

Article

Defect-Intent Ambiguity Addressing for Training-Free Deterministic PCB Defect Localization via Template Selection and Dissimilarity Mapping

Saiyan Saiyod , Woottichai Nonsakhoo , Zhengping Li and Piyanat Sirisawat

Hardware-Human Interface and Communications Laboratory (H2I-Comm Lab), Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen 40002, Thailand; nonsakhoo@cassia.kku.ac.th (W.N.); zhengping.l@cassia.kku.ac.th (Z.L.); piyanat_siri@cassia.kku.ac.th (P.S.)

* Correspondence: saiyan@kku.ac.th

Abstract

Automated optical inspection (AOI) for printed circuit boards (PCBs) requires localizing small, sparse defects under illumination drift and minor placement misalignment, while supporting fast, auditable pass/fail decisions. This paper presents a training-free, reference-based digital image processing framework with no learning/training stage that compares each defective query image with a small library of defect-free reference templates (for the same PCB layout/revision) using a small set of interpretable control parameters. A reference is selected by coarse-to-fine matching (fast pre-screening followed by SSIM refinement on a central region), and an optional global alignment is applied only when it increases SSIM to limit defect-driven over-correction. Defects are highlighted by a defect-likelihood field that fuses an SSIM-derived structural dissimilarity map with a normalized absolute-difference map, followed by connected-component extraction to produce confidence-ranked bounding boxes. The method achieves Precision = 0.9663, Recall = 0.9987, and F1 = 0.9822 at the best-F1 operating point (0.149 false positives per image). Under the adopted box-matching protocol, average precision reaches 0.984. Precision–recall and FROC curves are reported to support threshold selection under different false-alarm budgets.

Keywords: printed circuit board (PCB); automated optical inspection (AOI); defect localization; defect detection; image processing; template selection; structural similarity; difference map; correlation; quantile thresholding

1. Introduction

Automated optical inspection (AOI) is a core quality control stage in printed circuit board (PCB) manufacturing, where decisions must be made reliably at production speed [1,2]. In practice, true defects are sparse and localized (e.g., missing copper, spurious copper, scratches, and pinholes), yet benign variability in illumination and minor board-placement changes is common [3]; therefore, a useful inspection method must suppress nuisance variability while remaining sensitive to subtle, localized structural deviations. AOI is typically positioned inside a broader in-line quality strategy that aims to reduce scrap and rework while keeping false alarms manageable in production [4].

In many industrial quality control (QC) workflows, the primary decision is pass/fail (accept/reject): if any verified defect is present, the board is rejected. Therefore, reliable defect localization and evidence are central, while fine-grained defect-type classification



Academic Editor: Tat-Hean Gan

Received: 30 January 2026

Revised: 22 February 2026

Accepted: 26 February 2026

Published: 28 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

(e.g., Missing_hole, Mouse_bite, Short) is often optional and can be treated as a downstream step for reporting, process monitoring, or root-cause analysis [5]. At the same time, surveys of PCB defects and inspection practices highlight that defect taxonomies and visual manifestations can vary across processes and products, which complicates universal classification claims and emphasizes the value of interpretable evidence [6].

A key practical disparity is that the notion of a “defect” can be ambiguous without context from the intended design [7]. In PCB engineering, many localized copper features are introduced deliberately to satisfy electromagnetic interference and compatibility (EMI/EMC) and signal/power integrity (SI/PI) constraints rather than to “look clean” in an image [7–10]. For example, an apparent copper bridge may be an unintended short in one location but an intentional connection, net-tie, or controlled strap in another. Likewise, patterns that resemble spurious copper in isolation can be purposeful copper pours and fills (e.g., ground pours for shielding and controlled return-current paths), via stitching and via fences to reduce radiation and crosstalk, guard traces, copper thieving/balancing for manufacturability, or local tuning structures used to manage impedance, ringing, and high-frequency noise.

Spur-like features and short stubs can also be intentional (e.g., test access, reinforcement/teardrops, thermal relief, or current-spreading structures), even though similar geometries may be considered defects in other contexts. Because these intent-driven features may be small and localized, appearance-only inspection can confuse design intent with anomaly [6,11].

This defect-versus-intent ambiguity motivates reference-based inspection: comparing a query board against a defect-free exemplar of the same layout provides the necessary design-context baseline [12]. The template encodes the intended routing, copper distribution, and layout-specific structures used for EMI/EMC and SI/PI, allowing the inspection decision to focus on deviations from the intended reference (true defects) rather than on visually unusual but intentional patterns [7,8].

In production, the primary constraint is often not peak accuracy in a curated benchmark but predictable behavior under everyday drift and change [4,13]. Practical AOI systems are judged by time-to-deploy on a new board, validation burden, compute footprint, and the ability to explain failures when they occur [13]. End-to-end learning-based models are widely used; however, in some deployment settings their internal decision logic can be harder to audit and their lifecycle cost can include labeling, retraining, monitoring, and re-qualification [5,14]. When an inspection decision is disputed, it is valuable to attribute the response to concrete causes such as illumination shift, residual misalignment, template mismatch, or a true localized defect [15,16].

Accordingly, a recurring deployment theme is to reduce manual labeling burden and model maintenance effort while still improving robustness, which motivates both training-free pipelines and low-label learning regimes [17,18].

These constraints motivate a complementary design point: analysis-first, physics- and mathematics-grounded image processing that exposes its decision signal. Classical industrial inspection continues to rely on deterministic components such as thresholding, morphology, and template matching because they are auditable and controllable [19,20]. In this view, a defect-free template functions as a measurement reference standard: inspection becomes change detection against the intended layout rather than category recognition.

The main obstacle is that simple subtraction is sensitive: small geometric mismatch and brightness drift can dominate the difference image and produce widespread false positives [15]. Robust reference-based inspection therefore benefits from structural similarity measures and careful handling of alignment [21]. The structural similarity index (SSIM) [22] emphasizes structure rather than raw intensity and can be evaluated locally

to form an SSIM map that highlights localized changes. Alignment can further reduce nuisance differences, but in defect localization it must be used carefully because defects can bias the estimated transform; a pragmatic safeguard is to accept alignment only when it yields a measurable similarity improvement [23].

Related work on PCB AOI spans broad surveys and focused PCB-specific reviews that summarize classical image processing pipelines and, more recently, deep learning-based approaches [1,2,6,11,14,24]. Classical AVI/AOI surveys establish the foundational problem setting and recurring algorithmic motifs (preprocessing, registration/matching, and post-processing for decision rules) [25]. More recent reviews focus on electronics manufacturing and PCB inspection specifically, including the rise of deep learning and the associated practical considerations (data requirements, interpretability, and lifecycle/maintenance cost) [11,14,24]. Across these perspectives, a consistent theme is that production deployment often rewards predictable behavior and actionable diagnostics, which keeps reference-based inspection relevant even when learning-based models are available.

Beyond RGB surface imagery, inspection and monitoring can also involve specialized imaging and task-specific pipelines, for example in solder-joint and electronic assembly defect analysis, where the sensing modality and measurement objective differ from bare-board surface AOI [26,27].

A long-standing and still common formulation is reference/template-based change detection, where the query is compared to a defect-free exemplar and differences are extracted using similarity measures and post-processing [12,20,21]. In this family, normalized cross-correlation and related similarity measures are widely used to compare a query against a template, both for alignment/matching and for highlighting likely change regions [12,28]. Because AOI images can differ by small shifts, rotations, and illumination drift, many works explicitly study robustness of similarity/dissimilarity measures and the sensitivity of subtraction-based signals to misregistration [15,29]. Feature-based matching and alternative similarity measures (beyond raw subtraction) are also widely explored as ways to increase tolerance to nuisance variation while preserving localized change cues [16,30,31]. Rotation or uncertainty-aware designs further emphasize that handling geometric variability is a practical requirement rather than a corner case [32]. This practical issue is directly aligned with our use of structural similarity and alignment safeguards: the goal is to suppress broad, nuisance-driven responses while preserving localized defect evidence [22,23].

Within classical (training-free) inspection, several works explicitly combine subtraction/matching with deterministic operators for categorization and reporting [19,21,33]. For example, an algorithmic scheme for concurrent detection and classification applies fuzzy c-means segmentation after image subtraction and then uses arithmetic/logic operations, the circle Hough transform, morphological reconstruction, and connected-component labeling to assign defect categories; it reports 100% detection and 99.05% classification accuracy in its experimental setting [33]. In a similar spirit, other subtraction-based pipelines target both detection and defect-type decisions by combining difference images with hand-designed operators (e.g., morphology, logical operations, and component analysis), illustrating the long-standing appeal of deterministic, auditable processing in PCB AOI [19].

In parallel, deep learning has been widely adopted for PCB defect detection and classification, commonly borrowing generic object-detection backbones and adapting them for small, low-contrast defects and industrial constraints [24,34–37]. One line of work applies one-stage object detectors (often YOLO-family variants) and proposes architectural changes to improve localization of tiny PCB defects while keeping inference efficient [38–44]. Additional one-stage variants integrate backbone and attention refinements for feature reuse and improved localization under clutter [45–48]. Another line adapts two-stage and feature-

pyramid designs (e.g., Faster R-CNN/FPN-style components) to improve multi-scale defect detection where targets are small and the background is cluttered [49–51]. Beyond detector backbones, several studies emphasize lightweight deployment, attention/feature-fusion refinements, and context modeling to trade off accuracy and throughput in practical settings [52–54]. Deep learning is also used for related PCB inspection tasks including defect classification, reconstruction/autoencoder-based inspection signals, and component-level understanding such as PCB segmentation for recognition [55–57]. Transformer- and DETR-inspired detectors are another recent direction, motivated by global context modeling and end-to-end detection design [58,59]. Beyond standard detector framing, learning-based approaches also include alternative objectives and cues (e.g., energy-based or edge-guided signals), domain adaptation strategies to handle distribution shift [60], and PCB-specific designs for bare-board defects [61–63].

Because labeled defect samples can be scarce and production distributions can drift, recent work also explores semi-supervised, unsupervised, and few-shot settings for PCB inspection [17,18,64]. These directions aim to reduce annotation burden and improve adaptability by leveraging unlabeled data, uncertainty modeling, or meta-learning-style transfer from limited support examples [17,18,64].

An important practical gap in PCB inspection is defect-versus-intent ambiguity. Appearance-only cues can confuse intentional layout features with true defects unless a layout-specific reference is available. Motivated by this, this work presents a training-free, reference-based image processing framework that retains the operational advantages of template inspection while improving robustness to everyday nuisance variation. While end-to-end learning-based defect detectors are widely used in industrial inspection across product domains, the proposed pipeline provides an auditable and controllable alternative that does not require a training lifecycle [65,66]. The method is computationally efficient and operationally transparent: it produces interpretable intermediate fields (similarity and difference maps) and relies on controllable operators whose behavior can be inspected and tuned without retraining.

The main contribution of this work lies in a training-free framework design that combines classical operators with explicit robustness safeguards targeting defect-versus-intent ambiguity under everyday nuisance variability. In particular, we emphasize ROI-based template selection, SSIM-gated alignment acceptance, and a fused defect-likelihood mapping that yields confidence-ranked detections for deployment-oriented PR/FROC threshold selection. The key contributions are as follows:

- Coarse-to-fine reference selection with central-region similarity: a fast pre-screening stage plus SSIM refinement computed on a central region to suppress border/outlier effects, with caching and downscaling for throughput.
- SSIM-gated alignment acceptance: an explicit validation rule that applies a global warp only when SSIM improves on the same central region, mitigating defect-driven over-correction.
- Fused, auditable defect-likelihood and operational evaluation: an explicit defect-likelihood field that fuses structural dissimilarity and normalized absolute difference with central-region-based rescaling for stable thresholding, producing confidence-ranked candidates that are characterized via PR/FROC and IoU-sensitivity rather than a single tuned threshold.

The remainder of this paper is organized as follows: Section 2 details the proposed method, Section 3 presents the experimental results and analysis (including the evaluation protocol), Section 4 discusses limitations and practical considerations, and Section 5 concludes this paper.

2. Proposed Methods

This section presents a training-free PCB defect-localization pipeline. For each query image, it selects a defect-free reference template and outputs (i) an interpretable defect-likelihood field and (ii) confidence-ranked bounding boxes. At a high level, the pipeline performs coarse-to-fine template selection, constructs the defect-likelihood field by fusing SSIM-based structural dissimilarity with normalized absolute difference, applies global alignment only when it improves SSIM, and extracts sparse candidates via quantile thresholding, morphology, and connected-components analysis (Figure 1).

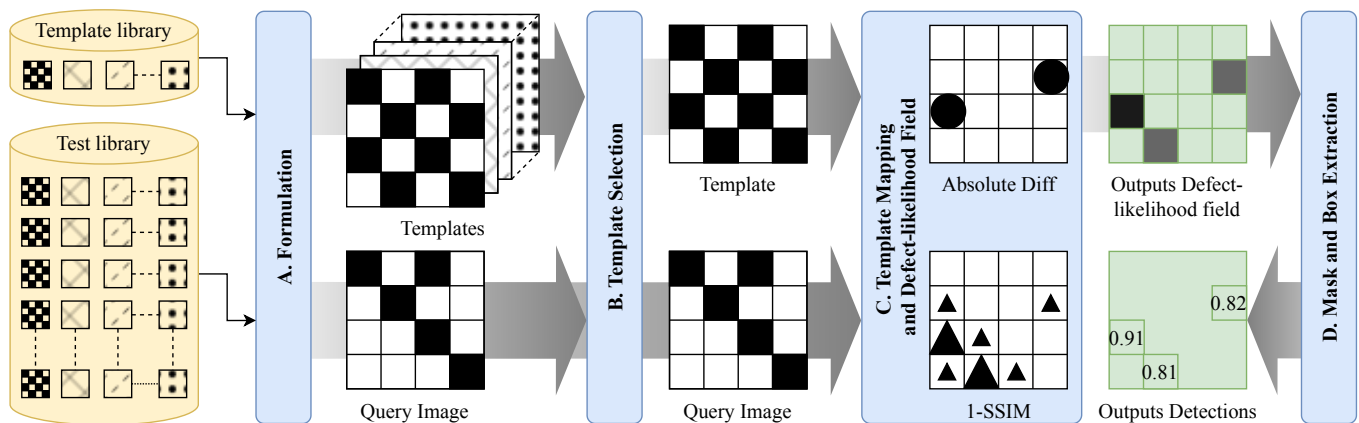


Figure 1. Overview of the proposed image processing framework for PCB defect localization.

