International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Based PCB Multi-Defect Detection Using Yolov8

Vishrutha Rani S N, Spoorthi P, Syeda Husna C A, Thaiseen M D, Vaishnavi H M
Department of Information Science and Engineering
JNN College of Engineering, Shivamogga, Karnataka, India- 577201

Abstract— Printed circuit boards (PCBs) are essential components of electronic devices, and their quality directly affects the reliability and performance of electronic systems. Conventional PCB inspection methods often rely on manual examination, which can be time-consuming and inconsistent when identifying small defects or missing components. This paper presents an artificial intelligence-based PCB inspection system using the YOLOv8 object detection framework. The system consists of two independent detection branches. The bare-PCB branch identifies six defect categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper, using a publicly available PCB defect dataset. The assembled-PCB branch detects missing resistors and missing capacitors using a manually collected dataset. Both models are integrated into a web-based inspection application developed using Flask. The application supports image upload and camera-based image capture and displays detection results through bounding boxes, class labels, and confidence scores. The bare-PCB model achieved a reported mean Average Precision of 94.7% at an Intersection over Union threshold of 0.5 and 51.0% across thresholds from 0.5 to 0.95. The assembled-PCB model is evaluated independently to assess its ability to identify missing components. The integrated system provides a practical framework for assisting PCB quality inspection through automated visual detection.

Index Terms— Assembled PCB, bare PCB, computer vision, defect detection, deep learning, YOLOv8.

I. INTRODUCTION

Printed Circuit Boards (PCBs) are fundamental components in modern electronic devices. They provide mechanical support and electrical connections for electronic components used in computers, communication systems, consumer electronics, automotive equipment, and industrial applications. As electronic products become smaller and more complex, maintaining PCB quality has become increasingly important.

PCB defects can affect electrical connectivity, product reliability, and overall system performance. Common bare-PCB defects include missing holes, mouse bites, open circuits, shorts, spurs, and spurious copper. In assembled PCBs, missing electronic components such as resistors and capacitors can also affect circuit functionality and require reliable inspection.

Traditional inspection methods include manual visual examination and conventional image-processing techniques. Manual inspection can be time-consuming and may produce inconsistent results, while conventional image-processing methods can be sensitive to lighting conditions and complex PCB patterns. Recent studies have therefore explored deep-learning approaches to automate defect detection and classification [1], [2].

Object detection models can identify defect categories and their locations within an image. The YOLO family of models uses a single-stage detection approach, making it suitable for applications requiring efficient visual inspection. YOLOv8 provides a framework for training object detection models to recognize different defect types and missing components using annotated images.

This paper presents an AI-based PCB inspection system using two YOLOv8 models. The first model detects six categories of bare-PCB defects, while the second identifies missing resistors and capacitors in assembled PCBs. Both models are integrated into a web application developed using Flask and HTML, CSS, and JavaScript, supporting image upload and camera capture. The application displays detection results using bounding boxes, class labels, and confidence scores. The bare-PCB model achieved a reported mAP@50 of 94.7% and mAP@50–95 of 51.0%.

1.1 Problem Statement

Manual PCB inspection can be time-consuming and prone to human error, particularly when identifying multiple defect types across large numbers of boards. Small surface defects and irregular circuit patterns make consistent detection challenging. Similarly, missing resistors and capacitors in assembled PCBs can affect circuit functionality. Therefore, an automated system is required to detect and localize bare-PCB defects and identify missing components in assembled PCBs, while presenting the results through an accessible web interface.

1.2 Objectives

The objectives of this work are:

1. To collect PCB images containing different types of defects.
2. To train and test a YOLOv8 model for bare-PCB defect detection.
3. To detect PCB defects from camera-captured and uploaded images.
4. To train a separate YOLOv8 model for detecting missing components in assembled PCBs.
5. To develop a web-based PCB inspection system with defect visualization.

II. CONTRIBUTION

The main contributions of this work are:

- Development of a six-class bare-PCB defect detection model based on YOLOv8.
- Comparison of YOLOv8n and YOLOv8s model variants for model selection.
- Evaluation of the selected bare-PCB model using standard object detection metrics.
- Integration of the trained models into a web-based PCB inspection interface.
- Development of a modular system with separate models for bare-PCB defect detection and assembled-PCB missing-component detection.

