

## EVALUATION OF PRINTED CIRCUIT BOARD DEFECT DETECTION ALGORITHMS BASED ON FASTER R-CNN AND YOLOV8

Pham Thu Ha, Luong Van Bang, Le Hoang Phuc, Ta Chi Hieu, Dao Thi Nga\*

Le Quy Don University

ARTICLE INFO		ABSTRACT
Received:	24/01/2025	Detecting printed circuit board defects is one of the critical tasks to ensure the quality of electronic devices, especially as printed circuit board sizes become increasingly compact, leading to higher demands for accuracy and speed in defect detection in industrial manufacturing. However, traditional inspection methods are time-consuming and inefficient for modern printed circuit boards. In recent years, deep learning techniques have demonstrated superior capabilities in detecting and classifying printed circuit board defects, providing a robust alternative to conventional methods. This paper presents an enhancement to the baseline model by incorporating modern techniques to analyze data in object detection tasks. By separately approaching the two models, Faster R-CNN and YOLOv8, we experimented with and compared their performance. Experimental results indicate that both models achieve promising performance; however, Faster R-CNN excels in accuracy accuracy (Faster R-CNN – 99.6%; YOLOv8 – 95%), while YOLOv8 stands out for its speed speed (Faster R-CNN – 1.37s/frame; YOLOv8 – 0.26s/frame).
Revised:	11/03/2025	
Published:	19/03/2025	
<b>KEYWORDS</b>		
Deep learning		
Computer vision		
Faster R-CNN and YOLOv8		
Defect detection		
Printed circuit board		

## ĐÁNH GIÁ CÁC THUẬT TOÁN PHÁT HIỆN LỖI BẢNG MẠCH IN DỰA TRÊN FASTER R-CNN VÀ YOLOV8

Phạm Thu Hà, Lương Văn Bằng, Lê Hoàng Phúc, Tạ Chí Hiếu, Đào Thị Nga\*

Trường Đại học Kỹ thuật Lê Quý Đôn

THÔNG TIN BÀI BÁO		TÓM TẮT
Ngày nhận bài:	24/01/2025	Phát hiện lỗi bảng mạch in là một trong những nhiệm vụ quan trọng để bảo đảm chất lượng của các thiết bị điện tử, đặc biệt khi kích thước bảng mạch in ngày càng nhỏ gọn dẫn đến những yêu cầu ngày càng cao về độ chính xác và tốc độ trong sản xuất công nghiệp. Tuy nhiên, các phương pháp kiểm tra truyền thống gây tốn thời gian và kém hiệu quả đối với các bảng mạch in hiện đại. Trong những năm gần đây, các kỹ thuật học sâu đã chứng minh khả năng vượt trội trong việc phát hiện và phân loại lỗi bảng mạch in, mang đến một giải pháp thay thế mạnh mẽ cho các phương pháp truyền thống. Bài báo này trình bày một cải tiến mô hình cơ sở bằng cách kết hợp các kỹ thuật hiện đại để phân tích dữ liệu trong bài toán phát hiện đối tượng. Bằng việc tiếp cận riêng biệt hai mô hình Faster R-CNN và YOLOv8, chúng tôi đã thử nghiệm và so sánh hiệu suất của hai mô hình. Kết quả thí nghiệm cho thấy Faster R-CNN có ưu thế về độ chính xác (Faster R-CNN - 99,6%; YOLOv8 – 95%), trong khi YOLOv8 nổi bật về tốc độ (Faster R-CNN – 1,37 s/khung hình; YOLOv8 – 0,26 s/khung hình).
Ngày hoàn thiện:	11/03/2025	
Ngày đăng:	19/03/2025	
<b>TỪ KHÓA</b>		
Học sâu		
Thị giác máy tính		
Faster R-CNN và YOLOv8		
Phát hiện lỗi		
Bảng mạch in		