2.1. Formulation

This work considers reference-based PCB inspection in which, for each defective query image, one or more defect-free exemplars of the same PCB layout are available. Let I denote a defective query image and $\mathcal{T} = \{T_k\}_{k=1}^K$ denote a library of defect-free templates. Here, templates refer to defect-free reference images for the same PCB layout/revision (not defect-type templates that enumerate defect categories). Consequently, the localization stage is class-agnostic across defect categories: any defect that manifests as a localized deviation from a valid defect-free reference can yield a response in the defect-likelihood field.

The goal is to output the following: (i) a spatial defect-likelihood field $M(u, v)$ (higher values indicate more likely defect) and (ii) a set of N axis-aligned bounding boxes with confidence scores $\mathcal{P} = \{(b_i, s_i)\}_{i=1}^N$, where each $b_i = [x \ y \ w \ h]$ denotes the top-left corner (x, y) and the box width/height (w, h) , and s_i is a confidence score.

To stabilize similarity computation and threshold estimation, statistics are evaluated on a central image region Ω parameterized by a margin with m pixels on all sides. The margin m is computed from an area-avoidance ratio $p \in [0, 1]$, which specifies the fraction of the total image area to ignore.

2.2. Template Selection

All images are converted to grayscale and normalized to $[0, 1]$ by an affine intensity rescaling. Scalar SSIM is denoted by $\text{SSIM}(\cdot, \cdot)$ [22]. Pixel coordinates are (u, v) (column, row) and the image domain is $\mathcal{X} = \{1, \dots, W\} \times \{1, \dots, H\}$.

Similarity and thresholds are evaluated on an inner region Ω parameterized by a symmetric margin m determined from an area-avoidance ratio $p \in [0, 0.95]$, where 0.95 is an experimental upper limit to ensure a nontrivial inner region. With a margin m on all four sides, the central area is $(W - 2m)(H - 2m)$ and the remaining area is

$$WH - (W - 2m)(H - 2m) = 2m(W + H) - 4m^2. \quad (1)$$

Setting this equal to pWH yields the (smaller) quadratic root

$$m = \frac{(W + H) - \sqrt{(W + H)^2 - 4pWH}}{4}, \quad (2)$$

and $\lfloor m \rfloor$ is used as an integer padding. The inner region is

$$\Omega = \{(u, v) \in \mathcal{X} : 1 + m \leq u \leq W - m, 1 + m \leq v \leq H - m\}. \quad (3)$$

In implementation, this continuous expression is converted to an integer ROI with explicit rounding and safety clamping; see Appendix A.1.

Given a query image I and template library $\{T_k\}_{k=1}^K$, the template is selected by maximizing SSIM on the inner region at a designated selection resolution. Let $\downarrow_D(\cdot)$ denote isotropic resizing to satisfy $\max(H, W) \leq D$ and let $\mathcal{N}(\cdot)$ denote grayscale+normalization. At selection scale D_{sel} ,

$$I_{\text{sel}} = \mathcal{N}(\downarrow_{D_{\text{sel}}}(I)), \quad T_{k,\text{sel}} = \mathcal{N}(\downarrow_{D_{\text{sel}}}(T_k)). \quad (4)$$

To reduce computation, similarity is evaluated only within Ω . Denoting cropping by $\mathcal{C}_\Omega(\cdot)$, the selection objective is

$$k^* = \arg \max_{k \in \{1, \dots, K\}} \text{SSIM}(\mathcal{C}_\Omega(I_{\text{sel}}), \mathcal{C}_\Omega(T_{k,\text{sel}})), \quad (5)$$

and the chosen template is $T^* = T_{k^*}$.

To reduce selection time, a two-stage scheme is used: (i) a cheap preselect score at a smaller max dimension D_{pre} to shortlist K_s candidates and (ii) SSIM refinement computed only on that shortlist. Two suitable preselect scores are as follows:

$$\text{corr}(a, b) = \frac{\langle a - \bar{a}, b - \bar{b} \rangle}{\|a - \bar{a}\|_2 \|b - \bar{b}\|_2 + \epsilon}, \quad (6)$$

$$\text{mad}(a, b) = -\frac{1}{|\Omega|} \sum_{(u,v) \in \Omega} |a(u, v) - b(u, v)|, \quad (7)$$

where a, b are vectorized values in Ω after mean subtraction (corr) or are directly compared (mad). Correlation-style matching is a standard building block for template-based inspection and fast preselection [12,28,67]. Further details and implementation considerations (including complexity, shortlist logic, and caching of preprocessed templates to amortize decoding/resizing across queries) are provided in Appendix A.2.

2.3. Template Mapping and Defect-Likelihood Field

After selecting T^* , a defect-likelihood field $M(u, v)$ is constructed on a processing grid. To control runtime, expensive stages (alignment, SSIM-map, and morphology) may be downsampled by selecting a target long-side processed length D_{proc} and defining an isotropic scale

$$s = \min\left(1, \frac{D_{\text{proc}}}{\max(H_I, W_I, H_T, W_T)}\right). \quad (8)$$

This yields

$$I_s = \mathcal{N}(\downarrow_s(I_{\text{sel}})), \quad T_s = \mathcal{N}(\downarrow_s(T^*)). \quad (9)$$

If needed, the query is resized to match the template grid. The implementation also supports resolution presets and scales morphology/area parameters consistently with the processing scale; see Appendix A.3.

Although the proposed framework can operate without explicit alignment, in practical AOI settings a lightweight global alignment is highly recommended whenever small placement shifts or mild rotations are expected. Because the defect-likelihood field is computed pixel-wise, even minor global misalignment can produce widespread responses that resemble false defects.

Let τ_θ denote a parametric global warp (translation or rigid) with parameters θ , and let the warped query be $I_s \circ \tau_\theta$. The alignment is estimated by maximizing an intensity-based similarity inside the inner region Ω ,

$$\hat{\theta} = \arg \max_{\theta} \rho(\mathcal{M}_\Omega(I_s \circ \tau_\theta), \mathcal{M}_\Omega(T_s)), \quad (10)$$

where ρ is a correlation-like similarity and $\mathcal{M}_\Omega$ denotes inner-region masking, i.e., pixels outside Ω are set to zero. More general geometric matching formulations exist (e.g., via epipolar-geometry constraints), but a lightweight global warp is sufficient under the fixture-controlled imaging assumed in AOI lines [68].

Because defect content can bias the estimated warp, the transform is not applied unconditionally. Instead, the candidate warp is validated using SSIM gain on the inner region,

$$\Delta = \text{SSIM}(\mathcal{M}_\Omega(I_s \circ \tau_{\hat{\theta}}), \mathcal{M}_\Omega(T_s)) \quad (11)$$

$$- \text{SSIM}(\mathcal{M}_\Omega(I_s), \mathcal{M}_\Omega(T_s)). \quad (12)$$

The alignment is accepted if $\Delta \geq \Delta_{\min}$ (a user-controlled safeguard); otherwise it is rejected. Accordingly, the aligned (or unaligned) query is defined as

$$\tilde{I} = \begin{cases} I_s \circ \tau_{\hat{\theta}}, & \Delta \geq \Delta_{\min}, \\ I_s, & \text{otherwise.} \end{cases} \quad (13)$$

This acceptance rule makes alignment a practical robustness aid while limiting over-correction in the presence of true defects.

To prevent outliers outside the central region from compressing the dynamic range used for thresholding, a min–max rescaling is used whose extrema are computed over Ω only. Define $X_{\min}^\Omega = \min_{(i,j) \in \Omega} X(i,j)$ and $X_{\max}^\Omega = \max_{(i,j) \in \Omega} X(i,j)$. The operator $\mathcal{R}_\Omega(X)$ therefore performs min–max normalization using extrema computed over pixels in Ω only, which prevents values outside Ω from compressing contrast in the region used for threshold selection. The stabilizer ϵ ensures numerical safety when $\max_\Omega X \approx \min_\Omega X$.

$$\mathcal{R}_\Omega(X)(u,v) = \begin{cases} \frac{X(u,v) - X_{\min}^\Omega}{X_{\max}^\Omega - X_{\min}^\Omega + \epsilon}, & (u,v) \in \Omega, \\ 0, & (u,v) \notin \Omega. \end{cases} \quad (14)$$

A local SSIM map is computed between $\tilde{I}$ and T_s . For a window centered at (u,v) , let $\mu_{\tilde{I}}, \mu_T$ be local means, let $\sigma_{\tilde{I}}^2, \sigma_T^2$ be local variances, and let $\sigma_{\tilde{I}T}$ be the local covariance. The SSIM map value can be written as

$$S(u,v) = \frac{(2\mu_{\tilde{I}}\mu_T + C_1)(2\sigma_{\tilde{I}T} + C_2)}{(\mu_{\tilde{I}}^2 + \mu_T^2 + C_1)(\sigma_{\tilde{I}}^2 + \sigma_T^2 + C_2)}, \quad (15)$$

where C_1, C_2 stabilize the ratio. This is converted to a structural dissimilarity map,

$$D_{\text{ssim}}(u,v) = 1 - S(u,v). \quad (16)$$

In our implementation, the SSIM map is computed using MATLAB R2023a's SSIM function with explicitly specified parameters (window size/weighting and stabilizing constants) for reproducibility; the exact values are reported in Appendix A.4. An absolute-difference map is also computed

$$D_{\text{abs}}(u, v) = |\tilde{I}(u, v) - T_s(u, v)|. \quad (17)$$

An additive fusion with weight w is used, followed by rescaling for stable thresholding:

$$M_{\text{raw}}(u, v) = D_{\text{ssim}}(u, v) + w \mathcal{R}_{\Omega}(D_{\text{abs}})(u, v), \quad (18)$$

$$M(u, v) = \mathcal{R}_{\Omega}(M_{\text{raw}}(u, v)). \quad (19)$$

In this interpretation, $M(u, v)$ is a defect-likelihood signal: localized defects produce concentrated peaks, while nuisance global mismatch (wrong template) tends to produce broad responses.

2.4. Mask and Box Extraction

Defects are sparse, so a high-quantile threshold is chosen using pixels in Ω only:

$$\theta = Q_q(\{M(u, v) : (u, v) \in \Omega\}), \quad (20)$$

where $q \in (0, 1)$ is a sensitivity parameter.

For a finite set of values $\{z_i\}_{i=1}^n$, the q -quantile Q_q is any value such that at least a fraction q of samples are $\leq Q_q$ and at least a fraction $(1 - q)$ are $\geq Q_q$. Using a high quantile to set θ is robust in this setting because true defects typically occupy only a small fraction of pixels within Ω .

The initial binary mask is

$$Y_0(u, v) = \mathbb{K}[M(u, v) > \theta] \mathbb{K}[(u, v) \in \Omega]. \quad (21)$$

To suppress speckle noise and consolidate defect blobs, a sequence of set operators is applied to Y_0 within the inner region. Let $B(r)$ denote a disk structuring element of radius r , and let $\text{AreaOpen}(\cdot; A_{\min})$ remove connected components smaller than $A_{\min}$. The sequence is

$$Y_1 = \text{AreaOpen}(Y_0; A_{\min}), \quad (22)$$

$$Y_2 = Y_1 \bullet B(r_c), \quad (23)$$

$$Y_3 = Y_2 \oplus B(r_d), \quad (24)$$

$$Y = Y_3 \wedge \mathbb{K}_{\Omega}, \quad (25)$$

where $\bullet$ is morphological closing, $\oplus$ is dilation, and $\wedge$ is pixel-wise AND with the inner-region mask $\mathbb{K}_{\Omega}$.

Connected components of Y are converted to axis-aligned bounding boxes $b = [x \ y \ w \ h]$. Each component is assigned a confidence score using a robust statistic of M inside the component. A stable choice is a high quantile:

$$s_i = Q_{q_s}(\{M(u, v) : (u, v) \in \mathcal{C}_i\}), \quad q_s \in (0, 1). \quad (26)$$

Boxes are sorted by s_i and mapped back to the original resolution if the processing scale $s < 1$. For the exact connected-component scoring rule and supported alternatives

(e.g., quantile/max/mean scoring), see Appendix A.5. Therefore, the final output of this stage is a confidence-ranked set of detections

$$\mathcal{P} = \{(b_i, s_i)\}_{i=1}^N, \quad (27)$$

where each $b_i = [x \ y \ w \ h]$ is an axis-aligned bounding box, and s_i is its confidence score derived from the defect-likeness field. In summary, the mask-and-box stage is controlled by a small set of interpretable parameters: threshold quantile q , minimum area $A_{\min}$, closing radius r_c , dilation radius r_d , and confidence quantile q_s (default $q_s = 0.95$). These parameters jointly control the false-positive/false-negative trade-off and the coarseness of the resulting boxes.