2.1 RELATED WORK

Deep learning has become an important approach to PCB defect inspection because it can learn image features and localize defects without requiring every detection rule to be manually designed. Existing studies investigate model architecture, multiscale feature extraction, attention mechanisms, domain adaptation, few-shot learning, and unsupervised detection.

Chen et al. [1] reviewed deep-learning-based PCB defect detection methods, datasets, and evaluation metrics. Their review highlights the benefits of learned visual features while identifying challenges involving dataset availability, computational requirements, and small defects. Ling and Mat Isa [2] surveyed traditional image processing, machine learning, and deep-learning methods, providing a broader overview of PCB inspection approaches.

Several studies improve object detectors to handle small and visually subtle PCB defects. Ge et al. [3] proposed a class-incremental learning approach based on PCB-YOLOX to accommodate newly introduced defect classes while reducing catastrophic forgetting. Yan et al. [4] introduced a domain-incremental learning framework for adapting to changes in PCB data distributions. These approaches address challenges that arise when defect types or production conditions change over time.

Wu and Lin [5] proposed YOLO-EMAC to improve small-target defect detection and feature generalization. He and Xie [6] developed ABF-YOLO, incorporating axial attention and bidirectional feature fusion to improve the representation of fine-grained defects. Liu [7] introduced an adaptive defect-aware attention network that combines small-defect feature preservation with localization and segmentation-related processing.

Other approaches explore few-shot learning and feature enhancement. Luo and Chen [8] proposed CIFPNet for few-shot PCB defect detection using contextual information and foreground prototype guidance. Wang et al. [9] investigated feature enhancement and multiscale fusion for few-shot surface defect detection. These methods are relevant where annotated examples are limited or defect appearance varies.

Transformer-based and attention-based approaches have also been investigated. Chen et al. [10] proposed Transformer-YOLO for PCB defect detection, while Li et al. [11] introduced PCB-DETR with a spatial attention offset module. Jiang et al. [12] proposed a pair-image-based defect detection network using a template to support PCB inspection. These methods explore different ways to use contextual information, spatial relationships, and reference images.

Efficiency is another important concern in industrial inspection. Mo et al. [13] proposed SGT-YOLO as a lightweight PCB defect detection method, while Kong et al. [14] introduced SMC-YOLO using multiscale features and dual loss functions. Liu et al. [15] investigated an improved YOLOv5s-based approach for industrial component defect detection. Such work highlights the need to balance detection performance and computational cost.

Other research investigates less conventional inspection settings. Chen et al. [16] presented U2D2PCB, an uncertainty-aware unsupervised approach that combines reconstructive and discriminative models. Ancha et al. [17] assessed YOLO models using a mixed PCB defect dataset intended to represent real-world detection scenarios. Selvam et al. [18] proposed YOLO-DefXpert using an improved YOLOv11 algorithm. Ong et al. [19] developed SEConv-YOLO for PCB and PCBA wire defect detection, demonstrating the relevance of efficient object detection to assembled-board inspection. Kim et al. [20] studied background-image acquisition for YOLO-based static-object recognition, a topic related to image acquisition but not a direct evaluation of PCB defect detection.

2.1.1 Research Gap

The reviewed studies demonstrate progress in detecting small defects, improving feature extraction, and handling variations in PCB datasets. However, model comparisons remain dependent on clearly defined datasets, class categories, evaluation metrics, and testing conditions. Another practical consideration is integrating defect detection into an accessible interface that presents results clearly to users.

This work focuses on implementing a six-class bare-PCB defect detection model and a separate assembled-PCB model for detecting missing resistors and capacitors. Both models are integrated into a web-based inspection application using Flask and web technologies. The system emphasizes practical model selection, defect detection, and accessible visualization of inspection results.

2.2 PROPOSED SYSTEM

2.2.1 System Overview

The proposed system uses a computer-vision pipeline to detect and localize defects in PCB images. A user supplies an image through the web interface, and the application processes it using the appropriate detection model before displaying the results.