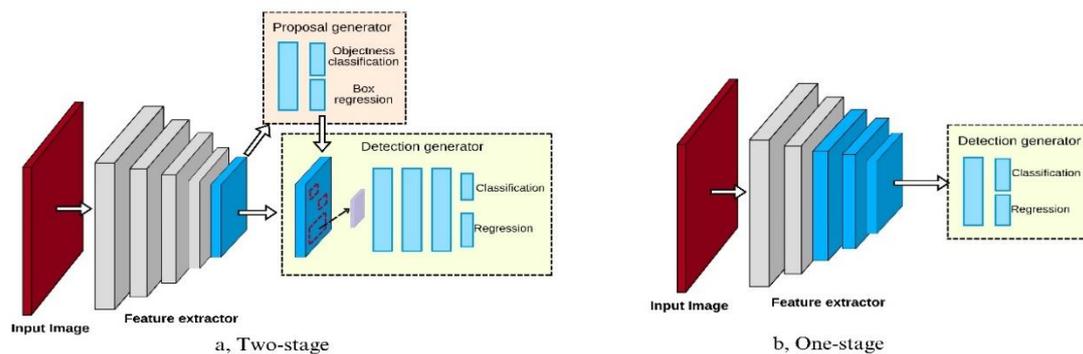
DOI: <https://doi.org/10.34238/tnu-jst.11949>

\* Corresponding author. Email: [daothinga@mta.edu.vn](mailto:daothinga@mta.edu.vn)

## 1. Introduction

Printed circuit board (PCB) is a fundamental component in electronic devices, providing both physical support and circuit connectivity. The PCB manufacturing process involves multiple stages, such as cutting, drilling, copper plating, etching, and electrical testing, which are prone to defects such as broken circuits, short circuits, and over-etching. Traditional defect detection methods, such as manual inspection and functional testing, are limited by low efficiency, high costs, and susceptibility to human error [1]. As PCBs become increasingly compact and complex due to advancements in semiconductor technology, the demand for precise and efficient defect detection methods has grown [2], [3].

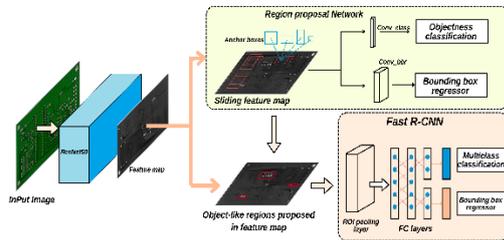
Deep learning has emerged as a promising solution for PCB defect detection [4], particularly in Automatic Optical Inspection (AOI) systems [5]. Deep learning models are typically categorized into one-stage (e.g., YOLO [6], [7], SSD [8]) and two-stage (e.g., Faster R-CNN [9], [10], R-CNN [11]) frameworks. As shown in Figure 1, one-stage models integrate object localization and classification within a single network, offering high speed and real-time performance. In contrast, two-stage models utilize a region proposal network (RPN) for region generation followed by object classification, achieving higher accuracy but slower speed. Among these, Faster R-CNN excels in detecting complex defects due to its high accuracy, while YOLOv8 offers superior real-time performance with an optimized design [12]. These models are increasingly integrated into AOI systems to improve the reliability and efficiency of automated PCB defect detection processes.



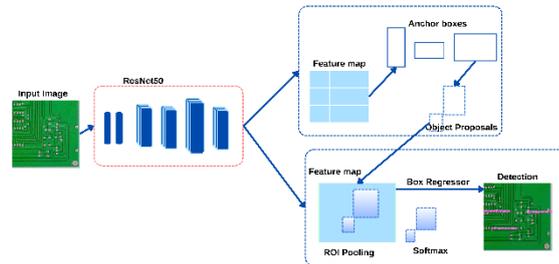
**Figure 1.** Comparison between one-stage and two-stage model architectures

Faster R-CNN (short for "Faster Region Convolutional Neural Network") is an advanced object detection architecture in the R-CNN family with a two-stage detection framework. The primary goal of this model is to develop a unified architecture capable of detecting objects in images and precisely locating them. The architecture of Faster R-CNN is depicted in Figure 2. In the first stage, a deep learning network is used to generate feature maps, which are then passed through a region proposal network (RPN) to identify regions in the image likely to contain objects. In the second stage, these proposed regions are processed through the region of interest (RoI) pooling layer to reshape them. Finally, several fully connected layers predict the class and bounding box offsets for the detected objects. This process is illustrated in Figure 2. YOLOv8 is an improved version of the YOLO (you only look once) algorithm, utilizing a one-stage detection framework and excelling in efficiency and real-time performance [13]. The model divides an image into an  $S \times S$  grid, employs anchor boxes to predict object classes and bounding box positions, and applies the non-maximum suppression (NMS) algorithm to eliminate duplicates. Compared to previous versions, YOLOv8 replaces the c3 module with the c2f module, retains the SPPF module, and transitions from an anchor-based to an anchor-free mechanism [14]. Its

architecture includes the backbone for feature extraction, neck for feature aggregation, and head for prediction generation [15].



