Journal of Northern Aegean
Technical Sciences and Technology
https://dergipark.org.tr/tr/pub/ ketbt

Performance Analysis of YOLO11 Models in PCB Defect Detection Tasks

Mehmet Dayıoğlu[a,*] , Ali Kemal Eyüboğlu[b] , Rıdvan Ünal[c]

[a] *Department of Electrical and Energy, Bergama Vocational School, Dokuz Eylül University, Izmir, Türkiye, ORCID: 0000-0001-8323-0730*
[b] *Department of Electrical and Energy, Bergama Vocational School, Dokuz Eylül University, Izmir, Türkiye, ORCID: 0000-0001-7637-2327*
[c] *Department of Electrical and Electronics Engineering, Faculty of Technology, Afyon Kocatepe University, Afyonkarahisar, Türkiye, ORCID: 0000-0001-6842-7471*

ABSTRACT

Printed Circuit Board (PCB) defect detection is critical in electronics manufacturing, as undetected faults can lead to severe quality control issues. Recent advancements in deep learning, particularly object detection models, have significantly improved inspection systems' accuracy and speed. This study explores the performance of the YOLO11 (You Only Look Once version 11) object detection architecture on a multi-class PCB defect dataset. Five YOLO11 variants—YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x—were trained and evaluated under identical conditions using high-resolution images containing six defect types. Metrics such as mAP@50, mAP@50-95, and FPS were used for evaluation. Results demonstrate that YOLO11l achieved the highest mAP@50-95 of 0.551, while YOLO11n achieved up to 166 Frame Per Second (FPS) on an NVIDIA A100 GPU, confirming its real-time capability. Comparative analysis against state-of-the-art models confirms that YOLO11 variants offer an effective trade-off between accuracy and efficiency. This study positions YOLO11 as a strong candidate for real-time PCB inspection systems.

Keywords: *PCB Defect Detection, Deep Learning, YOLO11.*

PCB Hata Tespit Görevlerinde YOLO11 Modellerinin Performans Analizi

Mehmet Dayıoğlu[a,*] , Ali Kemal Eyüboğlu[b] , Rıdvan Ünal[c]

[a] *Elektrik ve Enerji Bölümü, Bergama Meslek Yüksekokulu, Dokuz Eylül Üniversitesi, İzmir, Türkiye, ORCID: 0000-0001-8323-0730*
[b] *Elektrik ve Enerji Bölümü, Bergama Meslek Yüksekokulu, Dokuz Eylül Üniversitesi, İzmir, Türkiye, ORCID: 0000-0001-7637-2327*
[c] *Elektrik ve Elektronik Mühendisliği Bölümü, Teknoloji Fakültesi, Afyon Kocatepe Üniversitesi, Afyonkarahisar, Türkiye, ORCID: 0000-0001-6842-7471*

ÖZET

Baskılı Devre Kartı (PCB) hata tespiti, elektronik üretiminde kritik bir süreçtir; tespit edilemeyen hatalar ciddi kalite kontrol problemlerine yol açabilir. Derin öğrenme ve özellikle nesne tespiti modellerindeki son gelişmeler, denetim sistemlerinin doğruluk ve hızında önemli iyileştirmeler sağlamıştır. Bu çalışma, çok sınıflı bir PCB arıza tespit veri kümesi üzerinde YOLO11 (You Only Look Once versiyon 11) nesne tespiti mimarisinin performansını incelemektedir. YOLO11 ailesine ait beş farklı varyant—YOLO11n, YOLO11s, YOLO11m, YOLO11l ve YOLO11x—altı farklı arıza türü içeren yüksek çözünürlüklü görüntüler kullanılarak, aynı eğitim koşulları altında değerlendirilmiştir. Değerlendirme için mAP@50, mAP@50-95 ve FPS gibi metrikler kullanılmıştır. Sonuçlar, YOLO11l modelinin 0.551 mAP@50-95 ile en yüksek doğruluk değerine ulaştığını, YOLO11n modelinin ise NVIDIA A100 GPU üzerinde saniyede 166 kareye (FPS) kadar gerçek zamanlı performans sergilediğini göstermektedir. Yapılan karşılaştırmalar, YOLO11 ailesinin doğruluk ve verimlilik arasında etkili bir denge sunduğunu doğrulamaktadır. Bu çalışma, YOLO11 mimarisinin gerçek zamanlı PCB denetim sistemleri için güçlü bir aday olduğunu ortaya koymaktadır.

Anahtar Kelimeler: *PCB Hata Tespiti, Derin Öğrenme, YOLO11.*

**Sorumlu yazar*
E-posta: mehmet.dayioglu@deu.edu.tr

Geliş: 29 Nisan 2025
Kabul: 11 Haziran 2025

1. Introduction

Printed Circuit Boards (PCBs) are fundamental components in modern electronic devices, enabling functionality by providing electrical connections between various electronic components. However, defects can occur during or after the manufacturing process, and if left undetected, these defects can degrade system performance and lead to significant quality control issues.

To ensure product integrity, defect inspection is typically conducted using two primary approaches: manual inspection and Automatic Optical Inspection (AOI) systems [1-3]. Manual inspection involves visual checking by human operators and is feasible for small-scale production or prototyping. However, it is inherently prone to subjectivity, fatigue-induced errors, and limited repeatability, making it unsuitable for high-volume manufacturing. AOI systems, on the other hand, offer automated, high-speed visual analysis using high-resolution cameras and predefined rules or templates [4]. Although more scalable, traditional AOI systems are often sensitive to variations in lighting, orientation, and PCB layout. Their reliance on rigid image processing rules also limits adaptability to novel or irregular defect types.

To overcome the limitations of conventional approaches, researchers have explored classical computer vision and machine learning methods. Techniques such as edge detection, thresholding, morphological filtering, and template matching have been applied to highlight structural inconsistencies in PCB imagery [5-8]. While computationally efficient and interpretable, these methods tend to perform poorly under noise, background variation, or slight misalignments. Machine learning models like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests have also been employed [9-11], often in combination with hand-crafted feature extractors such as Histogram of Oriented Gradients (HOG) or Local Binary Patterns (LBP) [12]. While these hybrid systems improved accuracy over rule-based methods, their performance is limited by dependence on domain-specific feature engineering and insufficient generalization to complex PCB layouts.

The recent rise of deep learning has led to a paradigm shift in automated defect detection, particularly with the success of Convolutional Neural Networks (CNNs). Deep learning models have demonstrated superior performance in both classification and localization by learning hierarchical features directly from raw image data. Object detection architectures such as Faster R-CNN (Region Convolutional Neural Network, SSD (Single Shot Detector), and the YOLO (You Only Look Once) family have gained widespread adoption in industrial inspection tasks due to their ability to simultaneously identify and localize defects in real time [13-15].

YOLO-based models, in particular, are well-suited for PCB defect detection owing to their compact architecture, high inference speed, and ability to handle multiple objects in a single forward pass. From YOLOv1 to YOLOv8, successive versions have introduced improvements in accuracy, speed, and generalizability. However, challenges remain in detecting small, low-contrast, or densely distributed defects—especially in high-resolution PCB images with diverse component layouts.

The latest advancement in the YOLO family, YOLO11, incorporates several architectural improvements to address these challenges. These include an enhanced YOLO Feature Pyramid Network (YFPN) for improved multi-scale representation, Channel and Spatial Attention modules (e.g., C2f and PSA) for better focus on defect-relevant regions, and an optimized decoupled detection head to improve localization accuracy and convergence stability [16]. Additionally, YOLO11's lightweight variants, such as YOLO11n and YOLO11s, achieve high frame rates on edge devices, making them highly suitable for real-time deployment in constrained industrial environments.

This study presents a comprehensive evaluation of YOLO11 for multi-class PCB defect detection. Five model variants—YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x—are trained and benchmarked on a standardized dataset comprising six defect classes: missing hole, mouse bite, open circuit, short circuit, spur, and spurious copper. The evaluation considers accuracy (mAP@50 and mAP@50-95), inference speed (FPS), and model complexity (parameters and FLOPs). By comparing YOLO11 against state-of-the-art alternatives, the study aims to assess its practical utility and readiness for integration into real-world PCB inspection workflows.

2. Related Works

Early studies in PCB defect detection primarily relied on traditional image processing and manual feature extraction. Dave et al. [17] developed an embedded system utilizing image subtraction techniques for anomaly detection. Zhou [18] employed Principal Component Analysis (PCA) to improve image registration and defect localization, while Chang et al. [19] enhanced traditional segmentation approaches using Particle Swarm Optimization (PSO) combined with Speeded Up Robust Features (SURF).

