Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 11(5): 14487–14521.
DOI:10.3934/math.2026594
Received: 20 March 2026
Revised: 14 April 2026
Accepted: 24 April 2026
Published: 22 May 2026

Research article

A decision-level fusion hybrid deep learning framework for high-precision automated optical inspection of VCSEL semiconductor devices

Kyu-Jeong Choi[1,2] **and Jin-Taek Seong**[1,*]

[1] Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea

[2] Opticis Company Ltd., Seongnam-si 13354, Republic of Korea

* **Correspondence:** Email: jtseong@jnu.ac.kr; Tel: +82-62-530-5798.

Abstract: This paper presents a hybrid deep learning framework for the automated optical inspection (AOI) of vertical cavity surface-emitting laser (VCSEL) semiconductor devices. Manual inspection is limited by subjective inconsistency and operator fatigue, while high-end commercial AOI systems impose substantial costs that are often impractical for small and medium sized manufacturers. The proposed system adopts a decision-level fusion architecture in which defect-specific binary classifiers are aggregated through an OR-gate. This design prioritizes recall so that any defect flagged by at least one sub-classifier triggers rejection, reducing the risk of defect escape. An industrial dataset of 22,410 images with severe class imbalance (e.g., crack defects comprising less than 0.4% of all labels) was used for training, with targeted augmentation applied to minority classes. Five fold cross-validation yielded an overall accuracy of 98.7% and an F1-score of 93.5%. Deployment on an active production line reduced per-unit inspection time from 17.7 s to 1.5 s (an 89% reduction) and recorded a secondary defect escape rate of 0.00%.

Keywords: automated optical inspection; vertical cavity surface-emitting laser; decision-level fusion; hybrid deep learning; surface defect detection
Mathematics Subject Classification: 68T07, 68T45

1. Introduction

Semiconductor-based optical devices have structures at the micro-scale (μm). Defects arising during fabrication can cause performance degradation or electrical failure. Because detecting such fine-scale patterns by visual inspection is difficult, automated visual inspection systems with high-resolution imaging are widely used for quality assurance. However, manual inspection is limited by operator-dependent proficiency, fatigue, and subjective judgment; empirical studies indicate that

manual diagnostic accuracy typically plateaus at approximately 80% [1].

To address these limitations, automated optical inspection (AOI) technology has been widely adopted across the industrial sector. Standard AOI systems use high-magnification optical components, precision XY translation stages, and stabilized illumination modules for automated morphological analysis of semiconductor devices. However, high-end AOI systems for micro-scale optical devices require significant capital expenditure for specialized hardware and optical configuration. Furthermore, recurring costs for system maintenance and upgrades limit their adoption within small-to-medium enterprise (SME) manufacturing environments [2, 3].

Due to these limitations, recent research has increasingly adopted deep learning-based image analytics as a cost-effective and scalable alternative to traditional AOI frameworks [4,5]. Deep learning architectures can autonomously extract the defect morphology, spatial patterns, and locations from large-scale datasets [6]. This approach provides high diagnostic accuracy and consistent inspection quality while reducing operational overhead [7].

In this study, we propose a cost-effective AOI pipeline designed for the constraints of semiconductor optical device fabrication. Our methodology differs from single-model approaches by introducing a decision-level fusion framework. This system uses high-resolution imagery as the primary data source and an ensemble of fine-tuned YOLO [8, 9] and ResNet [10, 11] detectors. By aggregating the outputs of these detectors via an OR-gate, we prioritize recall. This configuration ensures that any detected anomaly is flagged for secondary review, reducing the defect escape rate to near-zero levels—a requirement for semiconductor applications where a single defective unit can affect the downstream module.

The primary contributions of this paper are summarized as follows.

- **Decision-level fusion framework:** We implement an OR-gate-based fusion logic that combines multiple defect-specific models, improving the F1-score and diagnostic reliability compared with single-stage detectors. A domain-optimized hybrid inspection framework is proposed that decomposes the multiclass vertical cavity surface-emitting laser (VCSEL) defect detection problem into four independent binary classification tasks, each equipped with a defect-type-specific classifier and a tailored input representation. A heterogeneous input routing strategy is designed in which raw images, chip-level regions of interest (ROIs), and U-Net segmented masks are selectively assigned to the classifiers according to the visual characteristics of each defect type, rather than using a shared input for all models. The framework adopts a parameter-free, decision-level OR-gate fusion that maximizes recall under the cost-asymmetric constraint of semiconductor manufacturing, where false negatives carry substantially higher costs than false positives. The modular architecture enables the incorporation of new defect types by appending additional binary classifiers without retraining the existing modules, supporting production-line adaptability.
- **Handling of imbalanced dataset:** We propose a targeted data augmentation strategy that simulates real-world manufacturing noise, alleviating the performance degradation caused by class imbalance.
- **End-to-end pipeline for downstream integration:** We provide a workflow that bridges raw defect detection and factory use, including the automated generation of digital wafer maps for downstream sorting processes.
- **Cost-efficiency validation:** We demonstrate that our deep learning-based approach achieves

inspection accuracy similar to high-end AOI systems while reducing equipment and maintenance costs.

2. Related work

2.1. Recent advances in semiconductor defect detection

Recent literature includes multiple studies on the application of deep learning architectures for semiconductor defect detection, with various methodologies evaluated across diverse manufacturing domains, as summarized in Table 1. To mitigate class imbalance between nominal and defective samples, previous researchers have used generative adversarial networks (GANs) [12] to synthesize artificial defect patterns, subsequently using YOLOv3 [13] for the identification of anomalies in wafer dies. This GAN-based data augmentation supplemented the limited defect data and improved detection accuracy by approximately 7% [1]. Furthermore, a hybrid architecture integrating ResNet50 [14] and Vision Transformer (ViT) [15] was introduced, which achieved a 6% improvement in classification accuracy by simultaneously capturing both localized textures and global semantic features [16]. By introducing a multi-patch structure to simultaneously learn the local and global features, this model achieved an accuracy improvement of approximately 6% compared with conventional ViT.

Table 1. Comparative analysis of deep learning architectures for surface defect detection in semiconductor manufacturing.

Author and year	Model	Dataset	Limitations	Performance	Ref
Chen et al., 2020	GAN + YOLOv3	Wafer images, total 669 (471 defect, 198 normal)	Limited defect types (particle-centric), lack of versatility	AP 88.72% (+7pt), FPS 13.45	[10]
Wang et al., 2023	IH-ViT	IC appearance images (1501 normal, 542 defect, 11 types → Aug. 7588)	Accuracy in 70% range, small dataset, fixed weights	Accuracy 72.51% (ResNet50 +2.8pt, ViT +6.06pt)	[13]
Ullah et al., 2024	Inverse feature matching CNN	Real-world process chip data (272 normal, 221 abnormal)	Data scarcity, lack of domain diversity	Accuracy 82.43%, F1-score 81.37%	[15]
Tang et al., 2025	RST-YOLOv8	PCB public dataset (693), chip data (584 → augmented 1594)	Limited chip types, domain expansion required	mAP@0.5: PCB 92.8%, chip 94.6%	[17]
Our work, 2026	YOLO, ResNet (decision-level fusion)	VCSEL chip image data (five types, 22410 images)	Limited to image-based, domain expansion required	Accuracy 98.7%, precision 95.5%, recall 91.6%, F1-score 93.5%	-

For cases with small dataset sizes and limited diversity, one study proposed a convolutional neural network (CNN) based chip defect detection model applying Inverse Feature Matching and masking techniques [17]. Using real process chip data, this approach improved the explainability of defect regions and location identification, achieving an F1-score of 81.37% [18]. To improve sensitivity toward sub-pixel surface defects, researchers integrated lightweight modules into the YOLOv8 [8] backbone, yielding a 5.4% increase in mean average precision (mAP) of 0.5 while balancing computational efficiency and detection accuracy [19].

Most prior studies trained models using a limited number of wafer or chip images, exhibiting limitations in data diversity, class imbalance, and small defect detection. In contrast, this paper utilizes a dataset with higher diversity compared with existing small-scale datasets. We improved accuracy by applying defect detection via YOLOv5 for each defect type and implementing an OR-gate-based decision-level fusion [20].

Research on integration with downstream processes following defect detection has also been conducted, as shown in Table 2. In downstream processes, results are required that allow the status of each chip (good or defective) to be viewed at a glance, similar to a map of the wafer. Therefore, chips are distinguished within the wafer image to create location coordinates for each, and these locations are populated with normal and defect information in a map format.

Table 2. Literature on downstream process integration and AOI systemization.

Author and year	Model	Dataset	Limitations	Performance	Ref
Yu et al., 2023	Segmentation, Clustered anomaly detection	Wafer die images under actual manufacturing (six types of defects)	Limited defect types, lack of process diversity	Accuracy 97.5%, precision 99.4%, recall 97.9%, F1-score 97.8%	[19]
Du et al., 2023	Modified YOLOv4	GMOC data under actual production	High-cost equipment, environment-dependent	Accuracy 97%, precision 99.5%, recall 100%	[20]
Fu et al., 2024	Position correction + CNN	Wafer / AOI system environment experiment	Limited dataset (single wafer type)	Accuracy 98.0%	[21]
Kim et al., 2024	Xception + CAM-based	Actual wafer buffer zone images (12,869 images, four classes)	Class imbalance, small defect detection limitations	Accuracy 96.9%, precision 89.6%, recall 98.1%, F1-score 94.0%	[22]
Our work, 2026	YOLO, ResNet (decision-level fusion)	VCSEL chip image data (five types, 22,410 images)	Limited to image-based, domain expansion required	Accuracy 98.7%, precision 95.5%, recall 91.6%, F1-score 93.5%	-

Research also exists on automatically identifying defective chips by clustering defect characteristics [21]. Furthermore, studies have developed three-dimensional (3D) AOI platforms to detect defects in realtime [22]. Recently, correction techniques for chip's orientation or position [23]

have been used to improve AOI recognition accuracy. Additionally, research on lightweight classification structures capable of estimating defects' location and size has been conducted to improve realtime performance [24].

These studies focused on image-based deep learning defect detection and investigated the potential for AOI systemization in production environments. However, limitations regarding small defects and imbalanced data persist, feedback loops to downstream AOI processes are insufficient, and evaluations are often limited to model accuracy. This work implements a conservative judgment structure through decision-level fusion, achieving high levels of accuracy and F1-score.

2.2. *Automated optical inspection and imbalance handling*

AOI is a non-destructive testing method that utilizes cameras and image analysis technology to automatically detect defects occurring during the production process. In contrast to manual inspection, which is limited by operator fatigue and subjective judgment, AOI frameworks provide reproducible and standardized evaluations under industrial conditions [25]. AOI systems consist of lighting, lenses, image sensors, and image analysis algorithms, each of which affects the defect detection accuracy. Recently, high-resolution sensors and artificial intelligence (AI)-based image processing technologies have been combined to enable the recognition of fine-scale defects. This technology is used to improve inspection efficiency and quality across various manufacturing sectors, including semiconductors, electronic components, and optical communication modules. In particular, deep learning-based AOI automatically classifies defect types and accumulates inspection data, which are then used for process improvement and quality prediction [26]. AOI is evolving beyond inspection equipment into a quality management system that integrates AI and data analysis.

In manufacturing data, normal product data generally constitutes the majority, while defect data accounts for only a minority. Class imbalance in manufacturing datasets often biases deep learning models toward the majority class during training, reducing sensitivity to low-frequency defect patterns [27]. Representative approaches to address this problem are as follows.