**Figure 2.** Architecture of Faster R-CNN



**Figure 3.** Faster R-CNN framework diagram

In this paper, we investigate methods for data analysis and augmentation to further enhance the performance of PCB defect detection algorithms. Through experimentation and evaluation, we aim to provide insights into the capabilities and practical applications of the Faster R-CNN and YOLOv8 models in industrial PCB defect detection.

## 2. PCB defect detection algorithms based on Faster R-CNN and YOLOv8

### 2.1. Faster R-CNN-based PCB fault detection algorithm

The overall process of the Faster R-CNN-based PCB defect detection is illustrated in Figure 3. First, the input image is resized while maintaining its original aspect ratio, ensuring no distortion and preserving the original information. The resized image is then fed into the main feature extraction network, where convolutional operations generate a feature map. This feature map is passed to the RPN to generate region proposals and their probabilities of containing objects by predicting both categories and locations. The feature map and region proposals are forwarded to the ROI pooling layer, where pooling operations standardize the varying sizes of local feature maps to a uniform dimension, facilitating unified data processing. These standardized feature maps are concatenated along the same channel, flattened, and finally input into a classifier to predict object categories and a regressor to estimate confidence scores.

In RPN, there are two outputs: the objectness score, which indicates whether a region contains an object or not, and the box location, representing the coordinates of the proposed bounding box. At the end of the model, the output layer includes two fully connected layers: one for the softmax classifier, which predicts the class of the object within the proposed region, and the other for bounding box regression, which refines the coordinates of the bounding box. This structure is illustrated in Figure 2.

### 2.2. Yolov8-based PCB fault detection algorithm

The overall process of the YOLOv8 algorithm applied to PCB defect detection is illustrated in Figure 4. First, the input image is resized to the original aspect ratio (640x640x3), ensuring that proportions and original information are preserved without distortion. This step maintains critical features within the image as it is fed into the model. Then, the resized image is passed to the backbone (CSPDarknet53) layer, where convolutional operations extract fundamental features such as edges, structures, and shapes, enabling the model to recognize objects within the image. The features extracted by the backbone are then forwarded to the neck layer, where they are processed to create feature maps at multiple levels using feature pyramid networks (FPN). This process combines information from both deep and shallow layers of the backbone network, allowing YOLOv8 to detect objects of varying sizes and scales within the image. The feature maps processed by the neck are then passed to the head layer, which predicts bounding boxes for each detected object along with probability scores for their respective classes. Once bounding

boxes and probabilities are predicted, YOLOv8 employs an NMS mechanism to eliminate duplicate or low-probability bounding boxes. Only bounding boxes with minimal overlap are retained. Finally, the result consists of non-overlapping bounding boxes, each associated with a confidence score and the detected object class. PCB defects are clearly identified, along with the model’s confidence level in these predictions.

Thus, the output of the YOLOv8 model is a vector that includes the following components:

$$y^T = [p_0, \langle t_x, t_y, t_w, t_h \rangle, \langle p_1, p_2, \dots, p_c \rangle] \tag{1}$$

where  $p_0$  is objectness score,  $\langle t_x, t_y, t_w, t_h \rangle$  is box coordinates,  $\langle p_1, p_2, \dots, p_c \rangle$  is class scores.

