Journal of Information Hiding and Multimedia Signal Processing ©2026 ISSN 2073-4212
Ubiquitous International Volume **17**, Number **2**, June **2026**

Evaluating YOLO Architectures for PCB Component Detection: A Comparative Analysis of Accuracy and Model Stability

Xuan-Huy Nguyen[1], Thanh-Tuan Nguyen[1]*, Thuc-Minh Bui[1]
[1] Department of Electrical and Electronic Engineering
Nha Trang University, Khanh Hoa 650000, Vietnam
huynx@ntu.edu.vn, tuannt@ntu.edu.vn, minhbt@ntu.edu.vn

*Corresponding author: Thanh-Tuan Nguyen

Received January 15, 2026, revised April 9, 2026, accepted April 13, 2026.

Abstract. *Automated Optical Inspection (AOI) for component verification on Printed Circuit Boards (PCBs) is a critical quality control process. This paper presents an empirical comparison of three contemporary YOLO architectures (v10, v11, and v12) to determine the optimal model for this computer vision task. Models were trained on a custom PCB dataset and evaluated based on both mAP (mean Average Precision) for accuracy and validation loss dynamics for generalization capability. Our findings show that all three architectures achieved competitive mAP@50-95 scores (YOLOv10: 0.7309; v11: 0.7295; v12: 0.7285). However, a significant disparity was observed in training stability. YOLOv10 demonstrated pronounced overfitting, with validation loss metrics nearly 100% higher than its counterparts. In contrast, YOLOv11 provided the most favorable trade-off, combining high-fidelity detection (mAP: 0.7295) with superior convergence stability, evidenced by the lowest validation classification loss (val/cls_loss: 0.3216). This indicates superior generalization. We conclude that YOLOv11 is the most robust architecture for this industrial application, offering the optimal balance between detection accuracy and reliable performance on unseen data.*
Keywords: Automated Optical Inspection; PCB Component Detection; YOLO Architectures; Model Stability; Deep Learning

1. **Introduction.** In the contemporary era of Industry 4.0, the comprehensive automation of electronics manufacturing processes has become a paramount objective [1]. Printed Circuit Boards serve as the foundational architecture for virtually all modern electronic devices, ranging from consumer gadgets to critical industrial machinery. Consequently, assuring their structural integrity and quality, specifically the verification of the correct presence, orientation, and placement of distinct components, constitutes a critical step in the production line. Traditionally, the industry has relied heavily upon Automated Optical Inspection systems to maintain rigorous quality control standards [2]. Historically, these inspection frameworks depended primarily on classical computer vision algorithms to identify defects [3]. These conventional methodologies typically utilize template matching techniques or explicit feature extraction methods that require extensive manual engineering. While these algorithmic approaches may function adequately within highly constrained and invariant environments, they exhibit significant fragility when faced with the complexities of modern manufacturing.

They frequently struggle to adapt to the increasing density of components, the vast diversity of component types, and complex lighting conditions or photometric variations. Such limitations often result in elevated false positive rates and necessitate substantial manual intervention, thereby creating significant bottlenecks in production throughput and efficiency [4]. The advent of Deep Learning, and particularly the rapid development of Convolutional Neural Networks, has catalyzed a fundamental paradigm shift in the field of computer vision [5]. These advanced neural architectures have dramatically surpassed

the performance capabilities of classical methods in visual recognition tasks. Within the specific domain of object detection, the You Only Look Once architectural family has demonstrated superior efficacy [6]. These models are renowned for their exceptional ability to detect objects with high accuracy while maintaining inference speeds sufficient for immediate processing requirements on the assembly line. The architecture has continuously evolved, with new variants such as version 10 and other refined iterations appearing in recent literature, promising further gains in both computational efficiency and detection precision. Nevertheless, selecting the optimal architecture for a specific industrial application such as PCB component detection remains a complex and challenging matter. A significant portion of existing research tends to focus predominantly on reporting peak performance metrics, such as mean Average Precision, as the singular benchmark of success. However, in a practical manufacturing environment, factors such as operational stability and generalization capabilities are equally, if not more, critical. Generalization refers to the ability of the model to perform reliably on new or unseen data that was not present during the training phase. A model that achieves the maximal mean Average Precision but suffers from severe overfitting becomes inherently unreliable upon deployment. Overfitting occurs when a model memorizes the specific details of the training data rather than learning representative features, leading to poor performance and divergent loss metrics on validation data [7].

The main contributions of this study are threefold. First, we provide a controlled comparison of YOLOv10, YOLOv11, and YOLOv12 for PCB component detection under the same dataset split, training schedule, and hardware setting. Second, we evaluate the models with a dual emphasis on detection accuracy and training stability, using validation loss behavior together with mAP rather than relying on peak accuracy alone. Third, we show that YOLOv11 offers the most reliable accuracy-stability trade-off for this dataset, whereas YOLOv10 attains a slightly higher peak mAP but exhibits markedly less stable localization behavior.

