A hierarchical hybrid architecture with CNN autoencoder for foreign object detection on jig surfaces in smartphone screen manufacturing

Luong Van Chung[1,*], Dong Xuan Tuan[1], Ha Trung Kien[2] and Dao Quy Thinh[1]

[1]*School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam*
[2]*Department of Electrical Engineering, School of Electrical and Electronic Engineering, Hanoi University of Industry*
[*]*Corresponding author E-mail: vanchung27392@gmail.com*
DOI: https://doi.org/10.64032/mca.v30i3.429

Abstract

Foreign object and surface defect detection on inspection jigs is critical in smartphone screen manufacturing, demanding both real-time speed and high accuracy. This paper proposes a Hierarchical Hybrid Pipeline comprising two phases: Phase 1 employs traditional image processing with rigid Euclidean alignment, absolute background subtraction, and robust-statistics-based dynamic thresholding (Median + k × MAD) for high-speed full-frame screening; Phase 2 employs an unsupervised CNN Autoencoder with Ensemble anomaly scoring (Local MSE + Feature-based MSE) for in-depth verification of suspect regions. Evaluated on a dataset collected from an actual Lamination production line, Phase 1 achieves 99.25% Accuracy on 1,200 images (∼200 ms processing time). Phase 2, evaluated on a test set of 140 abnormal patches and 58 normal patches, attains an AUC-ROC of 0.971 and F1-score of 0.961 with 97.1% Recall. The complete pipeline executes in ∼200 ms for nominal frames and up to 350 ms for worst-case scenarios on industrial CPU hardware, satisfying real-time production requirements.

Keywords: *Anomaly Detection; Automated Optical Inspection; CNN Autoencoder; Industrial Image Processing; Jig Surface Inspection; Robust Statistics.*

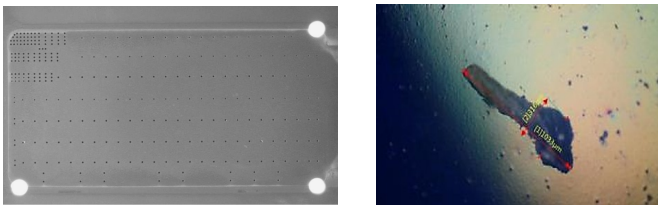
Abbreviations

AOI	Automated Optical Inspection
AUC-ROC	Area Under the ROC Curve
ROI	Region of Interest
CNN	Convolutional Neural Network
FNR	False Negative Rate
FPR	False Positive Rate
MAD	Median Absolute Deviation
MSE	Mean Squared Error
NCC	Normalized Cross-Correlation
ONNX	Open Neural Network Exchange

1. Introduction

In electronic component manufacturing, high-precision assembly stages operate within cleanroom environments. The inspection jig surface directly contacts the material; hence, any foreign object or surface defect on the jig may cause irreversible damage to the finished product. Automated Optical Inspection (AOI) systems enable 100% inspection at production speed, yet their effectiveness hinges on a software architecture capable of balancing processing speed and accuracy amid complex optical noise on metallic surfaces.

Key technical challenges include: (i) anodized aluminum surfaces with high specular reflectivity that generate false-positive signals; (ii) morphologically diverse foreign objects (dust, fibers, metallic fragments, adhesive residue) at the tens-to-hundreds μm scale; (iii) mechanical deviations from vibration and fixture tolerances causing inter-frame jig misalignment; (iv) real-time processing under 350 ms per 10-megapixel frame; (v) severe class imbalance with extremely rare defect occurrences [1].



(a) Image of the actual jig **(b)** Sample foreign objects

Figure 1: Image from the dataset.

Traditional image processing methods [2], [3] achieve high throughput via linear pixel operations but depend on static thresholds and are sensitive to illumination drift, yielding elevated false-positive rates. Surface defect detection on industrial components has been explored via CNN-based classifiers [4], [5], yet these supervised approaches require large labeled defect datasets that are unavailable in zero-defect manufacturing settings. Deep learning models [6] learn hierarchical feature representations directly from data but impose prohibitive computational costs for full-frame real-time processing. Anomaly detection approaches based on Autoencoders [7], [8], reconstruction-based discriminative methods [9], and memory banks [10] have

Received: 7 April 2026; Revised: 5 May 2026; Accepted: 18 May 2026.

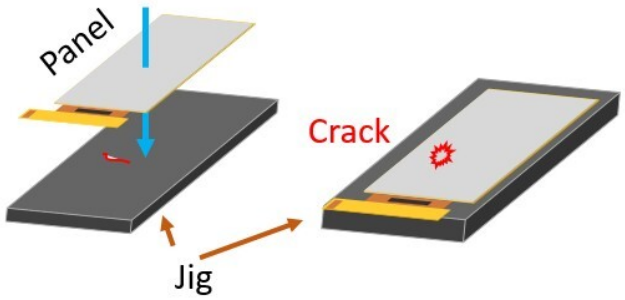


Figure 2: Defect generation mechanism.

demonstrated efficacy in defect detection without NG samples; SimpleNet [11] further achieves competitive AUC with a lightweight architecture. However, standalone deployment of these methods fails to simultaneously satisfy both speed and accuracy constraints on CPU-only industrial hardware.

Recent hybrid inspection approaches, such as those combining edge detection with deep classification networks or employing multi-stage CNN pipelines, have demonstrated improved accuracy in surface defect detection. However, most of these methods rely on GPU acceleration and have been validated only under controlled laboratory conditions, leaving a gap in practical deployment on resource-constrained industrial hardware. In contrast, the proposed hierarchical architecture is specifically designed to operate entirely on CPU-only platforms, achieving near-zero false negative rates in a real-world smartphone screen manufacturing environment-a combination that, to the best of our knowledge, has not been reported in prior literature.