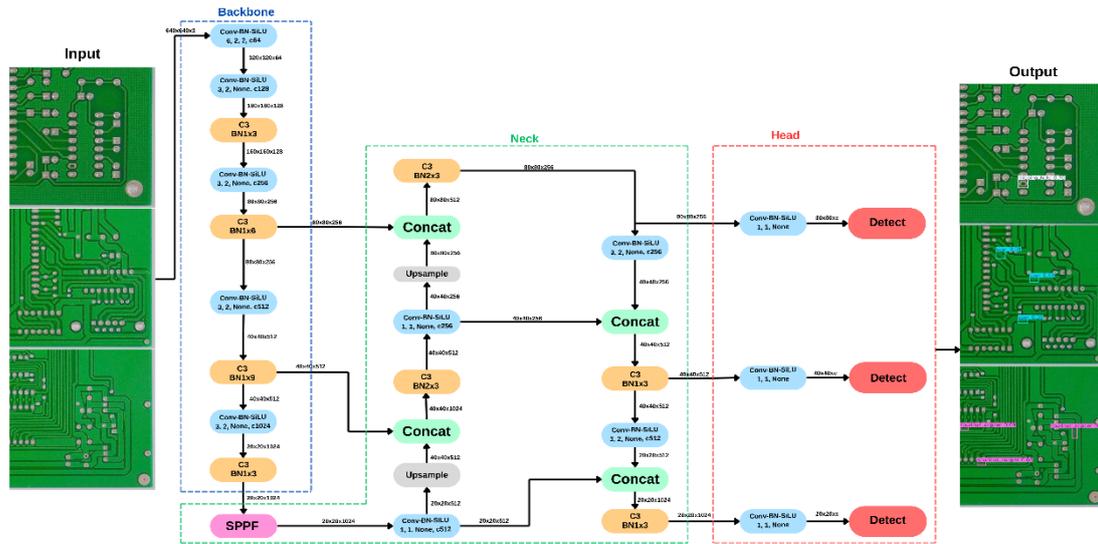


Figure 4. YOLOv8 framework diagram

### 3. Experiment and Results

#### 3.1. PCB defect detection dataset

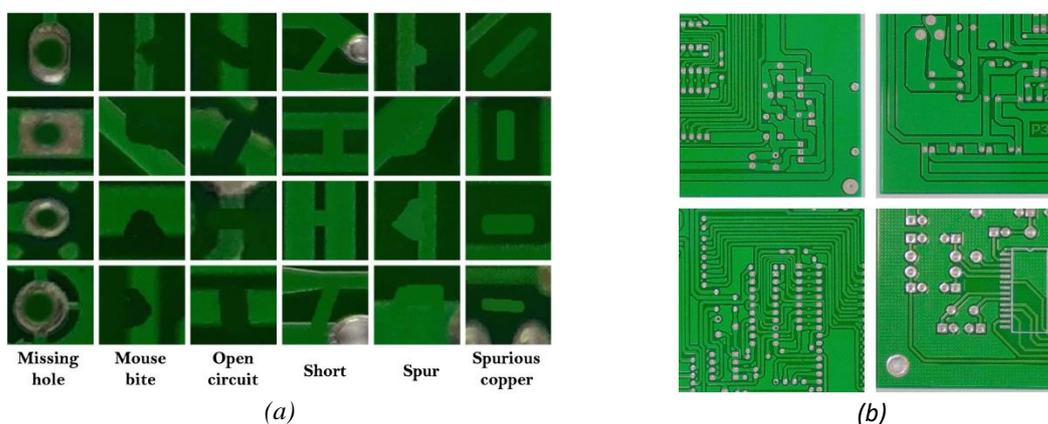
Our experiments were conducted using an augmented dataset based on the PCB defect dataset published by Peking University, which consists of 693 images with six different defect types (missing hole, mouse bite, open circuit, short, spur, and spurious copper) [16]. However, due to the small size of this dataset, data augmentation techniques should be applied before training to improve data generality. Therefore, we apply the augmented dataset presented in [17]. This dataset was cropped into 600x600 sub-images, forming the training set, validation set, and testing set with 8534, 1066 and 1068 images, respectively, as shown in Table 1.

Table 1. The class distribution in the dataset

Defect name	Image	Instances
Missing hole	1832	3612
Mouse bite	1852	3684
Open circuit	1740	3548
Short	1732	3508
Spur	1752	3636
Spurios copper	1760	3676
<b>Total</b>	<b>10,668</b>	<b>21,664</b>

To enhance diversity and improve model robustness, various image processing techniques were applied to this dataset for data preprocessing, including resizing, normalization, and data augmentation. The steps include image randomization, image rotation, and image transformation.