Resampling adjusts the data ratio between classes to increase the opportunity for the model to learn minority class patterns. It is generally divided into oversampling and undersampling [28]. While oversampling techniques such as the synthetic minority oversampling technique (SMOTE) are frequently used to balance class distributions [29, 30], this study uses deterministic image-based augmentation—including rotation and brightness modulation—to preserve the morphological integrity of real-world defect samples while improving spatial generalization. Previous studies often used methods that generate synthetic samples, such as SMOTE [31]. However, in this work, instead of synthetic methods like SMOTE, image-based data augmentation was applied to increase data diversity for the minority class (defects). Specifically, various transformation patterns of the minority class were generated through rotation [32], shifting [33], noise addition [34], and brightness adjustment [35]. This method improves generalization performance compared with simple duplication and allows for the learning of morphological diversity in defect patterns. However, since excessive augmentation can distort the original distribution, the experiments limited the number of augmentations for each defect type. Conversely, undersampling removes a portion of the majority class data to balance the classes [36, 37]. While this can reduce bias toward the majority class, it was not applied in this study due to the risk of losing important information.

Cost-sensitive learning does not directly adjust the sample ratios of classes but differentially applies

misclassification costs so that the model reacts more sensitively to losses in the minority class [38]. Weighted loss functions, which assign weights by class, are common [39] and are effective when the amount of data is limited. However, in this study, since the imbalance was addressed through data augmentation alone, separate cost correction was not applied.

Logit adjustment corrects imbalances in model output by incorporating the prior probability of each class [40,41]. Logit adjustment provides a correction of class-dependent biases by incorporating class-prior probabilities into the loss function. However, in this study, since stable classification performance was achieved through data-based oversampling, additional logit adjustment was not performed.

2.3. Comparison with existing multimodel fusion methods

Multimodel fusion for defect detection has been studied under several paradigms, including model averaging, stacking/boosting ensembles, multitask CNNs, and multimodel fusion with learned aggregation. Vasan et al. [42] proposed an ensemble-based deep learning model for weld defect detection, where multiple feature extractors share the same input and their outputs are averaged at the decision level. Choi et al. [43] applied a boosted stacking ensemble to wafer map pattern classification, training a meta-learner on the outputs of homogeneous base classifiers operating on shared feature representations. Bai et al. [44] reviewed multi-task CNN architectures for material defect detection, noting that shared-backbone designs enable parameter efficiency but may introduce intertask interference when the defect types have dissimilar morphologies. Fan et al. [45] combined AC-GAN-based data augmentation with multimodel fusion for laser weld defect detection across 10 classes, using learned confidence weighting for aggregation.

Table 3 summarizes the structural differences between these approaches and the proposed framework. The key distinction of the proposed approach is threefold. First, each defect type is assigned a dedicated binary classifier rather than being handled by a shared model, enabling independent optimization per defect morphology. Second, each classifier receives a heterogeneous input representation tailored to its target defect—raw images for missing defects, chip-level ROIs for crack defects, and U-Net segmented masks for window and metal defects—unlike conventional ensembles that operate on shared inputs. Third, the decision-level OR-gate aggregation is parameter-free, and a new defect type can be incorporated by appending a new binary classifier without retraining the existing modules.

Table 3. Structural comparison between the proposed framework and existing multimodel fusion approaches for defect detection.

Method category	Fusion level	Input strategy	Defect-type specificity	Extensibility	Ref
Model averaging	Decision	Homogeneous	No	Low	[42]
Stacking/boosting	Feature	Homogeneous	No	Low	[43]
Multitask CNN	Feature	Shared backbone	Partial	Medium	[44]
Multimodel fusion	Feature/decision	Shared/adapted	Partial	Medium	[45]
Proposed	Decision	Heterogeneous	Yes	High	—

3. Proposed decision-level fusion framework

3.1. Data acquisition and dataset construction

The dataset for this study is derived from high-resolution images of VCSEL units, initially fabricated in a wafer state and subsequently singulated into individual chips. These chips, each measuring 220 μm $\times$ 270 μm, are organized into a 35 $\times$ 40 matrix for standardized inspection. Following singulation, the chips undergo a classification process based on electro-optical parameters—including threshold current, optical power, and electrical resistance—using automated sorting equipment.

While conventional methods have relied on manual visual inspection via high-magnification microscopy, this research implements an automated acquisition pipeline. The system integrates a precision XY translation stage with a high-magnification optical camera to ensure consistent image capture across the chip array. Images are acquired at localized coordinates along the x and y axes under stabilized autofocus and constant luminance conditions to reduce environmental noise. The resulting raw data consists of 24-bit grayscale bitmap image file format (BMP) images with a spatial resolution of 768 $\times$ 576 pixels, as shown in Figure 1.

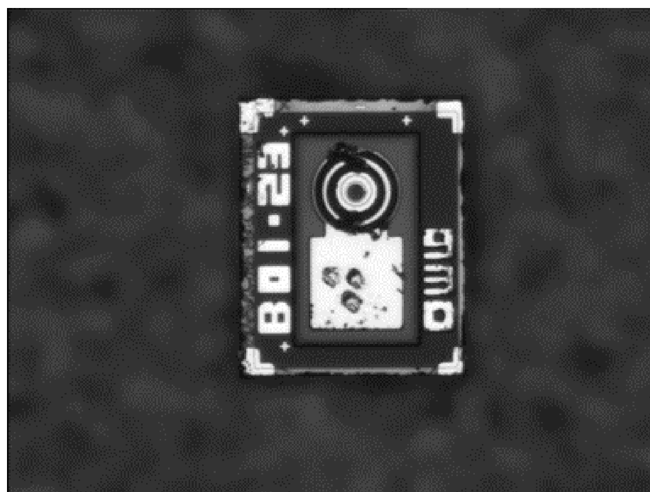


Figure 1. One example of an acquired image.

The image capture program was implemented to acquire images via the equipment shown in Figure 2. To facilitate downstream integration, each image is indexed with its spatial coordinates within the filename. This positional metadata is used for the autonomous generation of digital wafer maps, enabling the identification and sorting of defective units in the subsequent assembly processes.

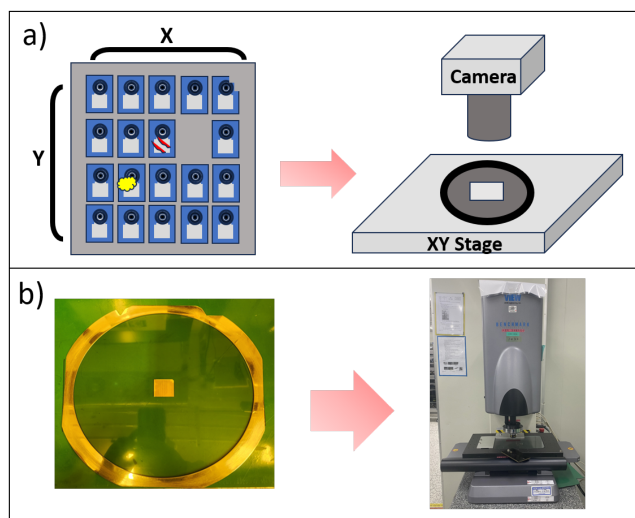


Figure 2. Chip image collection: a) Configuration of the equipment; b) actual configuration.

The dataset comprises 22,410 images with a class imbalance characteristic of industrial manufacturing environments. Specifically, the dataset contains 20,183 nominal (normal) samples (90.1%) and 2227 defective samples (9.9%), encompassing four defect types, namely crack, window, metal, and missing, as shown in Table 4.

Table 4. Quantity and ratio of data images by category.

Category	Crack	Window	Metal	Missing	Normal
Quantity	49	440	487	1251	20,183
Subtotal	2227 (defects)				20,183
Total	22,410				
Ratio (%)	0.2%	2.0%	2.2%	5.6%	90.1%

3.2. Data preprocessing and imbalance mitigation

3.2.1. Defect annotation and labeling

To facilitate supervised learning, localized defect regions were annotated using the LabelImg v1.8.1 tool, with the ground truth data exported in the standard YOLO format. For each defective sample, the annotation file contains the class index and relative bounding box coordinates. Normal samples are represented by empty files, indicating the absence of defects within the captured frame, as shown in Figure 3.

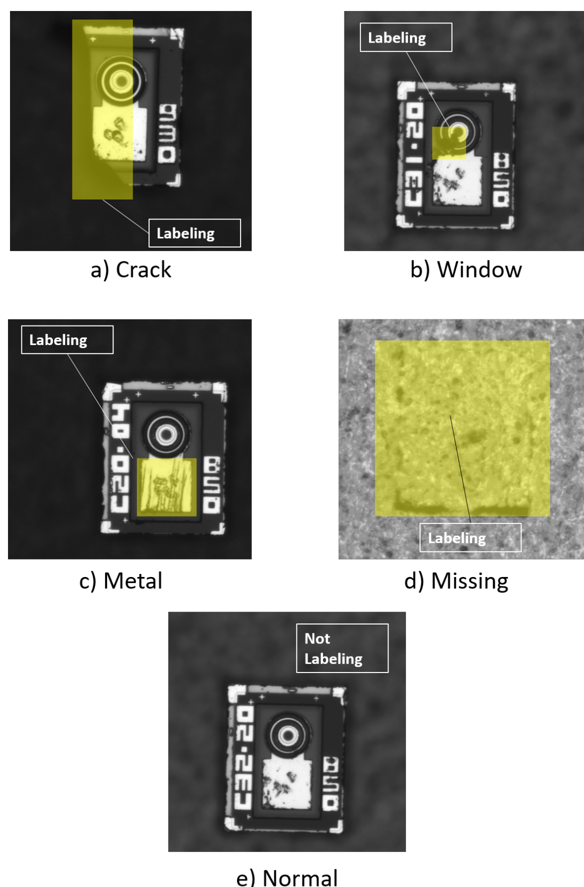


Figure 3. Five label types: a) Crack, b) window, c) metal, d) missing, and e) normal.

3.2.2. U-Net based ROI segmentation

The raw 768 × 576 grayscale images often contain background regions and variable positioning of the target chip within the camera frame, which may introduce background artifacts during feature extraction. To isolate the ROI and improve the signal-to-noise ratio, we implemented a semantic segmentation pipeline based on the U-Net architecture [46], as shown in Figure 4. Ground truth masks were generated using the labelme v5.2.1 annotation tool, defining the boundaries for the head and body regions of the VCSEL chip, as shown in Figure 5.

Table 5 summarizes the architecture, training configuration, and performance of the U-Net segmentation model used in this study. The U-Net follows a five stage encoder and four stage decoder structure with concatenation-based skip connections between corresponding encoder and decoder levels. The input is resized to 512 × 512 pixels with three channels, and the output layer produces a per-pixel classification over five classes (background, head, body, and two boundary regions). The model was trained for 40 epochs with a batch size of 4, using the AdamW optimizer with a learning rate of 1.00×10^{-3} and a weight decay of 1.00×10^{-6}. The loss function combines cross entropy loss and soft dice loss to handle both class imbalance and boundary precision. A validation split ratio of 0.2 was applied, and the best model checkpoint was selected on the basis of the highest validation mean Dice coefficient (mDice).

