Small Object Defect Detection for PCB Inspection under Scale Mismatch*

Dmytro Kyrychuk[1,†], Oleksandr Pshenychnyy[2,†] and Yevhen Martyn[3,†]

[1] SoftServe, Sadova Street 2D, 79021, Lviv, Ukraine
[2] Leobit, Stryiska Street 108, 79000, Lviv, Ukraine
[3] Lviv State University of Life Safety, Kleparivska Street 35, 79007, Lviv, Ukraine

Abstract

Automated surface inspection of Printed Circuit Boards (PCBs) requires detecting defects that occupy a fraction of a percent of the captured image. In production, the scale of these defects at inference time often differs from the training data. A physically smaller board produces defects with fewer pixels, irreversibly losing detail. A large panel captured at high resolution retains all detail, but the model's fixed input resize discards it. This study investigates how this distinction affects detection accuracy of YOLOv10 and whether Slicing Aided Hyper Inference (SAHI) can compensate for the resulting degradation. Two simulation protocols, zoom-out and grid tiling, are applied across multiple scale factors to a PCB defect dataset with six defect classes. The results show that SAHI effectiveness is bounded by whether pixel information was preserved or destroyed: tiling-based inference nearly fully restores detection when the original detail is intact. When pixel information has been lost, SAHI still improves detection by presenting each slice at the scale the model was trained on, but recovery is limited by the quality of the remaining content. The findings indicate that the source of scale mismatch, not its severity alone, determines the ceiling of tiling-based inference recovery.

Keywords

small object detection, YOLOv10, scale mismatch, slicing-aided hyper inference (SAHI), automated optical inspection (AOI)

1. Introduction

Automated inspection of printed circuit boards (PCBs) relies on detecting defects that are often extremely small relative to the captured image. In production environments, the physical size of inspected boards can vary, and high-resolution cameras may capture large panels containing multiple PCB units in a single frame.

In both cases, the scale of defects at inference time differs from what the detection model encountered during training. A board that is physically smaller than those in the training set produces defects that occupy fewer pixels in the image. A large panel captured at high resolution, on the other hand, preserves all pixel detail, but the model's fixed input resolution forces downscaling that discards it. Although both situations reduce the relative defect area, only the second offers a path to recovery.

In practice, automated optical inspection (AOI) systems operate under constraints that make scale mismatch unavoidable. A single production line may handle boards of different physical dimensions, yet the inspection pipeline typically uses a fixed camera setup and a single trained model. Increasing camera resolution preserves more detail but produces larger images that require either downscaling or more expensive processing.

Tiling-based inference strategies, such as Slicing Aided Hyper Inference (SAHI) [1], offer a way to process high-resolution inputs without retraining, but they multiply the number of forward passes and introduce latency. Whether this computational cost is justified depends on how much

* SmartIndustry 2026: 3rd International Conference on Smart Automation & Robotics for Future Industry, March 26-27, 2026, Lviv, Ukraine
1* Corresponding author.
† These authors contributed equally.
dimakyrytschuk@gmail.com (D. Kyrychuk); sasha.pshenychniy@gmail.com (O. Pshenychnyy); evmartyn@gmail.com (Y. Martyn)
0009-0002-1941-4337 (D. Kyrychuk); 0000-0001-8823-7472 (O. Pshenychnyy); 0000-0001-9095-7057 (Y. Martyn)

accuracy can actually be recovered, which in turn depends on whether the pixel information was lost at capture or merely discarded during preprocessing.

This study systematically evaluates both types of scale mismatch and their interaction with tiling-based inference. Two simulation protocols are designed: zoom-out, which irreversibly reduces pixel information, and grid tiling, which preserves it. A YOLOv10 [2] detector is tested under both conditions with and without SAHI [1], across multiple scale factors. The goal is to determine whether and to what extent tiling-based inference can compensate for each type of scale mismatch, and what practical trade-offs this introduces.

