Celal Bayar University Journal of Science

Evaluation of Lightweight Object Detection and Instance Segmentation Techniques for Industrial Printed Circuit Board Solder Quality Assessment

Tuncay Soylu[1]*, Emel Soylu[2]

[1]Samsun University, Department of Electrical and Electronics Engineering, 55700 Samsun, Türkiye.
[2]Samsun University, Department of Software Engineering, 55700 Samsun, Türkiye.
* tuncay.soylu@samsun.edu.tr
* Orcid No: 0000-0002-5541-2219

Received: April 2, 2025
Accepted: October 7, 2025
DOI: 10.18466/cbayarfbe.1669378

Abstract

The rapid progress of deep learning has transformed the detection of soldering defects in printed circuit boards (PCBs), outperforming traditional manual inspections and rule-based machine vision systems.This study evaluates the performance of state-of-the-art YOLO models—specifically YOLOv11 and YOLOv12—for both object detection and instance segmentation of solder defects in aerospace PCBs using the open-source SolDef_AI dataset. We compare multiple variants (nano, small, and medium) to assess their accuracy, efficiency, and suitability for industrial quality control. Our experiments reveal that YOLO-v11s-seg achieves the highest mean average precision (mAP50-95: 0.853 for detection, 0.822 for segmentation), demonstrating superior defect localization capabilities, particularly for challenging classes such as "poor_solder" and "spike." While YOLOv12 models exhibit competitive detection performance, they show slightly lower segmentation accuracy, indicating potential areas for architectural refinement. Smaller models (YOLO-v11n, YOLO-v12n) offer a favorable balance between speed and precision, making them viable for real-time applications. The findings highlight the effectiveness of YOLO-based deep learning in automating solder defect inspection, with implications for improving manufacturing quality assurance in electronics production.

Keywords: Image Processing, Object Detection, Quality Control, Soldering Defect Detection

1. Introduction

The detection of soldering defects in printed circuit boards (PCBs) has undergone a significant transformation with the advent of deep learning techniques, moving beyond traditional manual inspections and rule-based machine vision systems [1]–[3] . As electronic components continue to shrink in size while increasing in complexity, the need for more sophisticated inspection methods has become paramount. Current approaches leverage convolutional neural networks (CNNs) as their foundation, with modified architectures like ResNet and EfficientNet demonstrating remarkable success in classifying various soldering defects. These models have evolved to incorporate attention mechanisms and hybrid designs that combine CNN features with transformer elements, significantly improving the detection of microscopic defects that were previously challenging to identify [4]–[8].

Recent advancements in object detection frameworks have brought substantial improvements to real-time inspection capabilities. The YOLO series, particularly versions 5 through 8, has emerged as a preferred solution for industrial applications due to its balance between speed and accuracy, consistently achieving mean average precision scores above 95% while maintaining faster processing times compared to older architectures like Faster R-CNN [9]–[16]. Simultaneously, transformer-based detectors such as RT-DETR have shown superior performance in handling complex defect patterns, though their higher computational requirements currently limit their widespread industrial adoption [17]–[20]. For more detailed defect analysis, segmentation techniques using Mask R-CNN and its variants enabling precise identification of solder bridges and other fine anomalies [21].

The effectiveness of these deep learning models heavily depends on the quality and diversity of training data, leading to innovative solutions addressing data scarcity challenges. Synthetic data generation through GANs has proven particularly valuable, enhancing model generalization by 18-22% in recent implementations. Multimodal approaches that combine visual data with thermal imaging and 3D profilometry have further improved detection robustness, while active learning systems have dramatically reduced labeling costs by intelligently selecting the most informative samples for annotation. These data optimization strategies have been crucial in transitioning deep learning solutions from research environments to actual production lines.
Industrial implementation has seen the development of specialized edge AI solutions capable of real-time processing on compact hardware, with quantized YOLO models achieving 30 frames per second on Raspberry Pi platforms. The integration of these detection systems with robotic correction mechanisms represents the next frontier, enabling not just identification but immediate rectification of soldering defects. Looking ahead, research focuses on overcoming remaining challenges through few-shot learning for rare defect types, developing explainable AI systems for better root cause analysis, and optimizing computational efficiency through neural architecture search. The remarkable progress in this field is evidenced by current systems achieving 99.2% defect detection rates in production environments, establishing deep learning as an indispensable tool for quality assurance in modern electronics manufacturing. This technological evolution supports the industry's shift toward zero-defect manufacturing paradigms while adapting to the increasing miniaturization and complexity of electronic components.

A comprehensive examination of methodologies reveals significant improvements in detection accuracy due to the advent of deep learning frameworks. For instance, Zhang and Shen utilized an improved Faster-RCNN algorithm specifically for solder joint defect detection in connectors, achieving robust results compared to traditional methods [22]. Similarly, Nayak and Parameshachari reported a classification accuracy of 99.99% for PCB defects by employing a stacked autoencoder integrated with a Bi-LSTM network, showcasing substantial advancements over prior models that struggled to attain such precision [23]. Furthermore, Liao et al. highlighted a YOLOv4-MN3 model that effectively identifies PCB surface defects, establishing a new benchmark in defect detection methodologies [24]. Huang and Wei's efforts in curating a dataset specifically for PCB defects have provided a beneficial resource for validating numerous models [25].