The remainder of this paper is organized as follows. Section 2 presents the dataset and preprocessing methods utilized in this work. Section 3 details the experimental setup and the specific model architectures being compared. Section 4 analyzes and discusses the empirical results with a focus on training dynamics, and Section 5 provides the concluding remarks and outlines directions for future research.

2. **Related work.**

2.1. **Classical Approaches to AOI.**

The initial generation of solutions developed for the inspection of printed circuit boards was predominantly characterized by the utilization of classical computer vision algorithms [8]. These early methodologies primarily depended upon techniques such as template matching, morphological analysis, and descriptor methods that required explicit and manual engineering [9]. Prominent examples of these techniques include the Scale Invariant Feature Transform and Speeded Up Robust Features. These approaches operate on the premise of extracting specific visual features from an image and comparing them against a predefined reference database to identify components or detect defects.

Although these algorithmic approaches demonstrated functional utility within manufacturing environments that were highly constrained and controlled, they exhibited significant fragility when applied to more dynamic or variable settings. A primary limitation of these conventional methods is their distinct lack of robustness regarding photometric variations. They are frequently susceptible to errors caused by fluctuating lighting conditions, surface reflections from solder joints, or changes in contrast. Furthermore, they struggle to accommodate geometric transformations, including component rotation and scale variance, which are common occurrences during the assembly process [4].

Consequently, the escalating density of components and the persistent trend toward miniaturization which characterize modern electronics manufacturing have rendered these traditional methods largely inadequate. The reliance on rigid feature definitions means that these systems often fail to adapt to new component types without extensive reprogramming. This inflexibility frequently results in elevated rates of erroneous detections, specifically incorrect positive identifications where valid components are flagged as defects. Such limitations present substantial challenges regarding scalability and operational efficiency for industrial applications that demand high throughput and precision [10].

2.2. **Deep Learning in Object Detection.**

The advent of CNN initiated a fundamental paradigm shift in the domain of computer vision, dramatically surpassing the performance capabilities of classical methods in visual recognition tasks. Unlike traditional algorithmic approaches that rely on manual feature engineering, deep neural networks possess the inherent capability to learn hierarchical feature representations directly from raw input data. Within the specific landscape of modern object detection,

two dominant architectural families have emerged to address the challenge: two stage detectors and single stage detectors [5].

The first category comprises two stage detectors, exemplified by the Region Based Convolutional Neural Network family, which includes prominent architectures such as Fast RCNN and Faster RCNN [11]. These models decouple the detection problem into two distinct and sequential phases. In the initial phase, a Region Proposal Network generates a sparse set of candidate locations known as Regions of Interest. Subsequently, in the second phase, a dedicated classifier refines these proposals to determine specific class probabilities and precise bounding box coordinates. While this methodological approach typically yields high localization accuracy, it suffers from significant computational latency due to the complexity of the dual pipeline. This inherent delay renders such architectures suboptimal for inspection scenarios that demand immediate processing speeds or high throughput on production lines [12].

In contrast, single stage detectors, such as the Single Shot MultiBox Detector and the YOLO series, frame object detection as a unified regression problem [13]. These architectures eliminate the region proposal step entirely to streamline the inference process. Instead, they perform dense sampling across the input image and predict bounding boxes and class probabilities simultaneously in a single forward pass of the network. This distinct design philosophy enables superior inference speeds compared to their two stage counterparts. Consequently, these models are considered ideal candidates for automated optical inspection systems where maintaining real time performance is a mandatory requirement [14].

2.3. **YOLO Architectures for PCB Component Analysis and Research Gap.** Given its compelling equilibrium between inference speed and accuracy, the YOLO family has been investigated extensively for inspection tasks regarding printed circuit boards [15], [16]. Early academic inquiries applying version 3 and version 4 demonstrated feasibility in detecting larger components and gross defects. Subsequent research utilizing more recent iterations, such as version 5 and version 7, has shown improved efficacy in more challenging scenarios, including the detection of small, minute, and occluded components. These advancements suggest that single stage detectors are increasingly viable for complex industrial quality control [17].

Recent adjacent work also supports the importance of architecture choices under visually complex conditions. Dang reported a recent JIHMSP defect-detection study that used visual prompting to improve robustness in insulator inspection [23], while Sung et al. improved YOLOv8 with a lightweight neck for complex images and reported stronger detection performance under visually challenging conditions [24]. These studies reinforce the broader point that deployment-ready detection depends on design choices that preserve robustness, not only on nominal peak accuracy.