The system consists of two independent YOLOv8 models: one for detecting six types of bare-PCB defects and another for identifying missing capacitors and missing resistors in assembled PCBs. Each model uses its own trained weights and class configuration. Both models are integrated into the web application to provide visual detection results for PCB inspection.

2.2.2 System Architecture

The system can be represented by the following architecture:

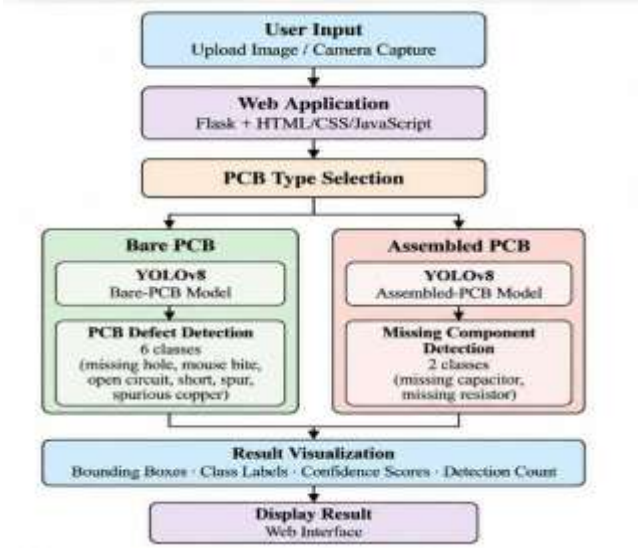


Fig. 1. System architecture of the proposed AI-based PCB defect detection system.

The system routes bare-PCB images to the six-class defect detection model and assembled-PCB images to the separate two-class missing-component detection model. Each model processes the selected image using its respective trained weights and class configuration. The output stage displays the input image and the annotated result, including bounding boxes, predicted class labels, confidence scores, and detection information.

2.2.3 Bare-PCB Defect Categories

The bare-PCB model recognizes the following six defect classes:

1. **Missing hole:** A required hole is absent from the PCB.
2. **Mouse bite:** An irregular indentation appears along the edge of the PCB.
3. **Open circuit:** A conductive trace is interrupted.
4. **Short:** An unintended conductive connection occurs between traces or regions.
5. **Spur:** An unwanted conductive protrusion appears on a trace.
6. **Spurious copper:** Unwanted copper material is present in a region of the board.

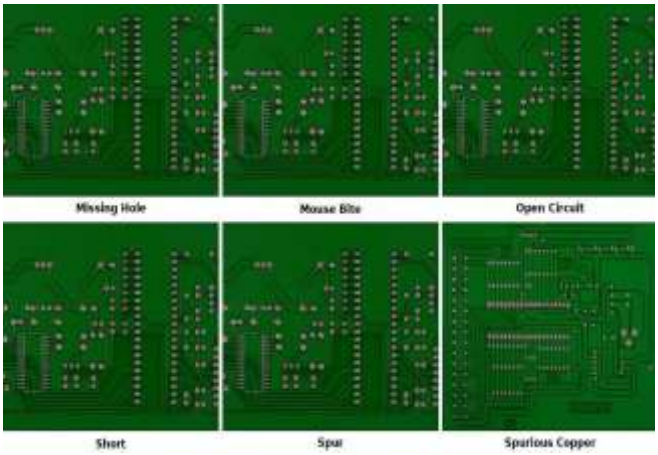


Fig. 2. Representative images of the six bare-PCB defect categories

2.2.4 Assembled-PCB Missing-Component Detection

Assembled-PCB inspection addresses a different visual problem from bare-PCB surface inspection. Instead of identifying trace and copper defects, the assembled-PCB model detects missing electronic components. The target classes are missing capacitor and missing resistor.

The assembled-PCB branch uses a separate YOLOv8 model configuration with its own labeled dataset, class mapping, and trained weights. Although both branches use YOLOv8, they are trained independently because their target classes and visual features differ. The assembled-PCB model is integrated into the web application to identify missing components and display the corresponding detection results.