With the rise of deep learning, CNN-based methods became dominant. Zhang et al. [20] leveraged deep feature learning to surpass shallow feature engineering approaches. Hu and Wang [21] improved Faster R-CNN by integrating a Feature Pyramid Network (FPN) and a Guided Anchor Region Proposal Network (GARPN) to enhance small-scale defect detection. Shao et al. [22] proposed MobileNet-YOLOv3, replacing Darknet-53 with MobileNet to achieve a more lightweight architecture suitable for PCB defect detection.

YOLO-based models soon gained prominence due to their real-time detection capabilities. Chaudhari et al. [23] enhanced YOLOv3 through multi-scale predictions and improved feature fusion. Liang et al. [24] introduced improvements to YOLOv5 by integrating Density-Based Spatial Clustering of Applications with Noise (DBSCAN), refined anchor generation, Global Attention Mechanism (GAM), and a Decoupled-Head design. Parallel to these developments, lightweight and fast detection models were explored. Zhang et al. [25] developed a channel-pruned YOLOv5s-based detector, achieving significant model size reductions with minimal accuracy loss.

Transformers and attention mechanisms have also influenced the field. Li et al. [26] introduced PCB-DETR, extending Deformable Detection Transformer (Deformable-DETR) structures with spatial attention offset modules. Pan et al. [27] proposed YOLOx-Plus, combining Simple Attention Module (SimAM) and Scaled Intersection over Union (SIoU) loss, while optimizing inference speed for FPGA (Field Programmable Gate Array)-based deployments. Extensions of the YOLOv8 framework have also been explored. Huang and Li [28] proposed YOLO-HB, incorporating Hybrid Attention Transformer (HAT) and Bidirectional Feature Pyramid Network (BiFPN) for enhanced multi-scale feature representation. Gu et al. [29] optimized YOLOv8 by integrating partial convolution modules. Zeng et al. [30] introduced LSDM-PCB, combining Residual Feature Attention Convolution (RFAConv) and Global-Local Mixed Attention (GLMA) to boost small defect sensitivity.

Alternative approaches based on generative models have also been explored. You [31] proposed a Generative Adversarial Network (GAN)-based pipeline enhanced with Edge-Enhanced Super-Resolution GAN (EESRGAN) to improve low-resolution PCB defect detection. For unsupervised learning, Chen et al. [32] developed U2D2PCB, an uncertainty-aware dual-network model combining reconstruction and discrimination tasks to detect defects without labeled data.

Lang and Lv [33] proposed SEPDNet, a simple and effective network customized for small target detection on PCBs. Zhu and Zhao [34] developed SRN_Net, integrating global context attention and residual aggregation to enhance detection accuracy of small PCB defects. Zhao et al. [35] introduced PSDDNet, a lightweight YOLOv8-based model optimized through multi-branch streaming convolution (MSC) and gather-distribute feature fusion mechanisms (GDLite), achieving high accuracy and fast inference with minimal parameters, making it highly suitable for real-time PCB surface inspection.

SSD-based approaches also contributed to PCB defect detection [36]. Kang et al. [37] proposed mSSD, an improved SSD model utilizing a ResNet50 backbone and a small-target feature prediction layer. Survey studies by Anoop et al. [38] and Ling and Isa [39] have comprehensively reviewed PCB defect detection methods, from early machine vision techniques to modern deep learning-based approaches, and highlighted ongoing challenges and future opportunities.

Collectively, these studies mark a transition from manual visual inspection to sophisticated deep learning-driven frameworks, emphasizing higher accuracy, faster inference, reduced model sizes, and improved robustness against small and complex defects. Despite these advancements, the fast-paced

development of detection models highlights the need for continuous benchmarking to assess their practical applicability.

Currently, although the YOLO11 family (YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x) has been introduced in the academic community, no systematic evaluation has yet been conducted specifically for PCB defect detection tasks, where precision on small and complex defects is critical. Therefore, this study aims to fill this gap by benchmarking the anticipated YOLO11 variants under identical experimental conditions and comparing their performance against established detectors. Through this comprehensive evaluation, practical insights are provided into the deployment potential of these models in real-world manufacturing environments.

3. Material and Method

This section details the dataset preparation, model architecture, training strategy, and evaluation metrics used in our application of YOLO11 for PCB defect detection.

3.1 Dataset Preparation

The dataset used in this study is based on the publicly available PCB defect dataset originally released by the Intelligent Robot Development Laboratory of Peking University [40]. This dataset has become a widely accepted benchmark in the literature for evaluating the performance of defect detection models in printed circuit boards. It comprises high-resolution images containing six types of common manufacturing defects: missing hole, mouse bite, open circuit, short circuit, spur, and spurious copper (Figure 1). Each image has been meticulously annotated with bounding boxes that localize defects.

To convert these annotations into a format compatible with YOLO11, we transformed the original labels into the standard YOLO format, which includes normalized coordinates of the bounding box center (x, y), width, height, and the corresponding class label. The dataset was then divided into training (70 %), validation (15 %), and testing (15 %) subsets, ensuring a representative distribution of all defect types across splits. This structured partitioning allows for consistent evaluation and comparison with prior work utilizing the same dataset.

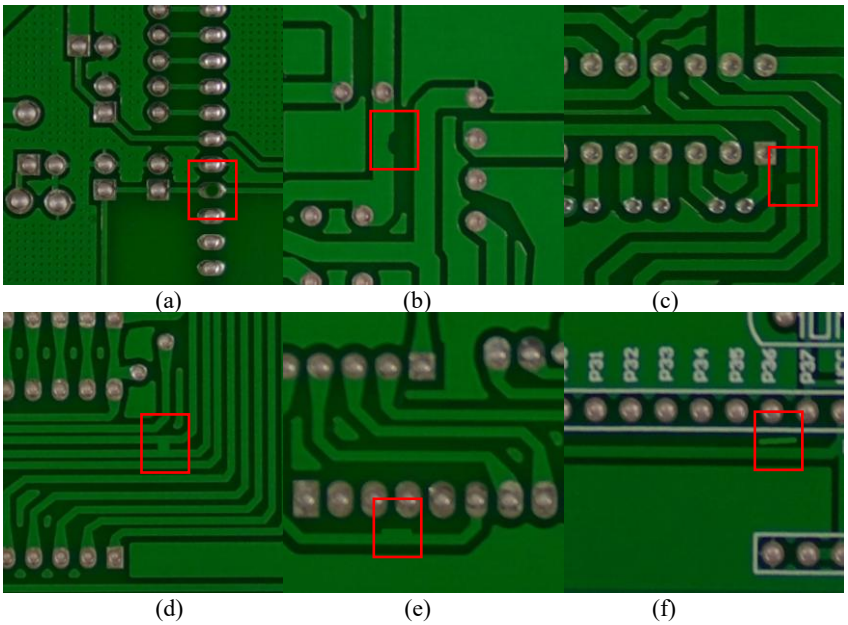


Figure 1. Illustration of the six PCB defect classes included in the dataset. (a) missing hole; (b) mouse bite; (c) open circuit; (d) short circuit; (e) spur; (f) spurious copper.

Table 1. Summary of the PCB defect dataset

Type of Defect	Number of Original Images	Number of Augmented Images	Total Images
Missing Hole	115	115	230
Mouse Bite	115	115	230
Open Circuit	116	116	232
Short Circuit	116	116	232
Spur	115	115	230
Spurious Copper	116	116	232
Total	693	693	1,386

The dataset used in this study consists of 1,386 images, with half of them being original and the other half generated through random rotation augmentation. The distribution of images across the six PCB defect classes is shown in Table 1.

3.2 Model Architecture

YOLO11 is a modern deep learning architecture designed for real-time object detection tasks. As illustrated in Figure 2, the model consists of three core components: the Backbone, the Neck, and the Head. These components operate sequentially, transforming the input image into rich feature representations and ultimately yielding object class predictions and precise bounding box localizations.