However, the existing literature in this specific domain exhibits two primary limitations. First, numerous studies function primarily as feasibility evaluations, focusing predominantly on reporting the final mean Average Precision as the singular benchmark of success while neglecting other critical performance indicators. Second, there is a distinct absence of systematic and comparative analyses regarding the latest architectures, specifically version 10 and its derivatives, under rigorously controlled and identical experimental conditions [18].

The predominant focus on peak accuracy within the literature obscures a critical aspect of deployability in industrial settings, specifically model stability and generalization. In a practical automated optical inspection system, a model that overfits is unreliable, even if it achieves high validation precision. This generalization gap is frequently overlooked. A model that fails to generalize will perform unpredictably on new board layouts or components not perfectly represented in the training set [19]. This research directly confronts this gap by providing a rigorous and empirical comparison of three contemporary architectures, specifically versions 10, 11, and 12 [20]. We move beyond a superficial comparison of precision scores to conduct a deep analysis of their training dynamics and convergence properties. The objective is to identify the architecture that presents the optimal equilibrium between high detection accuracy and the robust generalization required for reliable deployment.

3. **Methodology.**

3.1. **Dataset and Preprocessing.** The dataset used in this study is the publicly available PCB Fault Detection dataset [5], a collection of 1,399 high-resolution images of power-supply printed circuit boards captured under varying viewpoints and illumination conditions. The annotated corpus used in our experiments contains 9,354 labeled component instances across nine categories and is partitioned into 1,099 training images, 200 validation images, and 100 test images. Table 1 reports the training-split class distribution so that the dataset balance can be assessed directly.

TABLE 1. Training-split class distribution of the annotated PCB component instances used in the experiments.

Class	Instances	Proportion (%)
Cap1	1098	11.74
Cap2	1067	11.41
Cap3	1003	10.72
Cap4	872	9.32
MOSFET	1098	11.74
Mov	1094	11.70
Resistor1	0	0.00
Resistor2	960	10.26
Transformer	2162	23.11
Total	9354	100.00

Although most categories are reasonably represented, the training split contains no annotated Resistor1 instances, which limits class-specific evaluation for that category and should be considered when interpreting the aggregate metrics.

Prior to being fed into the neural network, all input images were standardized to a uniform resolution of 640 by 640 pixels to comply with the input dimension requirements of the architecture. Furthermore, to enhance data diversity and improve the generalization capability of the model, a series of extensive data augmentation techniques were applied during the training phase. These techniques included geometric transformations, such as random flips and rotations, as well as photometric distortions, including mosaic augmentation, Hue Saturation Value color space adjustments, and shifts in brightness or contrast.

To provide a clear insight into the structural composition and annotation quality of the dataset, a visual exemplar is presented below. This illustration displays the diversity regarding component type, physical scale, and spatial density that the models are required to learn. As demonstrated in the visualization, objects ranging from large Integrated Circuits to smaller entities such as Capacitors and Transistors are enclosed within precise bounding boxes. This meticulous annotation process forms the ground truth, which is essential for training the models to perform robust localization and classification tasks effectively.

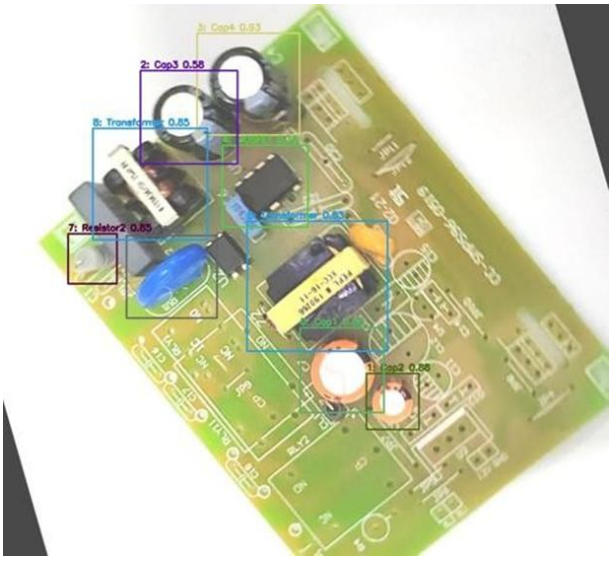

FIGURE 1. Annotated training sample displaying ground truth bounding boxes and component labels.

3.2. **Model Architectures.** The architectural configurations of the object detection models evaluated in this study, specifically versions 10, 11, and 12, adhere to a unified meta architecture. This fundamental structure comprises three primary components: a Backbone for feature extraction, a Neck for feature aggregation, and a Head for the final prediction of classes and bounding boxes. While sharing this

macroscopic skeleton, each iteration introduces distinct computational modules designed to optimize the equilibrium between inference speed and detection accuracy.