The novelty of this study is the empirical evidence that SAHI, established as a tool for high-resolution images with preserved detail [1], is also effective when objects occupy fewer pixels than in training data and detail is permanently lost at capture. By reducing the slice size so that each slice, when resized to the model's input resolution, magnifies objects back to their training pixel size, SAHI compensates for the scale reduction. This magnification does not restore lost features but presents degraded content in a form the model was trained to recognize, enabling partial recovery. The experiments further show that the source of scale mismatch, not its magnitude, determines the extent of this recovery.

2. Analysis of related publications

Deep learning has become the dominant approach for automated PCB defect detection. A systematic review of 56 studies published between 2015 and 2025 found that YOLO-family architectures appeared in 26 of the reviewed articles, making them the most widely adopted detection framework in this domain [3]. The same review confirmed that 78.6% of studies detect six or fewer defect types and 62.5% rely on custom datasets, indicating a concentration around the six standard bare-PCB defect types used in the HRIPCB dataset [4,5].

Recent work has pursued architectural improvements to push detection accuracy higher on these standard benchmarks. In [6], multi-scale attention, content-aware upsampling, and an improved loss function were integrated into YOLOv11, achieving 99.5% mAP50 on the HRIPCB dataset at 608×608 input resolution. In [7], YOLOv7-tiny was enhanced with coordinate attention and deformable convolutions, reaching 98.3% mAP50 at 128 FPS on the same dataset. In [8], YOLOv5s and YOLOv8s were applied with data augmentation strategies targeting illumination variation, achieving up to 95.18% mAP50 on a custom 19-class PCB dataset.

These studies share a common experimental design: training and testing are conducted at identical resolutions under controlled conditions, with no investigation of what happens when the scale at inference time differs from training.

The YOLO architecture itself has continued to evolve. YOLOv10 [2] eliminates non-maximum suppression (NMS) post-processing through consistent dual assignments, where a one-to-many head provides supervision during training while a one-to-one head enables end-to-end inference. It offers six model variants with efficiency and accuracy improvements over its predecessors, and all variants use a fixed 640×640 input resolution.

Small object detection (SOD) presents distinct challenges that standard architectural improvements do not fully address. A recent survey of SOD papers [9] identifies scale mismatch as a core challenge: fine-grained spatial information and weak features of small objects are lost in deeper network layers as feature maps are progressively downsampled.

The COCO benchmark defines small objects as those with area below 32×32 pixels [9], a threshold that PCB defects easily fall under when images are resized to fit model input dimensions. Feature pyramid networks and multi-scale fusion are the primary architectural remedies, but they remain insufficient under extreme scale mismatch [9].

For high-resolution inputs, tiling-based inference has emerged as a practical alternative. SAHI [1] partitions input images into overlapping patches, runs the detector on each patch independently, and merges predictions back to original coordinates via NMS. On aerial imagery benchmarks (VisDrone and xView), SAHI yields up to 6.8% AP improvement, and when combined

with slicing-aided fine-tuning, up to 14.5% AP improvement for small objects [1]. Computational time scales linearly with the number of slices while memory usage remains fixed [1].

Small object detection practical guidelines explicitly recommend tiled inference for both aerial imagery and industrial defect detection, advising against downsampling original inputs when possible [9]. However, the review in [3] notes that edge deployment often requires resizing high-resolution images to lower resolutions, which may compromise accuracy, yet neither SAHI nor scale mismatch at inference time are discussed.

A consistent gap emerges across these bodies of work. PCB defect detection studies [6-8] focus on architectural improvements under fixed-resolution conditions. SOD surveys [9] recommend tiling as a practical solution but do not systematically evaluate when it succeeds or fails. The SAHI framework [1] demonstrates effectiveness on aerial datasets but does not distinguish between scenarios where pixel information has been preserved versus irreversibly lost.

No prior work, to the best of the authors' knowledge, separates these two sources of scale mismatch or evaluates how each constrains the ceiling of tiling-based recovery. This study addresses that gap.