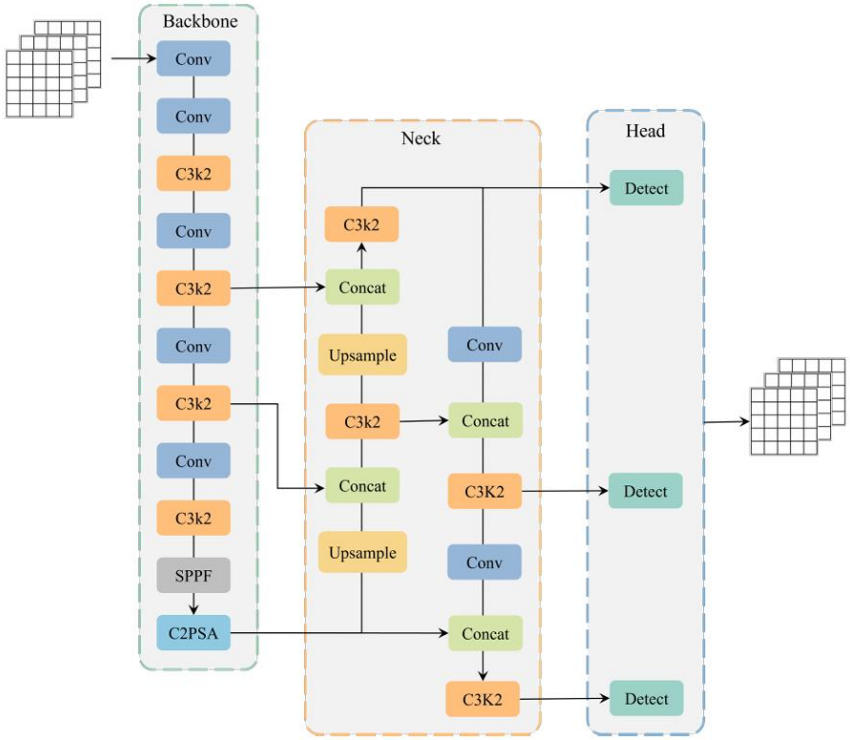


Figure 2. The architecture of YOLO11 [41]

Backbone

The backbone is responsible for extracting low- and high-level semantic features from the input image. It consists of repeated Conv (Convolutional) layers and C3k2 blocks. The C3k2 modules, featuring residual connections and multi-path convolutions, enable the model to learn spatially diverse and deep feature representations.

A key enhancement in the YOLO11 backbone is the inclusion of the SPPF (Spatial Pyramid Pooling - Fast) block, which consolidates contextual information from multiple receptive fields using parallel max-pooling operations. This is followed by the C2PSA (Channel and Spatial Attention) module, which improves feature map quality by selectively focusing on informative channels and spatial locations. These attention-based mechanisms are particularly useful for detecting fine-grained and subtle defects in complex PCB layouts [16].

Neck

The Neck functions as a feature aggregation and transmission layer. It fuses multi-scale feature maps obtained from the backbone using Upsampling, Concatenation, and additional C3k2 blocks. This hierarchical combination of features facilitates better object detection across scales [42].

The Neck follows a similar philosophy to Feature Pyramid Networks (FPN) and Path Aggregation Networks (PANet), ensuring both bottom-up and top-down information flow. As a result, fine spatial details and strong semantic signals are preserved, enabling robust detection of both small and large objects [43].

Head

The detection Head consists of three parallel branches that operate on feature maps of different resolutions. Each branch terminates in a Detect module, which outputs:

- Bounding box coordinates,
- Objectness scores, and
- Class probability distributions.

This multi-scale output structure allows YOLO11 to detect defects of varying sizes simultaneously and efficiently in a single forward pass.

Loss Function

The total loss $\mathcal{L}_{total}$ is expressed as a weighted sum of three components. The YOLO11 model is trained using a composite loss function that balances three key objectives in object detection: accurate localization of object boundaries, reliable objectness confidence, and precise classification of defect types. The total loss function is formulated as:

$$\mathcal{L}_{total} = \lambda_{loc}.\mathcal{L}_{CIoU} + \lambda_{obj}.\mathcal{L}_{obj} + \lambda_{cls}.\mathcal{L}_{cls} \tag{1}$$

where $\mathcal{L}_{CIoU}$, Complete Intersection over Union ($CIoU$) loss is used for bounding box regression. It incorporates the overlap area (IoU), the distance between the predicted and ground-truth box centers, and the aspect ratio consistency.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

In this formulation, b and b^{gt} denote the center points of the predicted and ground truth bounding boxes, respectively. The term $\rho(b, b^{gt})$ represents the Euclidean distance between centers, and c is the diagonal length of the smallest enclosing box covering both bounding boxes. The term v measures the similarity of aspect ratios between the predicted and actual boxes, and α is a positive scalar used to balance its influence. This comprehensive formulation allows the model to optimize for spatial accuracy, scale alignment, and geometric consistency [44].

$\mathcal{L}_{obj}$, objectness loss is calculated using Binary Cross-Entropy (BCE), measuring the confidence that an object exists in a predicted region.

$\mathcal{L}_{cls}$, classification loss evaluates the predicted class probabilities against ground truth labels. Either BCE or Focal Loss (FL) may be used, depending on the model configuration and class imbalance severity.

The weighting factors λ_{loc}, λ_{obj}, λ_{cls} are empirically selected to balance the contribution of each component and ensure effective learning during training.

3.3 Training Strategy

In this study, the YOLO11 model was trained on the PCB defect dataset [40] using Google Colab, leveraging both NVIDIA A100 and Tesla T4 GPUs to accelerate the training process. The training was conducted for 100 epochs with an image input size of 640×640 pixels and a batch size of 16. The model was optimized using the AdamW optimizer, with the learning rate ranging from 0.0001 to 0.01 throughout the training. The implementation was developed using Python 3.11.2 and PyTorch 2.0.1 with CUDA 11.8 support, and the Ultralytics YOLO11 framework (version 8.3.111) was employed for training and evaluation. [45]

To explore the performance variability and capabilities across different architectural scales, we trained all five variants of the YOLO11 family—YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x. These versions vary in terms of model size, number of parameters, and computational complexity, enabling a comparative analysis between speed and accuracy trade-offs. All training runs adhered to the same hyperparameter settings unless otherwise specified by the model's configuration. Optimization was performed using Stochastic Gradient Descent (SGD) with momentum, and learning rate scheduling was automatically managed by the YOLO11 training pipeline.

The performance of each variant was evaluated using standard object detection metrics such as mean Average Precision (mAP), mAP@50-95, and inference speed (FPS). The training times, model complexities, and accuracy results are reported in the Results and Discussion section to offer a comprehensive assessment.

3.4 Evaluation Metrics

To comprehensively evaluate the YOLO11 models applied in the PCB defect detection task, we utilized a range of quantitative evaluation metrics that are standard in the field of object detection. These metrics are crucial for assessing the model's ability to both correctly identify (classification) and accurately localize (detection) defect instances in PCB imagery. The following subsections provide detailed definitions, formulations, and roles of these metrics in our analysis.

Intersection over Union (IoU)

IoU measures the extent of overlap between the predicted bounding box and the ground truth bounding box. It is a critical metric for localization tasks, defined mathematically as:

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \times 100 \tag{3}$$

where B_p is the predicted bounding box, and B_{gt} is the ground truth bounding box. An IoU threshold (typically 0.5 or 0.75) is used to determine if a predicted box is considered a true positive.

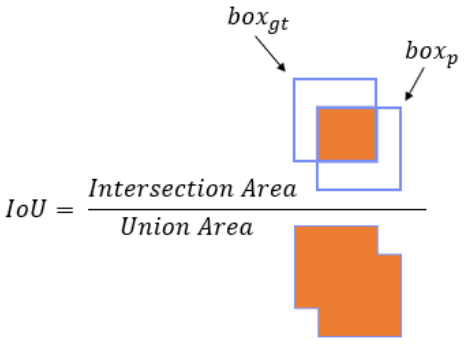


Figure 3. Intersection over union [46]

Figure 3 illustrates the IoU concept in a visual format. The orange area represents the overlapping region (intersection) between the predicted box_p and the actual ground truth box_{gt}. The union area, which includes both boxes, is used as the denominator. This visual summary also explains the rule-based decision: if $IoU > \propto$ the detection is a True Positive (TP); otherwise, it is a False Positive (FP).

Confusion Matrix

A confusion matrix is a tabular representation used to evaluate the performance of a classification model by comparing the model's predicted class labels with the true labels. In the context of multi-class problems—such as PCB defect detection involving multiple defect categories—the confusion matrix provides detailed insights into how well the model is performing on a per-class basis.