Primary contributions:

(1) Hierarchical hybrid pipeline: A novel two-phase architecture decoupling high-speed coarse screening from deep learning-based verification, effectively resolving the fundamental trade-off between real-time processing speed and high-precision anomaly detection.

(2) Robust statistical thresholding: A dynamic background-subtraction mechanism $T = \max(\text{Median} + k \cdot \text{MAD}, T_{\min})$ that suppresses optical noise and specular reflections on metallic surfaces, reducing false positives compared to mean-based thresholding.

(3) Ensemble anomaly scoring: An integration of Local MSE and Feature-based MSE that mitigates the "dilution effect" in standard autoencoders, enhancing sensitivity to microscopic and low-contrast defects.

(4) Industrial CPU deployment: The complete pipeline is evaluated on 1,200 real production images collected from an active manufacturing line, and optimized via ONNX Runtime to achieve $<$350 ms per 10-megapixel frame on standard industrial CPUs without GPU dependency.

2. Methodology

2.1 Dataset

The dataset was collected from an actual production line using a Basler industrial camera (GigE POE, grayscale Mono8, 10 megapixels). Inspection jigs are fabricated from anodized aluminum; captured image size is 2592×1944 pixels.

Training data for the Phase 2 CNN Autoencoder follows the *normal-only training* paradigm [8]. Normal samples were collected from multiple golden reference images captured across different jig instances under real production conditions, covering variations in surface texture, illumination, and minor alignment differences inherent to the manufacturing process.

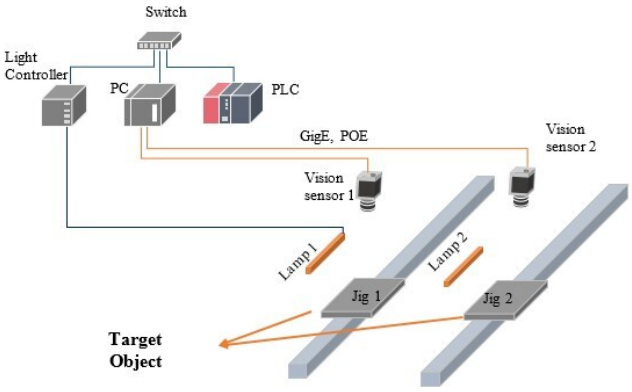


Figure 3: Applications of the foreign object detection system.

From these images, 48×48 pixel patches were extracted on a dense grid with stride 36 ($\approx$25% overlap). Only patches with $\geq$85% pixels inside the ROI were retained. The resulting 388 normal patches are not an arbitrary subset but represent the deterministic output of this sliding window procedure across the defined jig geometry. This specific quantity naturally emerges from the physical dimensions of the effective ROI combined with the chosen extraction parameters, thereby ensuring complete and uniform coverage of the inspectable surface without introducing unnecessary computational redundancy.

The abnormal test set comprises 140 real defect patches collected directly from the active production line over an extended period. These encompass four primary defect morphologies (35 patches each): Spot (circular bright/dark marks), Scratch (thin elongated marks), Cluster (groups of 5-12 particles), and Stain (elliptical blurred regions, 15-30% intensity deviation). Collecting and curating this rare anomalous data ensures the system is rigorously evaluated against actual industrial conditions, accurately reflecting the extreme class imbalance ($<$0.5% occurrence rate) inherent to zero-defect manufacturing.

In addition, a limited set of real NG samples collected from the production line is used for evaluation purposes. Although not sufficient for supervised training, these real samples provide complementary validation and confirm that the proposed method generalizes to real-world defect characteristics.

Both phases are evaluated exclusively on real industrial data: Phase 1 is evaluated on full-frame production images to validate high-speed screening, while Phase 2 is evaluated on the curated set of real NG patches to confirm anomaly detection performance. Data partitioning is summarized in Table 1.

The 388 normal patches are partitioned at an 85/15 ratio into 330 for autoencoder training and 58 for T_AI calibration. The test set comprises 140 real NG patches covering four defect morphologies (35 each). Phase 1 is evaluated on full-frame images; Phase 2 is evaluated at patch level, with the 58 validation patches additionally serving as negative ground truth in the confusion matrix.

Table 1: Dataset partitioning.

Set	Qty	Label	Purpose
Training	330	Normal	AE training
Validation	58	Normal	T_{AI} calibration
Test	140	Abnormal	Detection eval.

On-the-fly augmentation: horizontal/vertical flip (p=0.5), rotation $\pm 10^\circ$, brightness scaling $\times[0.85, 1.15]$, Gaussian noise σ=0.02, clamped to $[0,1]$.