3. Methodology

3.1. Evaluation framework

The methodology is built on the observation that scale mismatch at inference time can arise from two fundamentally different sources. In the first case, pixel information is permanently lost before detection, for example, when a physically smaller board produces defects that occupy fewer sensor pixels. In the second case, all pixel information is preserved in the captured image, but the model's fixed input resolution discards it during preprocessing.

Although both cases reduce the relative area of defects, only the second retains sufficient information for full recovery through post-hoc processing.

To isolate the effect of pixel preservation from the effect of scale reduction, the framework introduces two simulation protocols, zoom-out and grid tiling, that produce matched relative object areas through different mechanisms.

Zoom-out downsamples the image content, destroying pixel information permanently. Grid tiling instead assembles original images into a larger canvas, preserving pixel content identically to the training data.

By comparing detection accuracy and SAHI recovery at matched scale ratios where both protocols produce equivalent relative defect areas, the framework attributes any difference in recovery to the presence or absence of pixel information rather than to the degree of scale change.

The SAHI slice dimensions follow a single design principle: each slice should present the detector with a region equivalent to one training-scale image. For pixel-preserving mismatch, this fully reconstructs training conditions. For pixel-destroying mismatch, it magnifies degraded content back to training scale, providing a controlled upper bound on what tiling can recover when the original detail is gone.

3.2. Dataset

For this research, the publicly available PCB Defect Dataset [4,5] was used, containing 600×600 pixel images annotated with six defect classes (mouse_bite, spur, missing_hole, short, open_circuit, spurious_copper).

The dataset is split into train, validation, and test partitions. The test set comprises 1068 images with 1662 defect instances. All defect classes have average bounding box areas ranging from 0.14% to 0.41% of the image, which is represented in Table 1.

These percentages represent the ratio of each defect's bounding box area to the total image area of 600×600 pixels.

Table 1
Original dataset scale statistics

Defect	Train Count	Train Avg %	Test Count	Test Avg %
mouse_bite	2221	0.2117	262	0.1957
spur	2186	0.2784	279	0.3047
missing_hole	2197	0.2555	283	0.2490
short	2029	0.4059	275	0.4119
open_circuit	2137	0.1424	265	0.1582
spurious_copper	2221	0.2964	298	0.2866

3.3. Detection model

YOLOv10m [2] was selected as the detection backbone and trained for 30 epochs at 640×640 resolution on an NVIDIA Tesla T4 GPU. Best weights were selected by validation mAP50. No special augmentation strategies targeting small objects were employed, as the goal was to evaluate the baseline model's robustness to scale mismatch at inference time.

3.4. Scale mismatch simulation

Two transformations simulate the two sources of scale mismatch. Zoom-out destroys pixel information irreversibly, while grid tiling preserves it.

3.4.1. Zoom-out transformation

The zoom-out transformation simulates scenarios where defects occupy fewer pixels than in the training data, either because the camera is placed farther from the board or because the PCB itself is physically smaller with finer traces and details. Each original 600×600 test image is resized by a zoom factor $z \in \{0.7, 0.5, 0.3\}$, reducing both dimensions proportionally. The resized image is then placed at the center of a new 600×600 canvas filled with neutral grey, as shown in Figure 1. Bounding box labels are scaled by the same factor and shifted to account for the centering offset.

This procedure preserves the overall image dimensions but reduces the effective object size. Pixel information is permanently lost during the resize step, as downsampling interpolation discards fine-grained defect details that cannot be recovered at inference time.