Each row of the matrix represents the actual class, while each column represents the predicted class. The diagonal elements indicate correct predictions (True Positives), whereas off-diagonal elements represent misclassifications (False Positives and False Negatives). Figure 4 illustrates the confusion matrix table.

This matrix helps identify which defect classes are accurately classified and which ones tend to be confused. For example, subtle defects like spur or open circuit may be misclassified more often than clearly visible ones like missing hole. Analyzing such patterns supports targeted improvements in model training, such as using data augmentation or loss weighting for underperforming classes.

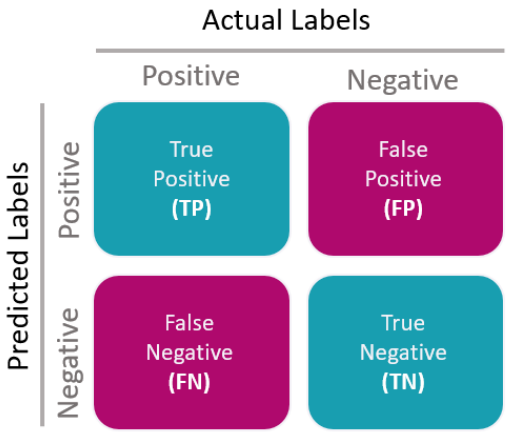


Figure 4. Confusion Matrix

Precision and Recall

These two metrics are used to evaluate classification performance:

- Precision indicates the proportion of true positive detections among all predicted positive instances:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

- Recall measures the proportion of actual positives that were correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Precision is crucial when the cost of false positives is high, whereas recall is vital when false negatives are costly.

Precision-Recall Curve

In addition to precision and recall as scalar metrics, the Precision-Recall (PR) curve provides a comprehensive view of the trade-off between these two metrics across varying classification thresholds.

It is especially informative in scenarios with imbalanced classes, such as PCB defect detection, where certain defect types may be significantly rarer than others.

The PR curve plots precision (x-axis) against recall (y-axis) for different confidence score thresholds. As the threshold decreases, the model tends to make more positive predictions, increasing recall but potentially reducing precision. A well-performing model maintains high precision and recall across thresholds, resulting in a curve that bends toward the top-right corner of the plot. An illustrative example of such behavior is shown in Figure 5.

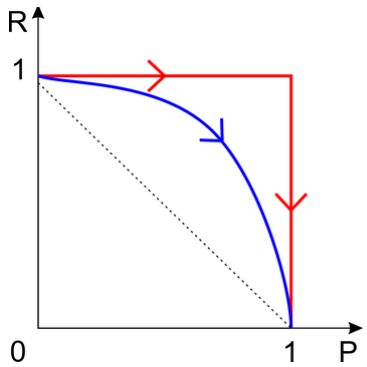


Figure 5. Examples of PR curve. The red curve represents an ideal PR curve, while the blue curve depicts a typical PR curve obtained in practical experiments. Arrows indicate the direction of increasing threshold levels [47].

Mean Average Precision (mAP)

Mean Average Precision (mAP) is a widely used evaluation metric in object detection tasks, measuring the detector's performance across all classes and multiple Intersection over Union (IoU) thresholds. It summarizes both the precision and recall characteristics into a single scalar value. The Average Precision (AP) for a single class is defined as the area under the precision-recall curve:

$$AP = \int_0^1 P(R)dR \tag{6}$$

where $P(R)$ denotes the precision as a function of recall R.

The overall mean Average Precision (mAP) across N classes are computed as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{7}$$

where AP_i represents the Average Precision for the i^{th} class, and N is the total number of classes.

Additionally, a more detailed evaluation metric, mAP_{50-95}, is often used. Instead of evaluating AP at a single IoU threshold, mAP_{50-95} computes the mean AP across ten IoU thresholds, ranging from 0.50 to 0.95 with an increment of 0.05, and is defined as:

$$mAP_{50-95} = \frac{1}{10} \times (AP@0.50 + AP@0.55 + \cdots + AP@0.95) \tag{8}$$

where $AP@t$ denotes the Average Precision at a specific IoU threshold t.

Higher values of mAP and mAP_{50-95} indicate better detection performance, combining both the model's classification accuracy and localization precision.

Inference Time

Inference time refers to the amount of time a trained model takes to process an input and generate a prediction. In the context of object detection, it measures the time required for the model to detect objects within a single image, typically reported in milliseconds (ms) or frames per second (FPS).

The average inference time per image can be calculated as:

$$t_{inference} = \frac{T_{total}}{N} \tag{9}$$

where T_{total} is the total time taken to infer N images during testing.

Alternatively, the inference speed is often reported as Frames Per Second (FPS), calculated as:

$$FPS = \frac{1}{t_{inference}} \tag{10}$$

Lower inference times and higher FPS values are crucial for real-time applications, particularly in industrial settings where rapid and accurate defect detection is necessary. In this study, inference times are evaluated across different YOLO11 variants to assess their suitability for real-time PCB inspection.

4. Results and Discussion

To validate the performance of the proposed YOLO11 variants, comprehensive experiments were conducted and compared with the most widely used object detection models. Table 2 summarizes the performance metrics, including mAP@0.5, mAP@0.5-0.95, inference speed (FPS), model size, computational cost (GFLOPs), hardware platform, and training time.

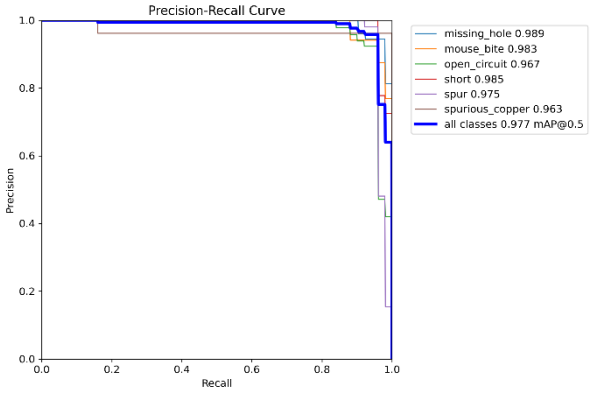
Table 2. Performance comparison of different models on the PCB defect detection dataset. (The best results in each column are highlighted in bold).

Model	**mAp$_{50}$ (%)**	**mAP$_{50-95}$ (%)**	**FPS**	**Parameters (M)**	**GFLOPs**	**System**	**Train Time**
YOLOv5 [48]	95.97	-	92.5	92.3	-	Nvidia RTX3090	-
YOLOx-Plus [49]	90.1	-	72.6	9.18	-	PYNQ-Z2 FPGA	-
YOLOv8n [50]	94.8	37.3	124	5.96	8.2		-
CDI-YOLO [51]	98.3	51.1	128	5.76	12.6		-
YOLOv8-s [52]	90.8	-	-	5.79	-	AMD E5-2698B	-
YOLOv7 [53]	92.8	-	48.6	-	-		-
Faster R-CNN [54]	73.14	-	6	522.98	-	Nvidia RTX2070	-
SSD [54]	81.53	-	14	100.27	-	Nvidia RTX2070	-
Transformer YOLO [54]	97.0		21	93.95		Nvidia RTX2070	
YOLO11n	97.7	51.9	**166**	**2.5**	**6.3**	Nvidia A100	**0.174**
YOLO11n	98.3	51.8	27	**2.5**	**6.3**	Tesla T4	1.031
YOLO11s	94.8	52.9	100	2.6	9.4	Nvidia A100	0.178
YOLO11m	98.8	53.8	38	20	67.7	Nvidia A100	0.215
YOLO11l	98.5	**55.1**	32	25.5	86.6	Nvidia A100	0.268
YOLO11l	**99.2**	54.6	6	25.5	86.6	Tesla T4	1.340
YOLO11x	98.6	52.9	21	65.8	194.4	Nvidia A100	0.332

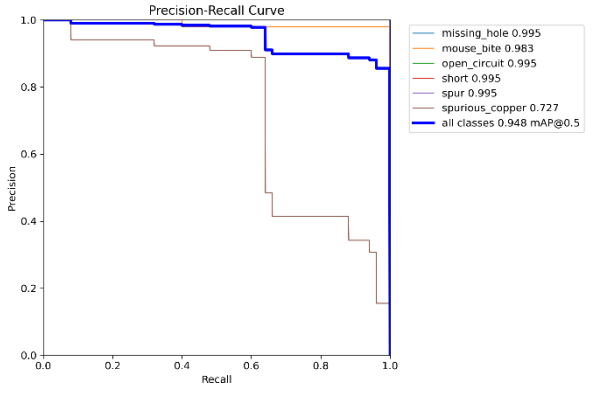
The experimental results demonstrate that the YOLO11 family achieves competitive performance compared to other object detection models reported in the literature. As presented in Table 2, the YOLO11n variant reached a mAP@50 of 0.983 and a mAP@50-95 of 0.518 on the Tesla T4

platform, while maintaining an inference speed of 27 FPS. On the NVIDIA A100, the same model reached up to 166 FPS, significantly outperforming traditional models such as Faster R-CNN (6 FPS) and SSD (14 FPS), and showing better speed-accuracy trade-off than YOLOv7 (48.63 FPS) and YOLOx-Plus (72.6 FPS).