The collective outputs of recent research in PCB solder defect detection reveal that leveraging deep learning techniques along with substantial datasets and traditional image processing methods significantly enhances the identification accuracy of such defects. These advancements underscore the evolving landscape of defect detection in manufacturing quality assurance.
While previous studies have extensively utilized earlier YOLO versions (e.g., YOLOv3, YOLOv5, YOLOv8) for PCB defect detection, the application of YOLOv12 remains underexplored in the literature. YOLOv12 introduces several architectural improvements, including enhanced feature extraction, better handling of small objects, and optimized computational efficiency, making it a promising candidate for high-precision PCB defect detection. However, to the best of our knowledge, no comprehensive study has yet evaluated the effectiveness of YOLOv12 specifically for solder defect detection. This study aims to fill this gap by implementing YOLOv12 on a PCB solder defect dataset and comparing its performance with previous YOLO versions. By doing so, we provide insights into the suitability of the latest YOLO architecture for real-world quality control applications in electronics manufacturing.

2. Materials and Method

2.1 Dataset

The SolDef_AI dataset [21], used in this study, is an open-source collection specifically designed for defect detection in the soldering processes of electronic components on PCBs for aerospace applications. It contains 1,150 high-quality images of surface-mount technology (SMT) components, including capacitors and resistors of various sizes, captured from three different viewpoints (top, 45-degree, and axonometric views) to provide a comprehensive visualization of soldered joints and component positioning. The dataset focuses on two main defect categories: misalignment of SMT components and soldering joint defects, such as excessive or insufficient solder material and the presence of spikes. Each image is annotated with bounding polygons and class labels, making it suitable for training and evaluating deep learning models, particularly for tasks like object detection and instance segmentation using algorithms such as Mask R-CNN.

The SolDef_AI dataset addresses the lack of publicly available datasets tailored to aerospace PCB manufacturing, which must adhere to stringent standards like ECSS-Q-ST-70-38C. By providing a diverse set of images with varying lighting conditions and perspectives, the dataset enables the development of robust defect detection models that can generalize across different SMT components and soldering defects. The dataset is openly accessible on Kaggle, can be accessed at: Kaggle SolDef_AI Dataset [26].

This study utilized 428 labeled images from the dataset. Data augmentation, including flipping and 90-degree rotations, was employed, resulting in an expanded

dataset of 1026 images. In the context of soldering quality assessment, several terms are used to describe various conditions. A 'Spike' refers to an undesirable sharp point or protrusion in the solder, indicating either improper spreading or excessive solder usage. 'Poor solder' signifies low-quality or defective soldering, encompassing issues such as weak connections, cold solders where the solder hasn't melted correctly, insufficient solder, or inadequate adhesion to the surface. 'No good' denotes a completely faulty or unacceptable soldering state, rendering the connection non-functional or unreliable. Conversely, 'Good' describes a high-quality, robust solder joint that meets the required standards. Lastly, 'Exc solder,' short for 'Excess Solder,' indicates an overabundance of solder on the circuit, beyond what is necessary. The dataset was subsequently partitioned into training (70%), validation (15%), and test (15%) subsets.vAs shown in Table 1, the training subset consists of 718 images, while the validation and test subsets contain 153 and 155 images, respectively. This partitioning ensures a robust evaluation of the model's performance on unseen data. Furthermore, the table delineates the distribution of labels within each subset, specifically 'exc solder', 'good', 'no good', 'poor solder', and 'spike'. Notably, the 'exc solder' and 'good' categories represent the most prevalent labels, with 452 and 442 instances, respectively, in the overall dataset. Distribution of data and labels are given in Table 1. The distribution of labels across the training, validation, and test sets is maintained proportionally to the overall dataset distribution, ensuring that each subset reflects the original data's characteristics. This balanced representation across the subsets is critical for training a model that generalizes well to unseen data and avoids bias towards any specific label.

Table 1. Distribution of data and labels

Data	Images	Excessive solder	Good	No good	Poor solder	Spike
Train	718	311	314	186	95	216
Valid	153	65	62	44	18	53
Test	155	76	66	44	22	41
Total	1026	452	442	274	135	310

Fig. 1 presents examples of labeling for each category within the dataset.

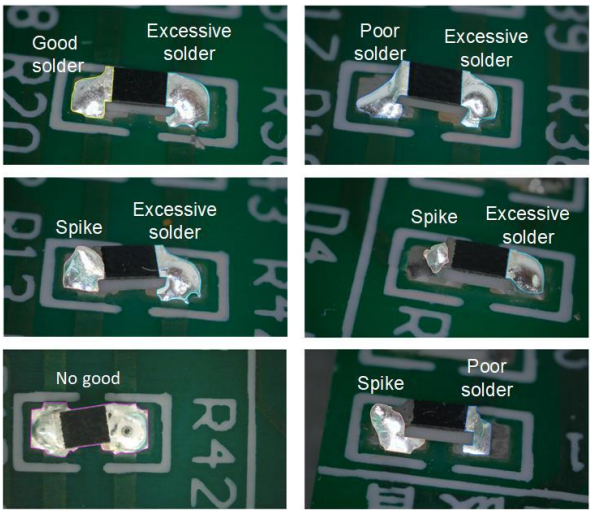

Figure 1. Annotation examples from dataset.

2.2 Method

YOLO is a popular deep learning model for real-time object detection. Unlike traditional object detection methods that use a two-stage approach (e.g., region proposal followed by classification), YOLO employs a single neural network to predict bounding boxes and class probabilities directly from full images in one pass. This makes it extremely fast and efficient, suitable for applications like autonomous driving, surveillance, and robotics. Since its introduction with YOLOv1 in 2015 by Joseph Redmon et al., the YOLO architecture has undergone significant improvements with each subsequent version. The official versions of YOLO continue up to YOLOv12, developed by Ultralytics.