2.2 Hierarchical hybrid architecture

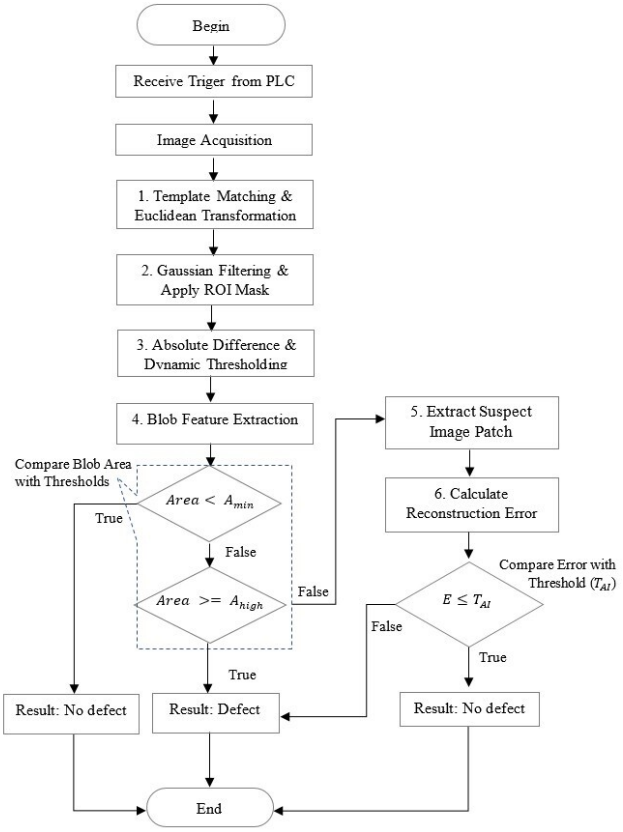


Figure 4: Hybrid hierarchical pipeline flow.

Let $\mathscr{I} \in \mathbb{R}^{H\times W}$ denote the inspection image and $\mathscr{G} \in \mathbb{R}^{H\times W}$ the golden image. The inspection task is formalized as a binary classification $f : \mathscr{I} \to \{\text{OK}, \text{NG}\}$ minimizing FNR $\to 0$ and FPR $< 5\%$ subject to the timing constraint $\tau < \tau_{\max}$.

The hybrid architecture decomposes $f = f_2 \circ f_1$, where f_1 (Phase 1) is a high-speed screener based on linear pixel operations ($\mathscr{O}(N)$, N=HW), and f_2 (Phase 2) is a CNN Autoencoder verifier operating only on the suspect subspace $\mathscr{S} \subset \mathscr{I}$ ($|\mathscr{S}| \ll |\mathscr{I}|$).

Phase 1 classifies $\mathscr{I}$ into: OK ($\forall b_k : A_k < A_{\min}$), NG ($\exists b_k : A_k \geq A_{high}$), or Suspect ($\exists b_k : A_{\min} \leq A_k < A_{high}$). Phase 2 activates only when $f_1(\mathscr{I}) = \text{Suspect}$.

2.3 Phase 1: Traditional vision screening

2.3.1 Geometric normalization

Under fixture tolerance ± 0.2 mm and a rigidly mounted camera, perspective and affine distortions are negligible. The 3-DOF rigid Euclidean transformation (translation $\Delta\mathbf{t}$, rotation $\Delta\theta$) is selected as the most parsimonious model [12].

Two fiducial marks $\{(\mathbf{p}_{0i}, T_i)\}_{i=1,2}$ are registered on $\mathscr{G}$, where $T_i \in \mathbb{R}^{P\times P}$ (P=100) is the template patch. Localization employs NCC [13]:

$$\text{NCC}(x,y) = \frac{\sum_{u,v}[T(u,v)-\bar{T}]\,[I(x+u,y+v)-\bar{I}_{xy}]}{\sqrt{\sum[T-\bar{T}]^2 \cdot \sum[I-\bar{I}_{xy}]^2}} \tag{1}$$

Search is confined to $\mathbf{p}_{0i} \pm R$ (R=400 px). When NCC falls below 0.50, the template is rotated $\pm 5^\circ$ in 1° steps to handle rotation up to 5°. From detected positions $\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2$:

$$\Delta\mathbf{t} = \tfrac{1}{2}\left[(\hat{\mathbf{p}}_1 - \mathbf{p}_{01}) + (\hat{\mathbf{p}}_2 - \mathbf{p}_{02})\right] \tag{2}$$

$$\Delta\theta = \arg(\hat{\mathbf{p}}_2 - \hat{\mathbf{p}}_1) - \arg(\mathbf{p}_{02} - \mathbf{p}_{01}) \tag{3}$$

The transformation matrix $\mathbf{M} = [\mathbf{R}_{\Delta\theta} \mid \mathbf{t}] \in \mathbb{R}^{2\times 3}$, $\mathbf{R}_{\Delta\theta} \in SO(2)$, is applied via inverse warp with bilinear interpolation. Alignment error < 0.5 pixel RMS [12]. The effective ROI is constructed as:

$$\Omega_{\text{eff}} = (\Omega_{\text{ROI}} \cap \Omega_{\text{valid}}) \setminus \Omega_{\text{excl}} \ominus SE^{25}_{\text{rect}} \tag{4}$$

2.3.2 Defect extraction and dynamic thresholding

A Gaussian filter G_σ (5×5, σ=1.0) is applied to both $\mathscr{I}_{\text{aligned}}$ and $\mathscr{G}$:

$$D(x,y) = |G_\sigma * \mathscr{I}_{\text{aligned}}(x,y) - G_\sigma * \mathscr{G}(x,y)| \tag{5}$$

The classification threshold is derived from robust statistics [14]:

$$T = \max\Big(\underbrace{\widetilde{D}_{\text{roi}} + k \cdot \text{MAD}(D_{\text{roi}})}_{\text{adaptive}}, \underbrace{T_{\min}}_{\text{floor}}\Big) \tag{6}$$

where $\widetilde{D}_{\text{roi}}$ is the median, $\text{MAD} = \text{med}_i\{|d_i - \widetilde{d}|\}$ [15], $k = 4.0$, and $T_{\min} = 15$. The coefficient $k = 4.0$ was selected based on empirical testing across $k \in [2.0, 6.0]$. Values below 3.5 increased false alarms due to surface texture, while values above 5.0 caused missed detections of subtle defects. This choice ensures 100% recall in Phase 1 and is consistent with robust statistical practices where $k \in [3,5]$ is typically used for outlier detection in non-Gaussian distributions. The Median and MAD possess a 50% breakdown point-remaining reliable when up to 50% of observations are outliers-outperforming mean/std (0%) and Otsu (∼20%) [16]. The floor $T_{\min}$ prevents threshold collapse when MAD ≈ 0.