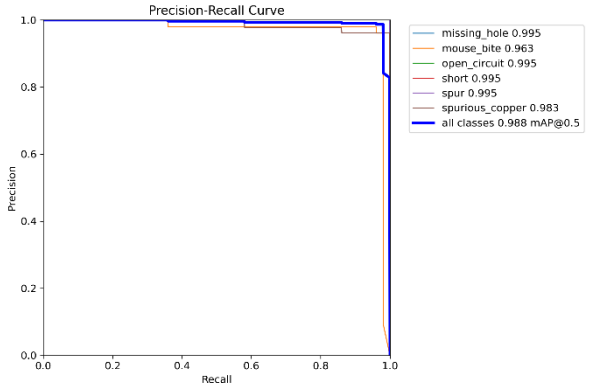
Although CDI-YOLO and YOLOv8n demonstrated high inference speeds (128 FPS and 124.8 FPS respectively), YOLO11 variants such as YOLO11m and YOLO11l achieved better overall accuracy while preserving real-time capability. For instance, YOLO11m achieved a mAP@50-95 of 0.538 with 38 FPS, and YOLO11l delivered the highest mAP@50-95 value of 0.551 at 32 FPS. Even the largest variant, YOLO11x, maintained a reasonable speed of 21 FPS on the A100 while achieving strong detection results.



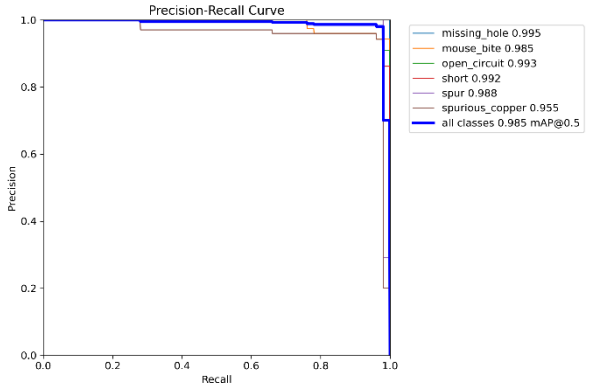
(a) YOLO11n (Nvidia A100)



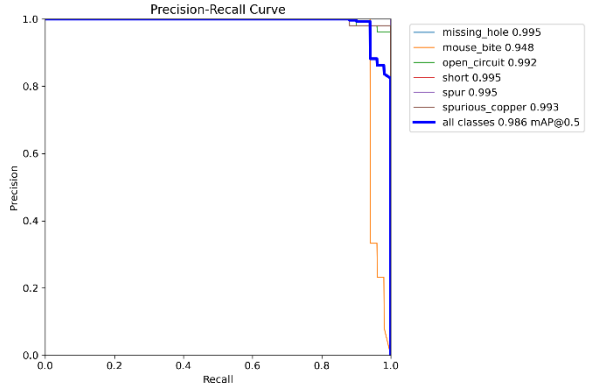
(b) YOLO11s (Nvidia A100)



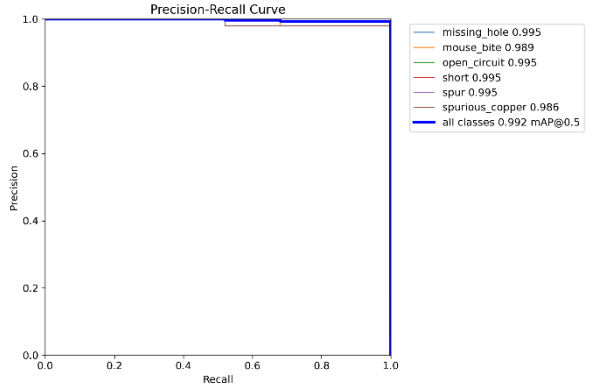
(c) YOLO11m (Nvidia A100)



(d) YOLO11l (Nvidia A100)



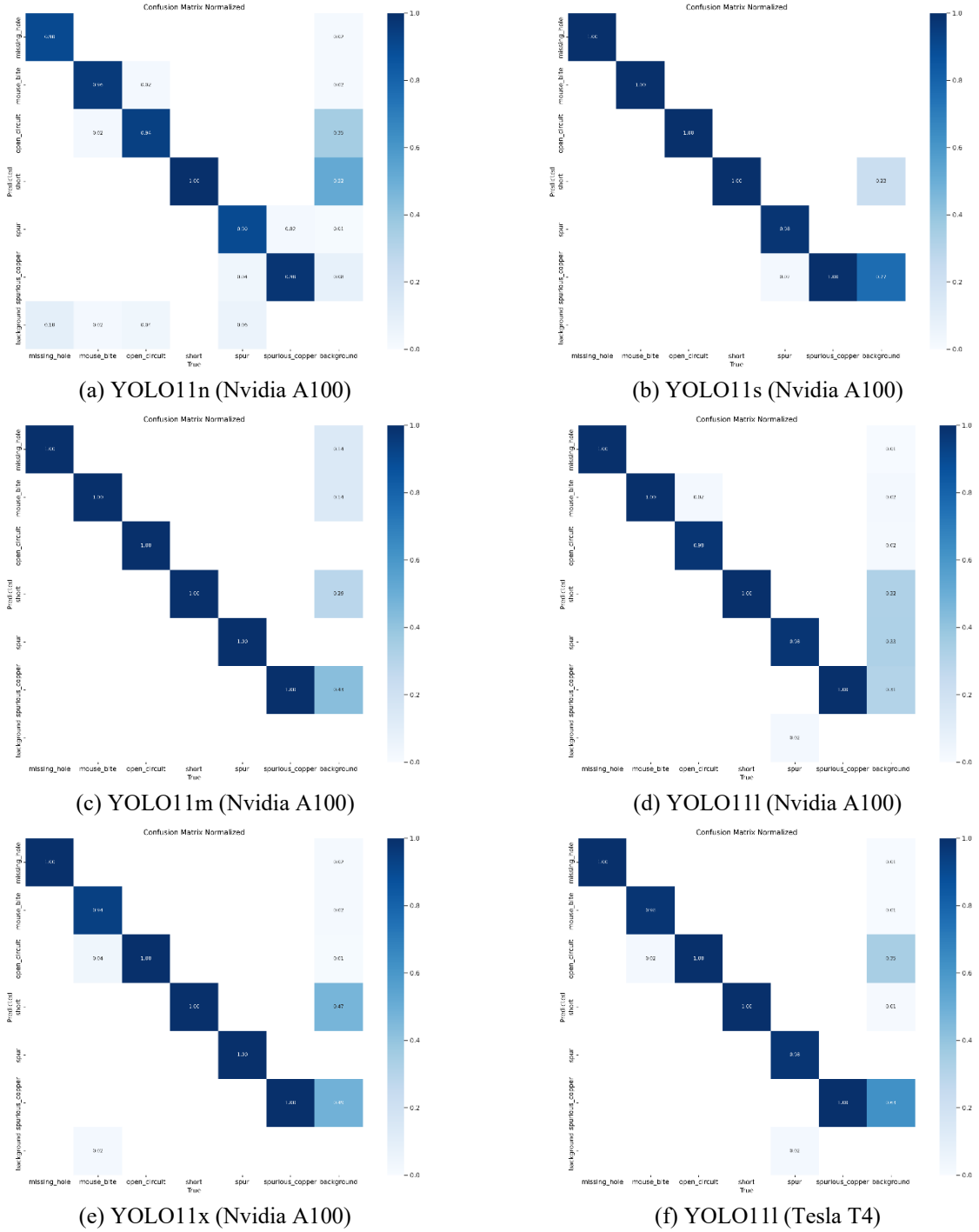
(e) YOLO11x (Nvidia A100)



(f) YOLO11l (Tesla T4)

Figure 6. PR curves of YOLO11 variants: (a) YOLO11n (Nvidia A100); (b) YOLO11s (Nvidia A100); (c) YOLO11m (Nvidia A100); (d) YOLO11l (Nvidia A100); (e) YOLO11x (Nvidia A100); (f) YOLO11l (Tesla T4)

PR curves were used in our analysis to visualize the classification performance of the YOLO11 models across all six defect classes. Figure 6 illustrates the PR curves for each class across all YOLO11 model variants. These results further highlight the effectiveness of the YOLO11l variant within the YOLO11 model family.



