Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

HighTech and Innovation Journal

Vol. 7, No. 2, June, 2026

Deep Learning-Based Surface Defect Detection Technology for Microdevices

Zhizhuo Shi [1*]

[1] *School of Computer Science and Technology, Shandong University, Qingdao 266237, China.*

Received 03 December 2025; Revised 16 May 2026; Accepted 20 May 2026; Published 01 June 2026

Abstract

Surface defect detection on microdevices is challenged by extremely small defect sizes, complex background interference, and strict requirements on both detection accuracy and computational efficiency. The objective of this study is to develop a lightweight yet high-precision detection framework suitable for resource-constrained industrial deployment. To this end, this paper proposes LiteKANformer, a multi-module lightweight Transformer-based architecture. Based on LiteKANformer, a novel detection framework named LKF-YOLO is constructed by embedding global contextual modeling into the backbone and optimizing multi-scale feature fusion in the neck network. Experimental analysis is conducted on a PCB surface defect dataset and a semiconductor chip defect dataset. Compared with the C3TR module, LiteKANformer achieves comparable detection accuracy while reducing the parameter count by approximately 3.1% and improving the inference frame rate by 2.3%. Furthermore, the proposed LKF-YOLO framework outperforms other mainstream detection models on the PCB dataset in terms of accuracy, recall, and real-time performance. The main novelty of this work lies in the co-design of activation representation, normalization strategy, and computation primitives within a unified lightweight Transformer block, providing an effective solution that balances detection precision and deployment efficiency for microdevice surface defect detection.

Keywords: Deep Learning; Computer Vision; Transformer; Defect Detection; YOLO.

1. Introduction

Microdevices are essential components in modern industrial products, and their quality directly affects system reliability. However, surface defects are difficult to avoid during manufacturing due to complex surface conditions and subtle defect patterns. Consequently, accurate and efficient defect detection with minimal computational overhead has become a challenge in both industrial practice and academic research.

Historically, surface defect detection for microdevices relied on manual inspection, requiring experienced workers and substantial factory investment in recruitment or training [1]. With the advancement of deep learning technology, researchers have initiated studies on deep learning-based surface defect detection. Bhardwaj et al. [2] applied deep learning to automate semiconductor wafer defect detection, demonstrating improved classification accuracy over manual inspection, but their approach relies heavily on limited training data and basic neural network architectures, which may restrict detection robustness and generalization to diverse defect types. Zhao et al. [3] employed a CNN with selective weight freezing to retain old defect patterns, improving online detection of unknown wafer defects, but this approach may limit adaptability to entirely new defect types. Gao et al. [4] proposed a variant Swin Transformer (Cas-VSwin

* Corresponding author: zhizhuoshi@outlook.com

https://doi.org/10.28991/HIJ-2026-07-02-022

Transformer) to strengthen cross-window connections and improve surface-defect detection performance over standard CNN models, but its reliance on hierarchical attention mechanisms can still struggle with very small or subtle defects due to limited local detail sensitivity. Ma & Cheng [5] proposed an improved 2D OTSU segmentation algorithm combining an adaptive genetic algorithm and integral image to accelerate image segmentation and reduce computation, but this classical image processing-based method lacks the deep feature representation capability needed for robust defect classification under complex industrial variations. Jiang et al. [6] proposed CINFormer, a hybrid Transformer network that injects multi-stage CNN features to maintain detailed local information while suppressing background noise for surface defect segmentation, but its reliance on injected features and complex attention mechanisms increases computational complexity and risk of redundant feature representations in large-scale industrial scenarios. Singh et al. [7] demonstrated that the pre-trained CNN EfficientNet-b0 outperforms other CNN architectures for detecting surface defects on machined components such as flat washers and tapered rollers, but its reliance on CNN feature hierarchies limits its ability to capture global context and long-range dependencies in complex defect distributions.

In recent years, both academic researchers and industrial professionals frequently opt for the YOLO (You Only Look Once) series when tackling routine object detection tasks, owing to its prominent merits; including compact model volume, excellent detection accuracy, and straightforward deployability. Cao et al. [8] proposed an improved YOLOv8-GD deep learning model to detect sparse and hard-to-distinguish photovoltaic surface defects by enhancing feature extraction and discrimination capabilities, but its reliance on YOLOv8 and model enhancements still struggles with extremely subtle defect variations and complex background noise. Tian et al. [9] proposed an improved YOLOv5n algorithm with an enhanced backbone, neck structure, and loss function to improve industrial surface defect detection accuracy, but its increased model complexity and parameter count may limit real-time performance and adaptability to diverse defect types. Ge et al. [10] proposed YOLO-MSD, a lightweight YOLO-based surface defect detection model with a four-scale backbone and streamlined feature-pyramid neck to enhance multi-scale feature extraction and cross-scale fusion for large-size industrial images, but its reliance on handcrafted multi-scale convolution modules still limits performance on extremely small or low-contrast defects and increases architectural complexity. Ma et al. [11] proposed ELA-YOLO, a YOLOv8-based defect detection model incorporating linear attention and a selective feature pyramid network to improve representation capability and multi-level feature fusion for steel surface defects, but its reliance on YOLOv8 and added attention mechanisms struggles with extremely subtle defect variations under complex lighting or reflective industrial conditions. Yuan et al. [12] proposed YOLO-HMC, an improved YOLOv5-based network integrating a HorNet backbone, MCBAM attention, and CARAFE upsampling to enhance PCB surface defect detection accuracy and efficiency, but its increased module complexity and lightweight design may still struggle to capture fine details in high-density defect regions and limit adaptability in resource-constrained environments. In summary, although existing improved models based on the YOLO series have improved defect detection performance in specific scenarios, there is generally room for optimization in identifying extremely small defects, adapting to complex environments, or balancing real-time performance with complexity.

Overall, previous studies have demonstrated that convolutional neural networks (CNNs) achieve outstanding performance in defect detection tasks due to their efficient capture of local texture features, establishing themselves as mainstream solutions. However, CNNs [13, 14] face significant limitations in integrating cross-regional defect information and modeling global contextual associations, which restricts their ability to represent overall defect distributions. Transformers, based on self-attention mechanisms, have recently shown strong performance in various computer vision tasks and are considered feasible alternatives to CNNs. Transformer-based surface defect recognition constructs global feature representations that enhance modeling of overall defect information [15], but this often comes at the expense of local detail extraction, such as defect textures and edge contours [16]. Surface defects in microdevices are typically subtle and concealed, making accurate capture of local details critical. While CNNs can efficiently detect local features across sequential images, they cannot fully exploit cross-frame contextual relationships; meanwhile, Transformers alone struggle to capture fine-grained local information. Combining Transformer modules with YOLO frameworks and enhancing CNN backbones represents a promising approach for integrating local feature extraction with global contextual modeling, which can motivate the development of more effective and lightweight models for microdevice defect detection.