2.5. Algorithm Summary

The proposed framework takes a defective query image and selects a reference template from a small library using a coarse-to-fine strategy (cheap pre-screening followed by SSIM refinement). If enabled, a global alignment is applied with an explicit acceptance test based on SSIM gain on the inner region, limiting transforms biased by defect content. A defect-likeness field is then constructed by fusing an SSIM-derived structural dissimilarity map with a normalized absolute-difference map and applying an Ω -based rescaling. Finally, high-quantile thresholding and simple morphological post-processing yield connected components that are converted into confidence-ranked bounding boxes. Algorithm 1 summarizes the full procedure, while the mathematical operators and data products are summarized in Figure 2.

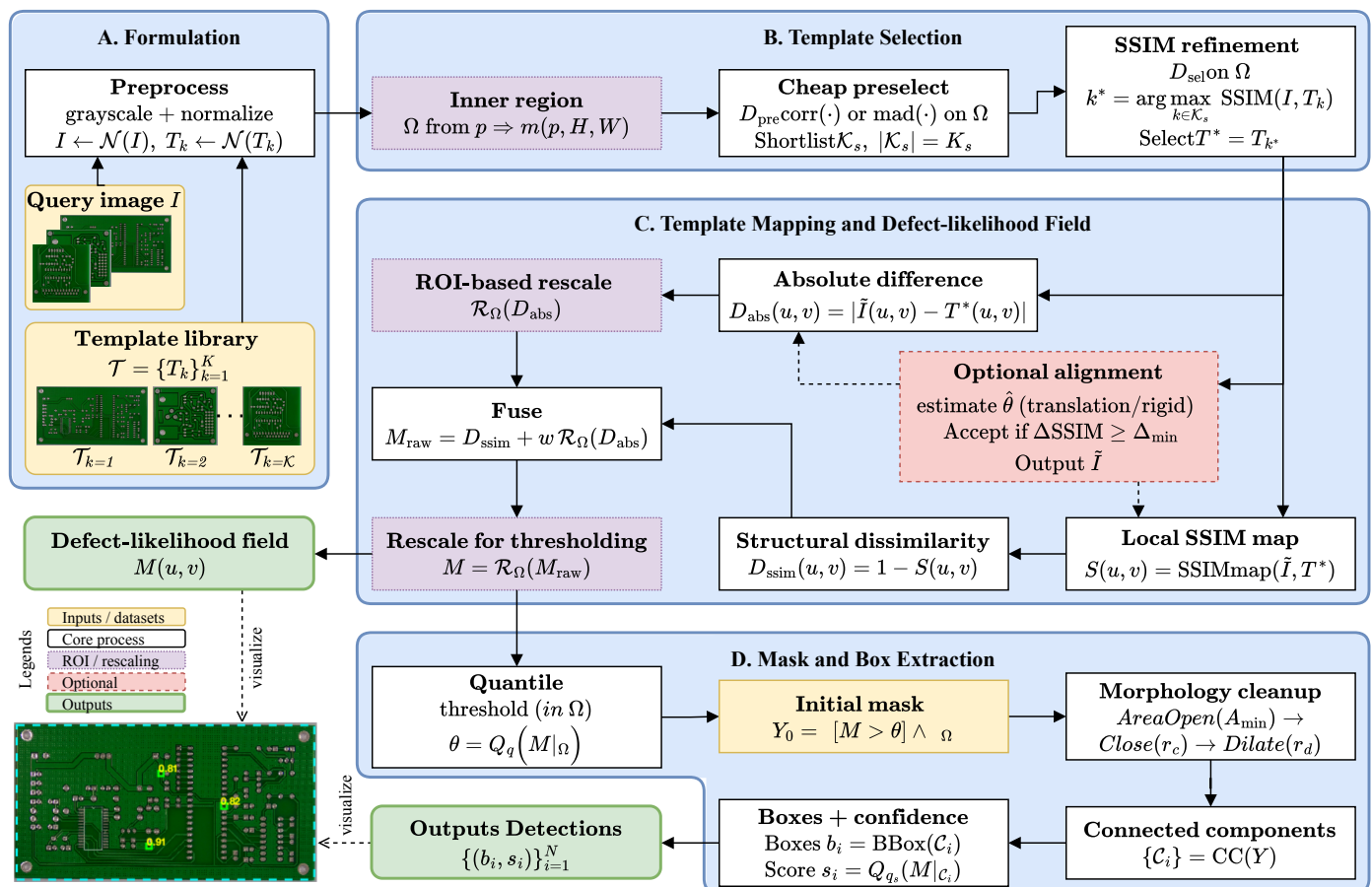


Figure 2. Mathematical overview of the proposed image processing framework.

Algorithm 1 Template selection and defect localization

Require: Query image I , templates $\{T_k\}_{k=1}^K$, parameters $(p, w, q, q_s, A_{\min}, r_c, r_d, D_{\text{pre}}, D_{\text{sel}}, D_{\text{proc}}, K_s, \Delta_{\min})$

Ensure: Detections $\mathcal{P} = \{(b_i, s_i)\}$ and defect-likelihood field M

- 1: Compute central region Ω from area-avoidance ratio p
- 2: Preselect shortlist $\mathcal{K}_s$ (size K_s) using a cheap score at resolution D_{pre}
- 3: Select template T^* by maximizing SSIM on Ω over $k \in \mathcal{K}_s$ at resolution D_{sel}
- 4: Set processing scale s from D_{proc} and build processed images (I_s, T_s)
- 5: Optional alignment: estimate transform $\hat{\theta}$ and compute SSIM gain Δ on Ω ; accept if $\Delta \geq \Delta_{\min}$
- 6: Define aligned query $\tilde{I}$ (either warped I_s or I_s)
- 7: Compute SSIM map $S(u, v)$ between $\tilde{I}$ and T_s
- 8: Compute dissimilarity maps $D_{\text{ssim}}(u, v) = 1 - S(u, v)$ and $D_{\text{abs}}(u, v) = |\tilde{I}(u, v) - T_s(u, v)|$
- 9: Fuse and rescale to obtain defect-likelihood field M
- 10: Threshold: $\theta = Q_q(\{M(u, v) : (u, v) \in \Omega\})$
- 11: Initial mask: $Y_0 = \mathbb{K}[M > \theta] \wedge \mathbb{K}_{\Omega}$
- 12: Cleanup: $Y = \text{AreaOpen}(Y_0; A_{\min}) \bullet B(r_c) \oplus B(r_d)$
- 13: Connected components $\{\mathcal{C}_i\} \rightarrow$ boxes $\{b_i\}$ and scores $s_i = Q_{q_s}(\{M(u, v) : (u, v) \in \mathcal{C}_i\})$
- 14: Map boxes back to original scale (if $s < 1$)

Computational Complexity

Let K be the number of available templates and $K_s \ll K$ be the shortlist size after preselection. Let $D_{\text{pre}}, D_{\text{sel}}, D_{\text{proc}}$ denote the maximum long-side resolutions used for (i) the cheap preselect score, (ii) SSIM refinement for template selection, and (iii) the downstream processing grid, respectively. Using the common square-grid proxy, the number of evaluated pixels scales as $\mathcal{O}(D^2)$ at resolution D . Restricting computations to the central region Ω (via ROI cropping/masking) primarily changes constants. Per query image, the coarse-to-fine template selection cost is

$$\mathcal{O}(K D_{\text{pre}}^2 + K_s D_{\text{sel}}^2), \quad (28)$$

where the first term corresponds to evaluating a cheap preselect score over all templates at D_{pre} , and the second term corresponds to SSIM refinement over the shortlist at D_{sel} (dominant when enabled). After selecting T^* , the downstream stages operate on the processing grid at D_{proc} : (i) optional global alignment uses a correlation-based registration objective (translation/rigid) and in practice scales near-linearly in the number of processed pixels (often with FFT-accelerated correlation), (ii) the SSIM map uses a fixed window size (constant window operations per pixel), and (iii) difference-map construction, Ω -based rescaling, quantile thresholding, morphology with fixed structuring-element radii, and connected-components labeling are all $\mathcal{O}(D_{\text{proc}}^2)$ up to small multiplicative constants. Overall, the per-image runtime scales as

$$\mathcal{O}(K D_{\text{pre}}^2 + K_s D_{\text{sel}}^2 + D_{\text{proc}}^2), \quad (29)$$

and the dataset runtime is linear in the number of evaluated images. In practice, template preprocessing (grayscale/normalization/resizing) can be cached, and choosing $D_{\text{pre}} \ll D_{\text{sel}}$ with $K_s \ll K$ keeps selection tractable even for moderate template libraries.

2.6. Experimental Setup and Evaluation Protocol

Experiments use the PKU-Market-PCB dataset [3]. Defect classes are used only for class-wise reporting; the localization pipeline itself is class-agnostic across defect categories (while assuming a valid defect-free reference template for the same layout/revision). The

dataset folder organization (templates, defective images, and Pascal Visual Object Classes (VOC) annotations) is described in Appendix A.6. The proposed method is implemented in MATLAB R2023a, and experiments are run on an Apple M1 processor with 8GB RAM. The implementation configuration and hyperparameters are reported in Appendix A.7

2.6.1. Dataset

Although the proposed method employs training-free image processing that produces an explicit defect-likelihood field $M(u, v)$ and interpretable intermediate maps, the final outputs are evaluated using standard box-level localization metrics, which facilitates comparison with established benchmarks in the AOI and object-detection literature. For broader context on automated optical inspection in electronics manufacturing and recent perspectives on PCB defect detection, see the comprehensive surveys in [69,70]. Table 1 summarizes the evaluated dataset, including image resolution and annotation statistics.

Table 1. Dataset summary.

Item	Value
Defective images (run)	693
Evaluated images (with GT XML)	693
Defect-free templates	10
Defect classes	6
GT objects (total)	2953
Median resolution ($W \times H$ in pixels)	2868×2159
Resolution range ($W \times H$ in pixels)	$2240\text{--}3056 \times 1586\text{--}2530$

2.6.2. Detections and Confidence

For each query image, the method outputs a set of axis-aligned detections $\mathcal{P} = \{(b_i, s_i)\}_{i=1}^{N_p}$, where each $b_i = [x \ y \ w \ h]$ is the bounding box of a connected component extracted from a thresholded version of $M(u, v)$ (Section 2.4). Each detection is assigned a confidence score s_i computed as a robust statistic of M values inside the component (a high-quantile score is used so that a higher s_i indicates stronger evidence of a localized peak). For the exact component-to-box confidence definition used in the implementation, see Appendix A.5.

2.6.3. Ground Truth and Intersection-over-Union (IoU) Matching

Ground-truth annotations are axis-aligned Pascal VOC boxes. For each query image, let $\mathcal{G} = \{g_j\}_{j=1}^{N_G}$ denote the set of ground-truth boxes. Detections are matched to ground truth using a greedy one-to-one assignment within each image: detections are processed in descending confidence order, and a detection b_i is matched to an as-yet-unmatched g_j if $\text{IoU}(b_i, g_j) \geq \tau_{\text{IoU}}$ (each detection matches at most one ground-truth box and vice versa). Unmatched detections are counted as false positives, and unmatched ground-truth boxes are counted as false negatives.

Unless noted otherwise, $\tau_{\text{IoU}} = 0.10$ is used because detections are derived from thresholded connected components rather than tight box regression.

2.6.4. Fixed Operating Point (Counts and F1)

Given a score threshold t , detections with $s_i \geq t$ are retained and dataset-level counts (after per-image one-to-one matching) are computed: true positives $\text{TP}(t)$, false positives $\text{FP}(t)$, and false negatives $\text{FN}(t)$. Precision, recall, and F1 are

$$\text{Precision}(t) = \frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \epsilon}, \quad (30)$$

$$\text{Recall}(t) = \frac{\text{TP}(t)}{\text{TP}(t) + \text{FN}(t) + \epsilon}, \quad (31)$$

$$\text{F1}(t) = \frac{2 \text{Precision}(t) \text{Recall}(t)}{\text{Precision}(t) + \text{Recall}(t) + \epsilon}. \quad (32)$$

where $\epsilon > 0$ is a small numerical stabilizer (e.g., 10^{-12}) used only to avoid division-by-zero in degenerate cases (e.g., when $\text{TP}(t) + \text{FP}(t) = 0$); it does not affect results when denominators are nonzero.

For completeness, we also report a detection accuracy that does not require defining true negatives (TN), which are not well-defined for variable-length detection outputs:

$$\text{Accuracy}(t) = \frac{\text{TP}(t)}{\text{TP}(t) + \text{FP}(t) + \text{FN}(t) + \epsilon}. \quad (33)$$

2.6.5. Avoiding Threshold Selection on the Reported Set

To avoid optimistic bias from selecting t on the same set used for reporting, operating-point metrics are reported using K -fold cross-validation: in each fold, the score threshold t^* is chosen to maximize $\text{F1}(t)$ on the calibration split (the union of the other $K - 1$ folds), and precision/recall/F1 are then computed on the held-out fold using this fixed threshold. This work reports the mean $\pm$ standard deviation across folds. For continuity with the original submission and to aid interpretation of class-wise behavior, this also includes a reference operating point obtained by selecting t^* on the full evaluated set; this reference is explicitly labeled and is not used as the primary estimate of generalization performance. Implementation details for the cross-validated operating point selection are provided in Appendix B.1.

2.6.6. Threshold-Sweep Metrics (PR/AP, AP-Versus-IoU, and FROC)

To summarize performance across confidence thresholds, the precision–recall curve is constructed by ranking all detections across the dataset by confidence s_i and sweeping down this global ranked list. True/false positives are determined using the matching protocol described above at the specified τ_{IoU} [71,72]. Average precision (AP) is computed from this PR curve using a precision envelope and piecewise integration over recall (the VOC07 11-point approximation is disabled in the evaluation script).