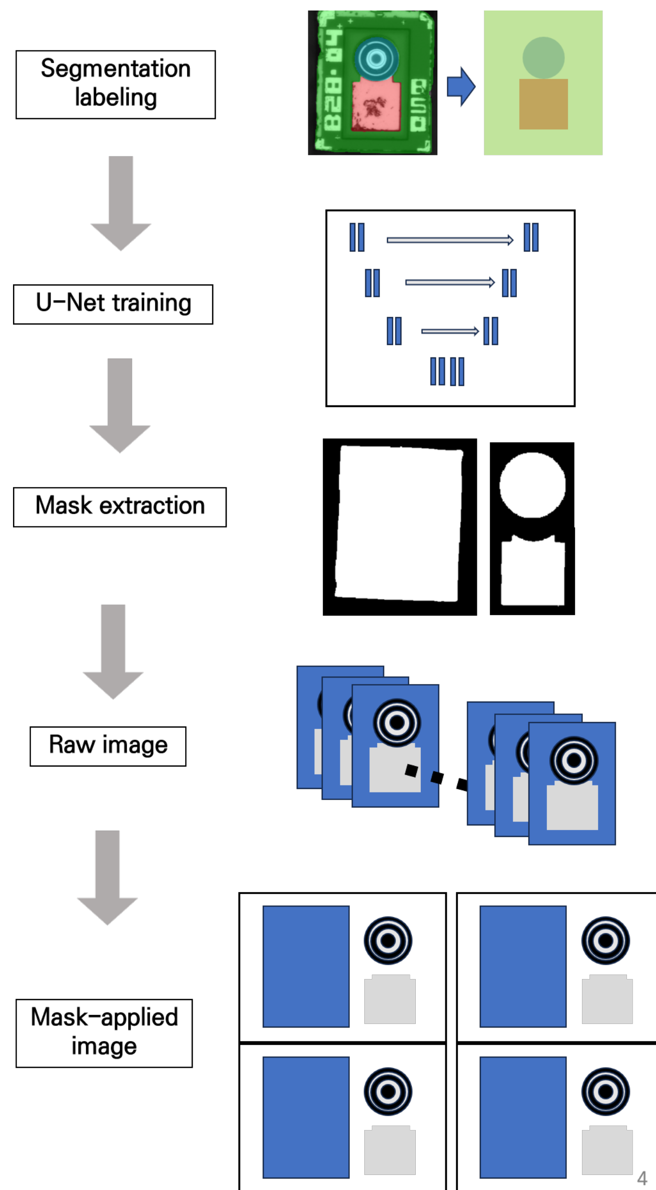


Figure 4. Image segmentation processing: Segmentation labeling, U-Net training, mask extraction, input of raw images, and mask-applied image.

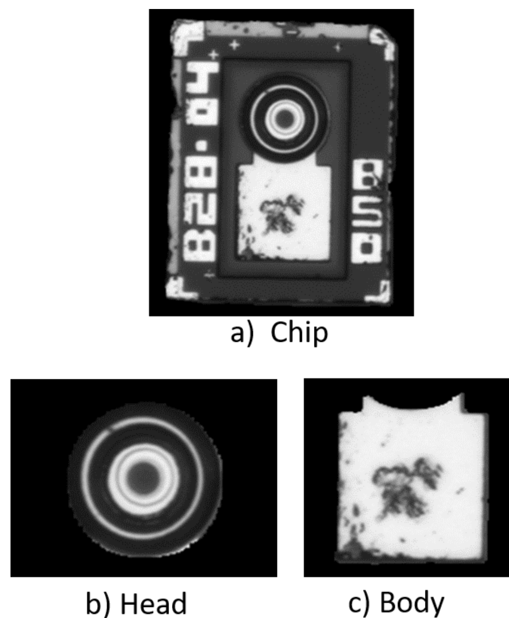


Figure 5. The original chip image a) is masked into two parts: b) head, and c) body.

Table 5. U-Net segmentation model training configuration and performance

Category	Item	Value
Input	Image size	512×512 pixels
Input	Channels	3
Output	Number of classes	5
Architecture	Encoder depth	5 stages
Architecture	Decoder depth	4 stages
Architecture	Skip connection	Concatenation
Loss	Loss function	Cross entropy loss + soft dice loss
Optimizer	Type	AdamW
Optimizer	Learning rate	1.00×10^{-3}
Optimizer	Weight decay	1.00×10^{-6}
Training	Epochs	40
Training	Batch size	4
Training	Validation split	0.2
Selection	Best model criterion	Highest validation mDice
Performance		
	Pixel accuracy	0.992
	Mean intersection over union (IoU)	0.957
	Dice coefficient	0.978

3.2.3. Data augmentation strategies

The raw industrial dataset exhibits class imbalance, where defective samples constitute only 9.9% (2227 out of 22,410 images) of the total population. To reduce the resulting bias toward the majority

class and improve generalization, we implemented a targeted data augmentation strategy. As illustrated in Figure 6, aggressive techniques such as random erasing or excessive structural noise were excluded to prevent the synthesis of artifacts that could be misclassified as foreign matter or surface scratches, preserving the morphological integrity of the original defect patterns.

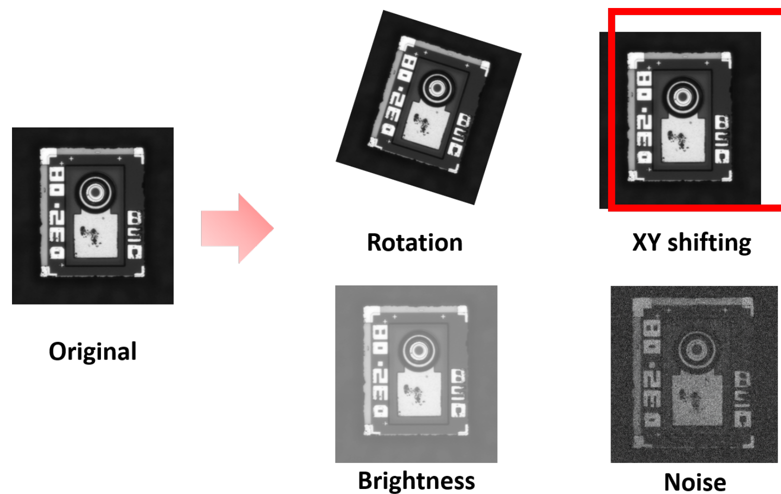


Figure 6. Image augmentation: Rotation, XY shifting, brightness, and noise injection.

To ensure orientation invariance and reduce sensitivity to mechanical misalignments, rotation was applied within ±10° using a uniform distribution. This augmentation emulates scenarios where the VCSEL chip is slightly misaligned during the automated capture process. By using the image center as the rotational axis and applying interpolation to fill vacant regions, the strategy improves spatial generalization and prevents the model from developing directional bias.

To account for spatial displacement and non-uniform chip placement, random translations were applied within a range of up to 12 pixels (approximately 5% of the total image dimensions). This encourages the framework to learn location-invariant features, ensuring detection performance regardless of the defect's position within the frame. Boundary-crossing pixels were managed via periodic interpolation from the opposite axis to maintain structural continuity.

To compensate for photometric variability arising from fluctuating illumination or varying material reflectance, a linear transformation defined as $\alpha \times img + \beta$ was applied to the pixel intensities. The contrast coefficient α and brightness offset β were sampled from the uniform distributions $[0.9, 1.1]$ and $[-10, 10]$ respectively, to simulate diverse luminance conditions. This ensures that the model maintains stable recognition performance under varying lighting conditions.

To improve resilience against sensor-induced artifacts and environmental perturbations, zero-mean Gaussian noise with a standard deviation (σ) between 0.5 and 1.5 was added to each pixel. By normalizing the noise injection within the standard pixel scale (0–255), we prevent excessive distortion while encouraging the model to prioritize the salient structural features over transient imaging noise.

The augmentation multiplier for each defect type was determined by the ratio of normal samples to defective samples so as to achieve an approximate 1:1 class balance; for instance, the crack class ($n = 39$) required a multiplier of $\lfloor 15{,}600/39 \rfloor = \times 400$. Ablation over ×100 to ×500 shows that validation loss is minimized at ×400 and classification recall remains highest at that level, while a slight increase in validation loss at ×500 indicates the onset of overfitting beyond the selected operating point

(see Table 6).

Table 6. Training and validation loss across augmentation levels and training fractions for the crack class.

	×100		×200		×300		×400		×500	
Frac.	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val
0.1	0.00150	0.03405	0.00755	0.01621	0.00013	0.00721	0.00012	0.00574	0.00019	0.00737
0.2	0.00179	0.02187	0.00211	0.00716	0.00007	0.00207	0.00095	0.00211	0.00012	0.00510
0.4	0.00087	0.01176	0.00072	0.00359	0.00031	0.00117	0.00023	0.00061	0.00004	0.00308
0.6	0.00049	0.00586	0.00062	0.00178	0.00035	0.00088	0.00021	0.00037	0.00010	0.00188
0.8	0.00089	0.00434	0.00045	0.00100	0.00025	0.00076	0.00005	0.00015	0.00006	0.00127

3.3. Hybrid architecture and fusion logic design

To classify four defect types (missing, crack, window, and metal) alongside normal units, the proposed framework applies a hybrid input strategy that routes different input representations to defect-type-specific pipelines. Segmented ROIs are used for window and metal defects to remove background artifacts: The U-Net body-region mask is used for window defects and the head-region mask for metal defects. The raw wafer-level image is used for missing defect analysis, as this defect corresponds to the physical absence of a chip and requires wafer-level spatial context. For crack detection, chip-level ROI crops are extracted from units that have passed the missing check.

High-dimensional feature vectors are extracted from these inputs using deep backbone architectures, specifically YOLO and ResNet models. These representations are subsequently classified using machine learning classifiers—including support vector machine (SVM) [47], K-nearest neighbors (KNN) [48], logistic regression [49], Random Forest [50], and gradient boosting [51]—to identify the optimal configuration for binary classification.

As illustrated in Figure 7, the framework applies decision-level fusion through an OR-gate across the four binary outputs. No feature-level operation (e.g., vector concatenation or weighted summation) is performed between the backbone networks; the OR-gate operates solely on the binary classification outputs. This design minimizes the defect escape rate, consistent with semiconductor manufacturing requirements where the cost of a missed defect exceeds that of a false alarm.

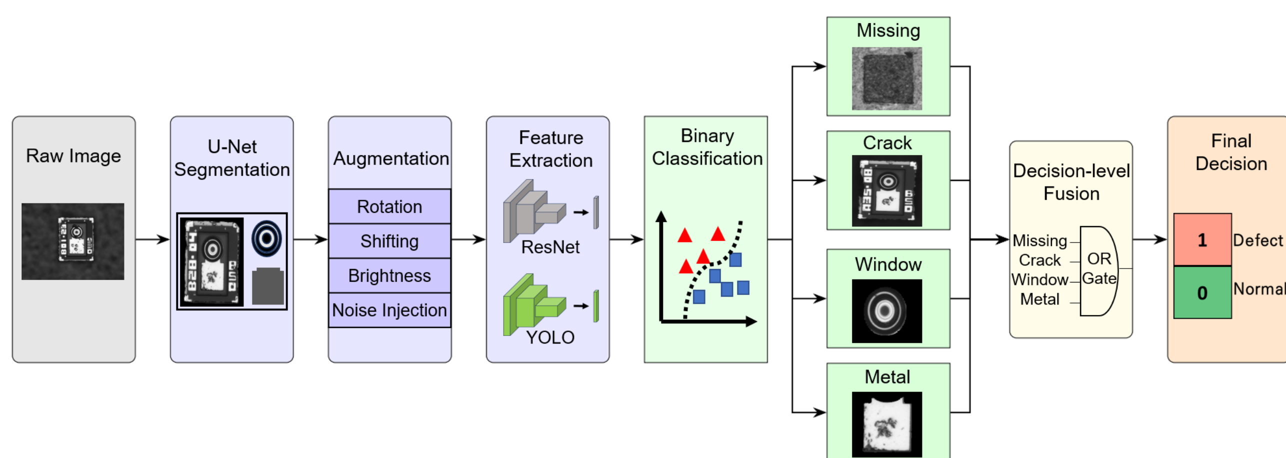


Figure 7. End-to-end data flow of the proposed framework, from raw image input through to U-Net segmentation, augmentation and feature extraction, and binary classification, and then to decision-level OR-gate fusion.

As shown in Figure 8, the feature extraction process consists of two components: A base model that extracts image features from the backbone (ResNet or YOLO), and an extended model that additionally extracts morphological defect features as vectors and concatenates them with the backbone-derived feature vectors before classification.

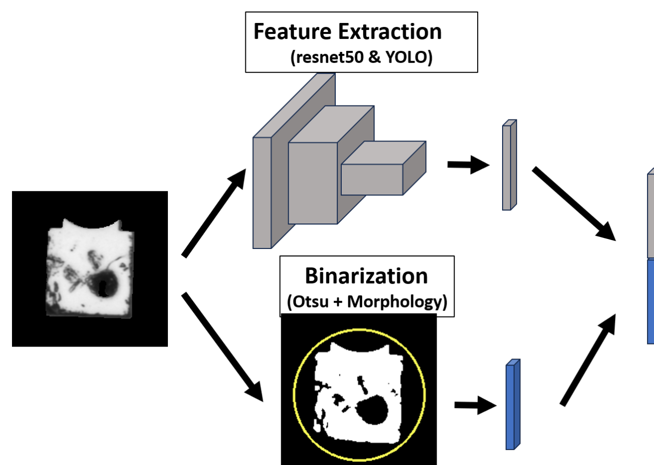


Figure 8. Detail of the feature extraction process.