The binary image undergoes sequential Opening $\to$ Closing with a cross structuring element SE_+ (3×3) [3]:

$$B'' = \left[(B \ominus SE_+) \oplus SE_+\right] \bullet SE_+ \tag{7}$$

SE_+ limits diagonal erosion, preserving 45° scratch morphology-an advantage over $SE_\square$ of equal size.

2.3.3 Blob analysis and tiered classification

8-connectivity CCL [17] identifies connected regions. Border-touching blobs (≤ 3 px) are rejected (warp artifacts).

Geometric features per blob k: area A_k, max extent $L_k = \max(w_k, h_k)$, centroid $\mathbf{c}_k$.

The tiered rule-based classifier (Algorithm 1) operates with short-circuit evaluation. The conjunctive condition (line 3) ensures low-area but high-length blobs (scratch tips) are not prematurely discarded-an improvement over single-criterion area-only filtering.

Algorithm 1 Tiered Blob Classification

Require: Blob set $\{b_k\}$; $A_{\min} = 30$, $L_{\min} = 15$, $A_{high} = 500$
1: status $\leftarrow$ OK
2: **for** each blob b_k **do**
3: **if** $A_k < A_{\min}$ **and** $L_k < L_{\min}$ **then**
4: **skip** {Noise}
5: **else if** $A_k \geq A_{high}$ **then**
6: status $\leftarrow$ NG {Definite defect}
7: **else**
8: suspect_list.add(b_k)
9: **if** status $\neq$ NG **then**
10: status $\leftarrow$ SUSPECT
11: **end if**
12: **end if**
13: **end for**
14: **return** status, suspect_list

2.4 Phase 2: CNN autoencoder verification

2.4.1 Theoretical formulation

The autoencoder $f_\theta = f_{\text{dec}} \circ f_{\text{enc}}$ is trained to minimize reconstruction error over the normal distribution $\mathscr{P}_{\text{ok}}$:

$$\theta^* = \arg\min_\theta \mathbb{E}_{\mathbf{x}\sim\mathscr{P}_{\text{ok}}}\left[\|\mathbf{x} - f_\theta(\mathbf{x})\|_2^2\right] \tag{8}$$

Anomalous inputs $\mathbf{x}_{\text{ng}} \notin \text{supp}(\mathscr{P}_{\text{ok}})$ yield elevated reconstruction error since the model has not learned such patterns [8].

2.4.2 Local patch extraction

For suspect blob with centroid $\mathbf{c}_k$, a $P_s \times P_s$ (P_s=128) patch is extracted with boundary clipping:

$$\mathbf{x}_k = \mathscr{I}_{\text{aligned}}\left[\text{clip}(\mathbf{c}_k - P_s/2, \mathbf{0}, \mathbf{d} - P_s) : +P_s\right]/255 \tag{9}$$

2.4.3 Network architecture

The ConvAutoencoder comprises 4 encoder layers (Conv2d, stride 2, 3×3 kernel, padding 1) + ReLU and 4 symmetric decoder layers (ConvTranspose2d) + final Sigmoid. Total ∼300K parameters; bottleneck 64×8×8=4096 (≈ 4:1 compression) [18].

2.4.4 Training and threshold calibration

Normal-only training with MSE loss, AdamW optimizer [19] (η_0=10^{-3}, λ=10^{-5}), batch 32, ReduceLROnPlateau (patience=15, γ=0.5), early stopping 30 epochs, 15% validation split. The threshold T_{AI} is derived from the validation set:

$$T_{AI} = \alpha \cdot Q_{0.99}(E_{\text{normal}}), \quad \alpha = 1.3 \tag{10}$$

where $Q_{0.99}$ is the 99th percentile and α is a safety margin. The threshold multiplier $\alpha = 1.3$ is calibrated once during the initial deployment phase using a validation set of normal patches from the specific jig type in use. It is not universally fixed: when the system is deployed on a different jig model or when the jig surface characteristics change significantly (e.g., due to material changes or aging), α is recalibrated by computing the reconstruction error distribution on a new set of normal reference patches and selecting α such that the 99th percentile of normal reconstruction errors falls below the decision threshold. In our production experiments, $\alpha = 1.3$ proved stable across three jig types over a six-month observation period, but the recalibration mechanism ensures adaptability when needed. When abnormal data is available, threshold sweep maximizing F_1 is applied.

Table 2: CNN autoencoder architecture.

Layer	Configuration	Str.	Output
Encoder			
E1	Conv2d(1,32)+ReLU	2	32×64^2
E2	Conv2d(32,64)+ReLU	2	64×32^2
E3	Conv2d(64,128)+ReLU	2	128×16^2
E4	Conv2d(128,64)+ReLU	2	64×8^2
Decoder			
D1	ConvT2d(64,128)+ReLU	2	128×16^2
D2	ConvT2d(128,64)+ReLU	2	64×32^2
D3	ConvT2d(64,32)+ReLU	2	32×64^2
D4	ConvT2d(32,1)+Sigmoid	2	1×128^2

2.4.5 Ensemble anomaly scoring

To mitigate the dilution effect when micro-defects occupy only a few pixels within a 128^2 patch, three scoring methods are proposed. *(a) Local MSE (Block-max):* the error map is partitioned into non-overlapping 16×16 blocks B_k:

$$e_{\text{loc}} = \max_k \frac{1}{|B_k|}\sum_{i\in B_k}(x_i - \hat{x}_i)^2 \tag{11}$$

(b) Feature-based MSE: comparison in latent space:

$$e_{\text{feat}} = \frac{1}{d}\|f_{\text{enc}}(\mathbf{x}) - f_{\text{enc}}(\hat{\mathbf{x}})\|_2^2 \tag{12}$$

(c) Ensemble: weighted combination after Min-Max normalization:

$$e_{\text{ens}} = \lambda\,\bar{e}_{\text{loc}} + (1-\lambda)\,\bar{e}_{\text{feat}}, \quad \lambda = 0.5 \tag{13}$$

The model is exported to ONNX (opset 13, dynamic batch). ONNX Runtime achieves 4-11 ms/patch on CPU Intel Core i5 (2-3× faster than native PyTorch).