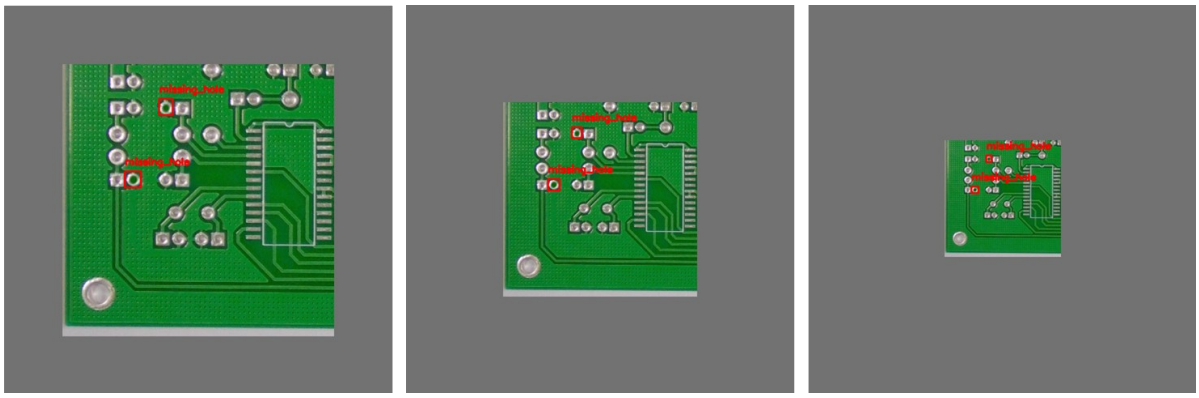

Figure 1: Zoom-out transformation at factors 0.7, 0.5, and 0.3.

Table 2
Object scales after zoom-out: bounding box area as percentage of total image area

Defect	Z=0.7, avg %	Z=0.5, avg %	Z=0.3, avg %
mouse_bite	0.0959	0.0489	0.0176
spur	0.1493	0.0762	0.0274
missing_hole	0.1220	0.0622	0.0224
short	0.2018	0.1030	0.0371
open_circuit	0.0775	0.0396	0.0142
spurious_copper	0.1404	0.0716	0.0258

3.4.2. Grid tiling transformation

The grid tiling transformation simulates a high-resolution camera capturing a large PCB panel. Each test image is replicated across a regular grid, producing a larger canvas. Three configurations were evaluated: 2×2 (1200×1200), 2×4 (1200×2400), and 5×5 (3000×3000). The 2×4 configuration produces a non-square canvas, so the model's letterbox resize to 640×640 adds padding that wastes a portion of the fixed input area.

Bounding box labels for each tile are offset by the tile position and scaled to the full canvas dimensions.

Unlike zoom-out, the original pixel content of each defect is fully preserved. The number of defect instances grows proportionally with the number of tiles (e.g., 2×2 produces four copies of each defect per image).

The examples of resulting tiled images are illustrated in Figure 2.

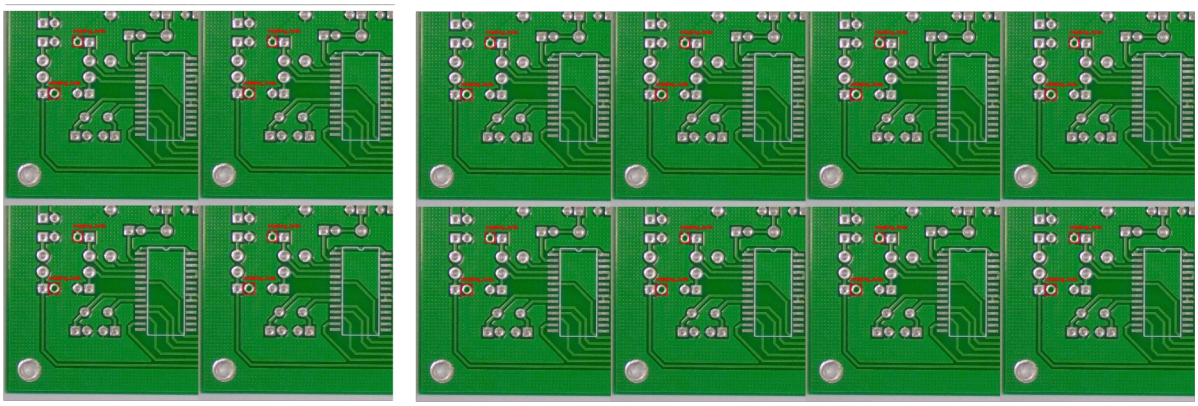

Figure 2: Grid tiling at 2×2 and 2×4 configurations. Grid with 5×5 configuration was too large to include.