As shown in Figure 9 and Table 7, the names of the models were defined based on the backbone model, the inclusion of binarization (fusion), and the classification model. The naming convention follows the form 'A_B_C'.

- **'A'** uses the abbreviation of the backbone model (e.g., 'ResNet50' is 'R50'), including the first letter of the model name and version information.
- **'B'** indicates the binarization status, which is either 'O' or 'X'.
- **'C'** uses the first letter of the classification model.

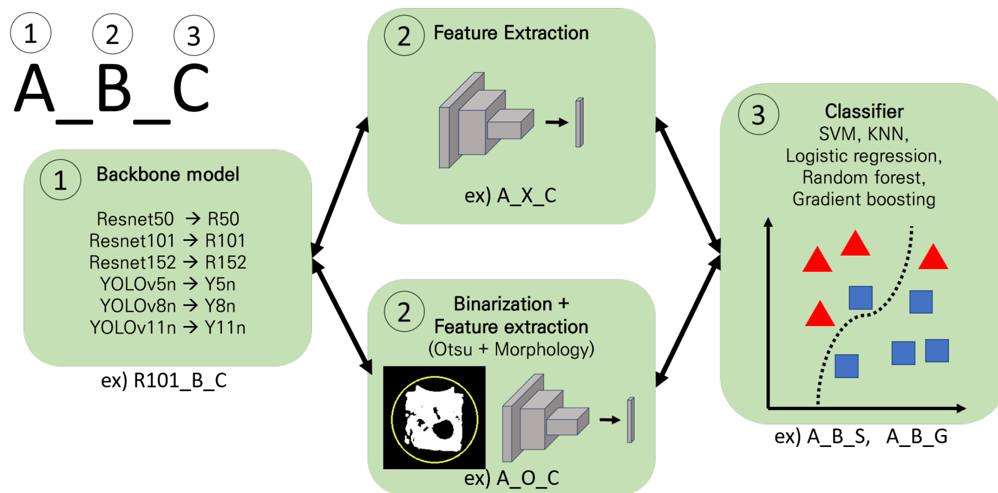


Figure 9. Model architecture and proposed naming convention.

Table 7. Definition of the model's naming conventions.

Model	Backbone (A)	Binarization (B)	Classifier (C)
R50_O_S	ResNet50	O	SVM
R50_X_S	ResNet50	X	SVM
R50_X_L	ResNet50	X	Logistic regression
R50_X_R	ResNet50	X	Random Forest
R50_X_G	ResNet50	X	Gradient boosting
R50_X_K	ResNet50	X	KNN
R101_O_S	ResNet101	O	SVM
R152_O_S	ResNet152	O	SVM
Y5n_O_S	YOLOv5n	O	SVM
Y5n_X_S	YOLOv5n	X	SVM
Y5n_X_L	YOLOv5n	X	Logistic regression
Y5n_X_R	YOLOv5n	X	Random Forest
Y5n_X_G	YOLOv5n	X	Gradient boosting
Y5n_X_K	YOLOv5n	X	KNN
Y8n_O_S	YOLOv8n	O	SVM
Y11n_O_S	YOLOv11n	O	SVM

The models were categorized according to the inclusion of binarization: Those consisting solely of the backbone model were designated as basic models, while those incorporating binarization were designated as hybrid models. In Table 7, R50_X_S represents a basic model, whereas R50_O_S represents a hybrid model.

3.4. *Justification of the OR-gate fusion strategy*

The proposed framework uses an OR-gate as the decision-level fusion operator. This design choice is motivated by the cost-asymmetric nature of semiconductor defect inspection, where the cost of a false

negative (a defective chip escaping to the downstream processes) far exceeds that of a false positive (a normal chip subjected to secondary reinspection). This relationship can be expressed as

$$\text{Total Cost} = N_{\text{FP}} \cdot C_{\text{FP}} + N_{\text{FN}} \cdot C_{\text{FN}}, \quad C_{\text{FN}} \gg C_{\text{FP}}. \tag{3.1}$$

Under this condition, the inspection objective is to minimize N_{FN}, which is equivalent to maximizing recall. The OR-gate achieves this by flagging a chip as defective whenever any one of the four defect-specific classifiers detects an anomaly:

$$D = b_1 \lor b_2 \lor b_3 \lor b_4, \tag{3.2}$$

where $b_i \in \{0, 1\}$ denotes the binary output of the i-th classifier (missing, crack, metal, window), and $D = 1$ indicates a defective classification.

To validate this design, we compared four decision-level fusion strategies on the same test set. The AND-gate and majority voting strategies require at least two classifiers to simultaneously flag a chip as defective ($\sum_{i=1}^{4} b_i \geq 2$). The weighted sum strategy computes the weighted average of the four classifier confidence scores using equal weights ($w_i = 0.25$ for all i) and applies a threshold to produce a binary decision. The results are presented in Table 8.

Table 8. Performance comparison of decision-level fusion strategies.

Fusion strategy	Accuracy	Precision	Recall	F1-score
OR-gate (proposed)	0.985	0.931	0.933	0.932
AND-gate (≥2 agreement)	0.899	1.000	0.027	0.053
Majority voting (≥2 votes)	0.899	1.000	0.027	0.053
Weighted sum (equal weights)	0.985	0.934	0.926	0.930

Three observations follow from the comparison. First, the AND-gate and majority voting strategies both yield a recall of only 0.027. Because each of the four classifiers in the proposed framework is defect-type-specific—trained exclusively for one defect category—a defective chip typically triggers only a single classifier. The requirement for two or more classifiers to agree is therefore rarely met, except in cases of co-occurring multi-type defects. This structural mismatch renders both strategies unsuitable for the defect-type-specialized architecture considered in this study and explains why their results are identical.

Second, the weighted sum strategy with equal weights achieves a recall of 0.926 and an F1-score of 0.930, both lower than the OR-gate (recall: 0.933, F1-score: 0.932). Although the weighted sum yields slightly higher precision (0.934 vs. 0.931), it introduces two practical disadvantages: (1) Weight assignment and threshold selection constitute additional hyperparameters that require calibration and may overfit to the training distribution and (2) the resulting soft decision boundary is less interpretable than the deterministic Boolean logic of the OR-gate, which is a relevant consideration in manufacturing environments subject to quality audit requirements.

Third, the OR-gate produces a precision of 0.931, implying a marginal increase in false positives compared with weighted sum strategy. In the VCSEL production context, these additional false positives are handled through the existing secondary manual re-inspection workflow at low operational cost (a few seconds per chip). Given that the cost of a single undetected defective chip (false negative) is substantially higher than that of re-inspecting a normal chip (false positive), the OR-gate's recall-maximizing property represents the preferred cost-sensitive trade-off for this application.

4. Experimental results and performance analysis

4.1. Dataset profile and defect characteristics

The VCSEL chip is partitioned into three regions—chip, head, and body—according to the defect types and the functional requirements of each region. The Chip region is evaluated for structural cracks; superficial scratches or minor foreign particles in this zone are excluded from defect categorization as they do not compromise the material properties. The head and body regions are inspected for both morphological scratches and contaminants.

A challenge in feature extraction arises from the nonfunctional textual artifacts—such as serial numbers (S/Ns) and wavelength specifications—inscribed on the bonding metal, as illustrated in Figure 10. These inscriptions exhibit high-frequency edge components that the model may interpret as structural noise or surface anomalies.

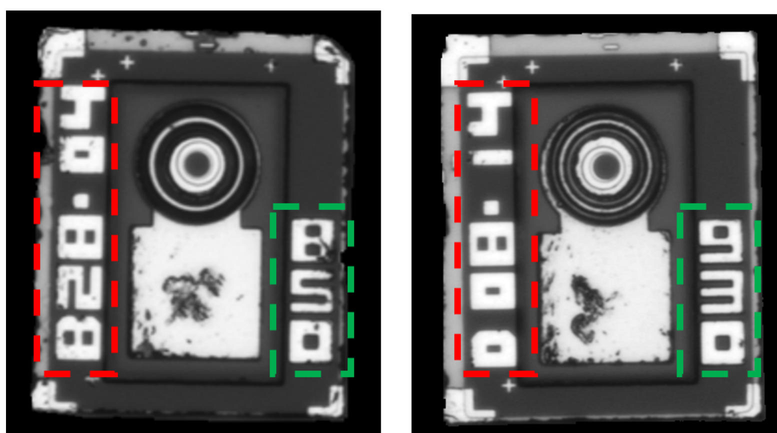


Figure 10. Chip image: S/N (red dotted line) and wavelength (green dotted line).

Furthermore, as shown in Figure 11, window defects present visual variance. While normal units Figures 11a) and 11b) exhibit imaging differences due to manufacturing tolerances, defective units Figures 11c) and 11d) manifest as malformations in the peripheral ring metal or the presence of opaque contaminants. Metal defects, detailed in Figure 12, are complicated by contact marks—remnants of pre-inspection electrical testing. These marks resemble foreign matter, and the grayscale imaging reduces the contrast difference between contact marks and actual metal defects, as the color-space information that would aid distinction is not available.

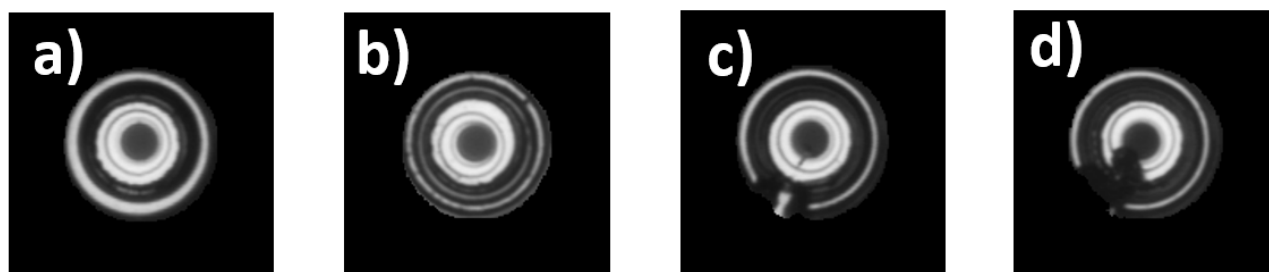


Figure 11. Characteristics of window defect images: a) and b) Normal types, c) ring metal defect, and d) damage to the ring metal and emitter area.

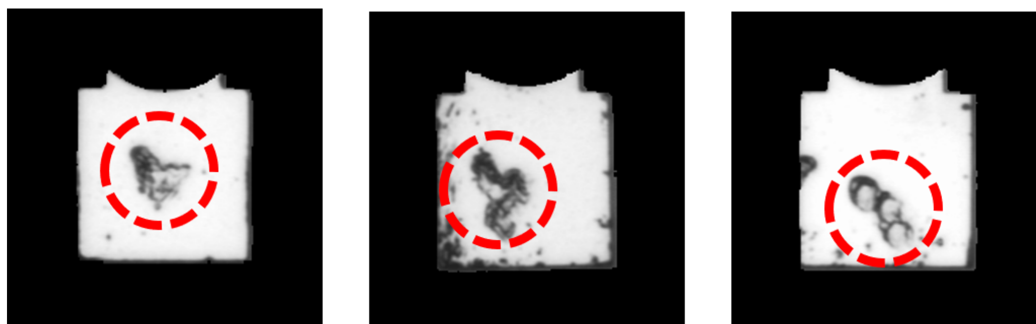


Figure 12. Characteristics of metal defect images.