This paper proposes a multi-module lightweight processing transformer module with the Kolmogorov–Arnold Networks [17], LiteKANformer. By integrating this module into the YOLO algorithm [18], a new model for surface defect detection of microdevices with complex surface features is achieved. LiteKANformer employs advanced KAN neural networks and is based on optimizing normalization and fully connected layers, thereby achieving model lightweighting. A brand new LKF-YOLO model based on the new module and YOLO framework is also designed. This model fully utilizes the contextual memory characteristics of Transformers while retaining the advantage of the CNN module, which achieves accurate recognition of defect information. It also employs advanced model lightweighting techniques, demonstrating outstanding advantages in both model lightweighting and high accuracy for surface defect recognition in microdevices. In detection experiments on the PCB defect dataset, the LKF-YOLO model achieved a ~1.7%–3.9% improvement in detection accuracy compared to common YOLO models while reducing GFLOPs.

Through testing on a semiconductor chip defect dataset, the model achieved an mAP of 99.4% and an mAP@0.5:0.95 of 93.7%. Compared to the C3TR (Cross Stage Partial Network with 3 convolutions) model in the YOLO series, it reduced GFLOPs by 3.1% and increased FPS by 2.3%.

The structure of this paper is as follows. Section 2 reviews the related work on surface defect detection, including Transformer-based approaches and YOLO-based detection frameworks. Section 3 describes the proposed microdevice surface defect detection system in detail, including the overall LKF-YOLO architecture and the design of the LiteKANformer module. Section 4 presents the experimental setup, datasets, evaluation metrics, and comparative as well as ablation experiments, followed by an in-depth discussion of the results. Section 5 concludes the paper by summarizing the main contributions and experimental findings. Finally, Section 6 outlines potential directions for future work.

2. Related Works

Transformers without Normalization*:* The normalization layer [19] plays a crucial role in Transformers. The design of the normalization layer's position and its structure itself can impact the training and accuracy of the Transformer [20]. However, recent studies have found that Transformer models without normalization can achieve equivalent or even superior performance [21]. Since layer normalization in Transformers often produces S-shaped input-output mappings resembling hyperbolic tangent functions, introducing the Dynamic Hyperbolic Tangent (DyT) operation ($DyT(x) = tanh(\alpha x)$) as a direct replacement for the normalization layer yields normalization-free Transformer models that rival normalized models in performance, often without requiring hyperparameter tuning.

Kolmogorov–Arnold Networks*:* Based on the Kolmogorov–Arnold theorem, Kolmogorov–Arnold Networks (KAN) emerge as an effective alternative to Multi-Layer Perceptron (MLP) [17, 22]. While MLP employs fixed activation functions on nodes (“neurons”), KAN utilizes learnable activation functions on edges (“weights”). As illustrated, each weight parameter in KAN is replaced by a univariate spline function, eliminating linear weights. Research demonstrates that this seemingly simple modification enables KAN to outperform MLP in accuracy and interpretability. Sainath Dey et al. put forward the Efficient-KAN (Eff-KAN) module [23]. This module substitutes multi-layer perceptron (MLP) layers with spline functions and derived from this design, the Hyb-KAN vision transformer (Hyb-KAN ViT) model achieves outstanding performance in domains including image recognition.

MatMul-free Transformers*:* Matrix multiplication (MatMul) typically reflects the computational cost of deep learning models. Numerous studies aim to reduce this computational burden. Kosson et al. [24] achieved multiplication-free training by replacing multiplication, division, and nonlinear operations with piecewise affine approximations. Guo [25] proposed a ternary spike neuron that not only replaces multiplication with addition for both activation and weights; thus enjoying the event-driven and multiplication-free operation advantages of the binary spike neuron, but also enhances information capacity. Zhu et al. [26] proposed a MatMul-free model, namely the BitLinear layer with ternary weights. By constraining weights within the set $\{-1, 0, +1\}$ and applying additional quantization techniques, matrix multiplication operations are replaced with addition and negation operations. This reduces computational cost and memory consumption, but preserves the network's expressive power.

YOLO (You Only Look Once)*:* The YOLO (You Only Look Once) framework stands out for its speed and accuracy compared with different object detection algorithms. YOLO completes detection tasks through a single pass through the network, and can quickly and reliably identify objects in images [27]. The YOLOv11 model innovatively uses the C3K2 module. Compared with the standard C3 module, C3K2 introduces a multi-scale convolution kernel C3K, where K is an adjustable convolution kernel size. This design can expand the receptive field, allowing the model to grasp a broader scope of contextual information, which is particularly well-suited for large-scale object detection or scenes with complex backgrounds, such as the surface defects of microdevices [28]. With the development of YOLO, there are more combinations with transformers and YOLO framework. Lv & Su [29] enhanced the precision of the task by adding transformer blocks (C3TR) to the YOLO model.

3. Method

3.1. Surface Defect Detection System for Microdevices

As shown in Figure 1, the microdevice surface defect detection system primarily comprises 5 parts, namely image acquisition, image processing, image feature extraction, defect classification, and result display modules.

The image acquisition module communicates with the visual induction sensor to capture image data of the microdevice surface and transmits them to the computer for subsequent processing. Then, the image preprocessing module performs preliminary processing of raw images to improve image quality and clarity. It is the foundation for subsequent feature extraction and defect classification. Surface defects within the image data are detected by the recognition module. The result display module presents detection outcomes in an intuitive manner, including defect location, size, and type, facilitating operator intervention and handling.

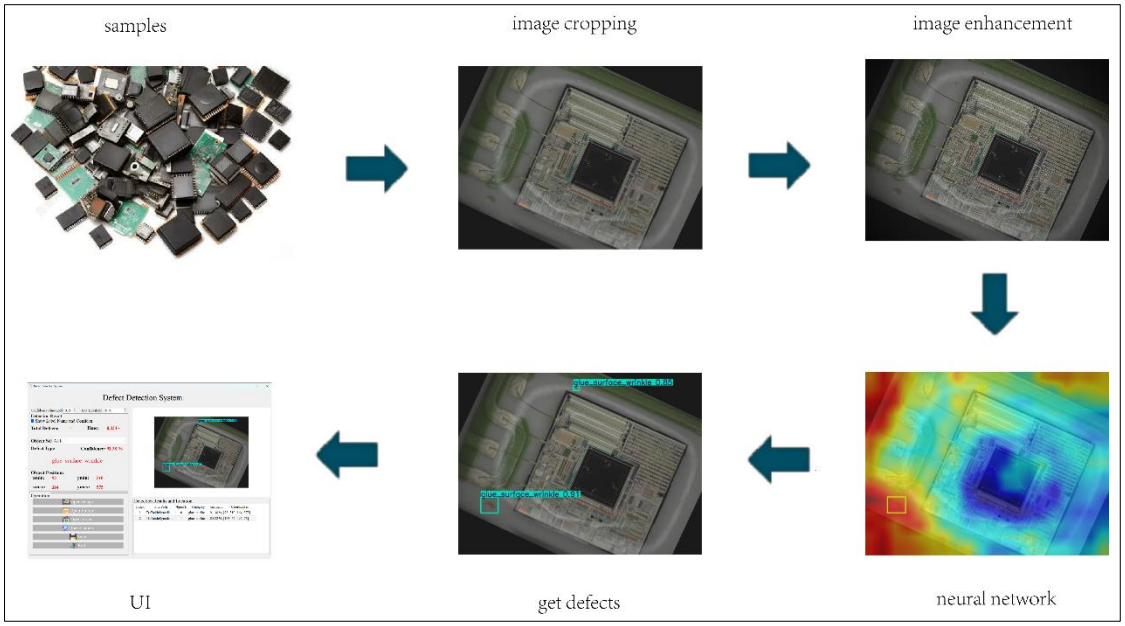


Figure 1. Operational Flowchart of the Microdevice Surface Defect Detection System