The YOLO architecture is built on three key components. The backbone acts as the primary feature extractor, using convolutional neural networks to convert raw images into multi-scale feature maps. The neck serves as an intermediate stage, refining and aggregating features across different scales. Finally, the head generates the final predictions for object localization and classification based on the processed feature maps [27]. General abstract of YOLO architectures is given in Fig. 2.

YOLOv11 is an advanced real-time object detection model that extends beyond traditional tasks to support posture estimation and instance segmentation. It enhances adaptability, accuracy, and efficiency, making it suitable for various industries. With improved performance and versatility, YOLOv11 represents a significant step forward in computer vision applications [27].

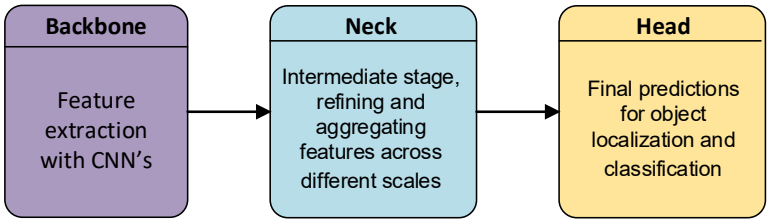

Figure 2. General abstract of YOLO architectures

In contrast to prior YOLO models that relied on traditional CNN paradigms, YOLO12 [28] introduces an attention-focused architecture. This advancement emphasizes the utilization of attention mechanisms, ensuring real-time inference for a broad spectrum of applications. By incorporating innovative methodological refinements within attention modules and the overall network framework, YOLO12 achieves exceptional object detection accuracy without compromising computational efficiency.

Object detection and instance segmentation represent distinct yet related tasks within the realm of computer vision. Object detection primarily focuses on the localization and classification of objects within an image. This is typically achieved by generating bounding boxes around detected objects, accompanied by corresponding class labels. In contrast, instance segmentation extends beyond mere detection by providing a pixel-level understanding of each object's shape and extent. It not only identifies the presence and category of objects but also delineates their precise boundaries through pixel-wise masks. This fine-grained segmentation allows for a more detailed and nuanced interpretation of the image content, enabling applications that require precise object delineation. In this study, we employed both object detection and instance segmentation techniques.For the purpose of identifying solder defects in PCB assemblies, we implemented the YOLOv11 and YOLOv12 object detection frameworks, as shown in Fig. 3.

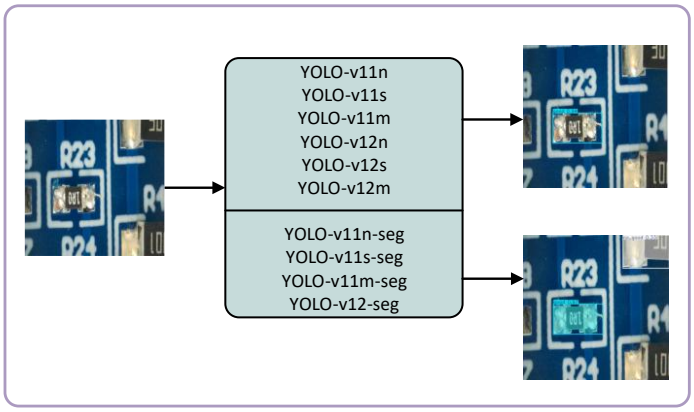

Figure 3. Performance evaluation of YOLOv11 and YOLOv12 for PCB solder defect detection

For the computational aspects of this study, we utilized Google Colaboratory, a cloud-based Jupyter notebook environment that provides access to computational resources free of charge. The runtime environment was configured to use Python 3. To accelerate model training, we allocated a T4 GPU as the hardware accelerator. Additionally, the High-RAM option was enabled to provide sufficient memory for the large datasets and complex neural networks used in our experiments. This configuration allowed us to efficiently conduct our deep learning experiments and train our models within a reasonable timeframe.

2.2.1 Hyperparameters

Epoch is number of training epochs (iterations over the entire dataset). Early stopping was employed, leading to the completion of all training runs and suggesting that the chosen epoch count was appropriate. Image size is input image size (pixels). Images are resized to this dimension during training. Batch size is the number of images processed in each training iteration. The training process for the YOLO-v11n, YOLO-v11s, YOLO-v11m, YOLO-v12n, YOLO-v12s, YOLO-v12m, YOLO11n-seg, YOLO11s-seg, YOLO11m-seg, and YOLO12-seg models was conducted over 500 epochs. The input image size was set to 640×640 pixels for all models, while the batch size was fixed at 16 during training.We chose the same hyperparameters for training these models because we wanted to maintain consistency in the training process. This allows us to accurately compare the performance of different model architectures without the confounding variable of varying hyperparameters. By keeping the epochs, image

size, and batch size constant, we ensure a fair evaluation of each model's effectiveness.

2.2.2 Performance Metrics

When evaluating a YOLO model for object detection and segmentation, several key performance metrics are used to assess its accuracy and effectiveness.
Bounding Box Metrics: These metrics evaluate how well the model detects objects and localizes them with bounding boxes [29].