The small percentages in Table 3 reflect the ratio of unchanged defect dimensions to the larger canvas area, not a reduction in actual defect detail. However, when the model resizes the full canvas (e.g., 3000×3000) to its fixed 640×640 input, defects that were originally 22–38 pixels become 5–8 pixels, falling below the detection limit.

Table 3
Object scales after grid tiling: bounding box area as percentage of total canvas area

Defect	2×2 grid, avg %	2×4 grid, avg %	5×5 grid, avg %
mouse_bite	0.0489	0.0245	0.0078
spur	0.0762	0.0381	0.0122
missing_hole	0.0622	0.0311	0.0100
short	0.1030	0.0515	0.0165
open_circuit	0.0396	0.0198	0.0063
spurious_copper	0.0716	0.0358	0.0115

Although the degradation mechanisms differ, zoom-out and grid tiling can produce identical relative object areas. For example, zoom 0.5 and grid 2×2 both yield approximately 0.04–0.10% average areas.

However, the grid retains full pixel information while zoom discards it irreversibly.

3.5. SAHI configuration

Slicing Aided Hyper Inference (SAHI) [1] partitions a test image into overlapping windows, runs the detector on each window independently, and merges detections via non-maximum suppression in the original image coordinate space.

The central design principle for the SAHI configuration is that each slice should present the model with a region corresponding to one training-scale image.

For zoom experiments, slice size is set to 600×z pixels, where z is the zoom factor. At zoom 0.7, slices are 420×420, at zoom 0.5 slices are 300×300, and at zoom 0.3 slices are 180×180. Each slice covers the area equivalent to one original training image within the zoomed canvas.

When resized to the model's input resolution (640×640), this provides magnification that partially compensates for the zoom. However, because pixel information was already lost during the zoom-out downsampling, magnification amplifies degraded content rather than restoring original detail.

For grid experiments, slice size is 600×600 pixels, matching the original tile dimensions exactly. Each slice captures one complete tile at its native resolution, identical to training conditions. This allows SAHI to fully reconstruct the training-time data distribution.

In both configurations, an overlap ratio of 0.2 is used to ensure that defects near slice boundaries appear fully in at least one adjacent slice.

3.6. Evaluation

For standard YOLO inference, the built-in YOLOv10 validation pipeline was used, which reports per-class and aggregate Precision, Recall, mAP50, and mAP50-95.

For SAHI evaluation, ground-truth YOLO labels were converted to COCO JSON format. SAHI predictions were saved as COCO-format result files and evaluated using the COCO evaluation API, producing per-class mAP50 and mAP50-95 directly comparable to the YOLO pipeline metrics. Throughout this study, mAP50 is used as the primary detection metric and mAP50-95 as the localization quality metric.

4. Experiments and results

4.1. Baseline results

Table 4
Baseline performance on the original test set

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	1068	1662	0.975	0.990	0.989	0.602
mouse_bite	131	262	0.977	0.985	0.991	0.589
spur	138	279	0.972	0.989	0.991	0.601
missing_hole	145	283	0.972	0.993	0.993	0.643
short	142	275	0.974	0.985	0.985	0.610
open_circuit	128	265	0.990	1.000	0.995	0.588
spurious_copper	145	298	0.964	0.984	0.980	0.580

The model achieves strong baseline performance with mAP50 = 0.989 across all classes. The gap between mAP50 and mAP50-95 (0.387) reflects the inherent difficulty of precise localization for small defects. Even under ideal conditions, tight bounding box alignment is challenging when objects average 0.14–0.41% of image area.