To quantify sensitivity to localization stringency, AP is also computed across multiple IoU thresholds. In particular, (i) a custom sweep $\tau_{\text{IoU}} \in \{0.10, 0.20, \dots, 0.90\}$ and (ii) a COCO-style sweep $\tau_{\text{IoU}} \in \{0.50, 0.55, \dots, 0.95\}$ are reported. When multiple IoU thresholds are considered, mean Average Precision mAP is reported as the mean AP across the specified thresholds.

For operational analysis, free-response receiver operating characteristic (FROC) curves are reported by sweeping score thresholds. For a threshold t , let N_{img} be the number of evaluated images, let N_{GT} be the total number of ground-truth boxes, and let $N_{\text{img}}^{\text{GT}}$ be the number of images containing at least one ground-truth box. The following quantities are defined:

$$\text{FP/image}(t) = \frac{\text{FP}(t)}{N_{\text{img}} + \epsilon}, \quad (34)$$

$$\text{Recall}_{\text{obj}}(t) = \frac{\text{TP}(t)}{N_{\text{GT}} + \epsilon}, \quad (35)$$

$$\text{Recall}_{\text{img}}(t) = \frac{N_{\text{img}}^{\text{TP}}(t)}{N_{\text{img}}^{\text{GT}} + \epsilon}, \quad (36)$$

where $N_{\text{img}}^{\text{TP}}(t)$ is the number of ground-truth-containing images for which at least one detection is a matched true positive (TP) at threshold t . In this paper, FROC plots and operating-point tables use object-level recall $\text{Recall}_{\text{obj}}(t)$ versus $\text{FP/image}(t)$; image-level recall is included for completeness. The exported threshold grid and image-level recall computation used to generate the FROC bundle are described in Appendix B.2.

At the dataset level, $N_{\text{GT}} = \text{TP}(t) + \text{FN}(t)$; therefore, $\text{Recall}(t)$ and $\text{Recall}_{\text{obj}}(t)$ are numerically equivalent. Accordingly, $\text{Recall}_{\text{obj}}(t)$ is used in the FROC definition to emphasize the object-level nature of the curve.

For completeness, explicit definitions for IoU on $[x \ y \ w \ h]$ boxes and the AP computation used in our evaluation are provided in Appendices B.3 and B.4.

3. Experimental Results and Analysis

This section reports qualitative examples and quantitative localization performance under the experimental setup and evaluation protocol described in the previous subsection. The method produces confidence-ranked detections $\mathcal{P} = \{(b_i, s_i)\}$ from an explicit defect-likelihood field $M(u, v)$; therefore, results are presented both at a fixed operating point (counts and precision/recall/F1) and across confidence thresholds (precision–recall, AP, and FROC).

All quantitative results in this section are reported on 693 evaluated query images under the adopted IoU matching protocol (Section 2.6). Threshold-sweep curves (PR/AP/FROC) are computed on the full set. For fixed-threshold operating-point metrics, this work reports K -fold cross-validated results (mean $\pm$ std) to avoid selecting thresholds on the same set used for reporting.

For reference and continuity with prior results, the manuscript also reports descriptive counts at a reference best-F1 threshold selected on the full evaluated set (explicitly labeled as reference).

3.1. Qualitative Results and Failure Modes

Figure 3 shows representative TP cases where the proposed mapping produces localized peaks in $M(u, v)$ that translate into axis-aligned bounding boxes overlapping the annotated defects. While the boxes are not optimized to tightly fit the defect boundary (they are derived from thresholded connected components), they consistently localize the defect region and provide a monotonic confidence score s_i for ranking.

Figure 4 shows representative false positive (FP) cases. In practice, FPs tend to arise from nuisance variability that produces structured responses in the SSIM/difference maps, such as residual global misalignment, illumination drift, or local texture/reflectance changes that are not labeled as defects. A qualitative review of these cases suggests that many FPs stem from two primary sources: (i) structured nuisance variability such as residual global misalignment, illumination drift [73], or local texture changes that are not true defects, and (ii) actual defects that are present in the images but are missing from the corresponding Pascal VOC XML annotation files. Because this work operates strictly under the provided ground-truth labels without resorting to relabeling or manual verification,

all reported metrics faithfully reflect the adopted box-matching protocol and evaluation set. The presence of unlabeled defects in the FP set underscores the importance of careful annotation in benchmark datasets and suggests that true recall may be higher than reported. These observations underscore the value of the optional alignment safeguard and careful template selection, which are most influential in reducing nuisance-driven artifacts. Moreover, the confidence-ranked outputs and FROC curves (Section 3.6) enable operational threshold selection based on a tolerable false-alarm budget in production, allowing practitioners to adjust operating points according to their acceptable FP/image rate and trade recall for lower false-alarm burden when necessary.

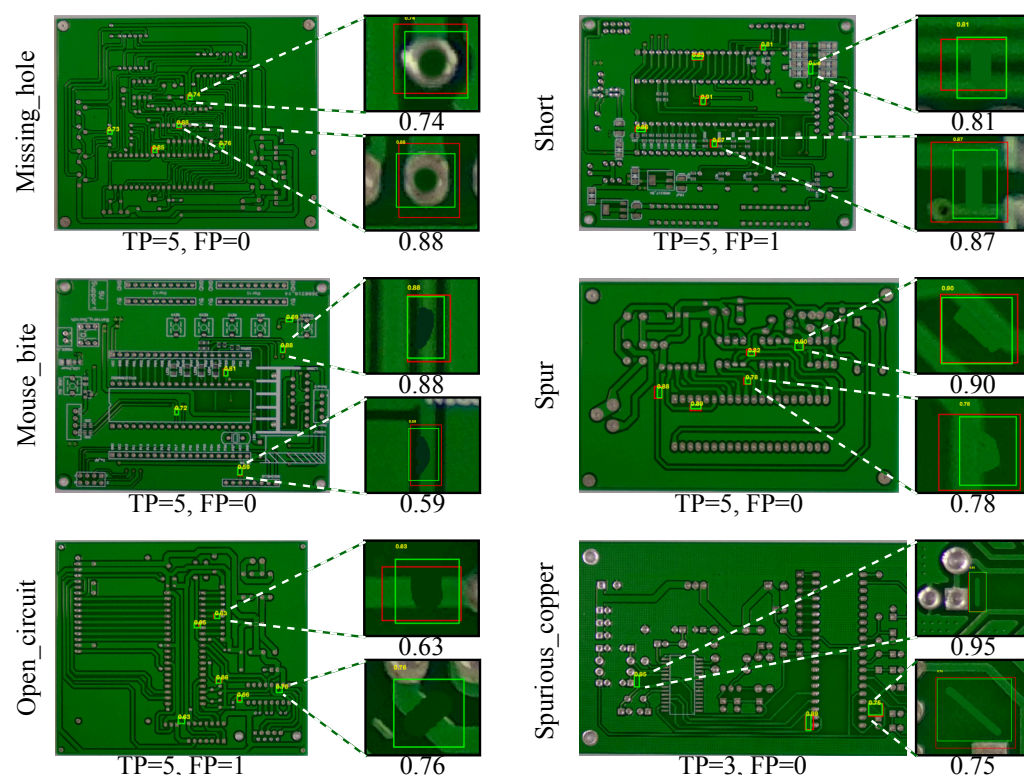


Figure 3. Example true positive cases (red: ground truth; green: proposed detection).

Figure 5 provides representative visual evidence for the observed Spur false negatives (all four FNs in the per-class breakdown). These cases are consistent with thin/fragmented structures producing a weaker or spatially discontinuous response under the current high-quantile thresholding and morphology settings, which can suppress small components or yield boxes that fail the one-to-one IoU matching criterion. Because the pipeline is deterministic and controlled by interpretable parameters ($q, A_{\min}, r_c, r_d, q_s$), this behavior reflects an explicit operating trade-off that can be adjusted (as also summarized by the FROC curve in Section 3.6).

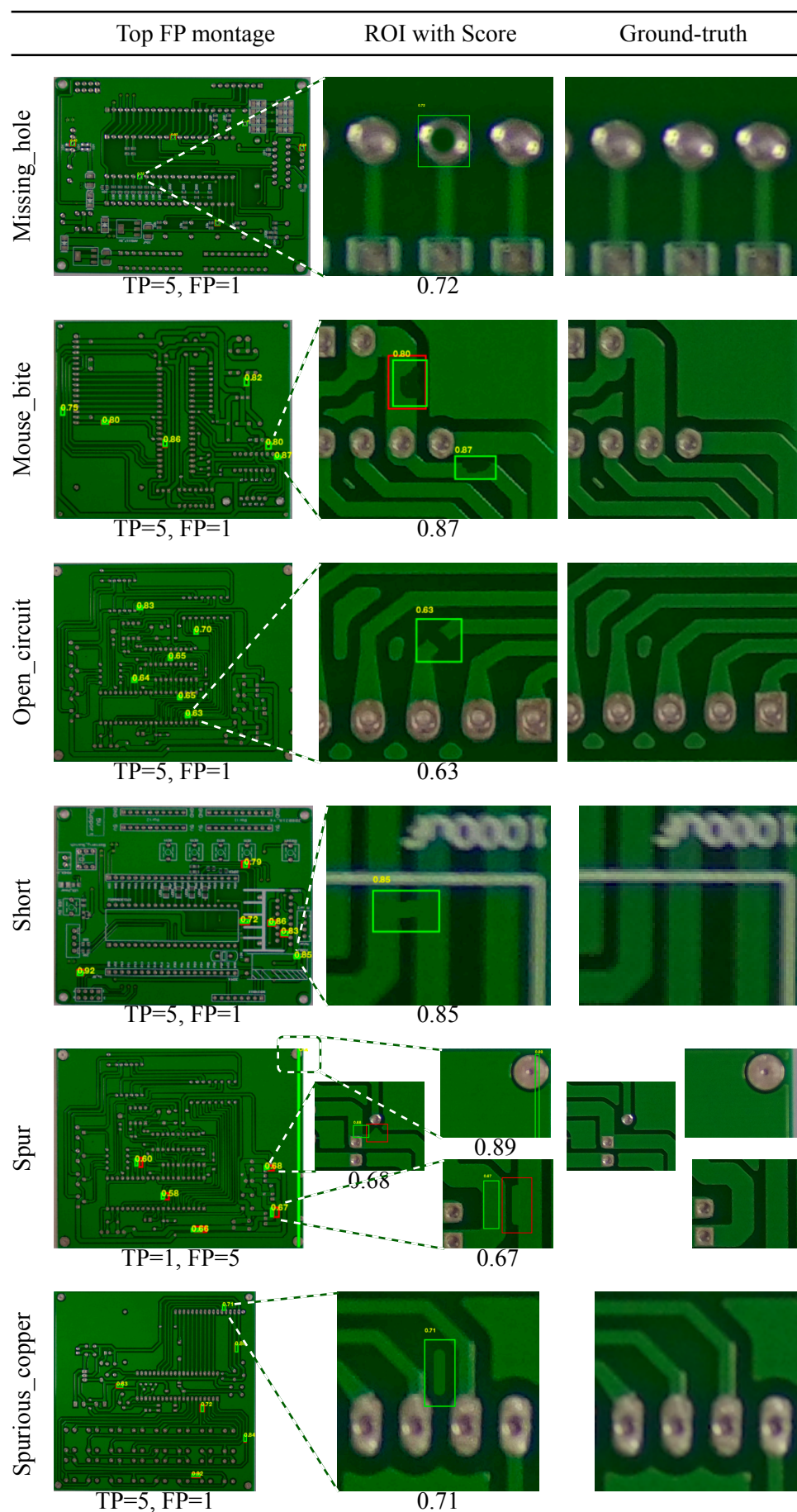


Figure 4. Example false positive cases (red: ground truth; green: proposed detection).

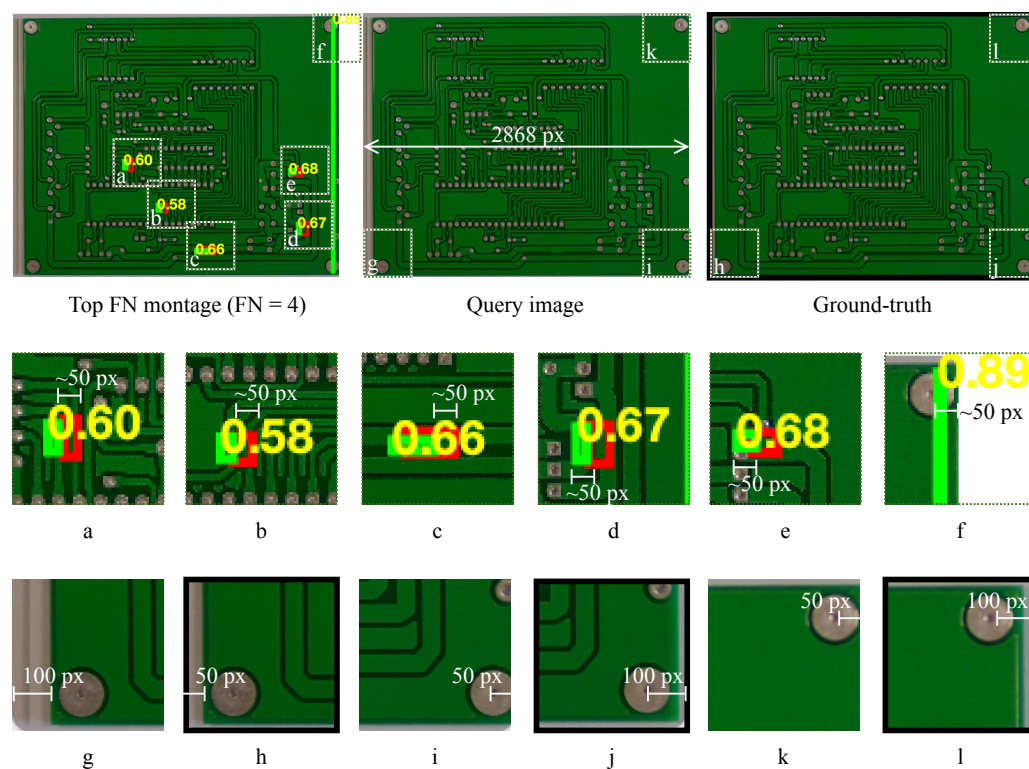


Figure 5. Example false negative case (red: ground truth; green: proposed detection; (a–f) are corresponding crop regions of the Top FN montage; (g,i,k) are crop regions of the query image; (h,j,l) are crop regions of the reference ground-truth image).