3.2. LKF-YOLO Model for Microdevice Surface Defect Detection

YOLO (You Only Look Once) is a single-stage object detection algorithm, and its core structure can be divided into three parts: backbone network, neck network, and head network. The core function of backbone networks is to extract multi-scale visual features from raw images and convert pixel-level inputs into feature maps with semantic information. The core function of the neck, which is located between the backbone network and the head, is to fuse feature maps of different scales and solve the contradiction of "shallow features with weak semantics and deep features with few details". This allows the model to utilize both detailed and semantic information, so it enhances the model's ability to detect targets of different sizes, especially small ones. The head directly outputs the category, position, and confidence of the target based on the fused feature map of the neck and completes the conversion from features to detection results. Therefore, improvements to the YOLO framework focus on enhancing the backbone network and neck network to boost feature extraction and small object detection capabilities. The structure is drawn in Figure 2.

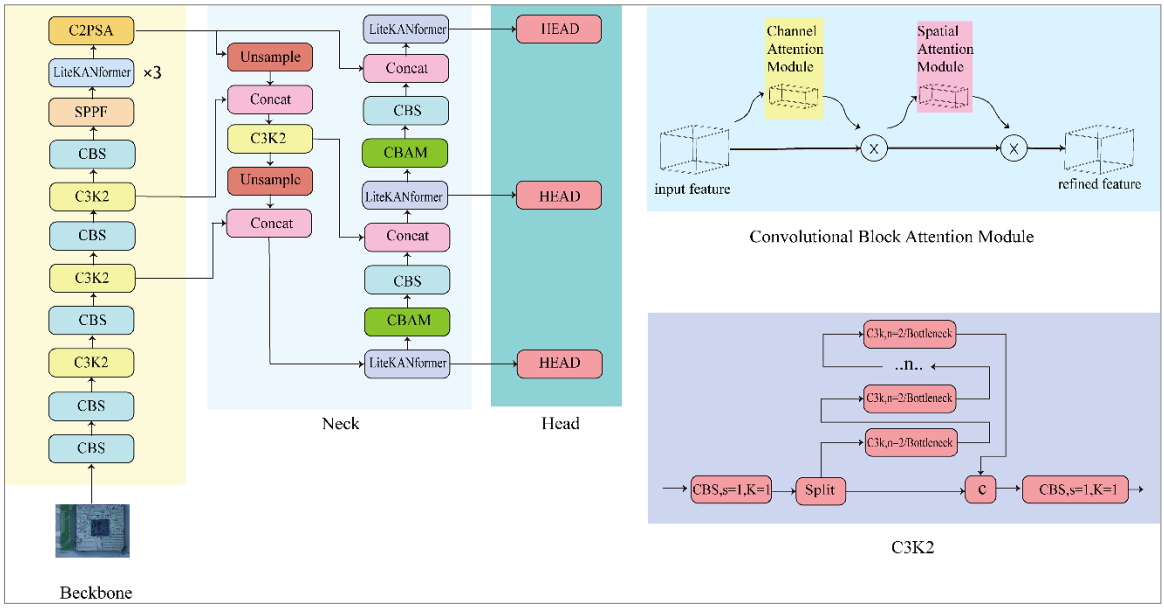


Figure 2. LKF-YOLO Model Architecture

Three LiteKANformer modules are introduced at higher layers of the backbone network to expand the receptive field. LiteKANformer learns more global and semantic features, improving small object detection. For surface defects on microdevices, which exhibit random distribution and unpredictable shapes, a transformer is also deployed at the connection between the neck network and detection head to enhance target localization in complex scenes.

Beyond LiteKANformer, the neck section incorporates the Convolutional Block Attention Module [30]. CBAM combines channel attention and spatial attention to provide more comprehensive and effective feature extraction capabilities.

Given an input feature map $F \in \mathbb{R}^{C\times H\times W}$, where C, H, and Wdenote the number of channels, height, and width respectively, the channel attention map $M_c(F)$ is computed as:

$$Mc(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right) \quad (1)$$

where, AvgPool(·) and MaxPool(·) denote global average pooling and global max pooling operations along the spatial dimensions, respectively. MLP(·) represents a shared multi-layer perceptron, and $\sigma(\cdot)$ denotes the Sigmoid activation function.

The resulting channel attention map $M_c(F) \in \mathbb{R}^{C\times 1\times 1}$ is applied to recalibrate channel-wise feature responses.

Subsequently, the spatial attention map $M_s(F)$is computed as:

$$Ms(F) = \sigma\left(f^{7\times 7}([AvgPool(F); MaxPool(F)])\right) \quad (2)$$

Here, $f^{7\times 7}$ denotes a 7×7 convolution operation, and [;] represents channel-wise concatenation of the pooled feature maps. The spatial attention map $M_s(F) \in \mathbb{R}^{1\times H\times W}$highlights informative spatial locations.

In terms of overall design, LiteKANformer benefits from self-attention in modeling long-range dependencies and global contextual relationships. It mainly plays a role in high-level semantic token-wise interactions that enhance global semantic consistency. In contrast, CBAM has a unique and complementary role. It acts on convolutional feature maps, and conducts channel-wise and spatial-wise reweighting to refine local feature responses. Therefore, CBAM emphasizes critical local regions and suppresses redundant information during multi-scale feature fusion, rather than pursuing global feature interactions. The hierarchical division of attention mechanisms effectively reduces redundancy, which makes it especially suitable for microdevice surface defect detection. Defects in this scenario are usually small, irregular, and situated in complex backgrounds, so both global context modeling and accurate local feature enhancement are required.

3.3. LiteKANformer Design

LiteKANformer builds upon the classical Transformer encoder architecture. For the normalization layer in Transformer, DyT (Dynamic Tanh) is used instead, because DyT can efficiently replace the normalization layer without the need to calculate activation statistics. It can also combine the convenience of use with stable training performance. In addition, this module replaces MLP with KAN layer. This neural network places the activation function on the edges of the network, and each weight is parameterized as a learnable univariate spline function, which is suitable for modeling complex systems. Compared to the original network layers, both modifications feature reduced parameter counts and lower computational requirements while achieving equivalent test accuracy. The module structure is drawn in Figure 3.