As shown in Figure 13, missing defects are categorized into two types: The chip is absent, or the chip is flipped. In Figures 13a) and 13b), traces of the chip's outline remain, making it impossible to determine the chip's presence solely on the basis of the outline image. In Figure 13a), where the chip is missing, the camera focuses on the bottom surface, resulting in a bright background with irregular patterns. In Figure 13b), the focus is on the flipped chip, resulting in lower brightness and a blurred background. Actual normal and defect images are shown in Figure 14, and types satisfying the defect criteria are designated as labels for each image.

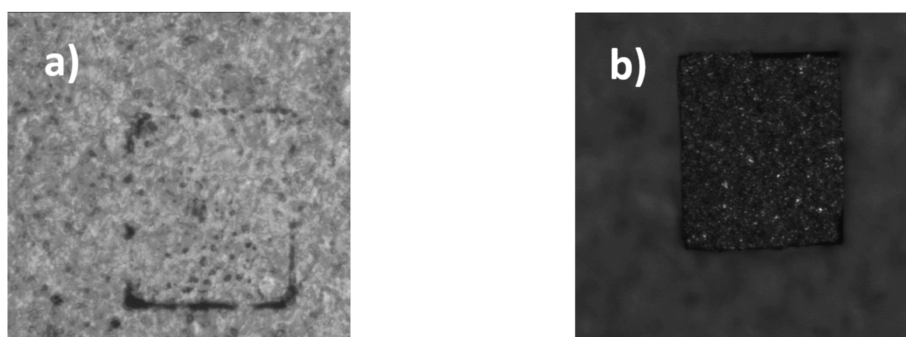


Figure 13. Characteristics of missing defect images: a) empty (chip absence), b) flipped (reversed chip orientation).

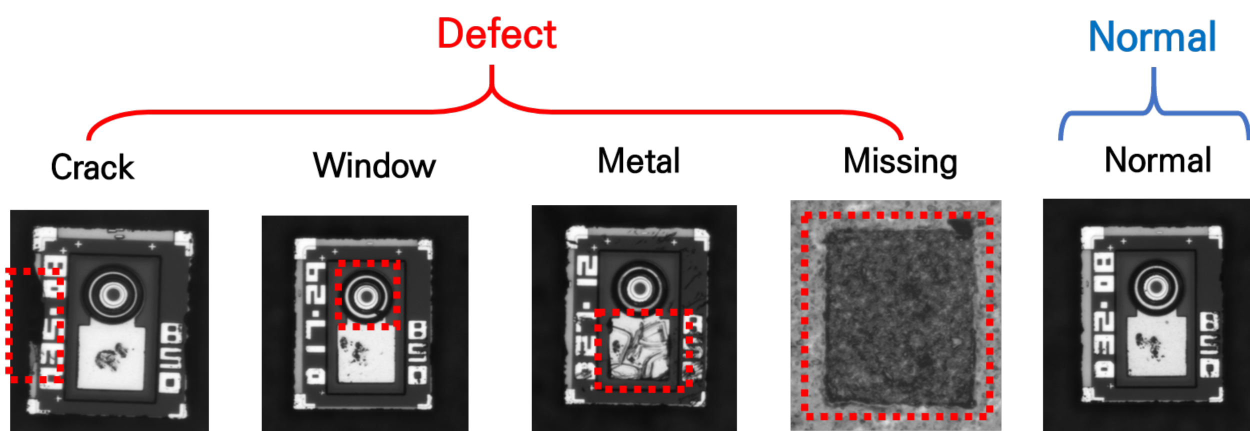


Figure 14. Representative examples of defective and normal chip images: Crack, window, metal, missing, and normal.

4.2. Experimental configuration and evaluation metrics

To ensure statistical robustness and evaluate generalization performance, a five fold cross-validation scheme was implemented. The dataset was initially partitioned into training and test sets using a stratified 8:2 ratio. The training partition was then subdivided into five disjoint folds; in each iteration, four folds were used for model training, while the remaining fold served as the validation set. This approach mitigates splitting bias and allows an assessment of the model's reproducibility.

For field verification, a hold-out set of 5250 chip images was used, with complete separation from the training and validation phases. The industrial validation set comprises 5250 unique chip images annotated with a total of 5314 category-level labels; the 64 additional labels ($\approx$1.2% of all images) originate from chips that exhibit two or more concurrent defect types, as shown in Table 9. Because each subclassifier evaluates every chip independently against its own binary ground truth, multilabel instances do not affect the per-class recall or precision reported in this study. The classwise stability of the five fold cross-validation is reported in Table 10. Across all four classes, the standard deviation of every metric is below 0.21%. Missing achieves zero variance because its sample count is large enough to eliminate fold-level fluctuation. Metal and window defects show slightly larger precision variance (0.178% and 0.204%, respectively) compared with cracks, which is consistent with their higher false-positive counts in the chip-level classification stage. The standard deviations of recall are at most 0.030%, confirming that detection stability is maintained across folds for all defect types.

Table 9. Category-level label distribution of the industrial validation set (5250 unique images with 5314 labels due to multilabel cases).

Category	Label count	Proportion (%)
Normal	4768	89.7
Crack	20	0.4
Metal	97	1.8
Missing	316	5.9
Window	113	2.1
Total (labels)	5314	100.0

Table 10. classwise stability analysis of five fold cross-validation (mean ± standard deviation %)

Class	Accuracy	Precision	Recall	F1-score
Crack	99.953 ± 0.037	99.906 ± 0.073	100.000 ± 0.000	99.953 ± 0.036
Metal	99.711 ± 0.087	99.458 ± 0.178	99.975 ± 0.027	99.716 ± 0.086
Missing	100.000 ± 0.000	100.000 ± 0.000	100.000 ± 0.000	100.000 ± 0.000
Window	99.687 ± 0.100	99.406 ± 0.204	99.980 ± 0.030	99.692 ± 0.098

As shown in Table 11, we implemented a class-balanced training strategy by maintaining a 1:1 ratio between nominal and defective samples. While the normal class remained unaugmented due to its sufficient sample count, the defect minority classes were expanded using four transformation modes: Rotation, shifting, noise injection, and brightness modulation. The augmentation was distributed across the training set with the following ratios: Rotation (20%), shifting (30%), noise (30%), and brightness

adjustment (20%). The missing defect category achieved optimal classification without additional data augmentation.

Table 11. Data augmentation and test set quantities by defect category (B: base, R: rotation, S: shifting, N: noise injection, BT: brightness).

Category	Data augmentation (defect)					Data normal	Training set (defect:normal)	Test set	
	B	R	S	N	BT			Defect	Normal
Missing	992	-	-	-	-	992	(992 : 992)	258	4224
Crack	39	3120	4680	4680	3120	15,600	(15,639 : 15,600)	12	4212
Metal	377	3016	4524	4524	3016	15,080	(15,457 : 15,080)	105	4119
Window	359	2872	4308	4308	2872	14,360	(14,719 : 14,360)	99	4125

4.3. Comparative performance evaluation

To evaluate the performance of the defect detection model, the confusion matrix in Table 12 was used to compute accuracy, precision, recall, and F1-score.

Table 12. Confusion matrix for binary classification

Actual class	Predicted class	
	Defect	Normal
Defect	True defect (TP)	False normal (FN)
Normal	False defect (FP)	True normal (TN)

As shown in Eq (4.1), accuracy represents the ratio of correctly classified normal and defective products. However, due to class imbalance, performance cannot be assessed on the basis of accuracy alone. Therefore, performance was evaluated using precision, recall, and the F1-score via Eqs (4.2)–(4.4). Because the number of defects is small, the F1-score and recall are given greater weight than accuracy. Since this is a defect detection model, recall must be high; however, considering productivity and cost, recall alone is not sufficient as the sole evaluation criterion.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP}. \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN}. \tag{4.3}$$

$$F1\text{-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{4.4}$$

For the basic model, we evaluated five classifiers with ResNet50 and YOLOv5 as backbones. The results were obtained from the test set after training with data augmented to a 1:1 ratio. Although the YOLO model includes defect location information, the ResNet model demonstrated higher

performance. Furthermore, SVM outperformed other classification models across both backbone families.

Figure 15 compares the basic model and the hybrid model, separated by ResNet and YOLO backbones. The hybrid model, which adds morphological features, showed improved performance compared with the basic model that used only ResNet or YOLO. Additionally, as shown in Figure 16, a comparison across backbone model versions identified YOLOv11 and ResNet101 as the best-performing configurations. Table 13 and Figure 17 summarize the performance of R101_O_S by defect type.

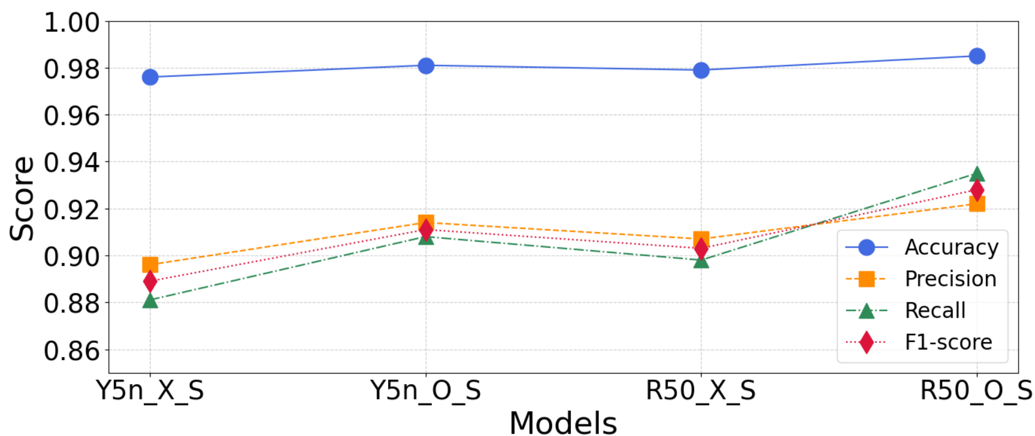


Figure 15. Performance comparison of basic and hybrid models.

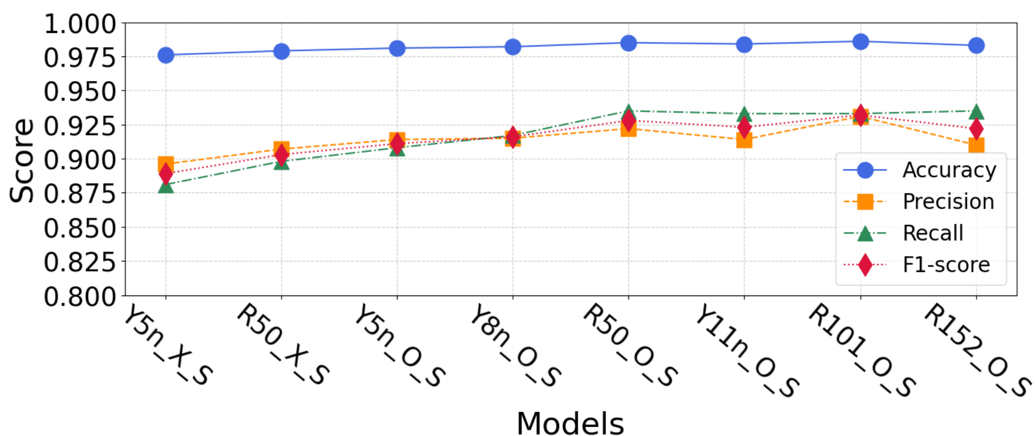


Figure 16. Performance comparison of various models.

Table 13. Performance by defect type for the R101_O_S model.