(a) YOLO11n (Nvidia A100)

(b) YOLO11s (Nvidia A100)

(c) YOLO11m (Nvidia A100)

(d) YOLO11l (Nvidia A100)

(e) YOLO11x (Nvidia A100)

(f) YOLO11l (Tesla T4)

Figure 7. Confusion matrix tables of YOLO11 variants: (a) YOLO11n (Nvidia A100); (b) YOLO11s (Nvidia A100); (c) YOLO11m (Nvidia A100); (d) YOLO11l (Nvidia A100); (e) YOLO11x (Nvidia A100); (f) YOLO11l (Tesla T4)

Figure 7 shows the normalized confusion matrices for each model variant. All variants demonstrate strong classification performance, with high values along the diagonals indicating accurate

predictions across classes. Among them, YOLO11m, YOLO11l, and YOLO11x show slightly higher diagonal concentration, reflecting their relatively better performance in distinguishing between defect types.

To illustrate real-world applicability, Figure 8 to Figure 12 shows a representative detection output from the YOLO11n, YOLO11s, YOLO11m, YOLO11l, YOLO11x models on a PCB image with multiple coexisting defect types. The model accurately identifies and localizes diverse defects in a single forward pass, underscoring the YOLO11 architecture's robustness in complex industrial scenarios.

These results collectively validate the efficacy of the YOLO11 family in balancing detection precision and computational efficiency, confirming their suitability for high-throughput, multi-class PCB inspection systems.

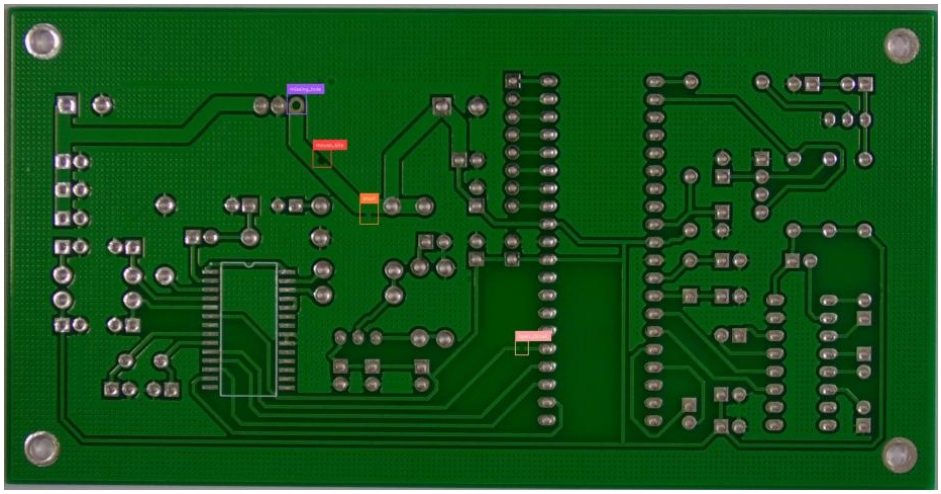


Figure 8. Detection output of the YOLO11n model on a PCB image containing multiple defect types.

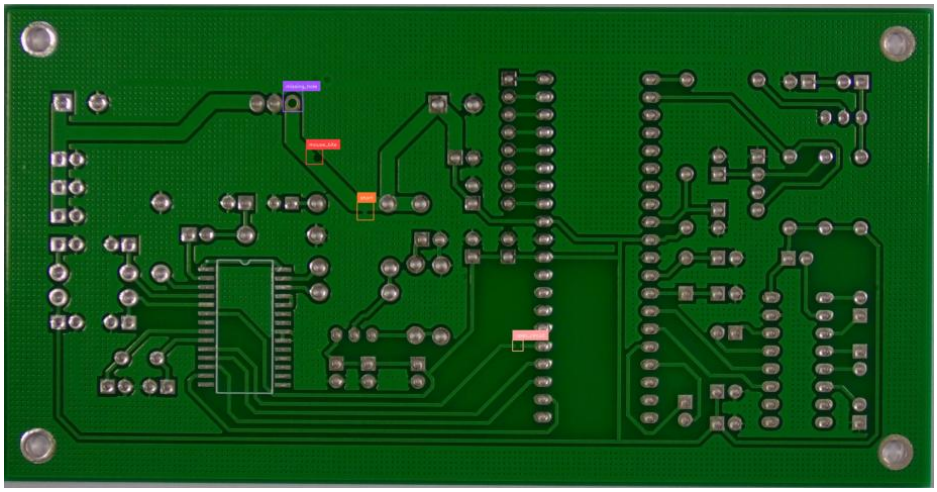


Figure 9. Detection output of the YOLO11s model on a PCB image containing multiple defect types.

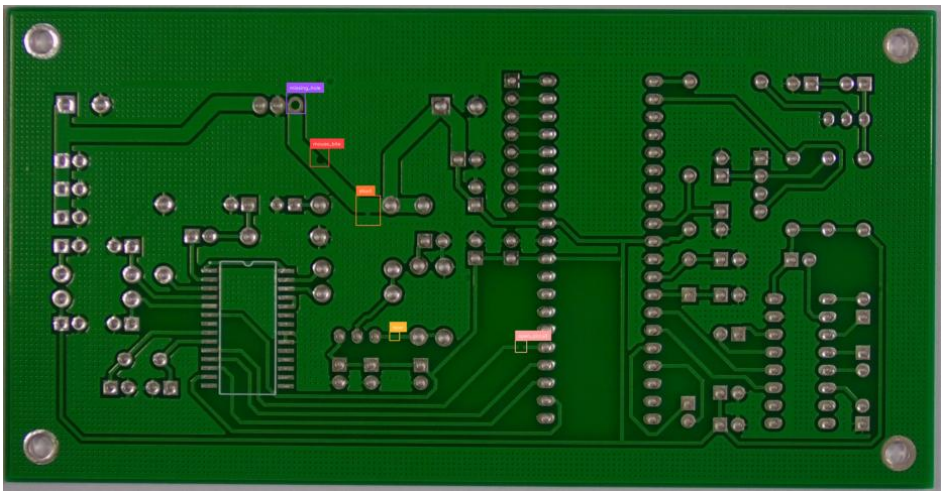


Figure 10. Detection output of the YOLO11m model on a PCB image containing multiple defect types.

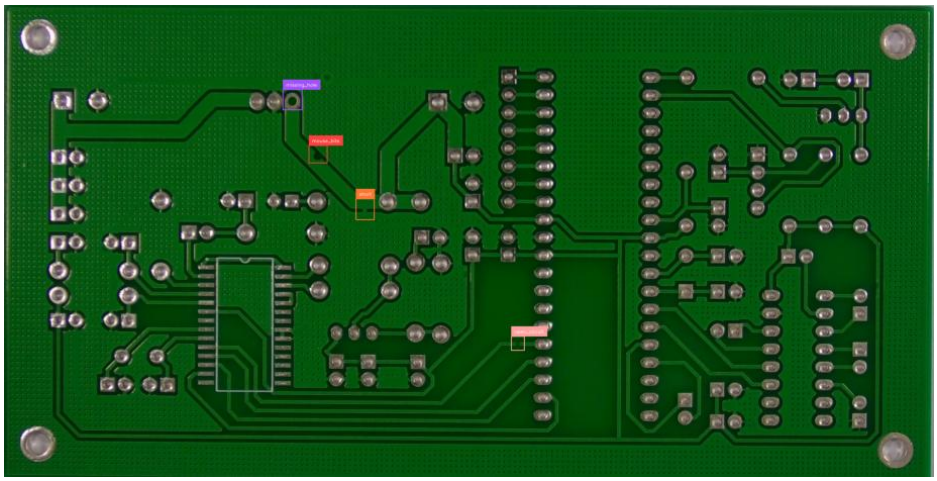


Figure 11. Detection output of the YOLO11l model on a PCB image containing multiple defect types.