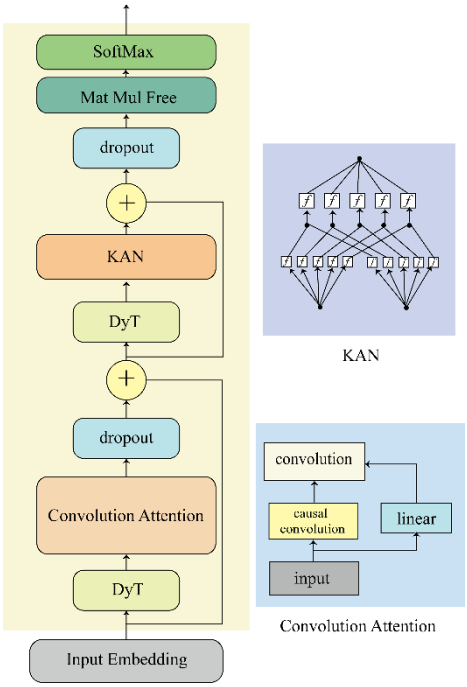


Figure 3. LiteKANformer Architecture Diagram

The module employs non-multiplicative fully connected layers. Traditional fully connected layers achieve feature transformation through multiplying input vector by weight matrix, whereas the core design of the zero-multiplication fully connected layers replaces multiplication with addition or accumulation operations. It quantizes the weights of the fully connected layers from continuous values to ternary values (-1, 0, 1), thereby avoiding the high computational cost of traditional floating-point multiplication. Using zero-multiplication fully connected layers can significantly improve training and inference speeds.

In contrast to existing lightweight transformer variants, which typically focus on reducing model size through parameter pruning, low-rank approximation, or simplified attention mechanisms, LiteKANformer introduces a fundamentally different lightweight design paradigm.

Specifically, LiteKANformer jointly integrates Kolmogorov–Arnold Networks (KAN) to replace conventional MLP layers with edge-wise learnable spline functions, Dynamic Tanh (DyT) to eliminate normalization layers without sacrificing training stability, and matmul-free fully connected layers to significantly reduce computational overhead. Unlike low-rank MLPs or linear-attention-based designs that primarily rely on linear approximations for efficiency, KAN provides enhanced non-linear representation capability with fewer parameters, enabling more accurate modeling of subtle texture variations and irregular defect patterns commonly observed in microdevice surfaces. Rather than optimizing individual components in isolation, LiteKANformer performs a co-design of activation representation, normalization strategy, and computation primitives within a unified transformer block, achieving an effective balance between expressive power and efficiency. This design is particularly well suited for microdevice surface defect detection, where global contextual modeling, fine-grained detail preservation, and deployment efficiency are simultaneously required.

3.4. Datasets

This study employs two microdevice surface defect datasets for experimentation: a PCB surface defect dataset and a semiconductor chip surface defect dataset. PKU-Market-PCB is a public PCB dataset released by Peking University [31], containing 1,386 images and six defect types: missing holes, mouse bites, open circuits, short circuits, stray copper, and pseudo-copper, as shown in Figure 4.

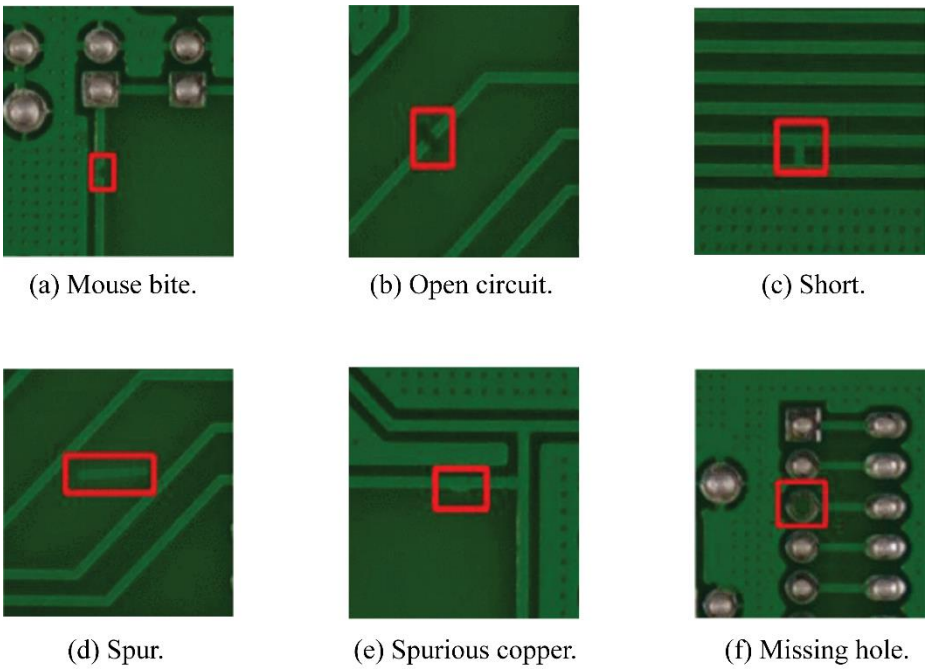


Figure 4. Defect types in the PCB dataset

Missing holes are pre-designed vias or mounting holes that fail to form, preventing proper fixation or signal flow. Mouse bites are irregular edge or trace defects caused by cutting or material damage. Open circuits result from broken traces that interrupt current flow, while short circuits arise when insulated traces are unintentionally connected, risking overload or burnout. Pseudo-copper refers to uneven, peeling, or poorly coated copper layers that impair conductivity and stability. Together, these defects cover structural, electrical, and material issues, providing comprehensive benchmarks for PCB defect detection.

The semiconductor defect dataset is a publicly available collection comprising 5,874 images of five categories of semiconductor chip defects. Bounding boxes for detection targets were annotated using the labelImg tool [32].

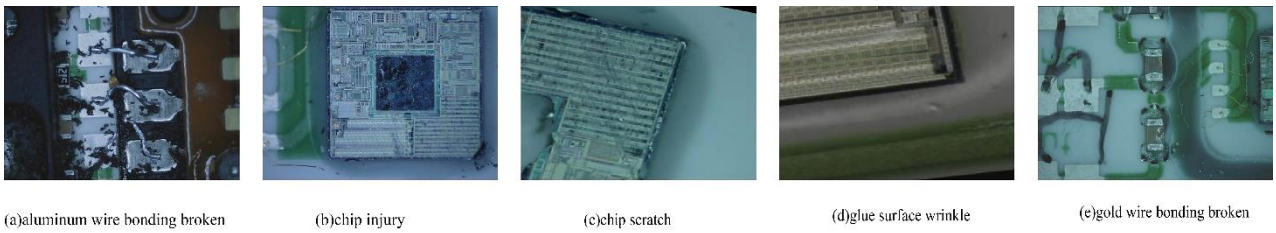


Figure 5. Five defect detection types: (a) aluminum wire bond breakage, (b) chip damage, (c) chip scratches, (d) adhesive layer wrinkles, (e) gold wire bond breakage