Precision (P): Measures how many of the predicted bounding boxes are actually correct. Equation of P is given in Eq. 2.1. P is Precision, TP is true positives, correctly detected objects, FP is false positives, incorrectly detected objects in this formula. Higher precision means fewer false detections [30].

$$P = \frac{TP}{TP+FP} \quad \textbf{(2.1)}$$

Recall (R) Measures how many actual objects were detected by the model. Equation of R is given in Eq. 2.2. Higher recall means fewer missed detections.

$$R = \frac{TP}{TP+FN} \quad \textbf{(2.2)}$$

- Mean Average Precision (mAP) is the most comprehensive metric for evaluating object detection performance. mAP@50 measures the average precision when the Intersection over Union (IoU) threshold is 50%. mAP@50-95 measures the average precision across IoU thresholds from 50% to 95%, in 5% increments. A stricter evaluation than mAP@50, giving insight into how well the model localizes objects with high accuracy. Higher mAP means better detection and localization.
- Segmentation Metrics (For YOLO-Seg Models): These metrics evaluate the mask predictions in segmentation tasks.
- Mask Precision (Mask P) & Mask Recall (Mask R): Similar to box precision and recall but applied to segmentation masks.
- Mask mAP@50-95: Evaluates the accuracy of segmentation across different IoU thresholds [29].

3. Results and Discussion

This section presents a comprehensive evaluation of YOLO-v11 and YOLO-v12 models for object detection and instance segmentation tasks. The performance metrics, including precision (P), recall (R), mean average precision at IoU=50% (mAP50), and mAP50-95, are analyzed across different model variants. Additionally, the computational efficiency of each model is assessed based on model size and overall accuracy.

As shown in Table 2, the YOLO-v11 series demonstrates strong detection capabilities. The YOLO-v11s model achieves the highest mAP50-95 (0.837) among the three variants, with particularly impressive recall (R=1.0) and mAP50 (0.986) for the "no_good" class. The YOLO-v11m model exhibits the best precision (0.944) but suffers from slightly lower recall (0.875). Notably, YOLO-v11n provides a balanced performance with a compact model size (5.25 MB), making it suitable for resource-constrained applications.

Table 2. Performance comparisons of YOLO-v11 models

Methods	Class	P	R	mAP50	mAP50-95
YOLO-v11n	all	0.888	0.892	0.933	0.826
	exc_solder	0.885	0.945	0.961	0.863
	good	0.854	0.850	0.939	0.923
	no_good	0.954	0.948	0.984	0.959
	poor_solder	0.809	0.833	0.858	0.705
	spike	0.940	0.884	0.923	0.682
YOLO-v11s	all	0.888	0.914	0.942	**0.837**
	exc_solder	0.799	0.978	0.944	0.851
	good	0.901	0.881	0.944	0.918
	no_good	0.852	**1.000**	**0.986**	0.965
	poor_solder	1.000	0.833	0.939	0.778
	spike	0.886	0.880	0.898	0.674
YOLO-v11m	all	0.944	0.875	0.955	0.855
	exc_solder	0.879	0.938	0.956	0.876
	good	0.890	0.887	0.955	0.929
	no_good	0.976	0.931	0.985	0.963
	poor_solder	1.000	0.791	0.956	0.795
	spike	0.978	0.826	0.922	0.711

According to Table 3, the YOLO-v12n model outperforms its counterparts with an mAP50-95 of 0.844, excelling in the "exc_solder" class (R=0.978, mAP50=0.980). However, YOLO-v12s and YOLO-v12m show inconsistent performance, particularly in the "poor_solder" and "spike" classes, where their mAP50-95 scores drop significantly.

Table 3. Performance comparisons of YOLO-v12 models

Methods	Class	P	R	mAP50	mAP50-95
YOLO-v12n	all	0.926	0.883	0.956	**0.844**
	exc_solder	0.901	**0.978**	**0.980**	0.894
	good	0.873	0.855	0.948	0.929
	no_good	0.931	0.916	0.974	0.952
	poor_solder	0.969	0.778	0.940	0.760
	spike	0.958	0.887	0.940	0.683
YOLO-v12s	all	0.884	0.894	0.943	0.803
	exc_solder	0.842	0.969	0.969	0.868
	good	0.843	0.871	0.929	0.886
	no_good	0.853	0.923	0.970	0.935
	poor_solder	0.942	0.833	0.939	0.687
	spike	0.939	0.873	0.906	0.640
YOLO-v12m	all	0.912	0.881	0.947	0.831
	exc_solder	0.906	0.954	0.974	0.879
	good	0.896	0.835	0.938	0.894
	no_good	0.909	0.932	0.978	0.945
	poor_solder	0.986	0.833	0.929	0.755
	spike	0.864	0.849	0.917	0.682

Table 4 highlights the segmentation performance of YOLO-v11 models. The YOLO-v11s-seg variant stands out with the highest mAP50-95 for both bounding box (0.853) and mask (0.822) predictions. It maintains perfect recall (1.0) for the "no_good" class, demonstrating robust segmentation accuracy. In contrast, YOLO-v11m-seg, despite its larger size (43 MB), underperforms in mask mAP50-95 (0.780), indicating inefficiency in handling segmentation tasks.