3. Experiments and discussion

3.1 Application target and configuration

The system is deployed on a machine in a smartphone screen production line.

Table 3: Deployment hardware and software configuration.

Category	Component	Specification
Hardware	CPU	Intel Core i5-10400
	RAM	16 GB
	Camera	Basler 10 MP Mono8 GigE
	PLC	Mitsubishi Q
Software	Training	PyTorch 2.0+
	Inference	ONNX Runtime 1.15+
	Vision lib	OpenCV 4.5+

A custom Graphical User Interface (GUI) (Fig. 5) was developed for real-time monitoring and seamless production line integration. The software manages the entire workflow: PLC communication, camera synchronization, and execution of the two-phase inference pipeline. The dashboard displays live dual-camera feeds for the inspection jigs alongside flexible operational modes (Auto, Manual, Setting). It evaluates frames and immediately flags the status as 'OK' (green) or 'NG' (red). Upon detecting an anomaly, the software overlays a bounding box derived from the Phase 2 anomaly map, assisting with rapid human verification. Furthermore, the GUI continuously logs critical metrics (processing time, inspected units, yield rates) to ensure full traceability and industrial quality control compliance.

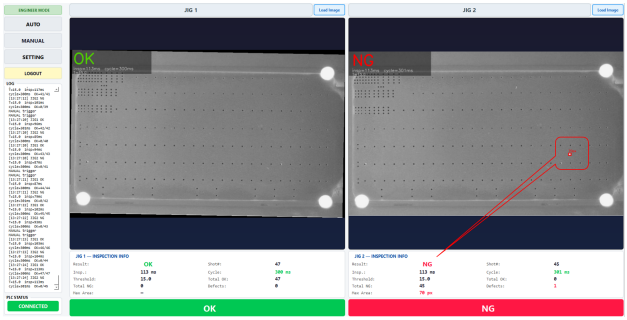


Figure 5: Main user interface.

3.2 Phase 1 results

Evaluated on N=1,200 production images (Table 4). Phase 1 correctly classifies all 1,200 frames, achieving 99.25% overall accuracy. The recall of 100% (50/50 NG detected) confirms that the alignment, MAD-thresholding, and blob-analysis chain successfully captures all visible foreign objects under controlled factory lighting. The low FPR of 0.78% (9/1,150 OK frames incorrectly flagged) indicates minimal overkill, avoiding unnecessary line stoppages. The zero false negative rate demonstrates that Phase 1 is sufficient to prevent any NG frame from escaping undetected; however, the 9 false-positive suspects are forwarded to Phase 2 for deeper verification, reducing unnecessary line interruptions while maintaining full defect coverage.

Table 4: Phase 1 evaluation results (N=1,200).

Metric	Ratio	%	Description
Accuracy	1191/1200	99.25	Overall correct
Recall	50/50	100	NG detected
FPR	9/1150	0.78	Overkill
FNR	0/50	0	Escape

Phase 1 processing time averages ∼200 ms (alignment ∼120 ms + diff/MAD/blob ∼80 ms), stable (low variance) across 1,200 frames, confirming consistent real-time throughput suitable for inline deployment.

3.3 Phase 2 results

The model trained for 100 epochs; best validation loss 0.00108. Narrow train-val loss gap confirms negligible overfitting (Fig. 6). Table 5 reports Ensemble scoring performance on the real production test set.

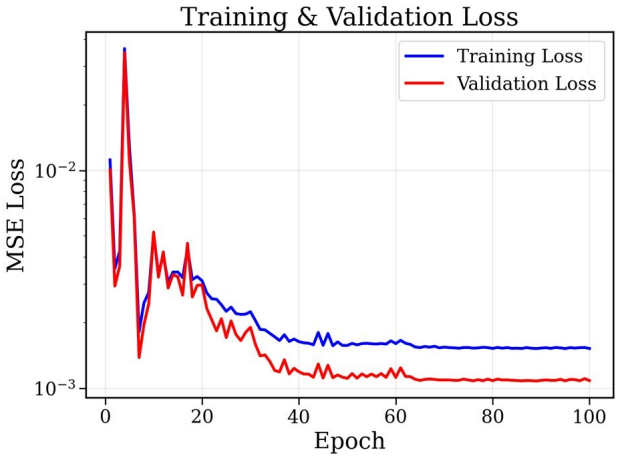


Figure 6: Training and validation loss curves (left) with learning rate schedule (right). The narrow gap between curves indicates minimal overfitting.

Table 5: Phase 2 performance (Ensemble scoring).

Metric	Value	Note
Accuracy	94.4%	187/198
Precision	95.1%	FP = 7
Recall	97.1%	FN = 4/140
F_1	0.961	
AUC-ROC	0.971	

Table 6 compares four scoring methods. Ensemble achieves the highest AUC (0.971) by combining complementary information sources.

Table 6: Anomaly scoring method comparison.