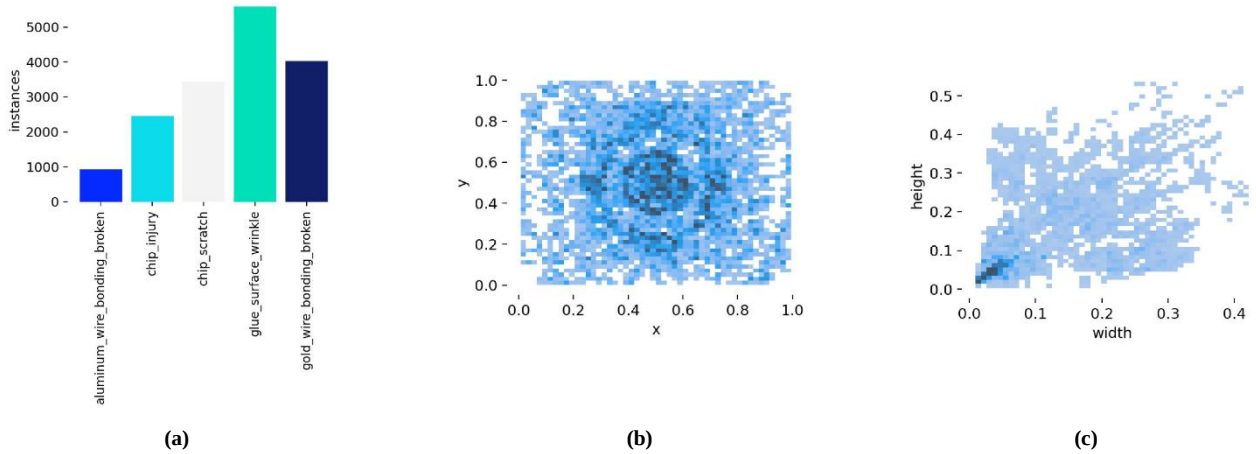


Figure 6. Dataset label distribution: (a) Type distribution, (b) Location distribution, (c) Size distribution

The dataset contains the main types of semiconductor chip defects. The type diagrams and label distributions are shown in Figures 5 and 6. Aluminum wire bonding is the core process in semiconductor packaging, and the main failure mode of this structure comes from its fracture. The structural damage that occurs on silicon substrates and surface circuits is called chip damage, which is its fatal defect. Chip scratches are damages on the surface of a chip caused by mechanical friction, and their influence depends on the depth of the scratches. Glue surface wrinkles refer to the wrinkles, indentations, or ripples that appear on the surface of the semiconductor package. These may pose internal structural risks. The breakage of gold wire bonding can lead to the failure of semiconductor packaging. To increase training data diversity and strengthen the model's generalization capability, several data augmentation operations were introduced during training. These included random image flipping, rotation, and scaling, as illustrated in Figure 7.

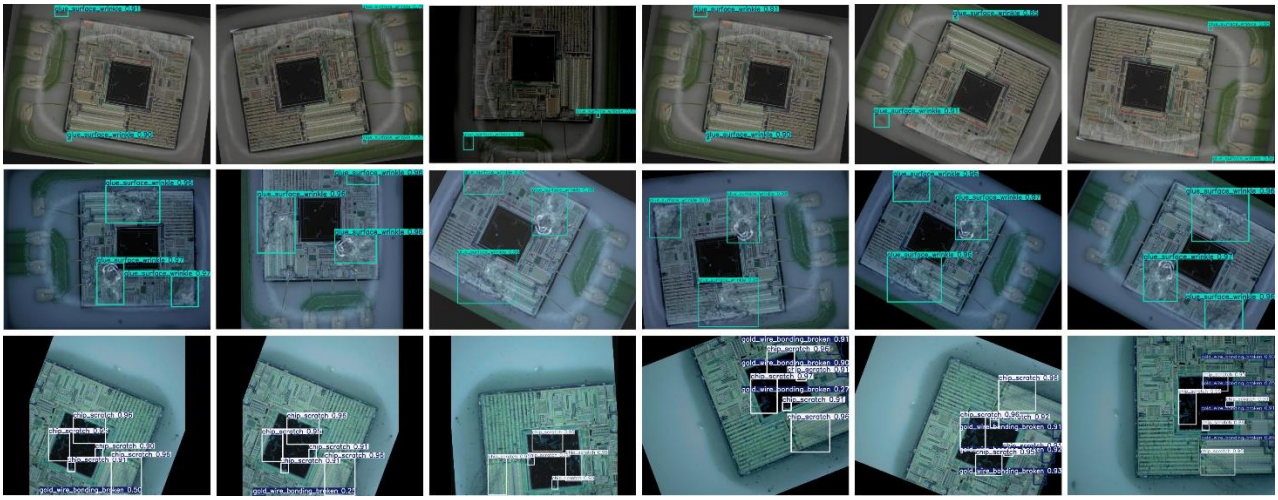


Figure 7. Data augmentation effect diagram

3.5. Evaluation Methods

This paper adopts accuracy, recall, and F1 score, average precision mean (mAP@50 and mAP@50 95) and the mean Intersection over Union (mIoU) as its evaluation metrics. These indicators mainly quantify performance based on the intersection area between predicted bounding boxes and real annotations [33].

Firstly, TP (True Positive) is defined as the predicted bounding box that correctly identifies real defects, with its intersection greater than the union (IoU) ≥ the set threshold. FP (False Positive) occurs when the IoU between the predicted bounding box and any real bounding box is less than 0.50, or when non-existent defects are incorrectly marked. FN (False Negative) refers to the true flaw missed by the model, which is the lack of predicted boxes that meet sufficient overlap requirements.

Precision reflects the proportion of correctly identified positive cases (true positives, TP) among all predicted positive cases (including TP and false positives, FP). Recall measures a model's ability to detect actual defects, and it represents the proportion of successfully identified real defects (true positives, TP) to all actual defects (including TP and false negatives, FN). F1 score is a balanced evaluation metric, which combines precision and recall and uses their harmonic mean to evaluate overall detection performance. Intersection over Union (IoU) quantifies the spatial consistency between predicted bounding boxes and ground truth boxes, which is calculated as the ratio of their overlapping area (intersection) to the union area of the two boxes.

$$mAP@50 = \frac{1}{N}\sum_{i=1}^{N} AP_i(IoU \geq 0.50) \tag{3}$$

$$mAP@50{:}\,95 = \frac{1}{10}\sum_{t=0.5}^{0.95} mAP_t \tag{4}$$

The two formulas above correspond to the definitions of mAP@50 and mAP@50:95. mAP@50 is a widely adopted metric for assessing object detection performance and represents the mean Average Precision calculated at an IoU threshold of 0.50. Specifically, the Average Precision (AP) is computed separately for each class and then averaged across all N categories. In comparison, mAP@50:95 extends this evaluation by averaging mAP values obtained at multiple IoU thresholds ranging from 0.50 to 0.95 with a step size of 0.05, resulting in a mean value over ten thresholds. By incorporating both relaxed and stringent overlap requirements, this metric provides a more comprehensive assessment of model robustness under varying detection precision conditions.

4. Experimental Process and Discussion

On the PKU-Market-PCB dataset, we conducted comparative and ablation experiments. The baseline group conducted experiments using the basic framework of the Ultralytics YOLO model [34] and additional tests were performed on the YOLO11s, YOLO11m, YOLO10s, YOLO5m, and LKF-YOLO models. The ablation experiment tested the effect of removing LiteKANformer and CBAM. For each model, the training was fixed at 300 epochs with a learning rate of 0.01, and a batch size of 2. The SGD optimizer was used and three experiments with random initializations were conducted on the same hardware device.