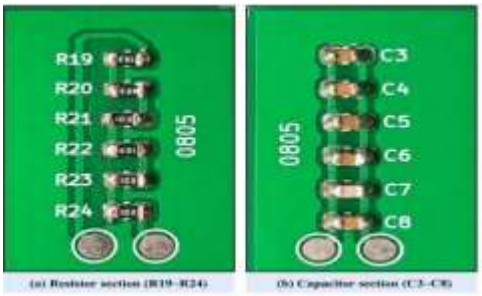


Fig. 3. Representative assembled-PCB images showing missing-component categories.

III. DATASET AND PREPROCESSING

3.1 Dataset Description

The bare-PCB model uses the Roboflow Universe dataset titled PCB Dataset Defect, which contains 693 images in the version used for the project. The dataset is provided in YOLO object-detection format, with annotations represented by bounding boxes. The original image resolution is reported as 2047 × 1070 pixels. It includes six defect categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper. The annotation files specify the class identifiers and normalized bounding-box coordinates each labeled object.

The assembled-PCB dataset was collected manually by capturing images of PCBs with electronic components. The images were annotated manually to identify missing capacitors and missing resistors. These annotations were used to prepare the dataset for training the separate YOLOv8 assembled-PCB detection model.

3.2 Annotation Format

Each YOLO annotation represents a defect using five values:[c, x_c, y_c, w, h]

where (c) denotes the class identifier, (x_c) and (y_c) represent the normalized center coordinates, and (w) and (h) represent the normalized bounding-box width and height. All coordinates are normalized relative to the image dimensions.

The class mapping used for the bare-PCB and assembled-PCB dataset is shown in Table and Table II respectively.

Class ID	Defect category
0	missing_hole
1	mouse_bite
2	open_circuit
3	short
4	spur
5	spurious_copper

Class ID	Defect category
0	missing capacitor
1	missing resistor

Correct class ordering is essential because the numeric identifiers in the annotations must match the class names in the dataset configuration.

3.3 Data Preparation

Both datasets are prepared in YOLO format and organized into training, validation, and test subsets. The training subset is used to update model parameters, the validation subset supports performance monitoring and model selection, and the held-out test subset is used for final evaluation.

The bare-PCB and assembled-PCB datasets are prepared separately, with their respective class mappings and bounding-box annotations. The exact number of images in each subset and the splitting procedure should be reported according to the final dataset configuration used for each model.

IV. YOLOv8-BASED METHODOLOGY

4.1 Model Selection

YOLOv8 is used as the object detection framework for both bare-PCB defect detection and assembled-PCB missing-component detection. Pretrained weights are used as the starting point for transfer learning, allowing the models to adapt learned visual features to their respective PCB datasets.

Two model variants, YOLOv8n and YOLOv8s, are considered for model selection. YOLOv8n is the smaller variant, while YOLOv8s offers a different balance between model capacity and computational requirements. Based on the results reported in the project presentation, YOLOv8s is selected as the accuracy-focused model.

4.2 Training Configuration

The training configuration is summarized in Table III.

Parameter	Configuration
Detection framework	Ultralytics YOLOv8
Model variants considered	YOLOv8n and YOLOv8s
Selected bare-PCB model	YOLOv8s
Input image size	800 × 800 pixels
Initialization	Pretrained weights
Maximum training epochs	150
Batch size	4
Early-stopping patience	30 epochs
Dataset format	YOLO object detection
Evaluation method	Held-out test dataset

The model is trained using annotated PCB images and their corresponding YOLO-format class labels and bounding boxes. During training, validation performance is monitored to support model selection, and the best-performing checkpoint is retained for evaluation and deployment.

4.3 Inference Pipeline

During inference, the application receives an input image through the web interface, either by uploading an image or capturing one using the camera. Based on the selected PCB type, the image is passed to the corresponding trained YOLOv8 model. The model predicts defect locations by generating bounding boxes, class labels, and confidence scores for the detected objects. Post-processing is then applied to filter and organize the predictions before presenting the results.