Image randomization involves applying random transformations such as horizontal flipping, vertical flipping, adjusting brightness, contrast, saturation, or adding random noise. These transformations introduce diversity to the dataset, helping the model improve its ability to generalize to unseen data and adapt to different environmental conditions. Image rotation is a technique that involves rotating images by a random angle within a specified range. This helps simulate various orientations of objects in the dataset, enabling the model to learn how to recognize objects from different viewpoints. Image transformation techniques, such as resizing, cropping, and shifting, are used to alter the spatial characteristics of the image. These transformations simulate changes in the size, shape, and position of objects, enhancing the model's ability to recognize objects under various conditions. Figure 5b shows the dataset after applying these image processing techniques.



**Figure 5.** PCB defects in the dataset [14] (a) Example and (b) Dataset augment

### 3.2. Experiment setting

This paper uses Google Colab to train the Faster R-CNN and YOLOv8 models by adjusting the number of epochs to be trained. To evaluate the performance of each model, we use metrics including mean Average Precision (mAP), Precision (P), and Recall (R). In this context, TP (True Positives) refers to the number of positive samples correctly classified, FP (False Positives) refers to the number of negative samples incorrectly classified as positive, and FN (False Negatives) refers to the number of positive samples incorrectly classified as negative. AP (Average Precision) is the area under the Precision-Recall (PR) curve, representing the integral of the curve. Finally, mAP (mean Average Precision) is the average value of the APs calculated across all classes or object categories in the task.

The calculation formulas are as follows:

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \end{cases} \quad \begin{cases} AP = \int_0^1 P(r)dr \\ mAP = \frac{\sum_{i=0}^n AP(i)}{n} \end{cases} \quad (2)$$

### 3.3. Results

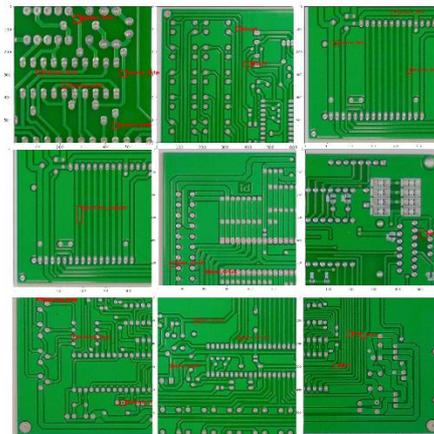
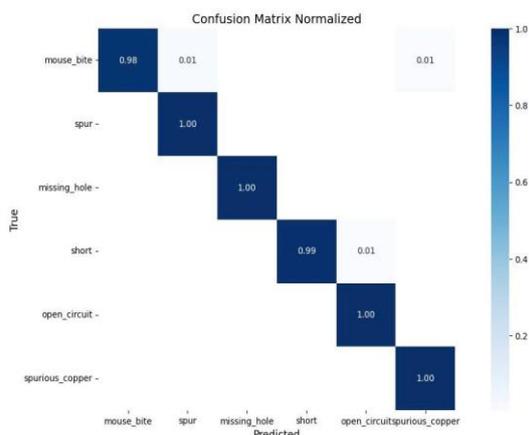
#### 3.3.1. Experimental results for Faster R-CNN-based PCB defect detection

The number of training epochs of the experiment is set to 8 since after the 8th epoch, the model training reaches a saturation status, indicating no significant improvement in performance. After each training epoch, the best current performance is recorded in Table 2. The confusion matrix at the IOU threshold of 0.25 is presented in Figure 6. Most defects have a detection accuracy close to 100%. Among them, mouse bite defects have lower accuracy compared to the

others, with detection accuracy of 98.4%. Totally, the average accuracy is 99.6%. Figure 7 shows some examples of the prediction with Faster R-CNN.