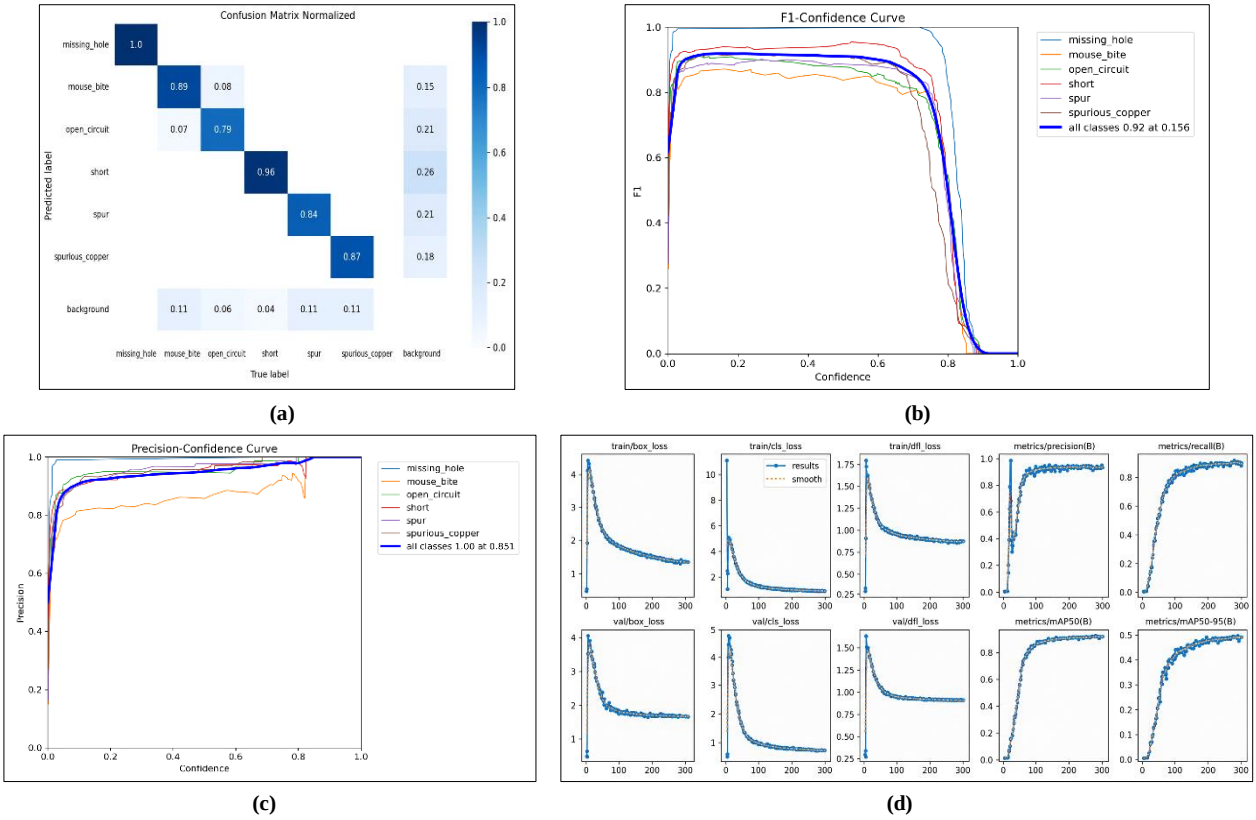
The performance of LiteKANformer was evaluated separately on a semiconductor defect dataset. In the native framework of YOLOv5, there exists a model incorporating the transformer module C3TR. This module was replaced with LiteKANformer for experiments to compare the performance gap between the two.

The dataset has been reasonably partitioned, with a ratio of approximately 7:1:2 between the training, testing, and validation sets and each set containes all types of defects. These datasets provide solid support for evaluating the generalization ability of models under different types of defects.

4.1. Training and Detection Performance

As shown in Figure 8, during the training phase, the training loss values showed a significant downward trend and tended to stabilize and the change pattern of the loss values during the validation phase was similar to that during the training phase.

From the evaluation metrics, the precision and recall values rapidly increase and approach saturation, which indicates that the model's recognition accuracy and recall ability for targets are gradually optimized to a stable level. The rapid stabilization of mAP@50 and mAP@50:95 reflects the excellent performance of this model in detecting targets within a wide range of IoU threshold intervals and a great comprehensive detection ability. Overall, the model exhibits a favorable convergence trend during the training process and showcases robust object detection performance.



(a) **(b)**

(c) **(d)**

Figure 8. Model Training Process. (a) Normalized Confusion Matrix (b) F1 Curve (c) Precision Curve (d) Round-Evaluation Values Curve

4.2. Comparative Experiments

4.2.1. Comparison with YOLOv10 on the PCB Dataset

Table 1 presents the comparison results between the micro-device surface defect detection model and YOLOv10 on the PCB dataset.

Table 1. Controlled Experiment Results

Model	Precision	Recall	mAP@50	mAP@50:90	GFLOPs
LKF-YOLO	95.2%	92.3%	94.2%	49.9%	23.4
YOLOv5m	91.3%	91.4%	93.7%	48.1%	47.9
YOLOv10	92.8%	86.6%	92.7%	48.7%	24.5
YOLOv11m	93.0%	92.3%	93.8%	49.8%	68.5
YOLOv10s	93.5%	90.2%	92.8%	48.1%	23.7

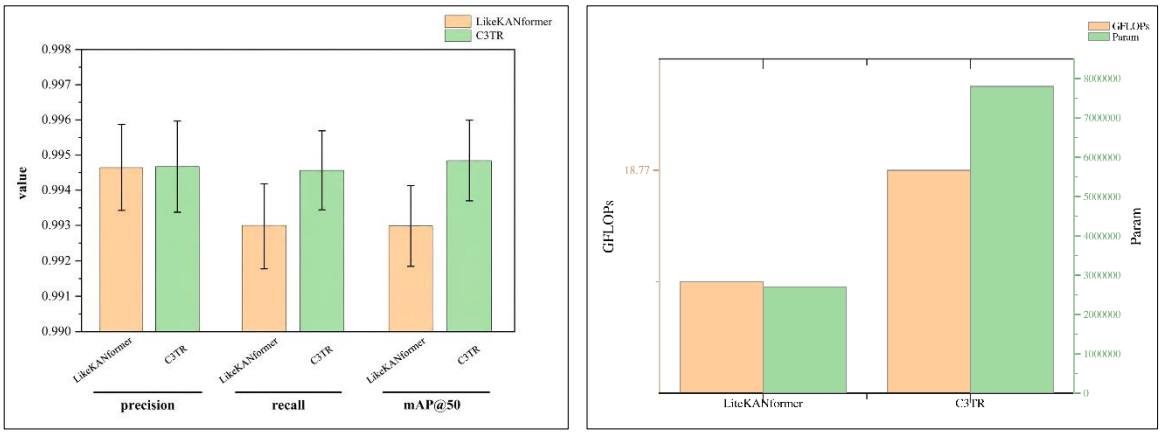
In terms of precision, LKF-YOLO (95.2%) achieves the highest value among all compared models, surpassing YOLOv10 (92.8%), YOLOv5m (91.3%), YOLOv11m (93.0%), and YOLOv10s (93.5%). The results indicate that LKF-YOLO produces fewer false positives and is more suitable for applications that require highly reliable classification results, LKF-YOLO has an absolute advantage in terms of accuracy. In terms of recall rate, LKF-YOLO ranks among the top with 92.3%, equaling YOLOv11m and surpasses YOLOv10, YOLOv5m, and YOLOv10s. It reflects stronger comprehensiveness in defect detection with a lower false negative rate. In terms of mAP metrics, LKF-YOLO performs better than YOLO models, which indicates that LKF-YOLO maintains competitive accuracy in detection tasks. This result also highlights LKF-YOLO's outstanding adaptability to various shapes, sizes, and edge blur defects since the mAP metrics better capture performance under stricter IoU thresholds.