YOLOv10 distinguishes itself primarily through the introduction of an NMS free training strategy. This design utilizes consistent dual assignments to eliminate the need for Non Maximum Suppression during inference, thereby reducing latency. Architecturally, it employs the C2fCIB module, which integrates Compact Inverted Block structures to enhance feature extraction with minimal computational overhead. Unlike the attention mechanisms seen in later versions, YOLOv10 relies heavily on standard convolutional efficiency [20].

YOLOv11 represents a significant refinement in feature integration and stability. It replaces the standard C2f blocks found in previous iterations with the C3k2 module. This updated block allows for customizable kernel sizes, offering greater flexibility in receptive field modeling. Furthermore, YOLOv11 incorporates the C2PSA block within its architecture. This component utilizes spatial attention mechanisms to prioritize informative regions of the image, which contributes directly to the superior classification stability and generalization capabilities observed in our results [21].

YOLOv12, as depicted in Figure 2, introduces a paradigm shift by integrating attention driven mechanisms directly into the backbone. It features the A2C2f block (Area Attention) and R ELAN (Residual Efficient Layer Aggregation Network) to capture long range dependencies more effectively than pure convolutional approaches. While this architecture theoretically offers the highest potential for feature representation, our empirical data suggests that the complexity may induce rapid convergence at the cost of the stability robustness found in version 11 [22]. Figure 2 illustrates the detailed schematic of the YOLOv12 architecture, which represents the most recent evolution in this series. We utilize this visual representation as a structural baseline to delineate the comparative characteristics of the three evaluated models.

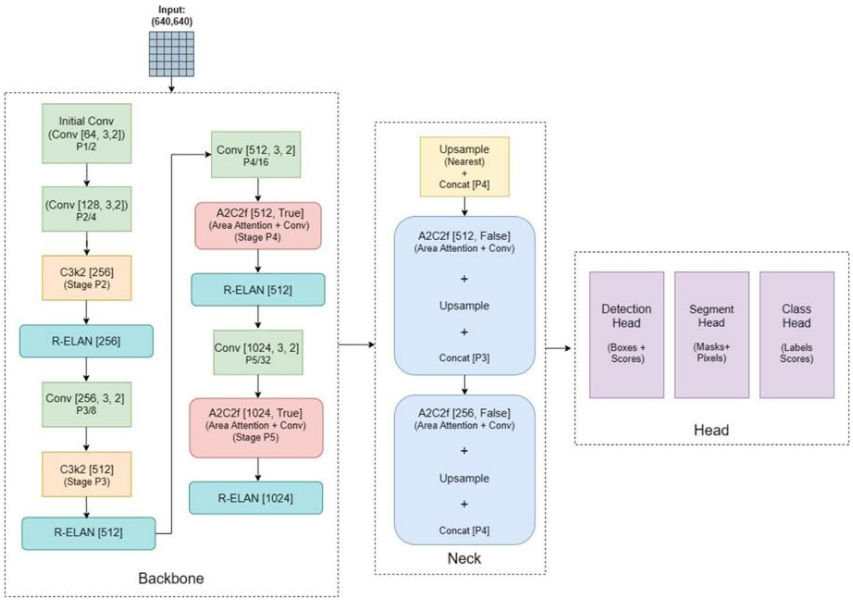

FIGURE 2. The architectural schematic of YOLOv12.

3.3. **Experimental Setup and Evaluation Criteria.** To ensure the reproducibility of our results and to facilitate a rigorous comparative analysis, all experiments were conducted within a strictly controlled environment. This section delineates the hardware specifications, training configurations, and the mathematical framework utilized for performance assessment.

All model training and inference processes were executed on a computational workstation equipped with an NVIDIA GeForce RTX 4090 Graphics Processing Unit. The software environment was implemented using the PyTorch deep learning framework alongside the Ultralytics library, supported by CUDA 12.2 for hardware acceleration[1]. To maintain consistency across all architectural evaluations, a unified set of hyperparameters was enforced for every experimental run. The input image dimensions were fixed at 640 $\times$ 640 pixels. A batch size of 16 was selected, and the training duration was set to 150 epochs. Optimization was performed using Stochastic Gradient Descent. The learning rate schedule employed a

Cosine Annealing strategy to adjust the learning rate (η) dynamically, commencing with an initial value (η_0) of 0.01.