The web interface displays the annotated image alongside the original input image to facilitate visual comparison. Each detected defect is marked with a bounding box and its predicted class label, together with a confidence score indicating the model's confidence in the prediction. The application also displays the defect count based on the final detections. This pipeline provides users with a visual representation of the detected defects and supports the inspection of both bare and assembled printed circuit boards.

V. Web Application

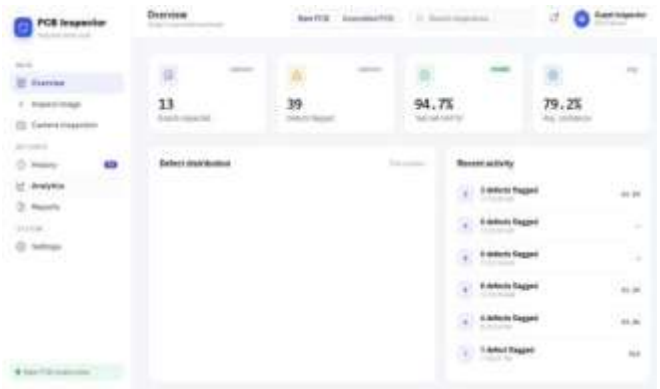


Fig. 4. Representative images of the Web Application

The trained defect-detection model is integrated into a web application using Flask for the backend and HTML, CSS, and JavaScript for the user interface. The application provides a graphical interface through which users can submit PCB images for inspection and view the corresponding detection results.

The user workflow consists of the following steps:

1. Open the PCB inspection interface.
2. Select the PCB inspection type supported by the application.
3. Upload a PCB image or use the available camera-capture functionality.
4. Submit the image for model inference.
5. View the annotated image and associated detection details.

For the completed bare-PCB workflow, the application displays the predicted defect classes, bounding boxes, confidence scores, and other available detection information. Camera-based operation depends on browser permissions and the implementation of the camera-capture functionality.The assembled-PCB workflow uses a separate dataset and a separately trained model to detect missing capacitors and missing resistors.

5.1 PERFORMANCE EVALUATION

5.1.1 Evaluation Metrics

The performance of the proposed defect-detection model is evaluated using standard object-detection metrics. These metrics assess the correctness of the predicted defect classes and the localization accuracy of the predicted bounding boxes.

Precision: Precision measures the proportion of predicted positive detections that are correct and is defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall: Recall measures the proportion of actual objects that are correctly detected by the model and is expressed as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Here, (TP), (FP), and (FN) denote true positives, false positives, and false negatives, respectively, according to the selected confidence threshold and bounding-box matching criteria.

Intersection over Union (IoU): IoU measures the overlap between a predicted bounding box and its corresponding ground-truth bounding box. It is calculated as

$$\text{IoU} = \frac{|B_p \cap B_g|}{|B_p \cup B_g|}$$

where (B_p) represents the predicted bounding box and (B_g) represents the ground-truth bounding box. A higher IoU indicates greater overlap between the predicted and actual object locations.

Average Precision (AP): AP summarizes the precision–recall relationship for an individual defect class across different confidence thresholds.

Mean Average Precision at IoU 0.50 (mAP@50) is calculated by averaging the AP values across all defect classes at an IoU threshold of 0.50.

Mean Average Precision at IoU thresholds from 0.50 to 0.95 (mAP@50–95): This metric averages AP across IoU thresholds from 0.50 to 0.95 in increments of 0.05 and then across the evaluated classes. It provides a stricter assessment of detection performance by considering multiple localization thresholds, following the COCO-style evaluation convention.

Confusion Matrix: A confusion matrix summarizes the distribution of correct and incorrect predictions across defect classes. It helps identify classes that are frequently confused and provides insight into the model's classification performance. In object detection, the interpretation of the matrix also depends on how unmatched predictions and ground-truth objects are represented.

5.1.2 Bare-PCB Defect Detection Results

Table IV. Reported performance of the YOLOv8s model

Metric	Reported result
mAP@50	94.7%
mAP@50–95	51.0%

The precision–recall curve included in the project results reports an mAP@50 value of 0.947, equivalent to 94.7%. This differs slightly from the value of 94.7% reported on the main results slide. Therefore, the final value should be verified against the saved evaluation output before submission. In this section, 94.7% is retained provisionally to remain consistent with the main results slide.