From the perspective of model complexity, LKF-YOLO performs much better than YOLOv5m and YOLOv11m with a GFLOP value of 23.4 and it is also slightly lighter than YOLOv10. Only YOLOv10s exhibits a lower computational load, but this comes at the cost of reduced detection accuracy. Therefore, LKF-YOLO achieves a more favorable balance between precision and computational efficiency. In other words, it provides powerful detection performance with minimal resource requirements. This makes it better suited for deployment on devices with limited computing power, while preserving both the robustness and accuracy of defect detection tasks.

4.2.2. Comparative Experiments with C3TR on Semiconductor Defect Datasets

These experiments were conducted by replacing the C3TR (in the YOLOv5 framework) module with LiteKANformer. The core performance of LiteKANformer and C3TR in object detection tasks was quantitatively evaluated using precision, recall, and mAP@50 metrics, and standard deviation was calculated to reflect model performance stability.

As shown in Figure 9, LiteKANformer and C3TR achieved precision values of 0.99464 and 0.99465 respectively. It demonstrates that they have identical performance and both models exhibit high reliability. For recall indicators, C3TR is similar to LiteKANformer, with a difference of 0.00158 between 0.99454 and 0.99296. Overall, the accuracy of LiteKANformer and C3TR is similar, both within the high-precision range of 0.99 or above and they both meet the basic requirements for high-precision detection tasks. However, the computational load (GFLOPs) and parameter count (Param) comparison between LiteKANformer and C3TR reveals significant differences in efficiency and resource consumption, which provides crucial insights for adapting models to deployment scenarios.



(a) **(b)**

Figure 9. Evaluation Comparison of LiteKANformer and C3TR Modules (a) Comparison of evaluation metrics using LiteKANformer and C3TR modules (b) Performance comparison using LiteKANformer and C3TR modules

In terms of computational complexity, LiteKANformer achieved 6.323 GFLOPs, far lower than C3TR's 18.77 GFLOPs (a reduction of 66.3%). This means that LiteKANformer requires fewer floating-point operations in a single forward inference pass and it effectively reduces the load on hardware computing units. In terms of parameters, compared to C3TR's 7.83 million parameters, LiteKANformer has only 2.72 million parameters, which is a reduction of 65.2% in parameters. This optimization directly reduces the memory requirements of the model, which makes it particularly advantageous to deploy in memory constrained scenarios such as edge computing devices. As shown in Figure 9(a), LiteKANformer maintains detection accuracy comparable to C3TR, but reducing computational load and parameters to approximately one-third of C3TR's level.

4.3. Ablation Experiment

To further evaluate the contribution of key components within LKF-YOLO, ablation experiments were conducted by selectively removing CBAM and LiteKANformer. Model A removed the CBAM attention mechanism. Model B removed the LiteKANformer module. As shown in Table 2, removing either component results in a significant decrease in all evaluation metrics.

Table 2. Ablation Experiment Results

Model	Precision	Recall	mAP@50	mAP@50:90
LKF-YOLO	95.2%	92.3%	94.2%	49.9%
Model A	91.7%	83.5%	89.2%	45.8%
Model B	89.3%	78.8%	88.2%	44.1%

In terms of precision, LKF-YOLO achieved 95.2%, while Model A and Model B only achieved 91.7% and 89.3%. In terms of recall, LKF-YOLO's recall rate was 92.3%, but Model A's was only 83.5% and Model B's was only 78.8%. These results indicate that both modules have made significant contributions to improving the model's detection performance. The mAP@50 metric of LKF-YOLO reached 94.2%, while Model A and Model B saw their mAP@50 decrease to 89.2% and 88.2% respectively. It confirms that the addition of these two modules can improve the fit of

detection. More importantly, for the mAP@50:90 metric, LKF-YOLO (49.9%) significantly outperformed Model A (45.8%) and Model B (44.1%). This result better reflects the overall robustness under stricter accuracy levels. When the contributions of the two modules are viewed separately, it can be seen that the presence of LiteKANformer plays a more important role than CBAM in the detection accuracy of this model.

These results collectively validate that the addition of LiteKANformer module and the CBAM module can greatly enhance the model's representational ability and detection stability. The LiteKANformer module enhances global feature interaction and multi-scale perception, while the CBAM module improves feature attention and localization accuracy. Their combined effect enables LKF-YOLO to achieve higher detection accuracy and a stronger generalization ability.

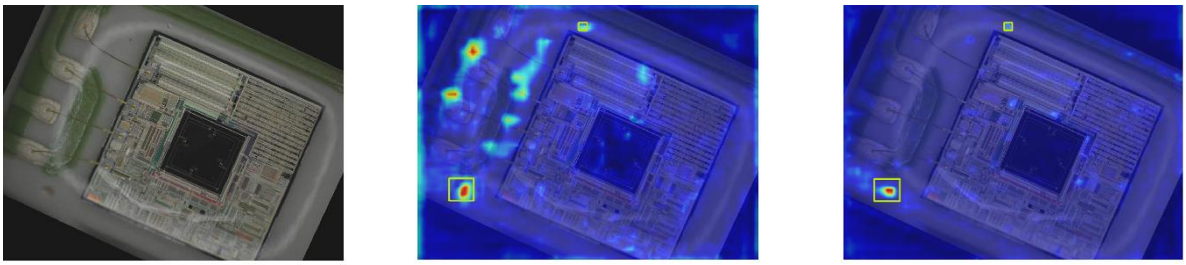
4.4. Discussion of Experimental Results

From the perspective of the training process, the model gradually improved and stabilized in the learning task as the number of iterations increased. The change in loss value indicates that there is no significant overfitting and the model has strong generalization ability.

Experimental results across multiple datasets further demonstrate the effectiveness of the proposed design. LiteKANformer and the resulting LKF-YOLO framework achieve strong performance in terms of precision, recall, and mAP. As reported in Table 1, the proposed method attains a precision of 95.2% and a recall of 92.3%, while effectively reducing both false positives and false negatives. Moreover, LKF-YOLO consistently outperforms the baseline YOLO model in mAP, indicating robust detection capability across defects of varying scales and shapes. Despite this performance gain, the model complexity of LKF-YOLO remains substantially lower than that of larger models such as YOLOv5m and YOLOv11m. On semiconductor defect datasets, the LiteKANformer module reduces both parameter count and computational cost by more than 65%, while achieving accuracy comparable to C3TR, highlighting its suitability for lightweight deployment scenarios.

The ablation results further highlight the importance of CBAM and LiteKANformer modules. In the ablation experiment, it is found that removing any of them would significantly reduce precision, recall, and mAP and the removal of LiteKANformer has a more significant impact on accuracy. This indicates that the two modules play a crucial role not only in spatial localization, but in enhancing feature attention and global interaction.