The primary metric for quantifying detection performance is the mean Average Precision (mAP) and Recall (R.) values, which are calculated based on the number of True Positives (TP), False Positives (FP), and False Negatives (FN). The quantitative evaluation of the detection performance is governed by four fundamental metrics. Precision measures the accuracy of positive predictions, while Recall assesses the ability of the model to identify all relevant instances. Subsequently, the Average Precision for a specific class is determined by integrating the area under the Precision Recall curve, and the final mean Average Precision is calculated by averaging these values across all distinct classes. These metrics are mathematically defined in the following sequence: In the present study, the reported mAP is an unweighted macro average across classes, so each component category contributes equally regardless of its instance frequency. This means that performance on less frequent categories can influence the summary score as much as performance on dominant categories.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \int_0^1 P(R)\, dR \tag{3}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

To provide a comprehensive evaluation, we report two standard variations of this metric. The first is $mAP50$, which is calculated at a fixed Intersection over Union threshold of 0.50. The second is the more rigorous $mAP50{:}95$, representing the average precision computed over multiple thresholds ranging from 0.50 to 0.95 in increments of 0.05, thereby offering a stricter assessment of localization rigor.

Beyond standard accuracy scores, this study places significant emphasis on model stability and generalization capabilities. To assess these factors, we continuously monitored the validation loss dynamics throughout the training process. A divergence in these metrics is interpreted as a primary indicator of overfitting. The total loss function (L_{total}) generally comprises the weighted sum of individual loss components:

$$L_{total} = \lambda_{box} L_{box} + \lambda_{cls} L_{cls} + \lambda_{dfl} L_{dfl} \tag{5}$$

We specifically focus on two critical components within this equation. The Validation Box Loss, denoted as L_{box}, quantifies the error in bounding box regression typically utilizing Complete Intersection over Union, where a high or divergent value on the validation set indicates a failure to accurately localize objects on unseen data. Concurrently, we monitor the Validation Classification Loss, denoted as L_{cls}, which measures the correctness of component categorization using Binary Cross Entropy to reflect the ability of the model to assign the correct class label reliably.

4. **Experiment and Results.**

4.1. **Quantitative Performance Analysis.**

To establish a baseline for peak performance, we identified the optimal epoch for each model based on the highest achieved mean Average Precision score on the validation set. Table 2 summarizes the key performance metrics recorded at these respective peak epochs.

Table 2. Peak Performance Comparison of YOLO Models

Metric	YOLO v10	YOLO v11	YOLO v12
mAP50-95	0.7309	0.7295	0.7285
mAP50	0.8683	0.8715	0.8673
Box loss	1.3597	0.6754	0.6709
Cls loss	0.5570	0.3216	0.3246

The analysis of the data presented in Table 2 reveals a complex narrative regarding model efficacy. Regarding detection accuracy, version 10 achieves the highest strict precision score of 0.7309. However,

this represents a negligible margin of only 0.14 percent over version 11. Conversely, version 11 achieves the highest standard precision score of 0.8715. These figures suggest that all three models possess highly competitive detection capabilities when evaluated solely on precision metrics. Nevertheless, a critical divergence appears when examining the validation loss metrics. At its peak epoch, version 10 exhibits a validation box loss of 1.3597 and a classification loss of 0.5570. These values are substantially higher, effectively double, than those observed in version 11 and version 12. This discrepancy signals a fundamental issue with generalization. Although version 10 registers a high precision score, the elevated box loss indicates that it struggles to localize objects accurately on unseen data. This behavior implies that the model may have memorized spatial features from the training set rather than learning robust generalization patterns. In contrast, version 11 demonstrates superior convergence stability with a significantly lower classification loss of 0.3216, indicating a more reliable equilibrium between accuracy and structural learning.

4.2. **Analysis of Training Dynamics and Stability.** The static metrics presented in Table 2 provide only a partial perspective of model performance. To fully comprehend the behavioral characteristics and reliability of the architectures, a temporal analysis of the training progression is mandatory. We examined the evolution of accuracy and loss metrics across all 150 epochs to identify convergence properties and potential generalization gaps. Accuracy Progression As illustrated in Figure 3 and Figure 4, all three architectures exhibit rapid convergence during the initial training phases. Version 12, distinguished by its complex attention mechanisms, reaches its performance apex relatively early, approximately around epoch 48. However, following this peak, it displays a slight but discernible trajectory of performance decay. This phenomenon serves as an early quantitative indicator of potential overfitting, suggesting that the model begins to memorize noise within the training data rather than learning robust features. Conversely, version 10 and version 11 demonstrate more sustained convergence behavior. Version 11 achieves a robust plateau, maintaining consistent peak performance without significant fluctuation, thereby suggesting a highly stable learning state. This post-peak oscillation suggests that the more complex YOLOv12 design may benefit from a smaller initial learning rate or stronger regularization, but such hyperparameter retuning is left for future work because the present comparison intentionally keeps the training recipe fixed across all three models.