4.2. Zoom-out experiments

Table 5
Zoom-out results: YOLO vs SAHI at three zoom factors

Scale	Method	mAP50	mAP50-95	Δ mAP50
1.0 (baseline)	YOLO	0.989	0.602	—
0.7	YOLO	0.990	0.579	+0.001
0.7	SAHI	0.984	0.547	-0.005
0.5	YOLO	0.951	0.491	-0.038
0.5	SAHI	0.968	0.494	-0.021
0.3	YOLO	0.295	0.105	-0.694
0.3	SAHI	0.742	0.344	-0.247

Three distinct performance regimes emerge. At zoom 0.7, YOLO maintains full detection accuracy (mAP50 = 0.990), and SAHI is marginally counterproductive (0.984) due to slicing artifacts without meaningful magnification benefit. At zoom 0.5, a mild degradation appears (mAP50 = 0.951), and SAHI provides a modest improvement to 0.968 through effective magnification of the content region. At zoom 0.3, YOLO performance collapses (mAP50 = 0.295). SAHI recovers mAP50

to 0.742, a 2.5 times improvement, but still falls 25% short of baseline. The residual gap reflects the irreversible pixel information loss incurred during the zoom-out downsampling.

4.3. Grid tiling experiments

Table 6
Grid tiling results: YOLO vs SAHI at three grid configurations

Grid	Method	Instances	mAP50	mAP50-95	Δ mAP50
1x1 (baseline)	YOLO	1662	0.989	0.602	—
2x2	YOLO	6648	0.959	0.499	-0.030
2x2	SAHI	6648	0.975	0.508	-0.014
2x4	YOLO	13296	0.128	0.042	-0.861
2x4	SAHI	13296	0.958	0.524	-0.031
5x5	YOLO	41550	0.025	0.008	-0.964
5x5	SAHI	41550	0.966	0.545	-0.023

For larger grids without a zoom-out analogue, YOLO degradation is even more severe. At 5×5, YOLO mAP50 drops to 0.025 (effectively random), since the 3000×3000 canvas is resized to 640×640, reducing each defect to fewer than 5 pixels. However, SAHI recovery in the grid scenario is substantially higher. With 600×600 slices that exactly reconstruct the original tile dimensions, SAHI achieves mAP50 of 0.975, 0.958, and 0.966 for 2×2, 2×4, and 5×5 grids respectively, all within 1.4–3.1% of the baseline. The larger the YOLO performance drop, the greater the SAHI recovery. At 5×5, SAHI improves mAP50 by a factor of nearly thirty-nine.

5. Discussion

YOLOv10 tolerates moderate scale mismatch but exhibits threshold-like collapse beyond it. In zoom-out, performance holds at z=0.7 but drops to 0.295 at z=0.3. In grid tiling, 2×2 remains viable (0.959) but 2×4 collapses to 0.128 and 5×5 to 0.025. The 2×4 case compounds the problem with aspect ratio distortion: the non-square canvas (1200×2400) forces letterbox padding during resize, wasting input area and increasing effective downscaling along the longer dimension.

SAHI recovery depends critically on the source of mismatch, not its magnitude. Grid 5×5 defects occupy smaller relative areas than zoom 0.3 defects, yet SAHI recovers the grid to 0.966 versus only 0.742 for zoom, because the grid preserves pixel information that zoom-out destroyed.

For AOI systems capturing large panels, SAHI with training-matched slice dimensions nearly eliminates the accuracy penalty without retraining. However, this comes at a direct computational cost: each slice requires a full forward pass at the model's input resolution, so total inference time scales linearly with the number of slices. Since slice dimensions are effectively fixed by the training resolution for optimal accuracy, larger images produce proportionally more slices with no option to reduce the count without sacrificing detection quality. The overlap ratio further increases the slice count to prevent missed detections at boundaries, adding to the linear scaling.

6. Conclusion

This study evaluated YOLOv10 detection accuracy under scale mismatch using two simulation protocols applied to a PCB defect dataset with six defect classes across multiple scale factors. Zoom-out simulated physically smaller boards by downscaling images, irreversibly reducing pixel detail. Grid tiling simulated large panels by assembling multiple images into a larger canvas,

preserving the original pixel content of each defect. SAHI with training-matched slice dimensions was applied to both protocols to measure the extent of accuracy recovery.