Table 4. Performance of detection and segmentation methods

Methods	Class	Bounding Box				Mask			
		P	R	mAP50	mAP50-95	P	R	mAP50	mAP50-95
YOLO-v11n-seg	**Class**	Box(P)	Box(R)	mAP50	mAP50-95	Mask(P)	Mask(R)	Mask mAP50	Mask mAP50-95
	all	0.884	0.907	0.938	0.838	0.885	0.908	0.937	0.807
	exc_solder	0.884	0.954	0.968	0.885	0.884	0.954	0.968	0.830
	good	0.853	0.934	0.955	0.934	0.838	0.918	0.944	0.888
	no_good	0.911	0.929	0.979	0.960	0.911	0.929	0.979	0.976
	poor_solder	0.832	0.833	0.855	0.690	0.832	0.833	0.853	0.626
	spike	0.942	0.887	0.932	0.723	0.962	0.906	0.944	0.718
YOLO-V11s-seg	**all**	0.871	0.925	0.954	**0.853**	0.868	0.922	0.949	**0.822**
	exc_solder	0.911	0.950	0.971	0.882	0.911	0.950	0.971	0.847
	good	0.826	0.919	0.943	0.919	0.812	0.903	0.930	0.885
	no_good	0.829	**1.000**	0.975	0.955	0.829	**1.000**	0.975	0.971
	poor_solder	0.881	0.833	0.928	0.782	0.881	0.833	0.917	0.669
	spike	0.910	0.925	0.953	0.726	0.909	0.925	0.952	0.736
YOLO-V11m-seg	**all**	0.891	0.887	0.936	0.826	0.891	0.887	0.935	0.780
	exc_solder	0.915	0.954	0.968	0.867	0.915	0.954	0.968	0.802
	good	0.816	0.887	0.922	0.882	0.816	0.887	0.917	0.866
	no_good	0.988	0.864	0.978	0.953	0.988	0.864	0.978	0.971
	poor_solder	0.835	0.889	0.914	0.774	0.835	0.889	0.914	0.606
	spike	0.899	0.839	0.899	0.651	0.899	0.839	0.899	0.656

Celal Bayar University Journal of Science
Volume 21, Issue 4, 2025, p 128-138
Doi: 10.18466/cbayarfbe.1669378

As presented in Table 5 the YOLO-v12-seg model shows moderate performance, with an overall mask mAP50-95 of 0.748. The model struggles with the "poor_solder" class, achieving only 0.490 in mask mAP50-95, suggesting limitations in segmenting complex or irregular objects.

Table 5. Performance of detection and segmentation methods

Methods	Class	Bounding Box				Mask			
		P	R	mAP50	mAP50-95	P	R	mAP50	mAP50-95
YOLO-v12-seg	**all**	0.849	0.891	0.925	0.779	0.837	0.887	0.903	**0.748**
	exc_solder	0.875	0.985	0.963	0.865	0.871	0.985	0.963	0.804
	good	0.787	0.893	0.906	0.860	0.774	0.884	0.893	0.834
	no_good	0.855	0.886	0.957	0.930	0.849	0.886	0.957	0.942
	poor_solder	0.821	0.768	0.852	0.588	0.764	0.721	0.741	**0.490**
	spike	0.908	0.926	0.945	0.653	0.927	0.960	0.963	0.672

Table 6 summarizes the trade-offs between model size and accuracy. The YOLO-v11s and YOLO-v11s-seg models achieve the highest overall accuracy (86%) while maintaining reasonable model sizes (18.3 MB and 19.6 MB, respectively). In contrast, larger models like YOLO-v11m-seg (43 MB) and YOLO-v12m (38.9 MB) do not provide proportional accuracy improvements, highlighting the diminishing returns of increasing model complexity.

Table 6. Overall accuracy of models

Model	Model Size (MB)	Overall Accuracy (%)
YOLO-v11n	5.25	83.00%
YOLO-v11s	18.3	**86.00%**
YOLO-v11m	38.6	82.00%
YOLO-v12n	5.3	84.00%
YOLO-v12s	18	76.00%
YOLO-v12m	38.9	82.00%
YOLO-v11n-seg	5.78	81.00%
YOLO-v11s-seg	19.6	**86.00%**
YOLO-v11m-seg	43	74.00%
YOLO-v12-seg	15.3	77.00%

Based on the experimental results, YOLO-v11s emerges as the optimal model for detection tasks, offering the best balance between accuracy and computational efficiency. For segmentation tasks, YOLO-v11s-seg is the preferred choice due to its high recall and mAP50-95 scores. The confusion matrices for YOLO-v11s and YOLO-v11s-seg are depicted in Fig. 4 and Fig. 5, respectively.

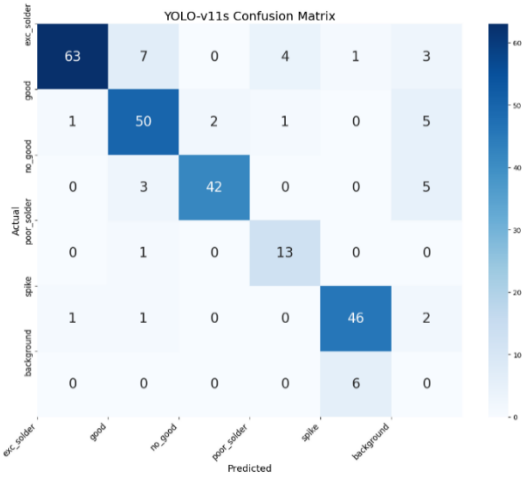

Figure 4. YOLO-v11s confusion matrix.

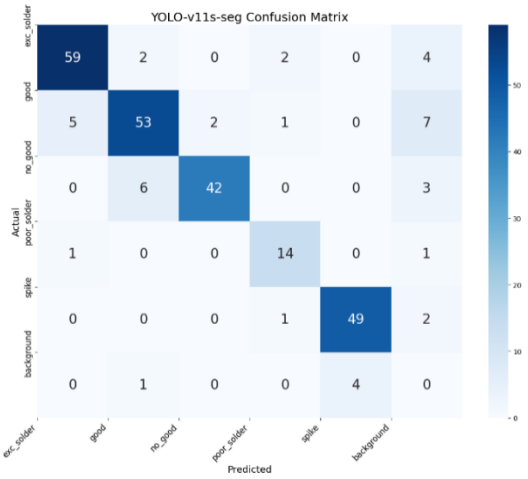

Figure 5. YOLO-v11s-seg confusion matrix.