Defect	Accuracy	Precision	Recall	F1-score
Missing	1.000	1.000	1.000	1.000
Crack	0.999	0.900	0.750	0.857
Metal	0.991	0.888	0.752	0.814
Window	0.993	0.842	0.859	0.850

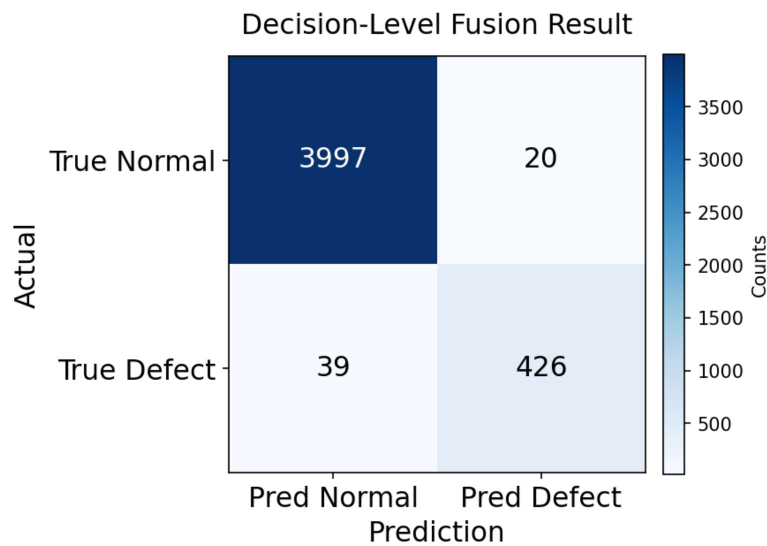


Figure 17. Confusion matrix for the R101_O_S model.

The missing defect type was accurately classified without data augmentation. Classification performance for the crack defect type was affected by the limited number of 12 defect samples in the test set. Due to the characteristics of grayscale images, normal and metal defect patterns appeared similar, leading to a relatively lower detection rate. However, while individual models showed lower detection rates, the decision-level fusion stage identified overlapping defects, improving overall performance. Table 14 presents the evaluation metrics for all models, trained according to the data augmentation quantities specified in Table 11.

Table 14. Performance of various models.

Model	Accuracy	Precision	Recall	F1-score
R50_O_S	0.985	0.922	0.935	0.928
R50_X_S	0.979	0.907	0.898	0.903
R50_X_L	0.978	0.905	0.892	0.898
R50_X_R	0.968	0.823	0.892	0.856
R50_X_G	0.962	0.783	0.890	0.833
R50_X_K	0.967	0.928	0.750	0.829
R101_O_S	0.985	0.931	0.933	0.932
R152_O_S	0.983	0.910	0.916	0.922
Y5n_O_S	0.981	0.914	0.908	0.911
Y5n_X_S	0.976	0.896	0.881	0.889
Y5n_X_L	0.967	0.812	0.900	0.854
Y5n_X_R	0.956	0.768	0.848	0.806
Y5n_X_G	0.940	0.673	0.858	0.755
Y5n_X_K	0.959	0.874	0.725	0.793
Y8n_O_S	0.982	0.915	0.917	0.916
Y11n_O_S	0.984	0.914	0.933	0.923

To evaluate the generalizability of the proposed OR-gate fusion framework, we replaced the backbone feature extractor while keeping all other components—the SVM classifier, OR-gate decision-level fusion, and the evaluation protocol—identical. Table 15 summarizes the results on the same hold-out set of 5250 chip images. Across all 10 configurations, accuracy ranges from 0.981 to 0.986 and F1-score from 0.911 to 0.935, indicating that the OR-gate fusion framework maintains stable classification performance regardless of the backbone architecture. The YOLO-based backbones (YOLOv5n, YOLOv8n, and YOLOv11n) yield lower F1-scores (0.911–0.923) compared with the classification-oriented backbones (0.921–0.935).

Table 15. Performance comparison across backbone architectures under the identical OR-gate and SVM framework.

Backbone	Accuracy	Precision	Recall	F1-score
ResNet101 (proposed)	0.985	0.931	0.933	0.932
ViT-B16	0.986	0.927	0.943	0.935
ViT-B32	0.985	0.919	0.936	0.928
EfficientNet-B0	0.985	0.915	0.944	0.929
EfficientNet-B1	0.986	0.924	0.946	0.935
EfficientNet-B2	0.983	0.901	0.942	0.921
EfficientNet-B3	0.984	0.900	0.951	0.925
YOLOv5n	0.981	0.914	0.908	0.911
YOLOv8n	0.982	0.915	0.917	0.916
YOLOv11n	0.984	0.914	0.933	0.923

4.4. Hyperparameter tuning and optimization

Table 16 details the hyperparameter settings for the R101_O_S model. The categories include values for SVM classifier parameters, backbone model parameters, augmentation ratios, and training multipliers. Table 13 presents the baseline results with the default hyperparameters, while Table 17 reflects the optimized configuration from Table 16. The final performance is accuracy of 0.987, precision of 0.955, recall of 0.916, and F1-score of 0.935.

Table 16. Summary of hyperparameters.

Category	Hyperparameter	Value
SVM	C search range	(0.1, 1, 3, 10, 30, 100)
	Gamma	0.03
	Total search combinations	6
	Kernel	radial basis function (RBF)
ResNet101	Train	ImageNet pretrained
	Input Size	224×224
	Normalization	Mean = [0.485, 0.456, 0.406], Std = [0.229, 0.224, 0.225]
Augmentation	Class	Missing $\times$ 0, crack $\times$ 400, metal $\times$ 40, window $\times$ 40
	Ratio	Rotation 20%, shifting 30%, noise 30%, brightness 20%

Table 17. Performance by defect type for the R101_O_S model with optimized hyperparameters.

Defect type	Accuracy	Precision	Recall	F1-score
Missing	1.000	1.000	1.000	1.000
Crack	0.999	0.900	0.750	0.818
Metal	0.991	0.825	0.810	0.817
Window	0.993	0.824	0.899	0.860

As shown in Figure 18, when compared with the results of the YOLOv5 single model under the same configuration and dataset (accuracy of 0.959, precision of 0.778, recall of 0.927, F1-score of 0.846), the proposed R101_O_S model demonstrated higher performance. This confirms that the combination of binarization and decision-level fusion yields better results than a single model alone.

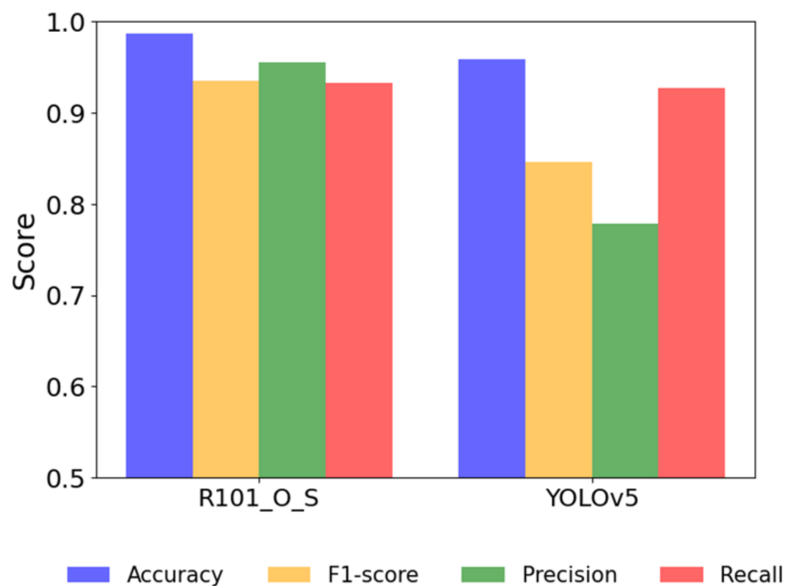


Figure 18. Performance comparison between the single YOLOv5 model and the proposed hybrid R101_O_S model.

4.5. Industrial deployment and real-world validation

Under the manual inspection protocol using high-magnification microscopy, the average throughput was approximately 17.7 seconds per unit. The proposed automated framework achieved a processing speed of approximately 1.5 seconds per unit. This metric covers the full operational cycle, including defect detection, defect removal, and data synthesis.

Specifically, in Table 18, segmentation (3199.7 s) and model inference (3285.6 s) together account for 6485.3 s, which is 83.5% of the total automated pipeline time (7769.8 s). These two stages replace the manual visual inspection step (62,360.9 s), achieving approximately a 9.6× speedup for the classification task itself. Beyond time reduction, the model provides consistent detection performance that is not subject to operator fatigue or subjective variability. Preprocessing (1250.4 s) and wafer map generation (34.1 s) account for 1284.5 s (16.5% of the pipeline). These are infrastructure components independent of the classification model. In particular, wafer map generation consumes only 34.1 s (0.4% of the pipeline), confirming that this stage contributes minimally to the overall time savings. The total processing time decreases from 92,924.9 s to 7769.8 s per wafer (89.1% reduction; 1.48 s vs. 17.70 s per chip). The majority of this reduction is attributable to replacing the two manual stages (visual inspection + defect removal: 92,924.9 s combined) with the four automated stages, of which the model stages constitute the dominant portion.

Table 18. Stage-wise processing time comparison (seconds, 5250 chips).

Category	Pipeline stage	Model applied (s)	Manual process (s)
Automation	Preprocessing	1250.4	—
DL/ML Model	U-Net segmentation	3199.7	—
DL/ML Model	Model inference	3285.6	—
Automation	Wafer map generation	34.1	—
Manual	Visual inspection	—	62,360.9
Manual	Defect removal	—	30,564.0
	Total	7769.8	92,924.9
	Per chip	1.48	17.70

In manual methods, the observation phase itself is relatively brief, but the time associated with post-detection sorting and data organization increases when defects are identified. The deployment of the proposed model reduced inspection and record management time. As shown in Figure 19, this implementation yielded an 89% reduction in total inspection time. Furthermore, the secondary defect rate in the downstream processes—attributable to human inspection error—dropped from 0.08% to 0.00% following the model's integration.

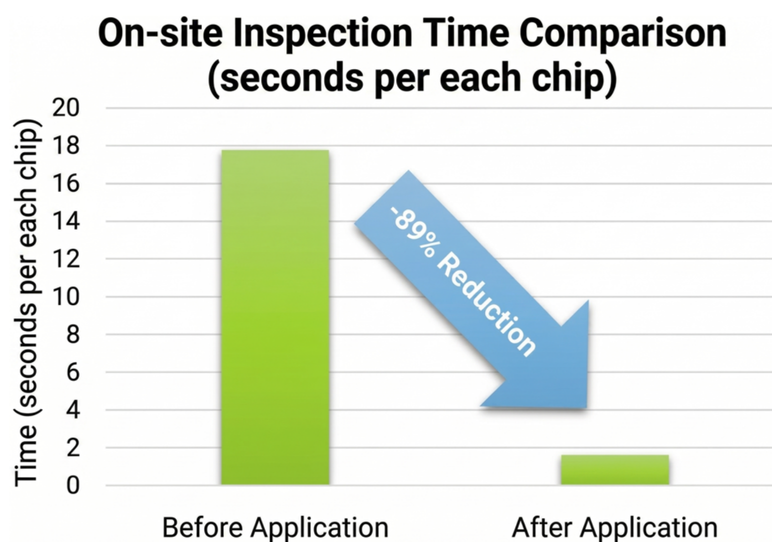


Figure 19. Comparison of inspection time after applying the proposed model.

The proposed framework classifies individual VCSEL chips into nominal and defective categories. These classification outputs are then synthesized into a digital wafer map that replicates the spatial layout of the chip array for downstream integration. Coordinate-based metadata—containing the missing, defective, and normal status for each unit—are generated in standardized formats (e.g., .txt and map files), enabling direct use by automated bonding equipment.

The manufacturing workflow previously required manual defect identification and physical extraction by operators to prevent downstream bonding failures. This process was time-consuming and introduced the risk of secondary mechanical damage during manual removal. Processing time increased with higher defect densities. In contrast, our method transmits the digital wafer map directly

to the bonding stage, bypassing physical intervention. During the automated bonding phase, the equipment references this map to selectively acquire only nominal units, eliminating the need for manual sorting and reducing the risk of erroneous insertion.

4.6. Analysis of experimental results

This section provides an analysis of the experimental results reported in Sections 4.3 and 4.4. Four aspects are examined: (1) the effect of backbone architecture selection, (2) the advantage of SVM over alternative classifiers, (3) defect-type-specific performance variation, (4) the contribution of decision-level fusion, and (5) error case (false positive and false negative) analysis.