**Table 2.** The classification performance of the Faster R-CNN model

Defect	Precision	Recall	mAP
Missing hole	1.000	1.000	1.000
Mouse bite	1.000	0.984	0.992
Open circuit	0.994	1.000	0.997
Short	1.000	0.995	0.998
Spur	0.989	1.000	0.995
Spurious copper	0.994	1.000	0.997
<b>Total</b>	<b>0.996</b>	<b>0.996</b>	<b>0.996</b>



**Figure 6.** Confusion matrix on the test set with Faster R-CNN

**Figure 7.** Example of the prediction in the test set with Faster R-CNN

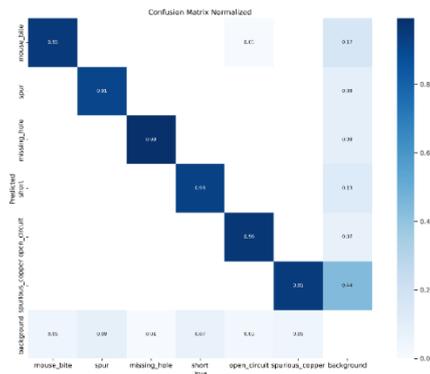
### 3.3.2. Experimental results for Faster YOLOv8-based PCB defect detection

The number of training epochs of the experiment is set to 15 since after the 15th epoch, the model training reaches a saturation status, indicating no significant improvement in performance. Tables 3 shows the final results of YOLOv8 after 15 training epochs.

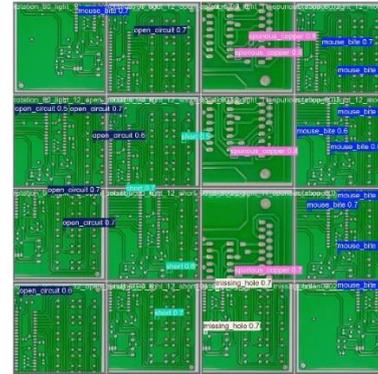
**Table 3.** The classification performance of the YOLOv8 model

Defect	Precision	Recall	mAP
Missing hole	0.980	0.973	0.977
Mouse bite	0.955	0.959	0.969
Open circuit	0.979	0.959	0.971
Short	0.964	0.939	0.954
Spur	0.960	0.949	0.967
Spurious copper	0.933	0.941	0.951
<b>Total</b>	<b>0.962</b>	<b>0.953</b>	<b>0.965</b>

After the training, the mAP value of the training part was 96.5%. The mAP value of the validation part was 95.7% and the testing part was 96.0%. The confusion matrix at the IOU threshold of 0.25 is presented in Figure 8. The highest detection accuracy of 99.3% belongs to missing holes. Meanwhile, spur defects have the lowest detection accuracy of 91.4%. Totally, the average accuracy on the test set is  $\frac{1579}{1662} = 95\%$ . Some examples of the prediction in the test set are shown in Figure 9. It can be seen that YOLOv8 can accurately identify and locate tiny PCB defects.



**Figure 8.** Confusion matrix on the test set with YOLOv8



**Figure 9.** Example of the prediction in the test set with YOLOv8

### 3.4. Performance comparison

It can be seen that both models achieve very high accuracy in detecting PCB defects. However, YOLOv8 has an advantage in speed, while Faster R-CNN excels in detection and classification rates for small defects that YOLOv8 struggles to identify, such as spur, spurious copper, and open circuit. For example, detection rate of Faster R-CNN is around 1.37s per frame while detection speed of YOLOv8 is 0.26s, which is measured on a PC with Intel CPU Core i7-9700 and 16GB RAM.

In industrial applications, a hybrid model combining Faster R-CNN and YOLOv8 could be developed to leverage the strengths of both models, ensuring both real-time performance and high accuracy [10]. This approach would involve using both Faster R-CNN and YOLOv8 as input models, performing detection and classification on the same image from the dataset. If Faster R-CNN provides a higher confidence score than YOLOv8, its prediction will be selected, and vice versa.

