduce parameter count. The combined effect of these modules resulted in a net increase of 0.04M parameters and a corresponding decrease of 1.3% in Map@0.5(%) performance. These findings provide valuable insights into the impact of the Adaptive Activate Conv and PCBSAdd modules on performance and parameter count.

We study findings indicate that applying the Adaptive Activate Conv in the Backbone component did not yield significant improvements. Consequently, we decided to implement it in the Neck instead. This adjustment resulted in a modest increase of 0.09M parameters. However, it led to a notable improvement of 1.2% in Map@0.5(%) performance. Additionally, incorporating the PCBSAdd module in the Neck reduced the model's parameter count and contributed to an additional 1% increase in Map@0.5(%) performance. By simultaneously incorporating the Adaptive Activate Conv and PCBSAdd modules in the Neck, we achieved a reduction of 0.03M parameters while significantly enhancing the Map@0.5(%) metric by 2.1%. These findings demonstrate the qualitative impact of these modules on both performance and parameter count.

To ascertain the superiority of the proposed ADPA-PCB, we conducted a thorough comparison with several state-of-the-art detection algorithms, including the classical two-stage detection algorithm Faster R-CNN, as well as the single-stage detection algorithms RetinaNet, CenterNet, YOLOv5, YOLOv7, and YO-LOv8. Table 5 presents a comprehensive analysis of these algorithms, considering metrics such as mAP@0.5, mAP@0.5:0.95, FPS, and Model Size (MB) under identical settings. While the single-stage algorithm Faster R-CNN exhibits superior detection accuracy compared to the two-stage detection algorithms, its model size is relatively large, rendering it less suitable for practical detection scenarios.

The utilization of two-stage detection algorithms has proven effective in meeting the real-world demands for PCB defect detection, demonstrating a balance between speed and accuracy. Our proposed ADPA-PCB model surpasses the performance of the single-stage algorithms, namely RetinaNet, CenterNet,YOLOv5, YOLOv7, and YOLOv8, with notable improvements of 5.2%,3.5%,2.1%, 3.5%, and 2%, respectively, in terms of mAP@0.5. Furthermore, ADPA-PCB exhibits substantial enhancements in terms of FPS, achieving improvements of 0.46, 23.33, and 15.76, respectively, when compared to the YOLOv5, YOLOv7, and YOLOv8 algorithms.

Table 4. Comparative experimental results of PCB defect datasets in mAP(%), F1,FPS(Frames Per Second),P,R and model size.

Algorithm	mAP@0.5%	mAP@0.5:0.95%	FPS	Model Size (MB)
Faster R-CNN	95.32	64.5	21.5	629.27
RetinaNet	88.8	52.9	19.6	145
CenterNet	90.6	51.4	7.1	77.2
Yolov5s	92	49.2	67.11	14.5
Yolov7	90.6	45.4	44.24	71.43
Yolov8	92.1	50.6	51.81	21.50
ADPA-PCB	**94.1**	**48.7**	**67.57**	**14.51**

From the results of the comparative experiments, it is evident that the two-stage object detection algorithm Faster-RCNN, while achieving higher detection accuracy compared to both single-stage methods and the proposed ADPA-PCB in this paper, is hindered by its significantly larger model size and parameter count. This limits its deployment on devices in real-world production scenarios. In contrast, single-stage object detection algorithms such as RetinaNet and CenterNet, although lagging behind in terms of detection accuracy and frames per second (FPS), offer a more practical option. Additionally, the YOLOv5, YOLOv7, and YOLOv8 models employ a simple and flexible design philosophy, making them easy to comprehend and modify. Comparing the proposed ADPA-PCB algorithm with YOLOv(5,7,8), our approach exhibits superior detection accuracy and model compactness, enabling better deployment on real-world production devices.

Based on the aforementioned comparisons, it can be deduced that ADPA-PCB showcases superior performance in the realm of PCB defect detection.

5 Conclusion and Future Work

This article improves YOLOv5 for PCB defect detection. It introduces Adaptive Activate Conv, enhancing neuron activation for learning PCB defect characteristics. PCBSAdd enhances the addition method, expanding image information without increasing channels. It reduces computational complexity and achieves 94.1% mAP. The method maintains a small model size and is deployable on production equipment.

PCB defect detection is crucial in electronic manufacturing for quality control and fault identification. It also supports automation, robotics, IoT, and smart devices. Accurate detection improves efficiency, quality, and reliability. However, challenges exist, such as complex defects, false positives/negatives, limited generalization, computational demands, cost, implementation, and evolving defect forms. Research and collaborations are essential to overcome these limitations.

Future research in PCB defect detection includes exploring advanced machine learning techniques, multi-modal data fusion, optimizing deep learning models, real-time detection and automation, dataset establishment, and environmental adaptability. These efforts aim to improve accuracy, efficiency, and robustness. Research will focus on advanced algorithms and models integrating multiple data sources for real-time detection and automation. Establishing larger datasets with accurate annotations and industry collaborations will facilitate algorithmic research and practical applications. These research directions will drive breakthroughs in PCB defect detection technology.

Acknowledgments. This work is partially supported by Major Project of Scientific and Technological Innovation in Hubei (Grant No. 2023EGA023, 2022BBA026, 2021BGD022，2020BGC028) and Hubei Chutian Talent Program of Robot positioning and navigation.

References

1. Anitha, D. B., and Mahesh Rao. "A survey on defect detection in bare PCB and assembled PCB using image processing techniques." 2017 International conference on wireless communications, signal processing and networking (WiSPNET). IEEE, 2017.
2. Ling, Qin, and Nor Ashidi Mat Isa. "Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey." IEEE Access (2023).
3. Li, Jing, et al. "Application research of improved YOLO V3 algorithm in PCB electronic component detection." Applied Sciences 9.18 (2019): 3750.
4. Kang, Litian, et al. "Research on PCB defect detection based on SSD." 2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). IEEE, 2022.
5. Niu, Junhao, et al. "A PCB Defect Detection Algorithm with Improved Faster R-CNN." Proceedings of the International Conference on Big Data & Artificial Intelligence & Software Engineering, Guangzhou, China. 2022.
6. Li, Brian, et al. "PCBDet: An efficient deep neural network object detection architecture for automatic PCB component detection on the edge." Electronics Letters 60.2 (2024): e13042.
7. Khalilian, Saeed, et al. "Pcb defect detection using denoising convolutional autoencoders." 2020 International Conference on Machine Vision and Image Processing (MVIP). IEEE, 2020.
8. Wu, Ligang, Liang Zhang, and Qian Zhou. "Printed circuit board quality detection method integrating lightweight network and dual attention mechanism." IEEE Access 10 (2022): 87617-87629.
9. Niu, Jie, et al. "An improved YOLOv5 network for detection of printed circuit board defects." Journal of Sensors 2023 (2023).
10. Tang, Junlong, et al. "PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOv5." Sustainability 15.7 (2023): 5963.
11. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).
12. Zhou, Yongbing, et al. "Review of vision-based defect detection research and its perspectives for printed circuit board." Journal of Manufacturing Systems 70 (2023): 557-578.
13. Bhanumathy, Y. R., et al. "Defect detection in PCBs using convolutional neural network." 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). IEEE, 2021.
14. Zhang, Yi-Fan, et al. "Focal and efficient IOU loss for accurate bounding box regression." Neurocomputing 506 (2022): 146-157.
15. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
16. Ahmed, Mohiuddin, Raihan Seraj, and Syed Mohammed Shamsul Islam. "The k-means algorithm: A comprehensive survey and performance evaluation." Electronics 9.8 (2020): 1295.