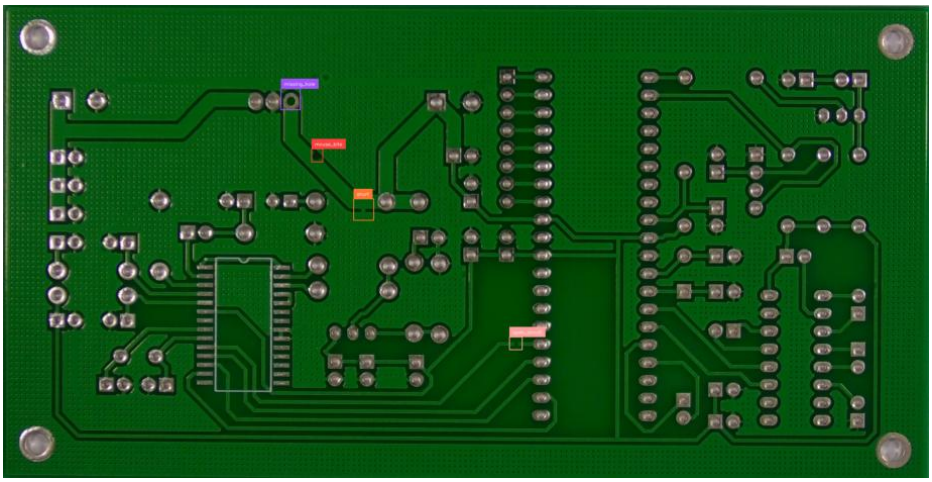


Figure 12. Detection output of the YOLO11x model on a PCB image containing multiple defect types.

5. Conclusions

This study provided a systematic evaluation of the YOLO11 object detection family for multi-class PCB defect detection. Through consistent training protocols and fair comparisons, we analyzed five model variants across multiple hardware configurations.

Key findings from this research include:

- YOLO11l achieved the highest overall detection accuracy (mAP@50-95 = 0.551), making it the most precise model for defect localization.
- YOLO11n demonstrated outstanding inference speed (up to 166 FPS), highlighting its potential for real-time deployment in resource-constrained environments.
- All YOLO11 variants showed competitive or superior performance compared to leading state-of-the-art models in both accuracy and efficiency.

The inclusion of advanced architectural features—such as attention mechanisms, enhanced feature fusion modules, and a decoupled detection head—contributed significantly to the models' ability to detect small, low-contrast, and coexisting defects in high-resolution images.

In conclusion, YOLO11 provides a robust framework for the automation of PCB inspection processes. It allows for flexibility in levels of model sophistication, as well as hardware interfaces, making it ideal for industrial settings where precision and speed are key. Research could also investigate further domain adaptation strategies for novel PCB layout representations, procedures for semi-supervised classification to reduce the labeling burden, and powerful and efficient methods for implementation on edge and embedded systems. Also, adding reasoning for spatial and temporal relationships for time-series analysis of PCBs along with explainable AI could strengthen performance and reliability for interpretable analysis while enhancing multi-modal data analysis fusion. Building self-adaptive architectures combined with continual learning paradigms is likely to enable enduring sustained effectiveness tailored to shifting needs in the context of agile responsive manufacturing systems. All these approaches offer a unified adaptive view of PCB inspection quality control, bringing intelligence, automation, and resilience into one system.

Acknowledgment

The authors would like to express their gratitude to the Huang and Wei, the researchers who developed the PCB defect dataset used in this study [40]. Their contribution has been fundamental in enabling advancements in automated defect detection research. Additionally, the authors acknowledge the Ultralytics team for their development and continuous improvement of the YOLO object detection models, which served as the backbone architecture for this research.

References

[1] V. K. Ancha, F. N. Sibai, V. Gonuguntla, R. Vaddi, (2024). Utilizing YOLO models for real-world scenarios: Assessing novel mixed defect detection dataset in PCBs, IEEE Access. 12, 100983–100990. https://doi.org/10.1109/ACCESS.2024.3430329.

[2] Q. Ling, N.A.M Isa, (2023). Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey, IEEE Access. 11, 15921–15944. https://doi.org/10.1109/ACCESS.2023.3245093.

[3] K. Singh, S. Kharche, A. Chauhan, P. Salvi, (2024). PCB defect detection methods: A review of existing methods and potential enhancements, Journal of Engineering Science & Technology Review. 17(1), 156-167. https://doi.org/10.25103/jestr.171.19.

[4] I.-C. Chen, R.-C. Hwang, H.-C Huang, (2023). PCB defect detection based on deep learning algorithm, Processes. 11(3), 775. https://doi.org/10.3390/pr11030775.

[5] L. Cai, J. Li, (2022). PCB defect detection system based on image processing. Journal of Physics: Conference Series, Qingdao, China, Conf. Ser. 2383, pp. 012077. https://doi.org/10.1088/1742-6596/2383/1/012077.
[6] G. Zhang, Y. Cao, (2023). A novel PCB defect detection method based on digital image processing. Journal of Physics: Conference Series, Suzhou, China, Conf. Ser. 2562, pp. 012030. https://doi.org/10.1088/1742-6596/2562/1/012030.
[7] S. H. I. Putera, Z. Ibrahim, (2010). Printed circuit board defect detection using mathematical morphology and MATLAB image processing tools. 2nd International Conference on Education Technology and Computer, Shanghai, China, pp. 359. https://doi.org/10.1109/ICETC.2010.5530052.
[8] M. Baygin, M. Karakose, A. Sarimaden, E. Akin, (2017). Machine vision based defect detection approach using image processing. 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, pp. 1–5. https://doi.org/10.1109/IDAP.2017.8090292.
[9] H. Hagi, Y. Iwahori, S. Fukui, Y. Adachi, M. K. Bhuyan, (2014). Defect classification of electronic circuit board using SVM based on random sampling, Procedia Computer Science. 35, 1210–1218. https://doi.org/10.1016/j.procs.2014.08.218.
[10] P. P. Londe, S. A. Chavan, (2014). Automatic PCB defects detection and classification using MATLAB, International Journal of Current Engineering and Technology. 4(3), 31–36.
[11] E. H Yuk, S. H. Park, C.-S. Park, J.-G. Baek, (2018). Feature-learning-based printed circuit board inspection via Speeded-Up Robust Features and Random Forest, Applied Sciences. 8(6), 932. https://doi.org/10.3390/app8060932.
[12] Y. Li, S. Li, (2017). Defect detection of bare printed circuit boards based on gradient direction information entropy and uniform local binary patterns, Circuit World. 43(4), 145–151. https://doi.org/10.1108/CW-06-2017-0028.
[13] J. Niu, J. Huang, L. Cui, B. Zhang, A. Zhu, (2022). A PCB defect detection algorithm with improved Faster R-CNN. International Conference on Big Data Applications and Services Engineering (ICBASE), Guangzhou, China, pp. 283–292.
[14] W. Shi, Z. Lu, W. Wu, H. Liu, (2020). Single-shot detector with enriched semantics for PCB tiny defect detection, The Journal of Engineering. 2020(10), 366-372. https://doi.org/10.1049/joe.2019.1180.
[15] V. A Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod, J.-S. Shieh, (2020). Defect detection in printed circuit boards using You-Only-Look-Once convolutional neural networks, Electronics. 9(9), 1547. https://doi.org/10.3390/electronics9091547.
[16] R. Khanam, M. Hussain, (2024). YOLOv11: An overview of the key architectural enhancements. arXiv. https://doi.org/10.48550/arxiv.2410.17725.
[17] N. Dave, V. Tambade, B. Pandhare, S. Saurav, (2016). PCB defect detection using image processing and embedded system, International Research Journal of Engineering and Technology (IRJET). 3(5), 1897–1901.
[18] L. Zhou, (2025). Research on PCB defect detection method based on principal component analysis. International Technology and Innovation Conference (ITOEC), Chongqing, China, pp. 1070–1074. https://doi.org/10.1109/ITOEC63606.2025.10967656.
[19] Y. Chang, Y. Xue, Y. Zhang, J. Sun, Z. Ji, H. Li, T. Wang, J. Zuo, (2024). PCB defect detection based on PSO-optimized threshold segmentation and SURF features, Signal, Image and Video Processing. 18, 4327–4336. https://doi.org/10.1007/s11760-024-03075-7.
[20] C. Zhang, W. Shi, X. Li, H. Zhang, H. Liu, (2018). Improved bare PCB defect detection approach based on deep feature learning, The Journal of Engineering. 2018(16), 1415–1420. https://doi.org/10.1049/JOE.2018.8275.
[21] B. Hu, J. Wang, (2020). Detection of PCB surface defects with improved Faster-RCNN and feature pyramid network, IEEE Access. 8, 108335–108345. https://doi.org/10.1109/ACCESS.2020.3001349.
[22] G. Shao, Y. Zhang, T. Li, J. Wu, J. Luo, F. Gao, J. Ma, T. Liu, (2021). An improved YOLOv3 network for PCB defect detection. China Automation Congress (CAC), Beijing, China, pp. 1819–1823. https://doi.org/10.1109/CAC53003.2021.9728216.