The experiments revealed that YOLOv10 maintains baseline accuracy under moderate mismatch but collapses sharply beyond a threshold, dropping to mAP50 of 0.295 at zoom 0.3 and 0.025 at grid 5×5. SAHI recovered grid 5×5 to mAP50 of 0.966 (within 2.3% of baseline) but only recovered zoom 0.3 to 0.742 (25% below baseline). The key conclusion is that pixel information preservation, not the magnitude of scale reduction, determines the ceiling of tiling-based recovery. Notably, SAHI proved effective not only on high-resolution images larger than the training input, but also on same-resolution images where defects appear smaller than during training. In the zoom-out scenario, the image dimensions remain 600×600, identical to training data, yet SAHI with slices matched to the zoomed content area (e.g., 180×180 at z=0.3) magnified the content back toward training scale, improving mAP50 from 0.295 to 0.742. A secondary finding is that mAP50-95 remains below baseline even when mAP50 is fully recovered, indicating that slice boundary effects introduce localization imprecision.

These results suggest that for AOI systems where camera resolution preserves full detail, SAHI can compensate for scale mismatch without retraining. Where pixel information is lost at capture, the problem must be addressed at the data acquisition stage. Future work may explore combining SAHI with super-resolution preprocessing to partially restore lost detail, as well as extending these findings to other detector architectures and defect domains.

Declaration on Generative AI

During the preparation of this work, the authors utilized Gemini to identify and rectify grammatical, typographical, and spelling errors. Following the use of this tool, the authors conducted a thorough review and made necessary revisions, and accept full responsibility for the final content of this publication.

References

[1] F. C. Akyon, S. O. Altinuc, A. Temizel, Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection, in: Proceedings of the IEEE International Conference on Image Processing, ICIP '22, IEEE, Bordeaux, France, 2022. doi:10.1109/ICIP46576.2022.9897990.

[2] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-Time End-to-End Object Detection, in: Advances in Neural Information Processing Systems, NeurIPS '24, Vancouver, Canada, 2024. doi:10.48550/arXiv.2405.14458.

[3] B. Montoya Magaña, Ó. Hernández-Uribe, L. A. Cárdenas-Robledo, J. A. Cantoral-Ceballos, Deep Learning Algorithms for Defect Detection on Electronic Assemblies: A Systematic Literature Review, Mach. Learn. Knowl. Extr. 8 (2026) 5. doi:10.3390/make8010005.

[4] R. Ding, L. Dai, G. Li, H. Liu, TDD-net: a tiny defect detection network for printed circuit boards, CAAI Trans. Intell. Technol. 4 (2019) 110–116. doi:10.1049/trit.2019.0019.

[5] N. Elter, PCB Defect Dataset. URL: https://www.kaggle.com/datasets/norbertelter/pcb-defect-dataset.

[6] Z. F. Elsharkawy, Enhanced YOLOv11 framework for high precision defect detection in printed circuit boards, Sci. Rep. 15 (2025) 42550. doi:10.1038/s41598-025-27415-w.

[7] G. Xiao, S. Hou, H. Zhou, PCB defect detection algorithm based on CDI-YOLO, Sci. Rep. 14 (2024) 7351. doi:10.1038/s41598-024-57491-3.

[8] Z. Ni, Y. Kim, Research on Printed Circuit Board (PCB) Defect Detection Algorithm Based on Convolutional Neural Networks (CNN), Appl. Sci. 15 (2025) 13115. doi:10.3390/app152413115.

[9] A. Aldubaikhi, S. Patel, Advancements in Small-Object Detection (2023-2025): Approaches, Datasets, Benchmarks, Applications, and Practical Guidance, Appl. Sci. 15 (2025) 11882. doi:10.3390/app152211882.