## 4. Conclusion

The key point of this research paper is not to identify the "best" model, as this depends on the user's preferences. The real question is which model and configuration provide the best balance between speed and accuracy for a specific application. Compared to Faster R-CNN, YOLOv8 offers more advanced applications. YOLOv8 proves to be a clearer and more efficient tool for object detection, as it provides end-to-end training. Both algorithms are quite accurate, but in some cases, YOLOv8 outperforms Faster R-CNN in terms of speed and efficiency. In the application of defect detection on PCBs, most users prefer YOLOv8 for its speed, accuracy, and efficiency. However, some users still favor Faster R-CNN for specific types of small-sized PCBs that do not require high-speed processing. With the results of our analysis and data augmentation to enhance the performance of both models, both have become promising choices for PCB defect detection applications.

## REFERENCES

- [1] Q. Zhang and H. Liu, "Multi-scale defect detection of printed circuit board based on feature pyramid network," in *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 911-914.
- [2] H. Suzuki, "Printed Circuit Board," Official Gazette of the United States Patent and Trademark Office, U.S. Patent 4,640,866, Mar. 16, 1987.
- [3] H. S. Cho, "Printed Circuit Board," Samsung Electro Mechanics Co., Ltd. Official Gazette of the United States Patent and Trademark, U.S. Patent 8,159,824, Mar. 16, 2012.
- [4] M. X. Zhao, Z. Wang, and Y. Wu, "A Dataset for Deep Learning Based Detection of Printed Circuit Board Defects," *Sci. Data*, vol. 11, no. 1, pp. 1–8, 2024, doi: 10.1038/s41597-024-03656-8.

- 
- [5] D. B. Anitha and M. Rao, "A survey on defect detection in bare PCB and assembled PCB using image processing techniques," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 39–43.
- [6] V. A. Adibhatla, "Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once," *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 4411–4428, 2021.
- [7] W. Chen, Z. Huang, Q. Mu, and Y. Sun, "PCB defect detection method based on transformer-YOLO," *IEEE Access*, vol. 10, pp. 129480–129489, 2022, doi: 10.1109/ACCESS.2022.3228206.
- [8] W. Liu and Y. Zhang, "Real-time PCB defect detection using SSD for industrial applications," *J. Electron. Imaging*, vol. 29, no. 2, 2020, Art. no. 023005, doi: 10.1117/1.JEI.29.2.023005.
- [9] B. Hu and J. Wang, "Detection of PCB Surface Defects with Improved Faster-RCNN and Feature Pyramid Networks," in *IEEE Access*, vol. 8, pp. 108335–108345, 2020.
- [10] F. Fan, B. Wang, G. Zhu, and J. Wu, "Efficient Faster R-CNN: Used in PCB solder joint defects and components detection," in *Proc. IEEE 4th Int. Conf. Comput. Commun. Eng. Technol. (CCET)*, Beijing, China, 2021, pp. 1–5, doi: 10.1109/CCET52649.2021.9544356.
- [11] H. Xu and W. Wang, "Defect detection in PCB manufacturing using R-FCN-based deep learning model," *IEEE Access*, vol. 9, pp. 123567–123577, 2021, doi: 10.1109/ACCESS.2021.3112345.
- [12] Y. Long, Z. Li, Y. Cai, R. Zhang, and K. Shen, "PCB defect detection algorithm based on improved YOLOv8," *Academ. J. Sci. Technol.*, vol. 7, no. 3, pp. 297–304, 2023, doi: 10.54097/ajst.v7i3.13420.
- [13] RangeKing, "Brief summary of YOLOv8 model structure," *GitHub*, Jan. 10, 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics/issues/189>. [Accessed Jan. 17, 2025].
- [14] TigerZ, "Target detection algorithm - YOLOV8 - algorithm details," *CSDN*, March. 2, 2023.
- [15] Z. Xiong, "A Design of Bare Printed Circuit Board Defect Detection System Based on YOLOv8," *Highlights in Science, Engineering and Technology*, vol. 57, pp. 203–209, Jul. 2023.
- [16] P. Huang and P. Wei, "A PCB dataset for defects detection and classification," *arXiv preprint arXiv:1901.08204*, 2019.
- [17] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-net: a tiny defect detection network for printed circuit boards," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 110–116, Jun. 2019, doi: 10.1049/trit.2019.0019.
- [18] P. V. Viet, "A combination of Faster R-CNN and YOLOv2 for drone detection in images," *TNU Journal of Science and Technology*, vol. 226, no. 06, pp. 90–96, May 2021.