Figure 4 presents the confusion matrix for YOLO-v11s. Rows correspond to the ground-truth classes, while columns represent the predicted classes. Diagonal cells indicate correct classifications, whereas off-diagonal cells denote misclassifications. The most frequent confusions are observed between poor_solder ↔ good

Celal Bayar University Journal of Science
Volume 21, Issue 4, 2025, p 128-138
Doi: 10.18466/cbayarfbe.1669378

and poor_solder ↔ excessive_solder. In Figure 5, the YOLO-v11s-seg matrix shows that the inclusion of mask information reduces false positives in the good/no_good classes; however, misclassifications persist for spike and poor_solder due to the small target size and subtle visual characteristics.

The results indicate that smaller models (e.g., YOLO-v11s) can outperform larger variants when optimized for specific tasks. The segmentation models generally exhibit lower mAP50-95 scores compared to their detection-only counterparts, emphasizing the challenges of precise mask prediction. Future work could explore hybrid architectures or post-processing techniques to improve segmentation accuracy, particularly for challenging classes like "poor_solder."

Figure 6 and Figure 7 illustrate qualitative results of the YOLO-v11s and YOLO-v11s-seg models on PCB test data. Both models demonstrate strong capability in detecting major defect types with high confidence, particularly for clearly distinguishable categories such as *excessive_solder* and *no_good*. While YOLO-v11s effectively localizes defects using bounding boxes, the segmentation-based YOLO-v11s-seg provides additional information on the spatial extent and shape of the defects, which enhances interpretability. At the same time, both models show certain limitations when dealing with underrepresented or fine-scale categories such as *spike* and *poor_solder*, where confidence levels may decrease or partial misclassifications can occur. Overall, the visual examples confirm the quantitative results, showing that segmentation improves boundary-level accuracy, while further refinements in data balance and feature learning are required for consistently robust performance across all defect classes.

Figure 6. Visual examples of YOLO-v11s model on PCB test data.

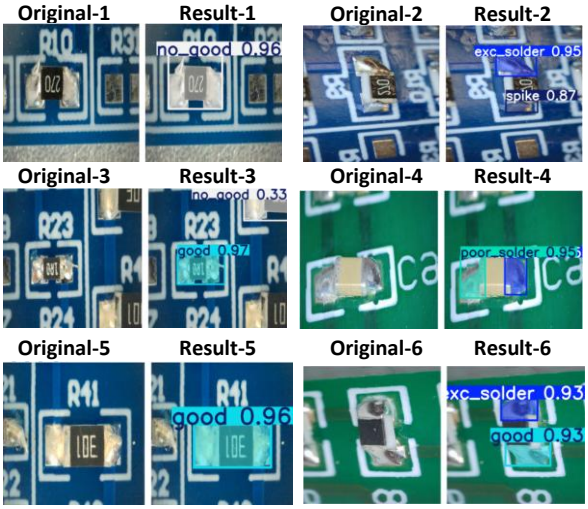

Figure 7. Visual examples of YOLO-v11s-seg model on PCB test data.

4. Discussion

To highlight the relative strengths and limitations of our proposed approach, we compared it with recent state-of-the-art methods, as summarized in Table 7**.** Recent studies reveal a steady progression from conventional deep learning frameworks toward more lightweight and specialized architectures. For instance, optimization-based CNNs[3] and improved Faster R-CNN models[22]represented early milestones in reliable solder defect detection, while YOLOv4-based approaches[24] demonstrated the feasibility of real-time inspection in production settings. Subsequent advances, such as stacked autoencoders with Bi-LSTM[31] and end-to-end CNN frameworks[4], achieved extremely high classification accuracies, underscoring the potential of hybrid and sequence-learning architectures. More recent contributions include SDD-Net[2], large-kernel convolutional networks[5], and attention-based fusion models[7], each of which improved robustness in detecting small or visually ambiguous defects. In parallel,[8] introduced synthetic sample generation through ControlNet and Swin Transformer, highlighting the increasing role of data augmentation in industrial inspection tasks. In addition, the most recent works, such as Xu et al.'s lightweight YOLO11n with attention modules [32], Li et al.'s SCF-YOLO designed for real-time PCB surface inspection [33], Zhu et al.'s VR-YOLO robust to viewpoint variations, further demonstrate the momentum toward practical and scalable inspection systems [34]. Compared with these approaches, our YOLO-v11s and YOLO-v11s-seg models achieve a favorable trade-off between accuracy and efficiency, with particularly strong performance in segmentation tasks that capture shape-based defects. These results suggest that the integration of segmentation within compact YOLO architectures represents a promising direction for enhancing defect detection in complex manufacturing environments.

Table 7. Comparison of our study with recent works.