The reported mAP@50 of 94.7% indicates strong aggregate detection performance at an intersection over union (IoU) threshold of 0.50 on the evaluated dataset. The mAP@50–95 value of 51.0% is lower because this metric evaluates detections across multiple IoU thresholds, including stricter localization requirements. This difference highlights the greater challenge of accurately localizing defects across varying overlap criteria, particularly when defect regions are small or irregular.

However, the aggregate metrics alone do not establish uniform performance across all six bare-PCB defect classes. Class-wise precision, recall, and average precision (AP), together with the confusion matrix, can provide further insight into the model's performance for individual defect categories.

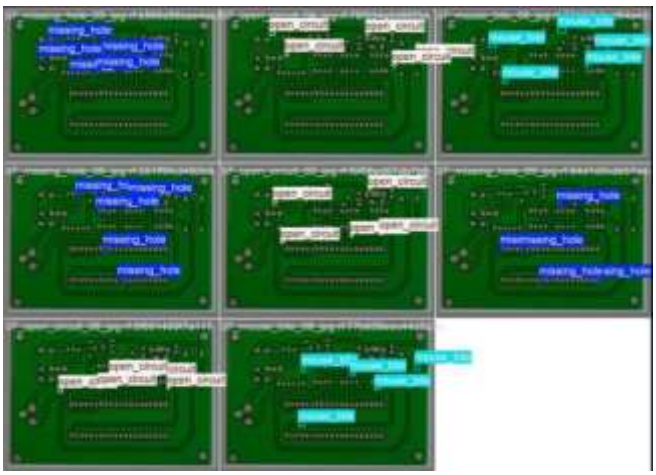


Fig. 5. Bare-PCB defect detection result with annotation

5.1.3 Assembled-PCB Defect Detection Results

The assembled-PCB defect-detection model is designed to identify missing components, specifically missing capacitors and missing resistors, on assembled printed circuit boards. The model uses a separately collected and manually annotated dataset of practice PCB images, with class labels corresponding to the two types of missing components. Unlike the bare-PCB model, which identifies six categories of PCB defects, the assembled-PCB model focuses on component-level defects. Its detection output includes the predicted defect class, bounding box, and confidence score for each identified missing component. The results can be visualized through annotated images to help users locate potential component omissions. The assembled-PCB model is evaluated separately from the bare-PCB model because the two models use different datasets and class definitions. Its effectiveness should be discussed based on the actual detection results obtained during testing.

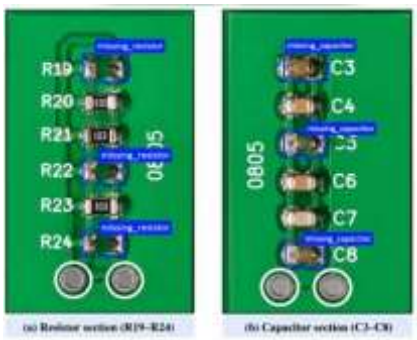


Fig. 6. Assembled-PCB missing component detection result with annotation

5.1.4 Results Interpretation

The YOLOv8s model demonstrates promising performance in bare-PCB defect detection, achieving a reported mAP@50 of 94.7%. The assembled-PCB model is also designed to identify missing capacitors and resistors, providing a complementary approach to component-level inspection. Together, the two models support automated PCB inspection across bare and assembled boards.

5.2 DISCUSSION AND FUTURE WORK

The results demonstrate the potential of YOLOv8 for six-class bare-PCB defect detection. The Flask-based web application provides a practical interface for image input and visualization of detected defects.

The modular architecture separates defect detection from the web interface, allowing the model to be retrained or replaced independently. However, evaluation on the 693-image dataset does not establish generalization to all PCB layouts, lighting conditions, or industrial environments. The difference between mAP@50 and mAP@50–95 also indicates that precise defect localization remains a challenge.

The system also considers assembled-PCB inspection for missing capacitors and resistors. This task requires a separate dataset, class mapping, and trained model.