[23] A. Chaudhari, V. Upganlawar, T. Barve, R. Vaidya, D. Shelke, (2024). Analysis of YOLO v3 for multiple defects detection in PCB. 2024 Parul International Conference on Engineering and Technology (PICET), Vadodara, India, pp. 1–6. https://doi.org/10.1109/PICET60765.2024.10716153.
[24] M. Liang, J. Wu, H. Cao, (2022). Research on PCB small target defect detection based on improved YOLOv5. 2022 International Conference on Sensing, Measurement & Data Analytics in the Era of Artificial Intelligence (ICSMD), Harbin, China, pp. 1–5. https://doi.org/10.1109/ICSMD57530.2022.10058458.
[25] Y. Zhang, H. Zou, J. Wang, Z. Lei, M. Zhou, (2023). Lightweight neural network-based real-time PCB defect detection system. 2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS), Yibin, China, pp. 1–6. https://doi.org/10.1109/SAFEPROCESS58597.2023.10295891.
[26] Q Li, L. Wu, H. Xiao, C. Huang, (2024). PCB-DETR: A detection network of PCB surface defect with spatial attention offset module, IEEE Access. 12, 158436–158445. https://doi.org/10.1109/ACCESS.2024.3486176.
[27] Y. Pan, L. Zhang, Y. Zhang, (2024). Rapid detection of PCB defects based on YOLOx-Plus and FPGA, IEEE Access. 12, 61343–61358. https://doi.org/10.1109/ACCESS.2024.3387947.
[28] X. Huang, W. Li, (2024). A novel PCB defect detection network based on the improved YOLOv8 with fusion of hybrid attention transformer and bidirectional feature pyramid network. 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC), Xiamen, China, pp. 207–211. https://doi.org/10.1109/ICAIRC64177.2024.10900305.
[29] X. Gu, C. Cao, Z. Xu, (2024). Research on PCB defect detection algorithm based on improved YOLOv8. 2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC), Qingdao, China, pp. 1103–1108. https://doi.org/10.1109/ICFTIC64248.2024.10913102.
[30] Q. Zeng, C. Zhao, P. He, H. Gao, (2024). LSDM-PCB: A lightweight small defect detection model for printed circuit board. 2024 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, pp. 673–679 https://doi.org/10.1109/ICIP51287.2024.10647590.
[31] S. You, (2022). PCB defect detection based on generative adversarial network. 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, pp. 557–560. https://doi.org/10.1109/ICCECE54139.2022.9712737.
[32] C. Chen, Q. Wu, J. Zhang, H. Xia, P. Lin, Y. Wang, M. Tian, R. Song, (2024). U2D2PCB: Uncertainty-aware unsupervised defect detection on PCB images using reconstructive and discriminative models, IEEE Transactions on Instrumentation and Measurement. 73, 1-10. https://doi.org/10.1109/TIM.2024.3386210.
[33] D. Lang, Z. Lv, (2025). SEPDNet: Simple and effective PCB surface defect detection method, Scientific Reports. 15, 10919. https://doi.org/10.1038/s41598-024-84859-2.
[34] L. Zhu, R. Zhao, (2025). A novel PCB surface defect detection method based on separated global context attention to guide residual context aggregation, Scientific Reports. 15, 9620. https://doi.org/10.1038/s41598-024-84961-5.
[35] X. Zhao, H. Zhang, C. Song, H. Li., H. Guo, (2025). Lightweight intelligent detection algorithm for surface defects in printed circuit board, Computers & Industrial Engineering. 203, 111030. https://doi.org/10.1016/j.cie.2025.111030.
[36] W. Shi, Z. Lu, W. Wu, H. Liu, (2020). Single-shot detector with enriched semantics for PCB tiny defect detection, The Journal of Engineering. 2020(3), 366–372. https://doi.org/10.1049/joe.2019.1180.
[37] L. Kang, Y. Ge, H. Huang, M. Zhao, (2022). Research on PCB defect detection based on SSD. 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, pp. 1315–1319. https://doi.org/10.1109/ICCASIT55263.2022.9986754.
[38] K. P. Anoop, N. S. Sarath, V. V. Kumar, (2015). A review of PCB defect detection using image processing, International Journal of Engineering and Innovative Technology (IJEIT). 4(11), 188–192.
[39] Q. Ling, N. A. M. Isa, (2023). Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey, IEEE Access. 11, 15921–15944. https://doi.org/10.1109/ACCESS.2023.3245093.

[40] W. Huang, P. Wei, (2019). A PCB Dataset for Defects Detection and Classification. arXiv. 10.48550/arXiv.1901.08204.
[41] W. Zhu, X. Han, K. Zhang, S. Lin, J. Jin, (2025). Application of YOLO11 model with spatial pyramid dilation convolution (SPD-Conv) and effective squeeze-excitation (EffectiveSE) fusion in rail track defect detection, Sensors. 25(8), 2371. https://doi.org/10.3390/s25082371.
[42] P. Hidayatullah, N. Syakrani, M. R. Sholahuddin, T. Gelar, R. Tubagus, (2025). YOLOv8 to YOLO11: A comprehensive architecture in-depth comparative review. arXiv. https://doi.org/10.48550/arXiv.2501.13400.
[43] N. Jegham, C. Y. Koh, M. Abdelatti, A. Hendawi, (2025). YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions. arXiv. https://doi.org/10.48550/arXiv.2411.00201
[44] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, (2020). Distance-IoU loss: Faster and better learning for bounding box regression. AAAI Conference on Artificial Intelligence, New York, USA, 34(07), pp. 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999.
[45] G. Jocher, J. Qiu, A. Chaurasia, (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.
[46] L. Rampini, F. R. Cecconi, (2024). Synthetic images generation for semantic understanding in facility management, Construction Innovation. 24(1), 33–48. https://doi.org/10.1108/CI-09-2022-0232.
[47] I. Dedovic, (2017). Efficient probability distribution function estimation for energy based image segmentation methods (Doctoral Thesis). RWTH Aachen University, Germany https://doi.org/10.18154/RWTH-2016-10831.
[48] J. Tang, S. Liu, D. Zhao, L. Tang, W. Zou, B. Zheng, (2023). PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOv5, Sustainability. 15(7), 5963. https://doi.org/10.3390/su15075963.
[49] Y. Pan, L. Zhang Y. Zhang, (2024). Rapid detection of PCB defects based on YOLOx-Plus and FPGA, IEEE Access. 12, 61343–61358. https://doi.org/10.1109/ACCESS.2024.3387947.
[50] J. An, Z. Shi, (2024). YOLOv8n-enhanced PCB defect detection: A lightweight method integrating spatial–channel reconstruction and adaptive feature selection, Applied Sciences. 14(17), 7686. https://doi.org/10.3390/app14177686.
[51] G. Xiao, S. Hou, H. Zhou, (2024). PCB defect detection algorithm based on CDI-YOLO, Scientific Reports. 14, 7351. https://doi.org/10.1038/s41598-024-57491-3.
[52] C. Liu, X. Zhou, J. Li, C. Ran, (2023). PCB board defect detection method based on improved YOLOv8, Frontiers in Computing and Intelligent Systems. 6(2), 1–6. https://doi.org/10.54097/fcis.v6i2.01.
[53] H. Guo, H. Zhao, Y. Zhao, W. Liu, (2024). PCB defect detection algorithm based on deep learning, Optik. 315, 172036. https://doi.org/10.1016/j.ijleo.2024.172036.
[54] W.Chen, Z. Huang, Q. Mu, Y. Sun, (2022). PCB defect detection method based on Transformer-YOLO, IEEE Access. 10, 129480–129489. https://doi.org/10.1109/ACCESS.2022.3228206.