Study (Year)	Dataset/Domain	Task	Main Method	Reported Metric(s)	Notes
Our study	SolDef_AI	Detection + Segmentation	YOLO-v11s / YOLO-v11s-seg	Overall accuracy ≈ 86%; poor_solder & spike remain challenging	Balanced speed/accuracy; segmentation improves shape-based defects but limited for fine solder errors
Xu et al., 2025	PCB solder joints	Detection	YOLO11n + RetBlock / SAF / AAF	mAP=0.850; mAP50=0.950; mAP75=0.950	Strong accuracy with a lightweight model
Li et al., 2025	PCB surface	Detection	SCF-YOLO (MobileNet-based)	Accuracy ≈95%; Recall >90%; FPS +60%	Designed for real-time applications
Zhu et al., 2025	PCB (various viewpoints)	Detection	VR-YOLO (YOLOv8 + CBAM)	mAP50: 98.9 (raw), 94.7 (under viewpoint shift)	Robust against viewpoint variations
Dai et al., 2020	PCB AOI	Detection	Hybrid DL approach	Stable defect detection results	Early application of DL in solder inspection
Sezer & Altan, 2021	PCB solder paste	Detection	Optimization-based CNN	High accuracy in defect detection	Industry-oriented approach
Zhang & Shen, 2021	PCB connectors	Detection	Improved Faster R-CNN	Robust detection vs. traditional	Strong on small solder joint defects
Liao et al., 2021	PCB surface defects	Detection	YOLOv4-MN3	High accuracy, low compute	Designed for real-time lines
Nayak & Parameshachari, 2022	PCB defects	Classification	Stacked Autoencoder + Bi-LSTM	Accuracy ≈ 99.99%	Very high classification accuracy
Bhattacharya & Cloutier, 2022	PCB defects	Classification	End-to-end CNN	High industrial accuracy	End-to-end DL framework
Ling et al., 2024	PCB soldering defects	Detection	SDD-Net	State-of-the-art IoU/mAP	New deep net architecture
Wang et al., 2024	PCB surface	Detection	Large kernel CNNs	High accuracy, strong generalization	Robust against noise
Vakili et al., 2024	PCB defects	Detection	Valid-IoU loss + DL	Performance improved via new loss	Novel metric-based improvement
Chen et al., 2024	PCB components	Detection	Multi-scale fusion + ECA	Better small defect detection	Attention-enhanced detection
Liu et al., 2024	PCB dataset	Detection + Data generation	ControlNet + Swin Transformer	Effective defect detection + augmentation	Combines sample generation with detection

Nevertheless, despite these advances, our experiments indicate that certain defect categories remain more challenging to detect, which points to unresolved gaps in the field.

In this study, the good and no_good classes are separated with high accuracy, whereas the majority of misclassifications occur in the poor_solder and spike categories. Three main reasons underlie this phenomenon:

- Data distribution and representation — the dataset is dominated by good and excessive_solder samples, while poor_solder is under-represented, which biases the decision boundaries.
- Target size and fine texture — defects such as spike occupy very small regions; single-scale training at 640×640 resolution can cause such patterns to be lost.
- Visual similarity — the contour and brightness similarity between poor_solder and excessive_solder increases the likelihood of confusion. These findings highlight the importance of multi-scale feature aggregation, class-balanced sampling, and augmentation strategies (e.g., zoom-in crops, mosaic, sharpening) to improve recognition of small or visually ambiguous defects.

5. Conclusion

This study presents a comprehensive evaluation of YOLO-based models for detecting and segmenting soldering defects in PCBs. Through extensive experiments on the SolDef_AI dataset, we compared multiple versions of YOLOv11 and YOLOv12, including their segmentation variants, to assess their effectiveness in identifying common soldering defects such as poor solder, excess solder, spikes, and misalignments.

YOLO-v11s-seg outperformed other models in both detection and segmentation tasks, achieving the highest

mean average precision (mAP) scores. This demonstrates the effectiveness of combining attention mechanisms with segmentation for fine-grained defect localization. YOLOv12 models showed competitive performance, particularly in bounding box detection, but exhibited slightly lower segmentation accuracy compared to YOLOv11-seg variants. This suggests that architectural refinements in YOLOv12 may require further optimization for mask prediction tasks. Smaller models (YOLOv11n, YOLOv12n) provided a good balance between speed and accuracy, making them suitable for real-time inspection systems where computational efficiency is crucial. Defect-specific performance analysis revealed that "no_good" (completely faulty solder) and "good" (proper solder) classes were detected with the highest precision, while "poor_solder" and "spike" defects posed greater challenges due to their subtle visual characteristics.

Future work could explore few-shot learning for rare defect types and edge-AI optimization to enable real-time processing on low-power industrial hardware. In conclusion, this study demonstrates that YOLO-based deep learning models, particularly those with segmentation capabilities, offer a reliable and scalable solution for PCB solder defect detection, contributing to improved quality assurance in electronics production.

Author's Contributions

Emel Soylu, Tuncay Soylu: conceptualization; methodology; data curation and augmentation; software implementation and experiments, evaluation and statistical analysis; visualization; writing—original draft; writing—review and editing; project administration.

Ethics

There are no ethical issues after the publication of this manuscript.

References

[1]. W. Dai, A. Mujeeb, M. Erdt, and A. Sourin, "Soldering defect detection in automatic optical inspection," Adv. Eng. Informatics, vol. 43, p. 101004, 2020.

[2]. Q. Ling, N. A. M. Isa, and M. S. M. Asaari, "SDD-Net: Soldering defect detection network for printed circuit boards," Neurocomputing, vol. 610, p. 128575, 2024.

[3]. A. Sezer and A. Altan, "Detection of solder paste defects with an optimization-based deep learning model using image processing techniques," Solder. \& Surf. Mt. Technol., vol. 33, no. 5, pp. 291–298, 2021.