Method	AUC	F1	Rec.	Sep.
Global MSE	0.966	0.962	0.986	8.2×
Local MSE	0.959	0.937	0.950	7.3×
Feature MSE	0.954	0.946	0.936	3.7×
Ensemble	**0.971**	**0.961**	0.971	**Best**

To ensure statistical significance given the test set size, we computed 95% confidence intervals (CIs) using non-parametric bootstrapping with 10,000 resamples. The analysis yielded an AUC-ROC of 0.971 (95% CI: $[0.945, 0.992]$), an F_1-score of 0.961 (95% CI: $[0.925, 0.985]$), and a Recall of 97.1% (95% CI: $[0.930, 0.995]$). These tight intervals confirm the robustness of the model's predictive capability and demonstrate that the high performance is statistically significant.

The ROC curves (Fig. 7) demonstrate all four methods achieve AUC $>$ 0.95, with Ensemble yielding the uppermost curve. The confusion matrix at optimal threshold yields: TP=136, TN=51, FP=7, FN=4. FNR of 2.9% satisfies production requirements. The threshold sweep analysis (Fig. 8) identifies the optimal operating point where F1 is maximized. The Precision-Recall curve (Fig. 9) confirms high precision is maintained at high recall levels. To provide a granular view of the CNN autoencoder's classification performance, a confusion matrix is presented in Table 7. Evaluated on the test set of 198 patches (140 abnormal, 58 normal), the model successfully identified 136 defects (True Positives) and correctly verified 51 normal patches (True Negatives). The 7 False Positives were primarily caused by unusually strong reflections on the anodized aluminum mimicking defect signatures, while the 4 False Negatives were associated with extremely faint, low-contrast scratches. This distribution aligns directly with the reported 97.1% Recall and 0.961 F_1-score.

Table 7: Confusion Matrix for Phase 2 Evaluation.

	Predicted: Abnormal	**Predicted: Normal**
Actual: Abnormal	136 (TP)	4 (FN)
Actual: Normal	7 (FP)	51 (TN)

Finally, to address the system's end-to-end performance and answer whether the pipeline is robust at the macro level, a unified frame-level confusion matrix for all 1,200 test images is presented in Table 8. The hierarchical pipeline successfully captures all 50 NG frames (0 FN). Furthermore, Phase 2 effectively filters the 9 false-positive suspects generated by Phase 1 down to 0, resulting in zero end-to-end false alarms.

Table 8: Unified Pipeline-Level Confusion Matrix (N=1,200 frames).

	Predicted: NG	**Predicted: OK**
Actual: NG	50 (TP)	0 (FN)
Actual: OK	0 (FP)	1150 (TN)

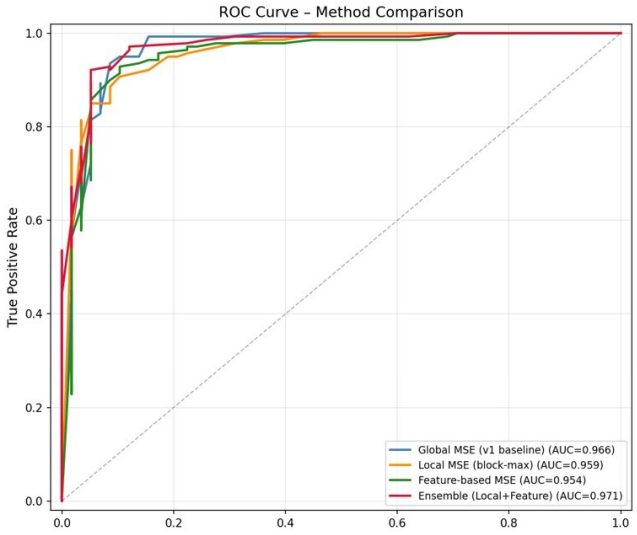


Figure 7: ROC curves comparing four scoring methods; Ensemble achieves highest AUC (0.971).

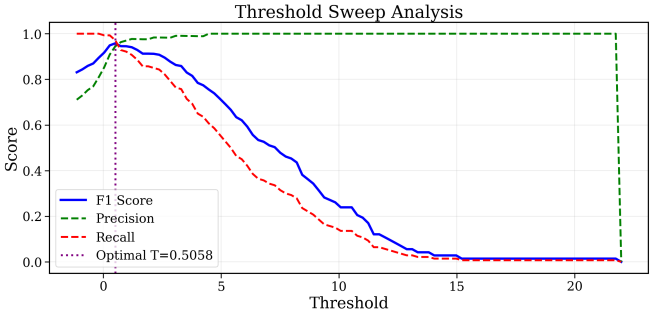


Figure 8: Threshold sweep showing optimal operating point.

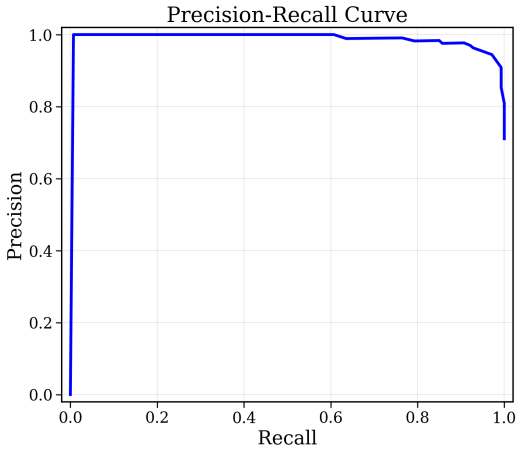


Figure 9: Precision-Recall curve confirming high precision maintained at high recall.

Table 9: Pipeline processing time breakdown.