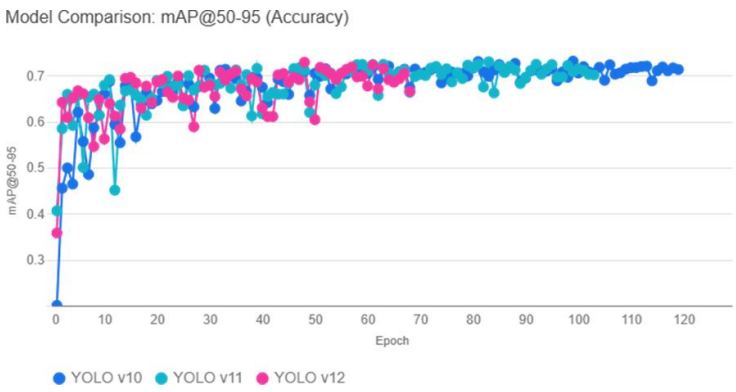

FIGURE 3. Temporal progression of mean Average Precision at strict threshold 0.50 to 0.95.

Stability and Generalization via Loss Metrics The analysis of validation loss curves constitutes the most critical evidence regarding model health. The distinction between the architectures becomes apparent when examining the Classification Loss depicted in Figure 5.

Both version 11 and version 12 perform exceptionally well, rapidly minimizing this error to converge at a low and stable plateau of approximately 0.32. This trajectory indicates that these architectures possess superior capability in correctly categorizing components within the validation set. In contrast, version 10 maintains a significantly elevated loss value of approximately 0.55 throughout the training process, implying a relative deficiency in classification generalization on unseen data.

However, the most significant anomaly is observed in the Bounding Box Loss shown in Figure 6. This metric reveals a catastrophic failure in the generalization of version 10. While version 11 and version 12 display ideal convergence characteristics with their regression error stabilizing at a low value of 0.67, version 10 exhibits severe instability. Its box loss remains disproportionately high, hovering between

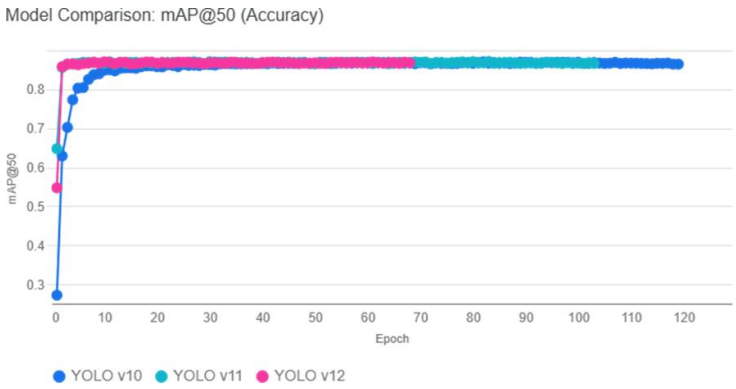

FIGURE 4. Temporal progression of mean Average Precision at standard threshold 0.50.

1.35 and 1.50 for the entire duration of the training run. This divergence serves as substantial empirical evidence that version 10 is suffering from severe overfitting regarding the bounding box regression task. Although it successfully classifies objects to achieve a high precision score, it fails to localize them with the necessary rigor on validation data, rendering it unreliable for practical deployment.

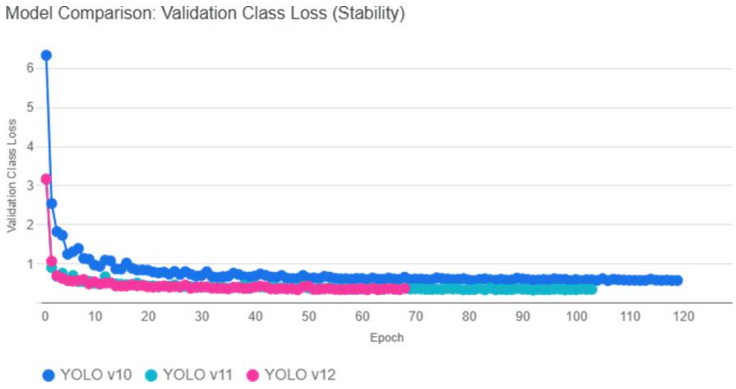

FIGURE 5. Comparative analysis of validation classification loss dynamics illustrating model convergence.

FIGURE 6. Comparative analysis of validation box loss dynamics revealing the generalization gap in version 10.

4.3. **Qualitative Assessment.** To complement the quantitative metrics presented in the preceding sections, a qualitative examination of the inference capabilities of version 11 was conducted using the unseen test dataset. This visual analysis serves to validate the practical applicability of the model

beyond statistical aggregates. Figure 7 presents a curated selection of prediction outcomes that visually affirm the robust performance of the architecture. The empirical evidence from these visualizations confirms that the model possesses a sophisticated understanding of component topology. It successfully identifies and localizes a diverse array of classes, ranging from large Integrated Circuits to smaller passive elements such as Resistors and Capacitors. A critical observation is the resilience of the model against environmental variability. As depicted in the samples, the detection logic remains consistent despite significant challenges, including high component density where objects are situated in close proximity, as well as variations in camera perspective and ambient lighting intensity.