[4]. A. Bhattacharya and S. G. Cloutier, "End-to-End Deep Learning Framework for Printed Circuit Board Manufacturing Defect Classification," Sci. Rep., 2022.

[5]. Y. Wang et al., "Surface Defect Detection of Printed Circuit Board With Large Kernel Convolutional Networks," 2024.

[6]. E. Vakili, G. Karimian, M. Shoaran, R. Yadipour, and J. Sobhi, "Valid-IoU: An Improved IoU-based Loss Function and Its Application to Detection of Defects on Printed Circuit Boards," Multimed. Tools Appl., 2024.

[7]. W. Chen, H. Zhao, and Z. Wang, "Defect Detection Model of Printed Circuit Board Components Based on the Fusion of Multi-Scale Features and Efficient Channel Attention Mechanism," Ieee Access, 2024.

[8]. Y. Liu, H. Wu, Y. Xu, X. Liu, and X. Yu, "Automatic PCB Sample Generation and Defect Detection Based on ControlNet and Swin Transformer," Sensors, 2024.

[9]. S. Liang et al., "Edge YOLO: Real-Time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," Ieee Trans. Intell. Transp. Syst., 2022.

[10]. G. Lavanya and S. D. Pande, "Enhancing Real-Time Object Detection With YOLO Algorithm," Eai Endorsed Trans. Internet Things, 2023.

[11]. M. J. Shaifee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A Fast You Only Look Once System for Real-Time Embedded Object Detection in Video," J. Comput. Vis. Imaging Syst., 2017.

[12]. Z. Guo, C. Wang, G. Yang, Z. Huang, and G. Li, "MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface," Sensors, 2022.

[13]. W. Tao, "Analysis the Improvements of YOLOv5 Algorithms: NRT-YOLO, MR-YOLO and YPH-YOLOv5," Appl. Comput. Eng., 2024.

[14]. X. Lang, Z. Ren, D. Wan, Y. Zhang, and S. Shu, "MR-YOLO: An Improved YOLOv5 Network for Detecting Magnetic Ring Surface Defects," Sensors, 2022.

[15]. J. Terven, D. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," Mach. Learn. Knowl. Extr., 2023.

[16]. M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature Toward Digital Manufacturing and Industrial Defect Detection," Machines, 2023.

[17]. L. Lü, X. Li, Y. Wu, and B. Chen, "Enhanced RT-DETR for Traffic Sign Detection:Small Object Precision and Lightweight Design," 2024.

[18]. Z. Li et al., "RT-DETR-SoilCuc: Detection Method for Cucumber Germinationinsoil Based Environment," Front. Plant Sci., 2024.

[19]. S. Tang, "Improvement of RT-DETR Model for Ground Glass Pulmonary Nodule Detection," PLoS One, 2025.

[20]. C. Tang, Y. Li, L. Wang, and W. Li, "Real-Time Traffic Light Detection Based on Lightweight Improved RT-DETR," 2024.

[21]. G. Fontana, M. Calabrese, L. Agnusdei, G. Papadia, and A. Del Prete, "SolDef_AI: An Open Source PCB Dataset for Mask R-CNN Defect Detection in Soldering Processes of Electronic Components," J. Manuf. Mater. Process., vol. 8, no. 3, p. 117, 2024.

[22]. K. Zhang and H. Shen, "Solder Joint Defect Detection in the Connectors Using Improved Faster-RCNN Algorithm," Appl. Sci., 2021.

[23]. J. P. R Nayak and B. D. Parameshachari, "Effective PCB Defect Detection Using Stacked Autoencoder With Bi-LSTM Network," Int. J. Intell. Eng. Syst., 2022.

[24]. X. Liao, S. Lv, D. Li, Y. Luo, Z. Zhu, and C. Jiang, "YOLOv4-MN3 for PCB Surface Defect Detection," Appl. Sci., 2021.

[25]. W. Huang and P. Wei, "A PCB dataset for defects detection and classification," arXiv Prepr. arXiv1901.08204, 2019.

[26]. "SolDef_AI: PCB dataset for defect detection." [Online]. Available: https://kaggle.com/datasets/f899d21ce26435a9aa74da20a1409641fcafee386bdaf980e451cdbd4d744e0c.

[27]. R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," arXiv Prepr. arXiv2410.17725, 2024.

[28]. Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," arXiv Prepr. arXiv2502.12524, 2025.

[29]. R. Padilla, S. L. Netto, and E. A. B. Da Silva, "A survey on performance metrics for object-detection algorithms," in 2020 international conference on systems, signals and image processing (IWSSIP), 2020, pp. 237–242.

[30]. S. Raschka, "An overview of general performance metrics of binary classifier systems," arXiv Prepr. arXiv1410.5330, 2014.

[31]. J. P. R. Nayak and B. D. Parameshachari, "Effective PCB Defect Detection Using Stacked Autoencoder With Bi-LSTM Network," Int. J. Intell. Eng. Syst., vol. 15, no. 2, pp. 285–294, 2022.

[32]. Y. Xu and others, "Lightweight PCB Solder Joint Defect Detection with YOLO11n Enhanced by RetBlock, SAF, and AAF," J. Intell. Manuf., vol., 2025.

[33]. H. Li, others, I. Mendizabal-Arrieta, and others, "SCF-YOLO: A Lightweight Real-Time Model for PCB Surface Defect Detection," Comput. Ind., vol., p. , 2025.

[34]. Q. Zhu and others, "VR-YOLO: Viewpoint-Robust PCB Defect Detection Based on YOLOv8 with CBAM," IEEE Access, vol., p. , 2025.