3.2. Template Selection Quality and Practical Failure Causes

The exported per-image diagnostics show that template selection is typically highly stable (template SSIM values are near 1.0 for the majority of images), but rare outliers can occur, which are consistent with occasional template mismatch or nontrivial capture variation. On the 693 evaluated images, the per-image template SSIM has median ≈ 0.9991 ($p5 \approx 0.9976$), while a small number of outliers fall below 0.99 (4/693 images) and one extreme case reaches as low as ≈ 0.4967 . These outliers are practically important because low similarity can inflate structured responses in $M(u, v)$ and therefore increase false positives under fixed post-processing parameters.

3.3. Quantitative Summary at the Adopted Operating Point

Table 2 reports per-class detection counts (TP/FP/FN) and the resulting precision/recall/F1 under the adopted IoU matching protocol at a reference best-F1 operating point selected on the full evaluated set. Overall, the method achieves TP = 2949, FP = 103, FN = 4, yielding Precision = 0.9663, Recall = 0.9987, and F1 = 0.9822 at this reference threshold. Cross-validated operating-point results (mean $\pm$ std across folds), which avoid threshold selection on the reported set, are provided in Table 3. In this revision, we treat these cross-validated operating-point results as the primary generalization-oriented summary, while the full-set best-F1 operating point is retained only as a clearly labeled reference point.

Across defect types, performance is broadly consistent. The main deviations are informative: all four missed defects (FN = 4) occur in the Spur class, suggesting a small subset of cases where the defect response is weaker or more fragmented under the current post-processing parameters. This behavior is expected in a deterministic pipeline and can be addressed by adjusting the interpretable parameters ($q, A_{\min}, r_c, r_d, q_s$) for the desired false-positive/false-negative trade-off.

Table 2. Reference operating-point detection counts and metrics (threshold selected on the full evaluated set).

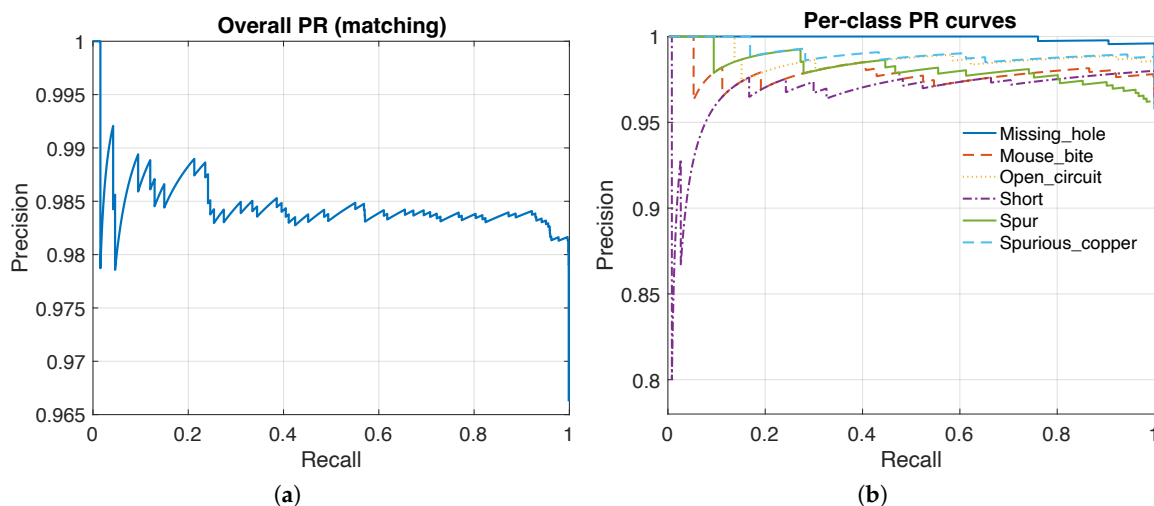
Class	TP	FP	FN	Prec.	Rec.	F1	Acc.
Missing_hole	497	22	0	0.9576	1.0000	0.9784	0.9576
Mouse_bite	492	20	0	0.9609	1.0000	0.9801	0.9609
Open_circuit	482	8	0	0.9837	1.0000	0.9918	0.9837
Short	491	18	0	0.9646	1.0000	0.9820	0.9646
Spur	484	19	4	0.9622	0.9918	0.9768	0.9546
Spurious_copper	503	16	0	0.9692	1.0000	0.9843	0.9692
Overall	2949	103	4	0.9663	0.9987	0.9822	0.9650

Table 3. Cross-validated operating-point metrics (mean $\pm$ std across folds).

Metric	Mean $\pm$ Std	Notes
Threshold	0.322 $\pm$ 0.053	K = 5, IoU = 0.10
Precision	0.980 $\pm$ 0.008	
Recall	0.997 $\pm$ 0.003	
F1	0.988 $\pm$ 0.004	
FP per image	0.088 $\pm$ 0.035	

3.4. Precision–Recall Characteristics Across Confidence Thresholds

Because detections are confidence-ranked by s_i , varying the acceptance threshold induces a precision–recall (PR) trade-off. Figure 6 reports the overall PR curve and the per-class PR curves, summarizing how precision changes as recall increases and which defect types degrade earlier as the threshold is lowered.

**Figure 6.** Precision–recall curves under the IoU matching protocol. (a) Overall PR curve. (b) Per-class PR curves.

In AOI practice, this view is useful because the operating point is chosen based on the acceptable false-alarm burden in production. In addition, the PR curves validate that the proposed confidence score is informative: a meaningful ranking yields a PR curve that maintains high precision over a wide recall range.

3.5. Sensitivity to Matching Stringency (AP Versus IoU)

Figure 7 reports average precision as a function of the IoU threshold. As the IoU requirement becomes stricter, AP decreases because the proposed detections are derived

from thresholded map components and are not explicitly optimized to tightly fit the ground-truth boxes. In particular, because our boxes are obtained by extracting connected components and enclosing them with axis-aligned rectangles, $\tau_{\text{IoU}} = 0.10$ should be interpreted as a permissive matching criterion for region-level localization (component fragmentation/merging can yield looser boxes even when the detected region overlaps the defect). Consequently, the decline in AP at higher IoU thresholds primarily reflects box tightness/placement mismatch rather than an inability to detect defect regions. Quantitatively, the exported AP–IoU sweep shows high AP at the adopted low-stringency regime (overall AP ≈ 0.984 at IoU = 0.10) but a pronounced decline at stricter thresholds (overall AP ≈ 0.572 at IoU = 0.50), approaching near-zero at very strict IoU (e.g., ≥ 0.80). This plot is therefore included to transparently characterize localization-stringency sensitivity rather than to claim pixel-tight bounding-box regression.

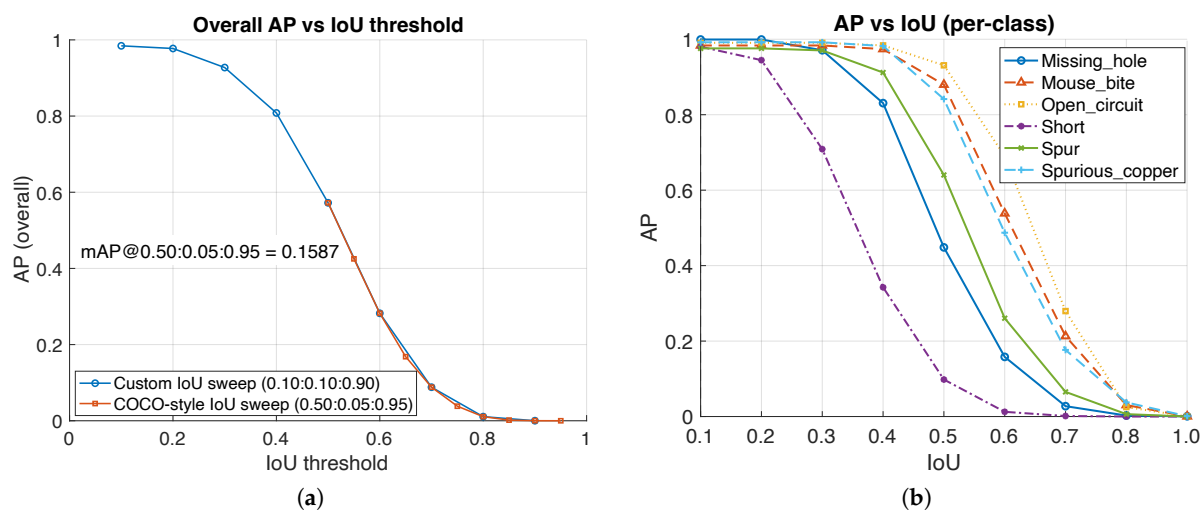


Figure 7. AP versus IoU threshold. (a) Overall AP versus IoU threshold, including custom IoU sweep and optional COCO-style mAP summary. (b) Per-class AP versus IoU threshold.

For completeness, Table 4 first summarizes representative overall AP values at selected IoU thresholds, and Figure 7 then visualizes the full AP-versus-IoU behavior, including both the overall curves (custom and COCO-style sweeps) and the per-class curves.

Table 4. AP at representative IoU thresholds (overall).

AP	0.05	0.10	0.20	0.30	0.40	0.50
IoU	0.985	0.984	0.977	0.927	0.808	0.572

3.6. False-Alarm Rate Versus Recall (Or FROC)

For deployment, a key question is how many false alarms must be tolerated to achieve a target recall. Figure 8 reports FROC curves, which summarize recall as a function of the false positive rate (e.g., false positives per image) as the confidence threshold varies. This view complements PR curves by emphasizing the operational cost of additional recall.

At the adopted operating point (Table 2), the system attains Recall = 0.9987 with FP/image ≈ 0.149 . In practice, Figure 8 is used to choose an operating threshold based on an allowed FP/image budget; the exported FROC curve enables selecting alternative thresholds to trade a small recall reduction for a lower false-alarm burden.

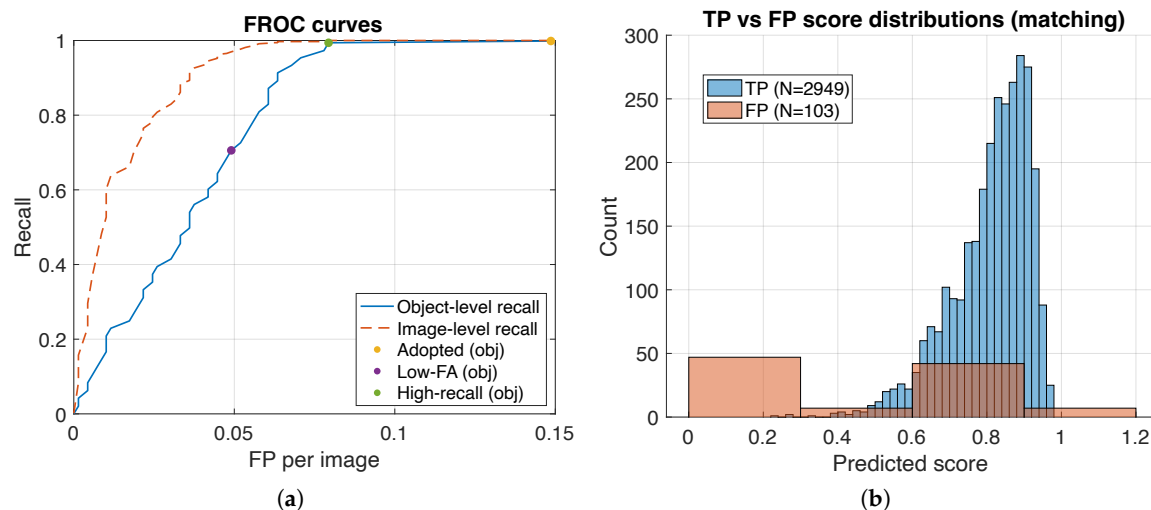


Figure 8. Operational views for threshold selection. (a) Free-response receiver operating characteristic FROC curves. (b) True positive and false positive score distributions histograms.

3.7. Score Separability of True and False Detections

Figure 8 compares the confidence-score distributions of true positives and false positives. A clear separation indicates that the confidence score s_i provides a meaningful ranking signal, enabling threshold selection to trade precision for recall. Overlap between TP and FP score distributions highlights ambiguous cases (e.g., residual misalignment or illumination-driven artifacts) that can be addressed by improving template selection, enabling alignment when appropriate, or adjusting the mask-and-box parameters.

3.8. Quantitative Runtime Breakdown

Beyond asymptotic scaling, an empirical runtime breakdown of the reference implementation by stage is reported in Table 5.

Table 5. Runtime breakdown by stage (reference implementation).