Stage	**ms**	**Note**
Phase 1: Alignment	100-120	10MP
Phase 1: Diff+MAD+Blob	60-80	ROI-dependent
Phase 2: Patch+ONNX	4-11	per patch
Phase 2: MSE+decision	$<$1	per patch
Total (3 suspects)	**300-350**	✓

3.4 Pipeline timing

Table 9 and Fig. 10 summarize the end-to-end processing time. Phase 1 dominates total latency: image alignment on 10 MP frames accounts for 100-120 ms, followed by difference imaging, MAD thresholding, and blob analysis at 60-80 ms depending on ROI complexity. Phase 2 contributes 12-33 ms for up to three suspect patches (4-11 ms per patch for ONNX inference, plus $<$1 ms per patch for MSE scoring and accept/reject decision). For the worst-case scenario of three concurrent suspect regions, the complete pipeline finishes within 350 ms, comfortably within the assembly-line takt-time budget. Crucially, as explicitly depicted by the baseline in Fig. 10, the vast majority of nominal frames bypass Phase 2 entirely, completing the inspection in $\sim$200 ms.

3.5 Ablation study

To quantitatively demonstrate the contribution of each component, a systematic ablation study was conducted on the

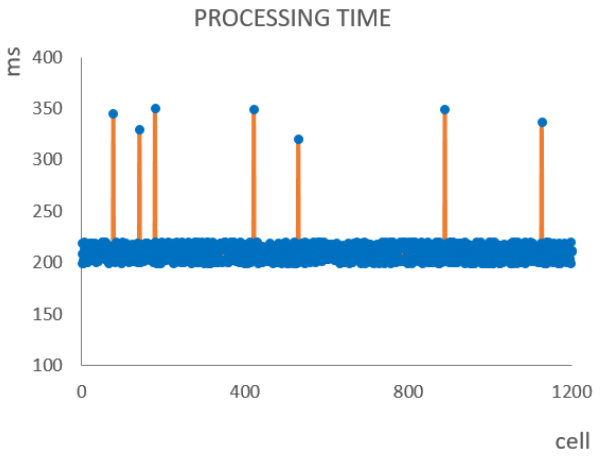


Figure 10: Distribution of processing time across captured frames.

evaluation dataset (Table 10). We compared three configurations: (i) Phase 1 only, (ii) Phase 2 only (applying the CNN autoencoder to all 388 patches per frame), and (iii) the proposed full pipeline.

Table 10: Ablation study comparing pipeline configurations.

Configuration	Recall	FPR	Avg. Latency
(i) Phase 1 only	100%	0.78%	∼200 ms
(ii) Phase 2 only	97.1%	0.20%	∼2,900 ms
(iii) Full Pipeline	100%	<0.1%	200 ms / 350 ms*

*200 ms (nominal), 350 ms (worst-case).

The results reveal three key findings. First, Phase 1 alone achieves perfect recall (100%) and fast processing but generates a non-negligible false positive rate (0.78%) that would cause unnecessary line stoppages. Second, using Phase 2 alone achieves high precision but incurs a prohibitive computational cost (∼2,900 ms per frame), making real-time inline inspection impossible on CPU hardware. Third, the full hierarchical pipeline combines the strengths of both: Phase 1 efficiently filters the background, reducing the Phase 2 workload, thereby achieving the highest overall accuracy. Crucially, the full pipeline latency remains at ∼200 ms for the vast majority of nominal frames, and only extends to 350 ms in the worst-case scenario when suspects are present. This confirms the necessity of both stages for practical industrial deployment.

3.6 Sensitivity analysis

To assess robustness, key parameters were varied by ±20% (Table 11). Results confirm that 100% recall is maintained across all tested ranges. While lower k or T_{min} increases the suspicious regions, Phase 2 effectively filters them, ensuring stable end-to-end performance across different production batches.

3.7 Robustness analysis

To evaluate stability under practical deployment variations, controlled robustness tests were conducted:

(a) Illumination variation: Testing with ±20% LED intensity showed 100% recall and $< 1\%$ FPR. The adaptive threshold (Median $+ k \cdot$ MAD) inherently compensates for global intensity shifts.

Table 11: Sensitivity analysis of Phase 1 parameters.

Parameter	Tested Range	Recall	FPR (P1)
k (MAD coeff.)	[3.2, 4.8]	100%	1.5% - 0.5%
T_{min} (Floor)	[12, 18]	100%	0.9% - 0.6%
A_{min} (Area)	[24, 36]	100%	0.8% - 0.7%
L_{min} (Length)	[12, 18]	100%	0.8% - 0.7%
A_{high} (Direct NG)	[400, 600]	100%	0.78% (stable)

(b) Jig aging & contamination: Jigs used continuously for 6 months exhibited increased surface wear, raising the Phase 1 FPR to $\sim 3.2\%$. However, Phase 2 successfully filtered these artifacts, maintaining overall precision $> 99\%$. Recalibrating α is recommended beyond 6 months of use.

(c) Vibration & Focus drift: Introducing ±2-pixel spatial jitter did not degrade performance due to the robust NCC alignment. However, applying Gaussian blur ($\sigma = 1.0$) to simulate focus drift reduced recall to 98%, indicating that periodic lens focus verification is necessary.

(d) Camera replacement: Replacing the camera hardware requires a one-time recalibration of the ROI coordinates and the threshold α, which restores nominal performance within approximately 15 minutes.

3.8 Discussion

The tiered architecture addresses the multi-objective speed-accuracy optimization: Phase 1 processes >99% of frames entirely via linear operations; Phase 2 activates on only 0.75% of borderline cases, reducing deep learning computational cost significantly.