Furthermore, the precision of the bounding boxes is particularly noteworthy. The predicted regions tightly encapsulate the target objects with minimal background noise, and the associated confidence scores consistently exceed the 0.80 threshold. This high degree of certainty serves as qualitative proof that the model has transcended mere memorization and has achieved a state of true generalization, enabling it to function reliably within dynamic industrial environments.

FIGURE 7. Qualitative detection results of the YOLOv11 model on the test dataset.

To address the reviewer's request for a direct side-by-side comparison, Figure 8 contrasts YOLOv10 and YOLOv11 on the same test images. The examples do not show a catastrophic miss-versus-hit contrast; instead, they show that YOLOv11 yields slightly tighter and more consistent localization, with higher average IoU in both examples (0.920 versus 0.885, and 0.915 versus 0.888). This visual tendency is consistent with the lower validation box loss of YOLOv11 and supports the interpretation that its localization behavior is more stable.

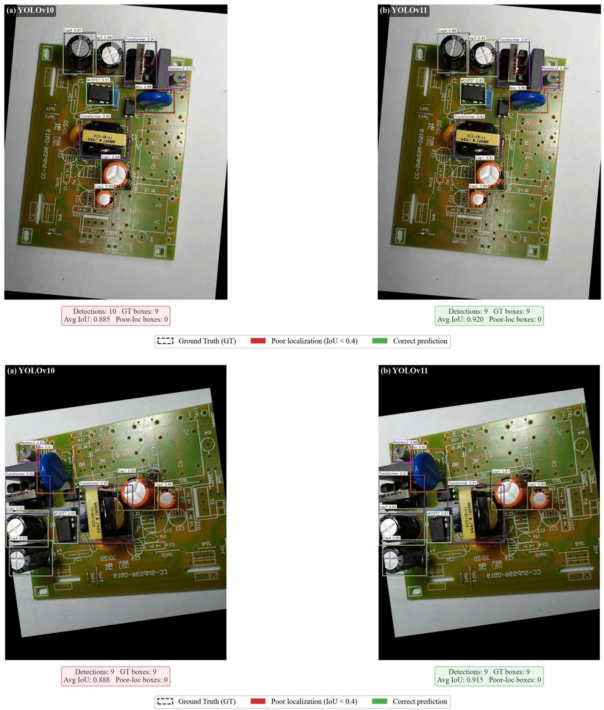

FIGURE 8. Side-by-side comparative localization examples for YOLOv10 and YOLOv11 on the same test images.

5. **Conclusion.** This study systematically compared three contemporary YOLO architectures, specifically versions 10, 11, and 12, to identify the optimal model for PCB component detection. The investigation focused heavily on dissecting the critical relationship between peak detection accuracy and the stability required for reliable industrial deployment. The empirical analysis revealed that relying exclusively on mean Average Precision is a flawed evaluation strategy for this specific domain. While version 10 achieved the highest accuracy score, it exhibited a catastrophic failure in generalization, characterized by a validation box loss that was approximately double that of the other models. This discrepancy indicates severe overfitting regarding the localization task. Conversely, version 11 proved to be the most robust architecture. It delivered highly competitive precision while maintaining superior training stability with the lowest classification loss, representing the ideal equilibrium for practical application.

This study has several limitations. First, the dataset is limited to a single PCB family and 1,399 images, so the conclusions should be interpreted as evidence for this inspection setting rather than as a universal ranking for all PCB products. Second, the absence of annotated Resistor1 instances in the training split prevents class-specific evaluation for that category. Third, the comparison uses one shared training recipe for all three architectures; this is useful for controlled benchmarking, but additional hyperparameter tuning may change the relative behavior of YOLOv12 in particular. Finally, the present analysis focuses on floating-point validation behavior and qualitative inspection, so embedded deployment issues such as quantization, latency on industrial hardware, and long-term maintenance under distribution shift remain for future work.

Looking forward, future research will address current limitations by expanding the dataset to cover a wider variety of component types and complex manufacturing conditions. Additionally, the subsequent phase of this work will focus on optimizing the identified version 11 architecture through quantization techniques to enable efficient processing on resource constrained embedded systems.

Acknowledgment. This research is funded by Nha Trang University for science and technology under grant number TR2025-13-44.