Stage	Time (s/Image)
Template selection (total)	0.242
Preselect	0.109
SSIM refine	0.033
Localization (total)	0.793
Preprocess	0.043
Resize	0.059
Align	0.368
Score map	0.048
Mask	0.073
Boxes	0.004
Total	1.035

4. Discussion

The proposed framework targets practical AOI where defect-free references are available and decisions must be reliable, auditable, and deployable without a training lifecycle. Operating-point metrics are summarized using K -fold cross-validation to avoid selecting thresholds on the reported set (Table 3), while a reference best-F1 operating point on the full set is retained for descriptive continuity (Table 2). Under the adopted IoU matching protocol, confidence-ranked outputs support operational interpretation via PR/FROC sweeps (Figures 6 and 8) and IoU-sensitivity via AP-versus-IoU analysis (Figure 7).

4.1. Why the Method Works in Practice

Robustness comes from explicitly controlling nuisance variability and using structure-aware comparison: template selection reduces systematic mismatch, optional alignment is gated by SSIM gain to avoid defect-driven over-correction, and the fused defect-likelihood field combines SSIM-derived structural dissimilarity with normalized absolute difference to handle both structural and radiometric changes (e.g., illumination drift). Together, these choices yield a confidence score that is informative for ranking, consistent with the PR and FROC behavior.

4.2. Class-Wise Behavior and Failure Modes

Per-class performance is broadly consistent; the main deviation is that the only missed defects ($FN = 4$) occur in the Spur class (Table 2), where responses can be weaker or more fragmented under the current post-processing. This is expected in a deterministic connected-components-to-box stage (threshold/area/morphology can suppress fine structures), and recall can be increased by adjusting $(q, A_{\min}, r_c, r_d, q_s)$ with the resulting FP/image quantified by the FROC curve.

4.3. Operating-Point Selection and Localization Stringency

Operating-point metrics are reported using K -fold cross-validation to avoid selecting thresholds on the same set used for reporting (Section 2.6). In practice, deployment typically selects thresholds based on an allowable false-alarm budget; Figure 8 and Table 6 make this trade-off explicit. Accordingly, FROC (object-level recall versus FP/image) is recommended as the primary operational view, with PR as a complementary precision summary. The adopted matching threshold $\tau_{IoU} = 0.10$ reflects that detections are derived from thresholded connected components rather than tight box regression; consequently, AP decreases as IoU becomes stricter (Figure 7), indicating a design choice favoring reliable region localization for QC over pixel-tight boxes. To avoid over-interpreting performance under low-stringency matching, the Results section summarizes AP at representative moderate IoU thresholds and provides the full AP-versus-IoU sweep. If tighter boxes are required, a lightweight post-processing refinement (e.g., contour-based tightening) can be added without changing the detection ranking.

Table 6. Example operating points derived from the exported FROC curve (object-level recall versus FP/image).

Operating Point	FP/Image	Recall	Notes
Adopted threshold	0.149	0.9987	From Table 2
Low-false-alarm	0.049	0.7057	scoreThr = 0.7712 (max recall with $FP/image \leq 0.05$)
High-recall	0.079	0.9939	scoreThr = 0.4441 (min FP/image with recall ≥ 0.99)

4.4. Limitations and Practical Considerations

The method assumes defect-free templates of the same PCB layout/revision, which is realistic in many manufacturing settings and helps resolve defect-versus-intent ambiguity; however, this assumption also defines the primary limitations:

- Layout/revision mismatch or wrong template: if the selected template does not correspond to the same board revision (or if no valid template exists), reference comparison can yield structured false-positive regions that mimic defects. Coarse-to-fine template selection mitigates mild mismatch, but it cannot correct true design changes; in such

cases, the correct remedy is updating the template library (or enforcing recipe-based template selection in fixed-product lines).

- Perspective/projective distortion and nonrigid deformation: the SSIM-gated alignment improves robustness to small global shifts/rotations, but larger viewpoint/perspective changes remain out of scope. When the imaging geometry is not fixture-controlled, residual misregistration can dominate difference/SSIM maps and reduce localization quality.
- Heavy illumination non-uniformity and reflectance drift: normalization and the fused SSIM/difference design reduce sensitivity to global brightness drift; however, strong spatially varying illumination (e.g., vignetting, hotspots, and specular highlights) can still introduce structured responses in the defect-likelihood field and increase false positives.
- ROI-based statistics and border effects: evaluating thresholds and similarity on the central region Ω improves robustness to border artifacts, but defects near the excluded border can have reduced influence on thresholding statistics. If border defects are critical, the ROI margin should be reduced (with the expected trade-off of higher nuisance sensitivity).
- Component-derived boxes are not tight boundaries: detections are produced from thresholded connected components, so bounding boxes are intended for reliable region localization rather than pixel-tight boundaries. This is why AP drops at stricter IoU thresholds (Figure 7); if tighter boxes are required, a lightweight contour-based tightening step can be added without changing the detection ranking.

Finally, this revision does not include trained deep learning baselines; because training-based results can be sensitive to split definition and training/engineering choices, we treat such comparisons as out of scope here and instead report explicit PR/AP/FROC and IoU-sensitivity analyses under a fully specified protocol.

4.5. Future Work

Future work includes the following: (i) automatic threshold recommendation for a specified FP/image budget from the exported FROC curve; (ii) lightweight box tightening to improve IoU without retraining; (iii) broader alignment models (e.g., rigid/affine) while retaining an SSIM-gain acceptance gate; (iv) per-template caching and template-library management for scale; and (v) computational/implementation optimizations (early exits, efficient SSIM/difference computation, downsampling-aware parameter scaling, and fixed-point/embedded-friendly variants) with evaluation on single-board computers or microcontrollers.

5. Conclusions

This work addresses defect-intent ambiguity in PCB AOI by presenting a training-free, reference-based defect localization framework designed for deployments where interpretability and controllable false-alarm rates are critical. The method selects a compatible defect-free reference via coarse-to-fine matching, applies global alignment only when it increases SSIM (to avoid defect-driven over-correction), and forms a defect-likelihood field by fusing SSIM-derived structural dissimilarity with normalized absolute differences; confidence-ranked detections are then produced via connected-component extraction.

To support deployment-oriented decision making, we report precision-recall and FROC curves rather than advocating a single universal threshold. At the adopted operating point, the method achieves 0.9987 recall at 0.149 FP/image, and operating-point metrics are additionally reported using K -fold calibration/test thresholding to avoid selecting thresholds on the same set used for reporting (with a full-set best-F1 threshold retained

only as a clearly labeled reference point). Under the adopted box-matching protocol with $\tau_{IoU} = 0.10$, average precision reaches 0.984, and AP-versus-IoU sweeps transparently characterize sensitivity to stricter localization requirements. The primary limitation is the need for defect-free templates of the same PCB layout/revision; within this setting, the approach provides a reproducible and auditable alternative to training-based inspection by flagging only statistically and structurally supported deviations for review.

Author Contributions: Conceptualization, investigation, writing—review and editing, S.S. and W.N.; methodology, software, formal analysis, writing—original draft presentation, visualization, W.N.; supervision, project administration, funding acquisition, S.S.; validation, S.S., W.N., Z.L. and P.S.; resources, data curation, Z.L. and P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code supporting the findings of this study will be finalized and released publicly upon acceptance/publication at: <https://github.com/nonsakhoo/PCB-defect-localization> (accessed on 24 January 2026). During peer review, an unofficial snapshot is made available for reviewers via the same repository URL.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Acc.	Accuracy
AI	Artificial Intelligence
AOI	Automated Optical Inspection
AP	Average Precision
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
F1	F1 Score
FN	False Negative
FP	False Positive
FROC	Free-response Receiver Operating Characteristic
GT	Ground Truth
IoU	Intersection over Union
mAP	mean Average Precision
PCB	Printed Circuit Board
PI	Power Integrity
PR	Precision–Recall
QC	Quality Control
RAM	Random-Access Memory
ROI	Region of Interest
SI	Signal Integrity
SSIM	Structural Similarity Index
TN	True Negative
TP	True Positive
VOC	Visual Object Classes
XML	Extensible Markup Language

Appendix A. Implementation Details and Parameter Scaling

This appendix describes implementation-faithful computations that affect reproducibility of the proposed method, including template preselection, ROI discretization, component-to-box confidence scoring, and processing-scale parameter scaling.

Appendix A.1. Discrete ROI Construction from the Area-Avoidance Ratio

The manuscript derives a continuous border thickness m from the area-avoidance ratio p . The reference implementation applies an explicit discretization and safety clamp.

Given image size $W \times H$ and $p \in [0, 0.95]$, define the discriminant

$$\Delta = (W + H)^2 - 4pWH, \quad \Delta \leftarrow \max(0, \Delta), \quad (\text{A1})$$

and compute the smaller root

$$m = \frac{(W + H) - \sqrt{\Delta}}{4}. \quad (\text{A2})$$

The integer padding is then

$$\text{padPx} = \lfloor \max(0, m) \rfloor, \quad (\text{A3})$$

followed by a clamp to prevent degenerate inner regions,

$$\text{padPx} \leftarrow \min\left(\text{padPx}, \left\lfloor \frac{\min(W, H)}{2} \right\rfloor - 1\right). \quad (\text{A4})$$

The resulting ROI is the axis-aligned inner rectangle

$$\Omega = \{(u, v) : 1 + \text{padPx} \leq u \leq W - \text{padPx}, 1 + \text{padPx} \leq v \leq H - \text{padPx}\}, \quad (\text{A5})$$

and pixels outside Ω are excluded from thresholding and post-processing.

Appendix A.2. Template Preselection Score and Shortlist Logic

To accelerate template selection, the implementation uses a cheap preselection score computed for all templates at a small working resolution and then refines SSIM only on a shortlist.

In the reference implementation, the preselection images are computed at max long-side D_{pre} and are restricted to the same central region Ω used elsewhere: pixels outside Ω are either set to zero (masking) or excluded by cropping, depending on whether SSIM is evaluated by masking or ROI-cropping.

Let I_{pre} and $T_{k, \text{pre}}$ denote the preprocessed (grayscale, normalized, and resized) query and template images at the preselection resolution. Two preselection scores are supported:

$$\text{mad}(I_{\text{pre}}, T_{k, \text{pre}}) = -\frac{1}{n} \sum_{p=1}^n |I_{\text{pre}}(p) - T_{k, \text{pre}}(p)|, \quad (\text{A6})$$

$$\text{corr}(I_{\text{pre}}, T_{k, \text{pre}}) = \frac{\langle I_{\text{pre}} - \mu_I, T_{k, \text{pre}} - \mu_T \rangle}{\|I_{\text{pre}} - \mu_I\|_2 \|T_{k, \text{pre}} - \mu_T\|_2 + \epsilon'}, \quad (\text{A7})$$

where p indexes vectorized pixels, n is the number of pixels, and μ_I, μ_T are the mean pixel values of the respective preselection images.

Both scores are maximized: corr is a normalized (mean-subtracted) correlation, while mad is defined as the negative mean absolute difference so that smaller pixel-wise devia-

tions yield larger scores. The stabilizer ϵ prevents numerical issues when the variance is near zero.

The shortlist is defined as the index set of the top- K_s templates under the chosen score,

$$\mathcal{K}_s = \text{TopK}(\{\text{score}(I_{\text{pre}}, T_{k,\text{pre}})\}_{k=1}^K, K_s), \quad (\text{A8})$$

and SSIM refinement is then computed only for $k \in \mathcal{K}_s$ on the selection resolution and inner region Ω .

The computational complexity of template preselection can be summarized as follows. Let K be the number of templates, and let P_{pre} denote the number of evaluated pixels in Ω at the preselection resolution; using the common square-grid proxy, $P_{\text{pre}} = \mathcal{O}(D_{\text{pre}}^2)$. Computing either mad or corr for a single template is linear in P_{pre} , so scoring all templates is

$$\mathcal{O}(K P_{\text{pre}}) = \mathcal{O}(K D_{\text{pre}}^2). \quad (\text{A9})$$

Constructing the shortlist can be implemented either by (i) sorting all K scores, which costs $\mathcal{O}(K \log K)$, or (ii) using a top- K_s selection routine (e.g., partial sort/heap/selection), which costs $\mathcal{O}(K \log K_s)$ (or $\mathcal{O}(K)$ expected time for selection-based methods) when $K_s \ll K$. The subsequent SSIM refinement cost is dominated by evaluating SSIM on K_s candidates at selection resolution, i.e.,

$$\mathcal{O}(K_s D_{\text{sel}}^2), \quad (\text{A10})$$

with smaller constants when SSIM is computed on the cropped ROI Ω rather than on the full image. In practice, grayscale conversion and resizing of templates can be cached to amortize per-query preprocessing and reduce I/O overhead.

Appendix A.3. Processing Resolution Presets and Parameter Scaling

To reduce runtime on high-resolution images, the implementation optionally runs the core localization stages on a downsampled processing grid and maps outputs back to the original resolution.

Let L be the maximum long-side length among the query and selected template, i.e.,

$$L = \max(H_I, W_I, H_T, W_T). \quad (\text{A11})$$

Given a preset target long-side length L_{target} (e.g., 1280 for “720p”, 854 for “480p”), the isotropic processing scale is

$$s = \min\left(1, \frac{L_{\text{target}}}{L}\right). \quad (\text{A12})$$

When $s < 1$, both images are resized by s prior to registration, score-map computation, and mask extraction.

To maintain approximately consistent behavior across scales, morphology and area parameters are scaled when enabled. Denoting original parameters $(A_{\text{min}}, r_c, r_d)$ and scaled parameters $(A'_{\text{min}}, r'_c, r'_d)$, the implementation uses

$$A'_{\text{min}} = \max(1, \text{round}(A_{\text{min}} s^2)), \quad r'_c = \max(1, \text{round}(r_c s)), \quad r'_d = \max(1, \text{round}(r_d s)), \quad (\text{A13})$$

where $\text{round}(\cdot)$ denotes rounding to the nearest integer (as in the reference implementation). Finally, predicted boxes are mapped back to the original grid by scaling their $[x \ y \ w \ h]$ coordinates by $1/s$.