Future work will focus on expanding both datasets, improving defect localization, and evaluating the models under varied lighting conditions and PCB layouts. Further work will also include completing the evaluation of the assembled-PCB model, measuring inference speed and model size, and testing the system on independently collected images to assess its suitability for practical PCB inspection.

5.3 CONCLUSION

This paper presents a YOLOv8-based approach for detecting six bare-PCB defect classes: missing hole, mouse bite, open circuit, short, spur, and spurious copper. The selected YOLOv8s model achieves a reported mAP@50 of 94.7% and mAP@50–95 of 51.0%. Integration with a Flask-based web application enables visualization of defect labels, bounding boxes, and confidence scores.The system architecture also supports assembled-PCB inspection for missing capacitors and resistors using a separate model.

REFERENCES

[1] X. Chen, Y. Wu, X. He, and W. Ming, “A Comprehensive Review of Deep Learning-Based PCB Defect Detection,” *IEEE Access*, vol. 11, 2023.
[2] Q. Ling and N. A. Mat Isa, “Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey,” *IEEE Access*, 2023.
[3] Q. Ge, R. Wu, Y. Wu, and H. Liu, “A Class-Incremental Learning Method for PCB Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 74, 2025.
[4] Z. Yan, R. Hao, B. Huang, L. Zhu, and H. Pan, “A Domain Incremental Learning Framework for PCB Continuous Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 74, 2025.
[5] W. Wu and colleagues, “A High-Performance and Enhanced Generalization Small Target Defect Detection Method for PCB Boards Based on YOLO-EMAC,” *IEEE Transactions on Instrumentation and Measurement*, 2025.
[6] X. He and M. Xie, “ABF-YOLO: A Fine-Grained PCB Defect Detection Framework Integrating Axial Attention and Bidirectional Feature Fusion,” *IEEE Transactions on Consumer Electronics*, vol. 71, no. 3, 2025.
[7] X. Liu, “An Adaptive Defect-Aware Attention Network for Accurate PCB-Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, vol. 73, 2024.
[8] L. Luo and colleagues, “CIFPNet: Contextual Information Aggregation and Foreground Prototype Guidance Network for Few-Shot PCB Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, 2025.
[9] “Few-Shot PCB Surface Defect Detection Based on Feature Enhancement and Multi-Scale Fusion,” *IEEE Access*, 2022.
[10] “PCB Defect Detection Method Based on Transformer-YOLO,” *IEEE Access*, 2022.
[11] “ PCB-DETR: A Detection Network of PCB Surface Defect With Spatial Attention Offset Module,” *IEEE Access*, 2024.
[12] “PIDDN: Pair-Image-Based Defect Detection Network With Template for PCB Inspection,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2025.
[13] “SGT-YOLO: A Lightweight Method for PCB Defect Detection,” *IEEE Transactions on Instrumentation and Measurement*, 2025.
[14] “SMC -YOLO: Surface Defect Detection of PCB Based on Multi-Scale Features and Dual Loss Functions,” *IEEE Access*, 2024.
[15] “Surface Defect Detection of Industrial Components Based on Improved YOLOv5s,” *IEEE Sensors Journal*, 2024.
[16]“U2D2PCB: Uncertainty-Aware Unsupervised Defect Detection on PCB Images Using Reconstructive and Discriminative Models,” *IEEE Transactions on Instrumentation and Measurement*, 2024.
[17] “Utilizing YOLO Models for Real-World Scenarios: Assessing Novel Mixed Defect Detection Dataset in PCBs,” *IEEE Access*, 2024.
[18] “YOLO -DefXpert: An Advanced Defect Detection on PCB Surfaces Using Improved YOLOv11 Algorithm,” *IEEE Access*, 2025.
[19] S. K. Ong, C. K. Tan, V. M. Baskaran, B. K. Puah, and K. H. Liew, “Enhancing Industrial PCB and PCBA Defect Detection: An Efficient and Accurate SEConv-YOLO Approach,” *IEEE Access*, vol. 13, 2025.
[20] “An Efficient Scheme to Obtain Background Image in Video for YOLO-Based Static Object Recognition,” *Journal of Web Engineering*, 2022.
[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.