REFERENCES

[1] B. Montoya Magaña, Ó. Hernández-Uribe, L. A. Cárdenas-Robledo, and J. A. Cantoral-Ceballos, "Deep Learning Algorithms for Defect Detection on Electronic Assemblies: A Systematic Literature Review," *Machine Learning and Knowledge Extraction*, vol. 8, no. 1, p. 5, Dec. 2025, doi: 10.3390/make8010005.

[2] A. A. R. M. A. Ebayyeh and A. Mousavi, "A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry," 2020, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ACCESS.2020.3029127.

[3] O. Y. Chiun and N. I. R. Ruhaiyem, "Object Detection Based Automated Optical Inspection of Printed Circuit Board Assembly Using Deep Learning," 2023, pp. 246–258, doi: 10.1007/978-981-99-0405-1_18.

[4] X. Tong, Z. Yu, X. Tian, H. Ge, and X. Wang, "Improving accuracy of automatic optical inspection with machine learning," *Frontiers of Computer Science*, vol. 16, no. 1, p. 161310, Feb. 2022, doi: 10.1007/s11704-021-0244-9.

[5] W. Huang and P. Wei, "A PCB Dataset for Defects Detection and Classification," Jan. 2019.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Dec. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[7] H. Zhang, Y. Li, D. M. Sharid Kayes, Z. Song, and Y. Wang, "Research on PCB defect detection algorithm based on LPCB-YOLO," *Frontiers in Physics*, vol. 12, Jan. 2025, doi: 10.3389/fphy.2024.1472584.

[8] Z. Ni and Y. Kim, "Research on Printed Circuit Board (PCB) Defect Detection Algorithm Based on Convolutional Neural Networks (CNN)," *Applied Sciences*, vol. 15, no. 24, p. 13115, Dec. 2025, doi: 10.3390/app152413115.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[10] M. Maślanka, D. Jancarczyk, and J. Rysinski, "Integration of Machine Vision and PLC-Based Control for Scalable Quality Inspection in Industry 4.0," *Sensors*, vol. 25, no. 20, p. 6383, Oct. 2025, doi: 10.3390/s25206383.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," available online: https://papers.nips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[12] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020, doi: 10.1016/j.neucom.2020.01.085.

[13] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[14] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[15] X. Yin, Z. Zhao, and L. Weng, "MAS-YOLO: A Lightweight Detection Algorithm for PCB Defect Detection Based on Improved YOLOv12," *Applied Sciences*, vol. 15, no. 11, p. 6238, Jun. 2025, doi: 10.3390/app15116238.

[16] S. Zhuo, J. Shi, X. Zhou, and J. Kan, "YOLOv7-TID: A Lightweight Network for PCB Intelligent Detection," *IEEE Access*, vol. 12, pp. 109957–109966, 2024, doi: 10.1109/ACCESS.2024.3439567.

[17] A. Bhattacharya and S. G. Cloutier, "End-to-end deep learning framework for printed circuit board manufacturing defect classification," *Scientific Reports*, vol. 12, no. 1, p. 12559, Jul. 2022, doi: 10.1038/s41598-022-16302-3.

[18] Z. Wei, F. Yang, K. Zhong, and L. Yao, "PCB-YOLO: Enhancing PCB surface defect detection with coordinate attention and multi-scale feature fusion," *PLOS ONE*, vol. 20, Jun. 2025, doi: 10.1371/journal.pone.0323684.

[19] X. Wang, S. S. Maidin, and M. Batumalay, "PCB defect detection based on pseudo-inverse transformation and YOLOv5," *PLOS ONE*, vol. 19, Dec. 2024, doi: 10.1371/journal.pone.0315424.

[20] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection," in *Advances in Neural Information Processing Systems 37*, San Diego, California, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, pp. 107984–108011, doi: 10.52202/079017-3429.

[21] H. Yang, J. Dong, C. Wang, Z. Lian, and H. Chang, "PCES-YOLO: High-Precision PCB Detection via Pre-Convolution Receptive Field Enhancement and Geometry-Perception Feature Fusion," *Applied Sciences*, vol. 15, no. 13, p. 7588, Jul. 2025, doi: 10.3390/app15137588.

[22] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," in *NeurIPS 2025 Conference on Neural Information Processing Systems, Poster Session*, 2025. [Online]. Available: https://neurips.cc/virtual/2025/loc/san-diego/poster/116765.

[23] P. H. Dang, "VPZL: Visual prompt-guided zero-shot learning for insulator defect detection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 16, no. 3, pp. 843–853, Sep. 2025.

[24] T. W. Sung, J. Li, C.-Y. Lee, and Q. Fang, "Improvement of YOLOv8 Object Detection Based on Lightweight Neck Model for Complex Images," *Image Analysis and Stereology*, vol. 44, no. 1, pp. 69–86, Mar. 2025, doi: 10.5566/ias.3514.