Appendix A.4. SSIM-Map Implementation Details

The SSIM map is computed using MATLAB R2023a’s SSIM to obtain the per-pixel similarity map (second output). For reproducibility, the reference implementation uses grayscale

images normalized to $[0,1]$ (dynamic range $L = 1$) and evaluates SSIM with explicit parameters: DynamicRange = 1, $K_1 = 0.01$, $K_2 = 0.03$, GaussianWeights = true, and WindowSize = 11 (square window). Under MATLAB's SSIM definition, $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$, so with $L = 1$ the stabilizing constants are $C_1 = 1 \times 10^{-4}$ and $C_2 = 9 \times 10^{-4}$. For robustness across MATLAB versions, the reference implementation attempts to call SSIM with the above name-value parameters and fall back to the default SSIM(A,B) if a given option (or per-pixel map output) is unsupported; if the per-pixel map output is unavailable, it falls back to using the scalar SSIM value to form a constant map.

Appendix A.5. Detection Confidence Score from Connected Components

The implementation forms detections by connected components of the binary mask and assigns each component a confidence score computed from the defect-likelihood map values within that component.

Let Y be the final binary mask. Let $\{C_i\}_{i=1}^{N_C}$ denote the connected components of Y . Let M denote the defect-likelihood (score) map used for thresholding. Each component yields an axis-aligned bounding box b_i and a scalar confidence score s_i computed by a configurable statistic over $\{M(u,v) : (u,v) \in C_i\}$. The default implementation uses a robust quantile score:

$$s_i = Q_{q_s}(\{M(u,v) : (u,v) \in C_i\}), \quad q_s = 0.95. \quad (\text{A14})$$

For completeness, the implementation also supports $s_i = \max_{(u,v) \in C_i} M(u,v)$ or $s_i = \frac{1}{|C_i|} \sum_{(u,v) \in C_i} M(u,v)$.

Appendix A.6. Data Availability

The dataset and code availability are summarized in the Data Availability statement in the main text. If access issues arise during peer review, the code and evaluation scripts can also be provided by the corresponding author upon reasonable request.

For reproducibility, the experiments in this paper use the PKU-Market-PCB dataset [3] with the following dataset organization under dev/PCB_DATASET/:

- Defect-free template images: PCB_USED(Defect-free)/;
- Defective query images: images(Defective)/ (organized by defect type, e.g., Missing_hole, Mouse_bite, Open_circuit, Short, Spur, Spurious_copper);
- Bounding-box annotations: Pascal VOC XML files under Annotations/ with the same defect-type subfolder structure as images(Defective)/.

Appendix A.7. Implementation Configuration Used in Experiments

For reproducibility, Table A1 summarizes the key configuration values used to generate the reported results and figures in this manuscript using the mathematical notation introduced in the main text and Appendix B.2. The rightmost column provides the corresponding configuration field in defaultConfig() (dev/v0_14_eval2_release1.m).

Table A1. Experiment configuration in manuscript notation (mapped to implementation cfg).

Symbol	Value Used in Experiments	Implementation Mapping
p	$p = 0.07$ (area-avoidance ratio for ROI border exclusion)	cfg.roi.avoidAreaRatio (with cfg.roi.enable = true)
q	$q = 0.995$ (threshold quantile on Ω for mask extraction)	cfg.mask.quantile
$A_{\min}$	$A_{\min} = 150$ px (minimum connected-component area)	cfg.mask.minAreaPx (scaled when resizing is enabled)
r_c	$r_c = 6$ px (morphological closing radius)	cfg.mask.closeRadiusPx (scaled when resizing is enabled)
r_d	$r_d = 2$ px (morphological dilation radius)	cfg.mask.dilateRadiusPx (scaled when resizing is enabled)
q_s	$q_s = 0.95$ (component confidence quantile)	cfg.bbox.scoreMethod = 'quantile', cfg.bbox.scoreQuantile = 0.95

Table A1. Cont.

Symbol	Value Used in Experiments	Implementation Mapping
w	$w = 0.35$ (absolute-difference weight in $M = (1 - \text{SSIM}_{\text{map}}) + w D_{\text{abs}}$)	<code>cfg.diff.absDiffWeight</code> (with <code>cfg.diff.useAbsDiff = true</code>)
$\Delta_{\min}$	$\Delta_{\min} = 0.0$ (accept alignment iff SSIM gain $\Delta \geq \Delta_{\min}$ on Ω)	<code>cfg.registration.validateTransform = true</code> , <code>cfg.registration.minSSIMGain = 0.0</code>
τ_{IoU}	$\tau_{\text{IoU}} = 0.10$ (IoU threshold for TP/FP/FN matching)	<code>cfg.eval.iouThreshold = 0.1</code>
D_{pre}	$D_{\text{pre}} = 128$ px (max long-side for preselect score)	<code>cfg.templateSelect.preselect.maxDim</code>
K_s	$K_s = 3$ (shortlist size after preselect)	<code>cfg.templateSelect.preselect.topK</code>
D_{sel}	$D_{\text{sel}} = 426$ px (max long-side for SSIM refinement; preset “240p”)	<code>cfg.templateSelect.downscalePreset = ‘240p’</code>
D_{proc}	$D_{\text{proc}} = 1280$ px (max long-side for processing; preset “720p”)	<code>cfg.resize.enable = true</code> , <code>cfg.resize.preset = ‘720p’</code>
τ_{θ}	translation-only transform model for alignment	<code>cfg.registration.type = ‘translation’</code>
ROI eval	SSIM/refinement computed on cropped ROI Ω (faster than masking)	<code>cfg.templateSelect.useROICropForSSIM = true</code>

Appendix B. Evaluation and Metric Definitions

This appendix collects explicit metric definitions and evaluation-protocol computations used to produce the reported PR/AP/FROC results under the adopted IoU matching protocol.

Appendix B.1. Operating-Point Threshold Selection (Option A: Best F1)

This paper reports a single reference operating point using Option A (best F1). This operating point is selected from the confidence-ranked detections produced by the pipeline.

Let $\{(s_i, y_i)\}_{i=1}^N$ denote all detections across the evaluated dataset, where $s_i \in \mathbb{R}$ is the confidence score, and $y_i \in \{0, 1\}$ indicates whether the detection is a TP under the adopted per-image greedy one-to-one IoU matching (with $y_i = 1$ for TP, and $y_i = 0$ for FP). Let N_{GT} be the total number of ground-truth boxes across the evaluated images.

Sort detections by score in descending order, producing an ordering π such that $s_{\pi(1)} \geq s_{\pi(2)} \geq \dots \geq s_{\pi(N)}$. Define the cumulative counts

$$\text{TP}(k) = \sum_{i=1}^k y_{\pi(i)}, \quad \text{FP}(k) = \sum_{i=1}^k (1 - y_{\pi(i)}), \quad k = 1, \dots, N. \quad (\text{A15})$$

The corresponding precision, recall, and F1 sequences are

$$P(k) = \frac{\text{TP}(k)}{\text{TP}(k) + \text{FP}(k) + \epsilon}, \quad (\text{A16})$$

$$R(k) = \frac{\text{TP}(k)}{N_{\text{GT}} + \epsilon}, \quad (\text{A17})$$

$$\text{F1}(k) = \frac{2 P(k) R(k)}{P(k) + R(k) + \epsilon}. \quad (\text{A18})$$

The best-F1 index is $k^* = \arg \max_k \text{F1}(k)$, and the operating-point threshold is chosen as

$$t^* = s_{\pi(k^*)}. \quad (\text{A19})$$

All detections with $s_i \geq t^*$ are retained to compute the reported operating-point counts and metrics.

For the cross-validated operating-point estimates reported in the main text, we use $K = 5$ folds. Evaluated images (those with available Pascal VOC ground-truth XML) are randomly permuted using a fixed implementation seed and assigned to folds in a round-robin manner; in each fold, t^* is selected on the calibration folds and metrics are computed on the held-out fold.

Appendix B.2. FROC Computation and Image-Level Recall

The evaluation exports FROC curves by sweeping a finite set of score thresholds derived from the score distribution.

Let $\mathcal{S} = \{s_i\}_{i=1}^N$ be the set of all detection scores across the evaluated dataset. This work defines a set of candidate thresholds by sampling quantiles of $\mathcal{S}$. Let $\mathcal{Q} = \{0, \frac{1}{49}, \dots, 1\}$ (50 evenly spaced quantile levels) and define

$$\mathcal{T} = \text{unique}(\{Q_q(\mathcal{S}) : q \in \mathcal{Q}\}), \quad (\text{A20})$$

then sort $\mathcal{T}$ in descending order.

For each threshold $t \in \mathcal{T}$, this work retains detections with $s_i \geq t$ and compute dataset-level counts $\text{TP}(t)$, $\text{FP}(t)$, $\text{FN}(t)$ by performing greedy one-to-one matching within each image (each ground-truth box matches at most one detection and vice versa) under the adopted IoU threshold. Let N_{img} denote the number of evaluated images that have available ground-truth annotations, and let N_{GT} denote the total number of ground-truth boxes across those images.

The object-level FROC quantities are

$$\text{FP/image}(t) = \frac{\text{FP}(t)}{N_{\text{img}} + \epsilon}, \quad \text{Recall}_{\text{obj}}(t) = \frac{\text{TP}(t)}{N_{\text{GT}} + \epsilon}. \quad (\text{A21})$$

Image-level recall is computed by counting images that contain at least one ground-truth box and checking whether any TP occurs in the image at threshold t . Let $N_{\text{img}}^{\text{GT}}$ be the number of images with at least one ground-truth box. Let $N_{\text{img}}^{\text{TP}}(t)$ be the number of those images for which at least one retained detection is matched as a TP at threshold t . Then

$$\text{Recall}_{\text{img}}(t) = \frac{N_{\text{img}}^{\text{TP}}(t)}{N_{\text{img}}^{\text{GT}} + \epsilon}. \quad (\text{A22})$$

In the manuscript, FROC plots use $\text{Recall}_{\text{obj}}(t)$ versus $\text{FP/image}(t)$; $\text{Recall}_{\text{img}}(t)$ is reported for completeness.

Appendix B.3. IoU for $[x \ y \ w \ h]$ Boxes

Let $b = [x \ y \ w \ h]$ and $g = [x' \ y' \ w' \ h']$. Define corners $(x_1, y_1) = (x, y)$, $(x_2, y_2) = (x + w, y + h)$ and similarly for g . The intersection width/height are

$$w_I = \max(0, \min(x_2, x'_2) - \max(x_1, x'_1)), \quad (\text{A23})$$

$$h_I = \max(0, \min(y_2, y'_2) - \max(y_1, y'_1)), \quad (\text{A24})$$

with intersection area $A_I = w_I h_I$. The IoU is

$$\text{IoU}(b, g) = \frac{A_I}{A(b) + A(g) - A_I + \epsilon}. \quad (\text{A25})$$

Appendix B.4. Average Precision from Confidence-Ranked Detections

Let all detections across the dataset be ranked by confidence s in descending order. For a fixed IoU threshold, each detection is labeled TP or FP by greedy matching within each image (each ground-truth box may match at most one detection). Let $\text{TP}(t)$ and $\text{FP}(t)$ be cumulative counts up to rank t , and let N_{GT} be the total number of ground-truth boxes. Precision and recall sequences are

$$P(t) = \frac{TP(t)}{TP(t) + FP(t) + \epsilon}, \quad R(t) = \frac{TP(t)}{N_{GT} + \epsilon}. \quad (A26)$$

AP is the area under the precision–recall curve. This work uses a standard VOC-style discrete approximation based on the precision envelope $\tilde{P}(r) = \max_{\hat{r} \geq r} P(\hat{r})$ and piecewise integration over recall. In our experiments, AP is computed using the all-points VOC interpolation (precision envelope with integration over all recall steps), not the VOC 2007 11-point variant.