4.6.1. Effect of backbone architecture and binarization

Table 14 shows that, under the same classifier (SVM) and input conditions, ResNet-based pipelines consistently outperform YOLO-based pipelines. We compare the two best hybrid configurations, R101_O_S (F1-score: 0.932) exceeds Y5n_O_S (F1-score: 0.911) by 0.021, with corresponding gains in recall (+0.025) and precision (+0.017). A similar pattern holds for basic models: R50_X_S (F1-score: 0.903) outperforms Y5n_X_S (F1-score: 0.889) by 0.014.

This difference is attributable to the architectural objectives of the two backbones. ResNet101 extracts 2048-dimensional feature vectors from global average pooling, producing a representation of the entire input image. VCSEL defects such as fine cracks and slight window deformations are characterized by subtle textural and structural changes distributed across the chip's surface. These features benefit from the high-dimensional, spatially aggregated representation that ResNet provides. In contrast, YOLOv5 is designed for real-time object detection with spatial localization; its feature maps are optimized for bounding-box regression rather than fine-grained texture discrimination. When the detection output is binarized (object detected $\rightarrow$ 1, else $\rightarrow$ 0) for downstream SVM classification, the spatial localization capability of YOLO is not fully utilized, reducing its relative advantage.

Among ResNet variants, ResNet101 (F1-score: 0.932) outperforms both ResNet50 (F1-score: 0.928) and ResNet152 (F1-score: 0.922). The improvement from ResNet50 to ResNet101 (+0.004) suggests that the additional network depth provides a richer feature hierarchy for distinguishing subtle defect patterns. The decline from ResNet101 to ResNet152 (-0.010) is consistent with overfitting on the relatively small training set, where the increased parameter count of ResNet152 does not yield additional discriminative benefit.

The hybrid model (with binarization-based morphological features) improves performance over the corresponding basic model across both backbone families: R50_O_S vs. R50_X_S (ΔF1-score = +0.025), Y5n_O_S vs. Y5n_X_S (ΔF1-score = +0.022). This indicates that the morphological features extracted through binarization provide complementary information to the backbone-derived features, particularly for defect types (window, metal) where the boundary between the defect region and the chip surface carries diagnostic relevance.

4.6.2. Classifier selection: SVM's advantage

Across both backbone families, SVM with the RBF kernel achieves the highest F1-score among the five classifiers evaluated (Table 14). For the ResNet50 Basic configuration, the F1-score ranking is: SVM (0.903) > logistic regression (0.898) > Random Forest (0.856) > gradient boosting (0.833) >

KNN (0.829). The same ordering holds for YOLOv5n: SVM (0.889) > LR (0.854) > RF (0.806) > GB (0.755) > KNN (0.793).

Two factors explain the advantage of SVM in this setting. First, the feature space is high-dimensional (2048 dimensions for ResNet101) relative to the number of training samples per defect type. SVM with RBF kernel constructs decision boundaries based on support vectors in the transformed feature space, which is effective when the sample-to-feature ratio is low. Second, the binary classification task (defective vs. normal for each defect type) produces a two-class problem where the margin-maximizing property of SVM is directly applicable.

KNN performs the poorest in both backbone settings (F1-score: 0.829 and 0.793). This is consistent with the curse of dimensionality: In a 2048-dimensional space, Euclidean distance-based neighbors become less discriminative as the distance between data points converges. Gradient boosting (F1-score: 0.833 and 0.755) also underperforms, which suggests that the sequential error-correction mechanism of boosting does not provide sufficient benefit when the base features are already high-quality representations from a pre-trained deep network.

4.6.3. Defect-specific performance variation

Table 17 reveals performance differences across the four defect types. The following analysis addresses each class.

Missing (F1-score: 1.000). The missing class achieves perfect classification without data augmentation. This result is consistent with the visual characteristics of missing defects described in Section 4.1: A chip's absence or chip flipping produces images with fundamentally different brightness distributions and background patterns (Figure 13), making them easily separable from normal chips. The training set contains 992 original samples (Table 11), which provides sufficient data diversity for the classifier.

Crack (recall: 0.750, F1-score: 0.818). The crack class exhibits the lowest recall among all defect types. Two factors contribute to this result. First, the original crack sample count is 39 (Table 11), the smallest among all classes. Despite ×400 augmentation, all augmented samples are geometric and photometric transformations of these 39 originals; the morphological diversity of crack patterns in the augmented set remains bounded by the original sample pool. Second, the test set contains only 12 crack defect samples. A recall of 0.750 corresponds to nine correct detections and three misses. In this setting, each misclassified sample reduces recall by approximately 8.3 percentage points ($1/12 \approx 0.083$), which amplifies the apparent performance variation. The five fold cross-validation results in Table 10 show that crack recall remains at 100.0% (± 0.000) across all folds in the training/validation phase, indicating that the model itself is stable; the lower test-set recall reflects the limited morphological coverage of the 39 original samples rather than model instability.

Metal (recall: 0.810, F1-score: 0.817). Metal defects are characterized by contact marks from pre-inspection electrical testing (Figure 12), which visually resemble foreign matter on the bonding surface. The grayscale imaging reduces the contrast difference between contact marks and actual metal defects, as described in Section 4.1. This ambiguity between normal contact marks and genuine defects accounts for the lower precision (0.825) and recall (0.810) compared with missing and window classes.

Window (recall: 0.899, F1-score: 0.860). Window defects include ring metal malformation and emitter area damage (Figure 11), which produce more visually distinct patterns than metal defects. The

higher recall (0.899 vs. 0.810 for metal) reflects this greater visual separability. However, the precision (0.824) is the lowest among all classes, indicating that some normal chips with manufacturing tolerance variations in the peripheral ring region are misclassified as window defects. This is consistent with the observation in Section 4.1 that normal window images exhibit imaging differences (Figures 11(a) and 11(b)).

4.6.4. Contribution of decision-level fusion

The final system-level performance (accuracy: 0.987, precision: 0.955, recall: 0.916, F1-score: 0.935) exceeds the individual R101_O_S model performance (Table 17) and the single YOLOv5 model (accuracy: 0.959, precision: 0.778, recall: 0.927, F1-score: 0.846), as shown in Figure 18.

The improvement from single-model to fused performance is attributable to two factors. First, the OR-gate aggregates the outputs of four independent binary classifiers, each specialized for a distinct defect type with a tailored input representation (the raw image for missing, the chip-level ROI for crack, the U-Net body mask for window, the U-Net head mask for metal). Because each classifier is optimized for its target defect morphology, the combined system benefits from complementary detection capabilities that a single multiclass model cannot replicate.

Second, the OR-gate fusion rule ($D = b_1 \vee b_2 \vee b_3 \vee b_4$) reduces false negatives that arise when individual classifiers miss defects outside their specialization. In the single YOLOv5 model, all four defect types must be detected by a single architecture, which leads to a precision of 0.778—lower than the fused system's 0.955. The fused system's precision gain (+0.177) indicates that the defect-type-specific classifiers produce fewer false positives per class than a single model attempting to detect all defect types simultaneously.

The recall comparison between the single YOLOv5 (0.927) and the fused system (0.916) shows a decrease of 0.011. This is because the single YOLOv5 model, operating on the full image, occasionally detects through a spatial context that is not available to the ROI-specific classifiers. However, the F1-score improvement (+0.089) confirms that the overall trade-off favors the fused system, where the precision gain more than compensates for the recall reduction.

Table 8 further demonstrates that the OR-gate is the only viable fusion strategy for this architecture: The AND-gate and majority voting yield recall of 0.027 due to the structural incompatibility with defect-type-specific classifiers (Section 3.4), while the weighted sum (F1-score: 0.930) performs below the OR-gate (F1-score: 0.932) and introduces additional hyperparameters.

4.6.5. Error case analysis

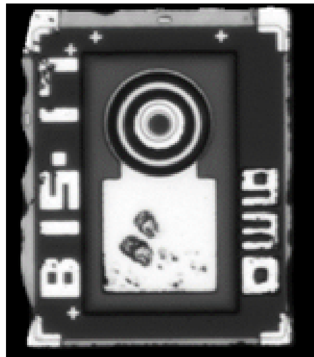
This subsection presents representative false positive (FP) and false negative (FN) cases for each defect type. The failure modes identified from the hold-out test set are summarized below.

False positive cases. Table 19 lists the failure mode for each FP case, and Figure 20 shows the corresponding chip images.

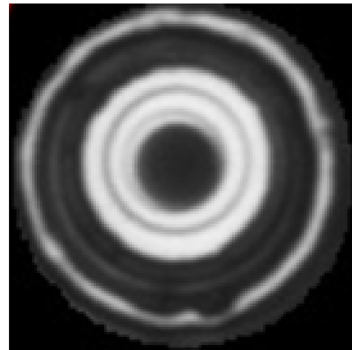
In the crack case Figure 20(a), the cutting lines left during the dicing process have irregular patterns that the crack classifier cannot distinguish from actual cracks. In the window case Figure 20(b), the ring metal is continuous, but its dark appearance in the image leads the window classifier to flag it as defective. In the metal case Figure 20(c), measurement marks and foreign particles on the metal pad share visual characteristics with actual metal defects.

Table 19. Failure modes of representative FP cases.

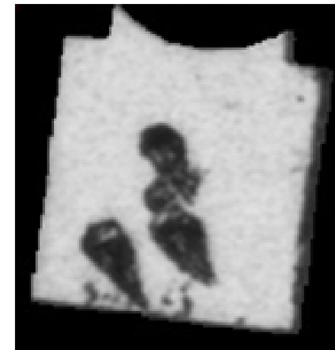
Defect type	Failure mode
Crack	Irregular cutting lines on the chip's surface were misclassified as cracks.
Window	The ring metal was physically intact, but appeared dark in the captured image, triggering a false alarm.
Metal	Measurement marks and foreign particles exhibited visual similarity to metal defects.



(a) Crack



(b) Window



(c) Metal

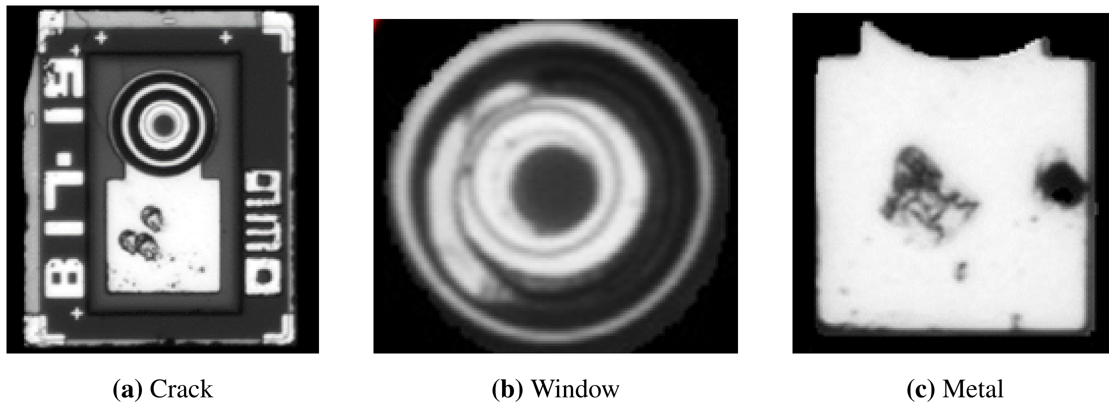
Figure 20. Representative false positive examples.

False negative cases. Table 20 lists the failure mode for each FN case, and Figure 21 shows the corresponding chip images.

Table 20. Failure modes of representative FN cases.

Defect type	Failure mode
Crack	The crack was too fine for the feature extractor to distinguish from the normal background.
Window	Insufficient training samples of this defect subtype limited the classifier's detection capability.
Metal	Measurement marks and foreign particles were visually similar to the metal defect, and the model failed to classify the sample as defective.