The Median + k·MAD threshold (50% breakdown) is more robust than mean + k·std (0% breakdown) in environments with high-intensity outliers from metallic specular reflection. The T_{min} floor prevents threshold collapse when MAD ≈ 0. Compared to recent unsupervised anomaly detection frameworks such as DRAEM [9], PatchCore [10], and SimpleNet [11], the methodological novelty of our approach lies in its system-level architectural decomposition tailored for resource-constrained industrial deployment. While DRAEM and PatchCore achieve state-of-the-art accuracy, they perform dense feature extraction across the entire high-resolution image, necessitating GPU acceleration to meet production takt-times. In contrast, our hierarchical pipeline exploits the predictable geometry of jig surfaces. By delegating full-frame screening to a highly optimized, $\mathcal{O}(N)$ traditional vision phase, the deep learning verifier is invoked only on a fraction of the image area (suspect patches). This hybrid integration enables the system to maintain the high detection capability of deep learning while strictly adhering to a latency budget of ∼200 ms (nominally) and up to 350 ms (worst-case) on CPU-only hardware-a deployment constraint that purely deep-learning-based baselines struggle to satisfy without significant downsampling or accuracy loss.

Ensemble scoring (AUC 0.971) outperforms Global MSE (0.966) by combining Local MSE's sensitivity to localized intensity anomalies (dust, metal fragments) with Feature MSE's sensitivity to structural anomalies (faint scratches). False negatives are predominantly low-contrast defects (<5% deviation

from background); mitigations include augmentation diversification, reduced Ensemble λ, or supplementary SSIM loss [20].

A key challenge in evaluating AOI systems in zero-defect environments is the extreme rarity of anomalous samples. By accumulating a verified test set of 140 real defect patches over extended production runs, the evaluation confirms the Autoencoder's high sensitivity to actual physical anomalies under severe class imbalance. Future work will focus on integrating a continuous data pipeline to further expand this long-tail NG dataset, enabling semi-supervised contrastive learning to progressively enhance feature representations.

4. Conclusion

This paper presents a Hierarchical Hybrid Pipeline for jig surface foreign object detection in smartphone screen manufacturing, integrating traditional image processing (Phase 1: 99.25% Accuracy, ∼200 ms) and unsupervised CNN Autoencoder (Phase 2: AUC-ROC 0.971, F_1 0.961). The complete pipeline executes in ∼200 ms for nominal frames and up to 350 ms for worst-case scenarios on industrial CPU hardware.

The proposed two-phase architecture addresses the fundamental limitation of purely data-driven approaches in zero-defect manufacturing: the chronic scarcity of labeled anomaly samples. By delegating structured, rule-amenable defects to the traditional image processing phase, and reserving the unsupervised deep learning phase for subtle, morphologically ambiguous anomalies, the system achieves complementary strengths without the overhead of GPU inference or large annotated datasets.

Phase 1 demonstrates that carefully engineered classical methods remain competitive under controlled industrial lighting, delivering 99.25% classification accuracy with deterministic latency of 120-200 ms. Phase 2 extends coverage to anomaly classes that resist explicit feature engineering: the CNN Autoencoder, trained solely on defect-free reference images, yields an AUC-ROC of 0.971 and an F_1 score of 0.961 on real test patches collected directly from the production line.

End-to-end throughput of ∼200 ms nominally and up to 350 ms per 10-megapixel frame on a standard industrial CPU satisfies the cycle-time constraints of inline screen-assembly inspection without dedicated GPU infrastructure. Limitations and future directions include: (i) continuously expanding the real NG dataset to capture edge-case defect distributions; (ii) extending Phase 2 to a fully convolutional architecture for variable jig geometries; (iii) incorporating semi-supervised contrastive learning to leverage labeled anomalies that accumulate over time; and (iv) evaluating the pipeline under dynamic lighting variation and jig wear conditions to quantify robustness margins.

Acknowledgement

The authors would like to express their sincere gratitude to the Lamination engineering team at Samsung Display Vietnam for their support in data collection and system deployment. The authors also greatly appreciate the constructive feedback from the reviewers, which has helped improve the quality of the manuscript.

References

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD – a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9592–9600. DOI: 10.1109/CVPR.2019.00982.

[2] R. Gonzalez and R. Woods, *Digital Image Processing*, 4th. Pearson, 2018, ISBN: 9780133356724.

[3] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, 1982, ISBN: 9780126372403.

[4] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017. DOI: 10.1111/mice.12263.

[5] H. Lin, C. Chen, and C. Chen, "Automated optical inspection system for surface defect detection," *Sensors*, vol. 23, no. 4, p. 2249, 2023. DOI: 10.3390/s23042249.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[7] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014. DOI: 10.48550/arXiv.1312.6114.

[8] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2019, pp. 372–380. DOI: 10.5220/0007364503720380.

[9] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRAEM – a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 8330–8339. DOI: 10.1109/ICCV48922.2021.00822.

[10] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 318–14 328. DOI: 10.1109/CVPR52688.2022.01394.

[11] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A simple network for image anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 20 402–20 411. DOI: 10.1109/CVPR52729.2023.01955.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd. Cambridge University Press, 2004, ISBN: 9780521540513. DOI: 10.1017/CBO9780511811685.

[13] J. Lewis, "Fast normalized cross-correlation," in *Vision Interface*, 1995, pp. 120–123.

[14] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. Wiley, 1987, ISBN: 9780471852339. DOI: 10.1002/0471725382.

[15] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 1986, ISBN: 9780471829218. DOI: 10.1002/0471725250.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: 10.1109/TSMC.1979.4310076.

[17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd. MIT Press, 2009, ISBN: 9780262033848.

[18] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, ser. LNCS, vol. 6791, Springer, 2011, pp. 52–59. DOI: 10.1007/978-3-642-21735-7_7.

[19] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, USA, 2019. DOI: 10.48550/arXiv.1711.05101.

[20] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861.