References

- Chin, R.T.; Harlow, C.A. Automated Visual Inspection: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **1982**, PAMI-4, 557–573. [\[CrossRef\]](#)
- Moganti, M.; Ercal, F.; Dagli, C.H.; Tsunekawa, S. Automatic PCB Inspection Algorithms: A Survey. *Comput. Vis. Image Underst.* **1996**, 63, 287–313. [\[CrossRef\]](#)
- Peking University Open Lab on Human-Robot Interaction. *PKU-Market-PCB Dataset*. Available online: <https://robotics.pkusz.edu.cn/resources/dataset/> (accessed on 26 January 2026).
- Azamfirei, V.; Psarommatis, F.; Lagrosen, Y. Application of automation for in-line quality inspection, a zero-defect manufacturing approach. *J. Manuf. Syst.* **2023**, 67, 1–22. [\[CrossRef\]](#)
- Chen, X.; Wu, Y.; He, X.; Ming, W. A Comprehensive Review of Deep Learning-Based PCB Defect Detection. *IEEE Access* **2023**, 11, 139017–139038. [\[CrossRef\]](#)
- Sankar, V.U.; Lakshmi, G.; Sankar, Y.S. A Review of Various Defects in PCB. *J. Electron. Test.* **2022**, 38, 481–491. [\[CrossRef\]](#)
- Kim, M. Meander-DGS Effect on Electromagnetic Bandgap Structure for Power/Ground Noise Suppression in High-Speed Integrated Circuit Packages and PCBs. *Electronics* **2022**, 11, 211. [\[CrossRef\]](#)
- Ott, H.W. *Electromagnetic Compatibility Engineering*; Wiley: Hoboken, NJ, USA, 2009.
- Bogatin, E. *Signal and Power Integrity—Simplified*, 2nd ed.; Prentice Hall: Hoboken, NJ, USA, 2010.
- Johnson, H.W.; Graham, M. *High-Speed Digital Design: A Handbook of Black Magic*; Prentice Hall: Hoboken, NJ, USA, 1993.
- Ling, Q.; Isa, N.A.M. Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey. *IEEE Access* **2023**, 11, 15921–15944. [\[CrossRef\]](#)
- Tsai, D.; Lin, C.; Chen, J. The evaluation of normalized cross correlations for defect detection. *Pattern Recognit. Lett.* **2003**, 24, 2525–2535. [\[CrossRef\]](#)
- Li, Y.; Kuo, P.; Guo, J. Automatic Industry PCB Board DIP Process Defect Detection System Based on Deep Ensemble Self-Adaption Method. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2021**, 11, 312–323. [\[CrossRef\]](#)
- Park, J.; Kim, Y.; Seo, H.; Cho, Y. Analysis of Training Deep Learning Models for PCB Defect Detection. *Sensors* **2023**, 23, 2766. [\[CrossRef\]](#)
- Tsai, D.; Chiang, I.; Tsai, Y. A Shift-Tolerant Dissimilarity Measure for Surface Defect Detection. *IEEE Trans. Ind. Inform.* **2012**, 8, 128–137. [\[CrossRef\]](#)
- Gaidhane, V.H.; Hote, Y.V.; Singh, V. An efficient similarity measure approach for PCB surface defect detection. *Pattern Anal. Appl.* **2018**, 21, 277–289. [\[CrossRef\]](#)
- Pham, T.T.A.; Thoi, D.K.T.; Choi, H.; Park, S. Defect Detection in Printed Circuit Boards Using Semi-Supervised Learning. *Sensors* **2023**, 23, 3246. [\[CrossRef\]](#)
- Chen, C.; Wu, Q.; Zhang, J.; Xia, H.; Lin, P.; Wang, Y.; Tian, M.; Song, R. U2D2PCB: Uncertainty-Aware Unsupervised Defect Detection on PCB Images Using Reconstructive and Discriminative Models. *IEEE Trans. Instrum. Meas.* **2024**, 73, 5017710. [\[CrossRef\]](#)
- Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson: London, UK, 2018.
- Crispin, A.J.; Rankov, V. Automated inspection of PCB components using a genetic algorithm template-matching approach. *Int. J. Adv. Manuf. Technol.* **2007**, 35, 293–300. [\[CrossRef\]](#)
- Kaur, B.; Kaur, G.; Kaur, A. Detection and classification of Printed circuit board defects using image subtraction method. In Proceedings of the 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 6–8 March 2014; pp. 1–5. [\[CrossRef\]](#)
- Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, 13, 600–612. [\[CrossRef\]](#) [\[PubMed\]](#)
- Debella-Gilo, M.; Käb, A. Sub-pixel precision image matching for measuring surface displacements on mass movements using normalized cross-correlation. *Remote Sens. Environ.* **2011**, 115, 130–142. [\[CrossRef\]](#)

24. Zheng, X.; Zheng, S.; Kong, Y.; Chen, J. Recent advances in surface defect inspection of industrial products using deep learning techniques. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, 35–58. [\[CrossRef\]](#)
25. Huang, S.; Pan, Y. Automated visual inspection in the semiconductor industry: A survey. *Comput. Ind.* **2015**, *66*, 1–10. [\[CrossRef\]](#)
26. Dwivedi, G.; Pensia, L.; Lohchab, V.; Kumar, R. Nondestructive Inspection and Quantification of Soldering Defects in PCB Using an Autofocusing Digital Holographic Camera. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–8. [\[CrossRef\]](#)
27. Cai, N.; Lin, J.; Ye, Q.; Wang, H.; Weng, S.; Ling, B.W. A New IC Solder Joint Inspection Method for an Automatic Optical Inspection System Based on an Improved Visual Background Extraction Algorithm. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2016**, *6*, 161–172. [\[CrossRef\]](#)
28. Wei, S.; Lai, S. Fast Template Matching Based on Normalized Cross Correlation with Adaptive Multilevel Winner Update. *IEEE Trans. Image Process.* **2008**, *17*, 2227–2235. [\[CrossRef\]](#)
29. Cheng, H.; Liu, Z.; Hou, L.; Yang, J. Sparsity-Induced Similarity Measure and Its Applications. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 613–626. [\[CrossRef\]](#)
30. Hassanin, A.I.M.; Abd El-Samie, F.E.; El Banby, G.M. A real-time approach for automatic defect detection from PCBs based on SURF features and morphological operations. *Multimed. Tools Appl.* **2019**, *78*, 34437–34457. [\[CrossRef\]](#)
31. Kang, D.; Lai, J.; Han, Y. Improving surface defect detection with context-guided asymmetric modulation networks and confidence-boosting loss. *Expert Syst. Appl.* **2023**, *225*, 120121. [\[CrossRef\]](#)
32. Luo, Z.; Lei, L.; Li, H. Rotation-Angle-Based Principal Feature Extraction and Optimization for PCB Defect Detection Under Uncertainties. *IEEE Trans. Ind. Inform.* **2025**, *21*, 932–939. [\[CrossRef\]](#)
33. Onshaunjit, J.; Srinonchat, J. Algorithmic Scheme for Concurrent Detection and Classification of Printed Circuit Board Defects. *Comput. Mater. Contin.* **2021**, *71*, 355–367. [\[CrossRef\]](#)
34. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [\[CrossRef\]](#)
35. Zhong, Z.; Ma, Z. A Novel Defect Detection Algorithm for Flexible Integrated Circuit Package Substrates. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2117–2126. [\[CrossRef\]](#)
36. Wang, Y.; Chen, Z.; Wang, J.; Shang, P.; Sowmya, A.; Sun, C. Printed Circuit Board Defect Detection Based on Lightweight Deep Learning Fusion Model. *Sensors* **2025**, *25*, 7403. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Jun, H.; Jung, I.Y. Enhancement of Product-Inspection Accuracy Using Convolutional Neural Network and Laplacian Filter to Automate Industrial Manufacturing Processes. *Electronics* **2023**, *12*, 3795. [\[CrossRef\]](#)
38. Adibhatla, V.A.; Chih, H.; Hsu, C.; Cheng, J.; Abbod, M.F.; Shieh, J. Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks. *Electronics* **2020**, *9*, 1547. [\[CrossRef\]](#)
39. Yang, Y.; Kang, H. An Enhanced Detection Method of PCB Defect Based on Improved YOLOv7. *Electronics* **2023**, *12*, 2120. [\[CrossRef\]](#)
40. Du, B.; Wan, F.; Lei, G.; Xu, L.; Xu, C.; Xiong, Y. YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5. *Electronics* **2023**, *12*, 2821. [\[CrossRef\]](#)
41. Mo, C.; Hu, Z.; Wang, J.; Xiao, X. SGT-YOLO: A Lightweight Method for PCB Defect Detection. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 1–11. [\[CrossRef\]](#)
42. Wang, X.; Gao, J.-S.; Hou, B.-J.; Wang, Z.-S.; Ding, H.-W.; Wang, J. A Lightweight Modified YOLOX Network Using Coordinate Attention Mechanism for PCB Surface Defect Detection. *IEEE Sens. J.* **2022**, *22*, 20910–20920. [\[CrossRef\]](#)
43. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [\[CrossRef\]](#)
44. Wang, F.; Jiang, X.; Han, Y.; Wu, L. YOLO-LSDI: An Enhanced Algorithm for Steel Surface Defect Detection Using a YOLOv11 Network. *Electronics* **2025**, *14*, 2576. [\[CrossRef\]](#)
45. Yuan, Z.; Tang, X.; Ning, H.; Yang, Z. LW-YOLO: Lightweight Deep Learning Model for Fast and Precise Defect Detection in Printed Circuit Boards. *Symmetry* **2024**, *16*, 418. [\[CrossRef\]](#)
46. Li, C.; Qu, Z.; Wang, S.; Bao, K.; Wang, S. A Method of Defect Detection for Focal Hard Samples PCB Based on Extended FPN Model. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2022**, *12*, 217–227. [\[CrossRef\]](#)
47. Chen, B.; Dang, Z. Fast PCB Defect Detection Method Based on FasterNet Backbone Network and CBAM Attention Mechanism Integrated With Feature Fusion Module in Improved YOLOv7. *IEEE Access* **2023**, *11*, 95092–95103. [\[CrossRef\]](#)
48. Yang, J.; Liu, Z.; Du, W.; Zhang, S. A PCB Defect Detector Based on Coordinate Feature Refinement. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [\[CrossRef\]](#)
49. Hu, B.; Wang, J. Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network. *IEEE Access* **2020**, *8*, 108335–108345. [\[CrossRef\]](#)
50. Feng, B.; Cai, J. PCB Defect Detection via Local Detail and Global Dependency Information. *Sensors* **2023**, *23*, 7755. [\[CrossRef\]](#) [\[PubMed\]](#)

51. Jiang, W.; Li, T.; Zhang, S.; Chen, W.; Yang, J. PCB defects target detection combining multi-scale and attention mechanism. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106359. [[CrossRef](#)]
52. Wu, L.; Zhang, L.; Zhou, Q. Printed Circuit Board Quality Detection Method Integrating Lightweight Network and Dual Attention Mechanism. *IEEE Access* **2022**, *10*, 87617–87629. [[CrossRef](#)]
53. Chen, W.; Meng, S.; Wang, X. Local and Global Context-Enhanced Lightweight CenterNet for PCB Surface Defect Detection. *Sensors* **2024**, *24*, 4729. [[CrossRef](#)]
54. Liu, H.; Zhao, X.; Liu, Q.; Chen, W. An Optimization Method for PCB Surface Defect Detection Model Based on Measurement of Defect Characteristics and Backbone Network Feature Information. *Sensors* **2024**, *24*, 7373. [[CrossRef](#)]
55. Bhattacharya, A.; Cloutier, S.G. End-to-end deep learning framework for printed circuit board manufacturing defect classification. *Sci. Rep.* **2022**, *12*, 12559. [[CrossRef](#)]
56. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder. *Sensors* **2021**, *21*, 4968. [[CrossRef](#)]
57. Li, D.; Li, C.; Chen, C.; Zhao, Z. Semantic Segmentation of a Printed Circuit Board for Component Recognition Based on Depth Images. *Sensors* **2020**, *20*, 5318. [[CrossRef](#)]
58. Peng, J.; Fan, W.; Lan, S.; Wang, D. MDD-DETR: Lightweight Detection Algorithm for Printed Circuit Board Minor Defects. *Electronics* **2024**, *13*, 4453. [[CrossRef](#)]
59. Xie, Z.; Zou, X. MFAD-RTDETR: A Multi-Frequency Aggregate Diffusion Feature Flow Composite Model for Printed Circuit Board Defect Detection. *Electronics* **2024**, *13*, 3557. [[CrossRef](#)]
60. Wang, S. Domain adaptation using transformer models for automated detection of exterior cladding materials in street view images. *Sci. Rep.* **2025**, *16*, 2696. [[CrossRef](#)] [[PubMed](#)]
61. Deng, S.; Deng, L.; Sun, T.; Yu, S.; Wang, L.; Chen, B.; Hu, H.; Xie, Y.; Yin, H.; Xiao, J.; et al. EEDD: Edge-Guided Energy-Based PCB Defect Detection. *Electronics* **2023**, *12*, 2306. [[CrossRef](#)]
62. Lei, L.; Li, H.; Yang, H. Multiscale Convolution-Based Probabilistic Classification for Detecting Bare PCB Defects. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–8. [[CrossRef](#)]
63. Ding, R.; Dai, L.; Li, G.; Liu, H. TDD-net: A tiny defect detection network for printed circuit boards. *CAAI Trans. Intell. Technol.* **2019**, *4*, 110–116. [[CrossRef](#)]
64. Shao, X.; Liu, Z.; He, Q.; Yu, M.; Li, C. SM-FSOD: A Second-Order Meta-Learning Algorithm for Few-Shot PCB Defect Object Detection. *Electronics* **2025**, *14*, 3863. [[CrossRef](#)]
65. Lin, Q.; Takamasu, K.; Chen, M. An Optical Lens Defect Detection Method for Micro Vision Based on WGSO-YOLO. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–8. [[CrossRef](#)]
66. Shi, W.; Li, C.; Dai, J.; Niu, N. GCF-Net: Steel Surface Defect Detection Network Based on Global Attention Perception and Cross-Layer Interactive Fusion. *Electronics* **2025**, *14*, 1776. [[CrossRef](#)]
67. Annaby, M.H.; Fouda, Y.M.; Rushdi, M.A. Improved Normalized Cross-Correlation for Defect Detection in Printed-Circuit Boards. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 199–211. [[CrossRef](#)]
68. Zhang, Z.; Deriche, R.; Faugeras, O.; Luong, Q. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell.* **1995**, *78*, 87–119. [[CrossRef](#)]
69. Ebayyeh, A.A.R.M.A.; Mousavi, A. A Review and Analysis of Automatic Optical Inspection and Quality Monitoring Methods in Electronics Industry. *IEEE Access* **2020**, *8*, 183192–183271. [[CrossRef](#)]
70. Zhou, Y.; Yuan, M.; Zhang, J.; Ding, G.; Qin, S. Review of vision-based defect detection research and its perspectives for printed circuit board. *J. Manuf. Syst.* **2023**, *70*, 557–578. [[CrossRef](#)]
71. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
72. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
73. Chiba, N.; Hashimoto, K. Ultra-Fast Multi-Scale Shape Estimation of Light Transport Matrix for Complex Light Reflection Objects. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 6147–6152. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.