In the crack case Figure 21(a), the crack width is below the effective resolution of the ResNet101 feature map, producing feature vectors that overlap with those of normal samples. In the window case Figure 21(b), the defect subtype has few training samples, and the SVM decision boundary does not generalize to this variant. In the metal case Figure 21(c), the coexistence of measurement marks and foreign particles with the actual defect makes the feature representation ambiguous, and the classifier assigns the sample to the normal class.



(**a**) Crack (**b**) Window (**c**) Metal

Figure 21. Representative false negative examples.

5. Conclusions and future work

This study presented a hybrid deep learning framework for the automated optical inspection of semiconductor VCSEL's devices by integrating a ResNet101 backbone with machine learning classifiers. By incorporating U-Net-based ROI segmentation and morphological feature extraction via binarization, the framework achieved an accuracy of 0.987, precision of 0.955, recall of 0.916, and an F1-score of 0.935. The deployment of this framework enabled the automated generation of digital wafer maps, which optimized downstream manufacturing workflows and yielded an 89% reduction in total inspection time.

Future research directions for extending the proposed framework may include integrating VCSEL's electro-optical parameters (e.g., threshold current, slope efficiency) with image-based defect labels to examine the correlation between visual defect types and device-level performance degradation. Another potential direction is the incorporation of 3D surface profilometry data from confocal microscopy or white-light interferometry, as topographic features such as crack depth and metal protrusion height are not captured by the current two-dimensional imaging pipeline.

Author contributions

Kyu-Jeong Choi: Conceptualization, methodology, formal analysis, writing original draft; Jin-Taek Seong: Conceptualization, writing review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

During the preparation of this work, the authors used ChatGPT strictly to refine the language and improve the readability of the manuscript. After using this tool, the authors thoroughly reviewed and edited the text as needed, and take full responsibility for the final content of the publication.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2023-00242528).

Conflict of interest

The authors declare no conflicts of interest.

References

1. S. Sundaram, A. Zeid, Artificial intelligence-based smart quality inspection for manufacturing, *Micromachines*, **14** (2023), 570. https://doi.org/10.3390/mi14030570
2. *3D automated optical inspection (AOI) equipment market*, Market growth reports, 2025. Available from: `https://www.marketgrowthreports.com/market-reports/3d-automated-optical-inspection-aoi-equipment-market-100380`.
3. A. Adrian, *AOI equipment: unveiling the hidden costs that impact your ROI*, AllPCB Blog, 2025. Available from: `https://www.allpcb.com/blog/pcb-assembly/aoi-equipment-unveiling-the-hidden-costs-that-impact-your-roi.html`.
4. Z. H. Ren, F. Z. Fang, N. Yan, Y. Wu, State of the art in defect detection based on machine vision, *Int. J. Precis. Eng. Manuf.-Green Tech.*, **9** (2022), 661–691. https://doi.org/10.1007/s40684-021-00343-6
5. J.-H. Park, Y.-S. Kim, H. Seo, Y.-J. Cho, Analysis of training deep learning models for PCB defect detection, *Sensors*, **23** (2023), 2766. https://doi.org/10.3390/s23052766
6. D. Tabernik, S. Šela, J. Skvarč, D. Skočaj, Segmentation-based deep-learning approach for surface-defect detection, *J. Intell. Manuf.*, **31** (2020), 759–776. https://doi.org/10.1007/s10845-019-01476-x
7. S. B. Jha, R. Babiceanu, P. Shekhar, S. Namilae, Deep convolutional neural network-based automated optical inspection for aerospace components, *Digital Engineering*, **7** (2025), 100062. https://doi.org/10.1016/j.dte.2025.100062
8. J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González, A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas, *Mach. Learn. Knowl. Extr.*, **5** (2023), 1680–1716. https://doi.org/10.3390/make5040083
9. W. Y. Lv, Y. A. Zhao, Q. Y. Chang, K. Huang, G. Z. Wang, Y. Liu, Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer, 2024, arXiv:2407.17140.
10. V. De Ridder, B. Dey, S. Halder, B. Van Waeyenberge, Semi-diffusioninst: A diffusion model based approach for semiconductor defect classification and segmentation, *2023 International Symposium ELMAR*, Zadar, Croatia, 2023, 61–66. https://doi.org/10.1109/ELMAR59410.2023.10253920
11. F. Mohammad, D. Ryu, Semiconductor wafer map defect classification with tiny vision transformers, 2025, arXiv:2504.02494.

12. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Commun. ACM*, **63** (2020), 139–144. https://doi.org/10.1145/3422622

13. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767

14. K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, 770–778. https://doi.org/10.1109/CVPR.2016.90

15. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2021, arXiv:2010.11929.

16. X. B. Wang, S. Gao, Y. T. Zou, J. L. Guo, C. Wang, IH-ViT: Vision transformer-based integrated circuit appear-ance defect detection, 2023, arXiv:2302.04521.

17. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386

18. W. Ullah, S. U. Khan, M. J. Kim, A. Hussain, M. Munsif, M. Y. Lee, et al., Industrial defective chips detection using deep convolutional neural network with inverse feature matching mechanism, *J. Comput. Des. Eng.*, **11** (2024), 326–336. https://doi.org/10.1093/jcde/qwae019

19. W. J. Tang, Y. J. Deng, X. Luo, RST-YOLOv8: An improved chip surface defect detection model based on YOLOv8, *Sensors*, **25** (2025), 3859. https://doi.org/10.3390/s25133859

20. J. J. Hwang, H. W. Lee, Y. S. Park, Multi-sensor and decision-level fusion-based structural damage detection using 1-D CNN, *Sensors*, **21** (2021), 3950. https://doi.org/10.3390/s21123950

21. N. G. Yu, H. Z. Li, Q. Xu, A full-flow inspection method based on machine vision to detect wafer surface defects, *Math. Biosci. Eng.*, **20** (2023), 11821–11846, https://doi.org/10.3934/mbe.2023526

22. Y. C. Du, J. P. Chen, H. Zhou, X. L. Yang, Z. Q. Wang, J. Zhang, et al., An automated optical inspection (AOI) platform for three-dimensional (3D) defects detection on glass micro-optical components (GMOC), *Opt. Commun.*, **545** (2023), 129736. https://doi.org/10.1016/j.optcom.2023.129736

23. H. Fu, Y. M. Lai, C. R. Pan, S. W. Zhang, L. P. Bai, J. Li, A central array method to locate chips in AOI systems in semiconductor manufacturing, *Electronics*, **13** (2024), 1070. https://doi.org/10.3390/electronics13061070

24. T.-Y. Kim, S. Park, C.-K. Lim, J.-H. Choi, C.-H. An, L.-H. Song, et al., Deep learning-based detection of defects in wafer buffer zone during semiconductor packaging process, *Multiscale Sci. Eng.*, **6** (2024), 25–32. https://doi.org/10.1007/s42493-024-00103-z

25. A. B. Goti, Automated optical inspection (AOI) based on IPC standards, *International Journal of Engineering and Computer Science*, **13** (2025), 26928–26947.

26. C.-J. Fu, H.-L. Chen, H.-Y. Tseng, Application of artificial intelligence to improve chip defect detection using semiconductor equipment, *Eng. Proc.*, **98** (2025), 26. https://doi.org/10.3390/engproc2025098026

27. J. M. Johnson, L. M. Khoshgoftaar, Survey on deep learning with class imbalance, *J. Big Data*, **6** (2019), 27. https://doi.org/10.1186/s40537-019-0192-5

28. M. S. Shelke, P. R. Deshmukh, V. K. Shandilya, A review on imbalanced data handling using undersampling and oversampling technique, *International Journal of Recent Trends in Engineering and Research*, **3** (2017), 444–449.

29. N. Sarafianos, X. Xu, I. A. Kakadiaris, Deep imbalanced attribute classification using visual attention aggregation, In: *Computer vision–ECCV 2018*, Cham: Springer, 2018, 708–725. https://doi.org/10.1007/978-3-030-01252-6_42

30. J. Byrd, Z. C. Lipton, What is the effect of importance weighting in deep learning, *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, California, 2019, 872–881.

31. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, **16** (2002), 321–357. https://doi.org/10.1613/jair.953

32. E. Goceri, Medical image data augmentation: techniques, comparisons and interpretations, *J. Artif. Intell. Rev.*, **56** (2023), 12561–12605. https://doi.org/10.1007/s10462-023-10453-z

33. P. Y. Simard, D. Steinkraus, J. C. Platt, Best practices for convolutional neural networks applied to visual document analysis, *Seventh International Conference on Document Analysis and Recognition*, Edinburgh, UK, 2003, 958–963. https://doi.org/10.1109/ICDAR.2003.1227801

34. S. S. Lee, Noisy replication in skewed binary classification, *Comput. Stat. Data An.*, **34** (2000), 165–191. https://doi.org/10.1016/S0167-9473(99)00095-X

35. S. Wang, Evaluation of impact of image augmentation techniques on two tasks: Window detection and window states detection, *Results in Engineering*, **24** (2024), 103571. https://doi.org/10.1016/j.rineng.2024.103571

36. M. Buda, A. Maki, M. A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, *Neural Networks*, **106** (2018), 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

37. A. More, Survey of resampling techniques for improving classification performance in unbalanced datasets, 2023, arXiv:1608.06048.

38. C. Elkan, The foundations of cost-sensitive learning, *The 17th International Joint Conference on Artificial Intelligence*, Seattle, WA, USA, 2001, 973–978.

39. K. M. Ting, An instance-weighting method to induce cost-sensitive trees, *IEEE T. Knowl. Data En.*, **14** (2002), 659–665. https://doi.org/10.1109/TKDE.2002.1000348

40. A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, S. Kumar, Long-tail learning via logit adjustment, 2021, arXiv:2007.07314.

41. H. Cho, W. Koo, H. Kim, Prediction of highly imbalanced semiconductor chip-level defects in module tests using multimodal fusion and logit adjustment, *IEEE T. Semiconduct. M.*, **36** (2023), 425–433. https://doi.org/10.1109/TSM.2023.3283101

42. V. Vasan, N. V. Sridharan, R. J. Balasundaram, S. Vaithiyanathan, Ensemble-based deep learning model for welding defect detection and classification, *Eng. Appl. Artif. Intel.*, **136** (2024), 108961. https://doi.org/10.1016/j.engappai.2024.108961

43. J. Choi, D. Suh, M.-O. Otto, Boosted stacking ensemble machine learning method for wafer map pattern classification, *CMC-Comput. Mater. Con.*, **74** (2022), 2945–2966. https://doi.org/10.32604/cmc.2023.033417
44. J. Bai, D. Wu, T. Shelley, P. Schubel, D. Twine, J. Russell, et al., A comprehensive survey on machine learning driven material defect detection, *ACM Comput. Surv.*, **57** (2025), 1–36. https://doi.org/10.1145/3730576
45. K. Fan, P. Peng, H. P. Zhou, L. L. Wang, Z. Y. Guo, Real-time high-performance laser welding defect detection by combining acgan-based data enhancement and multimodel fusion, *Sensors*, **21** (2021), 7304. https://doi.org/10.3390/s21217304
46. O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, Cham: Springer, 2015, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
47. C. Cortes, V. Vapnik, Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. https://doi.org/10.1007/BF00994018
48. T. M. Cover, P. E. Hart, Nearest neighbor pattern classification, *IEEE T. Inform. Theory*, **13** (1967), 21–27. https://doi.org/10.1109/TIT.1967.1053964
49. D. R. Cox, The regression analysis of binary sequences, *J. R. Stat. Soc. B*, **20** (1958), 215–232. https://doi.org/10.1111/j.2517-6161.1958.tb00292.x
50. L. Breiman, Random forests, *Mach. Learn.*, **45** (2001), 5–32. https://doi.org/10.1023/A:1010933404324
51. J. H. Friedman, Greedy function approximation: A gradient boosting machine, *Ann. Statist.*, **29** (2001), 1189–1232. https://doi.org/10.1214/aos/1013203451

AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)