Using Grad-CAM [35] to generate heatmaps for critical regions of the trained model (Figure 10), the comparison before and after optimization shows a reduction in the heatmap and a decrease in temperature. This indicates that the model is lighter in weight and more efficient in computational reasoning. In addition, the hot zone of the new model at the defect location is more concentrated, which indicates an enhanced defect recognition ability.



(a) Original image (b) Heat map at the heads of YOLOv11 (c) Heat map at the heads of YOLO used LiteKANformer

Figure 10. Grad-CAM heatmaps: (a) the target image containing two adhesive surface wrinkles; (b) heatmap of the head region using the YOLOv11 model; (c) heatmap of the head region using the LiteKanformer model

Compared with existing surface defect detection methods, the proposed LiteKANformer and LKF-YOLO achieve a superior balance between accuracy, robustness, and computational efficiency. Traditional CNN-based approaches [2, 7] mainly rely on local hierarchical features, which limits their ability to capture global contextual information and results in reduced robustness under complex backgrounds or irregular defect distributions. By introducing LiteKANformer, LKF-YOLO enhances global feature interaction while remaining lightweight, leading to higher precision and recall across defects of varying scales and shapes. Compared with transformer-based models such as Cas-VSwin Transformer [4] and CINFormer [6], which improve global modeling at the cost of increased complexity, LKF-YOLO delivers comparable or better detection performance with significantly fewer parameters and lower computational overhead. In comparison with recent YOLO-based variants [10–12], which often improve performance by introducing additional multi-scale or attention modules, LKF-YOLO demonstrates stronger robustness to subtle and multi-scale defects without excessive architectural complexity. The ablation study confirms that LiteKANformer is the primary contributor to accuracy improvement, while CBAM further refines spatial attention and feature discrimination. LKF-YOLO effectively addresses the limitations of existing methods in detecting small and complex defects under practical industrial constraints.

Overall, LiteKANformer effectively achieves lightweighting, and its integration with the CBAM module enables LKF-YOLO to achieve a balance between precision, recall, robustness, and efficiency, making it an efficient and lightweight solution for microdevice and semiconductor defect detection.

5. Conclusion

This study addresses the long-standing challenge of balancing detection accuracy and model lightweighting in microdevice surface defect detection. To this end, LiteKANformer, a lightweight Transformer-based module that integrates Kolmogorov–Arnold Networks, normalization-free Dynamic Tanh, and non-multiplicative fully connected layers is proposed. By redesigning the Transformer feed-forward and normalization components, LiteKANformer enhances non-linear feature representation while significantly reducing parameter count and computational overhead. Based on this module, a novel detection framework named LKF-YOLO is constructed by embedding LiteKANformer into the backbone of the YOLO architecture and optimizing multi-scale feature fusion in the neck network. This design effectively combines the global contextual modeling capability of Transformers with the efficient local feature extraction of convolutional networks, making it particularly suitable for microdevice surface images characterized by small defect size, irregular morphology, and complex backgrounds.

Comprehensive experimental tests carried out using a PCB surface defect dataset and a semiconductor chip defect dataset confirm the validity of the proposed method. On the PCB dataset, LKF-YOLO delivers better performance regarding precision, recall, and real-time inference in comparison to mainstream YOLO-based detectors, while keeping a notably lower computational overhead. On the semiconductor defect dataset, LiteKANformer exhibits detection accuracy comparable to the transformer-based C3TR module, while reducing parameter count by approximately 3.1% and improving inference frame rate by 2.3%, highlighting its efficiency advantages. In addition, Grad-CAM visualization results indicate that LiteKANformer enables the model to focus more precisely on defect regions, confirming its enhanced discriminative feature representation capability. Overall, LiteKANformer and the proposed LKF-YOLO framework provide an effective and practical solution for high-precision defect detection under resource-constrained conditions, offering strong potential for real-world industrial deployment in microdevice and semiconductor manufacturing.

5.1. Outlook and Future Work

Although LiteKANformer has demonstrated strong performance in both detection accuracy and model lightweighting, several directions remain worthy of further investigation. As the experiments in this study were primarily conducted on a limited number of defect datasets, future research should explore the generalization capability of LiteKANformer across other microdevices and large-scale industrial products to comprehensively evaluate its cross-domain adaptability. Although LiteKANformer has been substantially optimized in terms of parameter count and computational complexity, methods for achieving real-time deployment on edge computing platforms still require further investigation. Future work may therefore focus on deeper optimization through the integration of model pruning, knowledge distillation, and quantization techniques.

Overall, LiteKANformer presents a novel approach to microdevice surface defect detection. It not only extends the combined application of Transformer and KAN architectures in academic research but also provides technical support for the development of efficient industrial defect detection systems. Future studies should further expand the model in terms of cross-domain generalization, lightweight deployment, multimodal detection, and interpretability enhancement to promote its application across a broader range of industrial inspection tasks.

6. Declarations

6.1. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.2. Funding

The author received no financial support for the research, authorship, and/or publication of this article.

6.3. Institutional Review Board Statement

Not applicable.

6.4. Informed Consent Statement

Not applicable.

6.5. Declaration of Competing Interest

The author declares that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

[1] Ling, Q., & Isa, N. A. M. (2023). Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey. IEEE Access, 11, 15921–15944. doi:10.1109/ACCESS.2023.3245093.

[2] Bhardwaj, R. (2023). Semiconductor Wafer Defect Detection using Deep Learning. PriMera Scientific Engineering, 4, 3–13. doi:10.56831/psen-04-097.

[3] Zhao, Z., Wang, J., Tao, Q., Li, A., & Chen, Y. (2024). An unknown wafer surface defect detection approach based on Incremental Learning for reliability analysis. Reliability Engineering and System Safety, 244, 109966. doi:10.1016/j.ress.2024.109966.

[4] Gao, L., Zhang, J., Yang, C., & Zhou, Y. (2022). Cas-VSwin transformer: A variant Swin transformer for surface-defect detection. Computers in Industry, 140, 103689. doi:10.1016/j.compind.2022.103689.

[5] Ma, J., & Cheng, X. (2023). Fast segmentation algorithm of PCB image using 2D OTSU improved by adaptive genetic algorithm and integral image. Journal of Real-Time Image Processing, 20(1), 10. doi:10.1007/s11554-023-01272-0.

[6] Jiang, X., Guo, K., Lu, Y., Yan, F., Liu, H., Cao, J., ... & Tao, D. (2023). CINFormer: Transformer network with multi-stage CNN feature injection for surface defect segmentation. arXiv preprint arXiv:2309.12639. doi:10.48550/arXiv.2309.12639.

[7] Singh, S. A., Kumar, A. S., & Desai, K. A. (2023). Comparative assessment of common pre-trained CNNs for vision-based surface defect detection of machined components. Expert Systems with Applications, 218, 119623. doi:10.1016/j.eswa.2023.119623.

[8] Cao, Y., Pang, D., Zhao, Q., Yan, Y., Jiang, Y., Tian, C., Wang, F., & Li, J. (2024). Improved YOLOv8-GD deep learning model for defect detection in electroluminescence images of solar photovoltaic modules. Engineering Applications of Artificial Intelligence, 131, 107866. doi:10.1016/j.engappai.2024.107866.

[9] Tian, J. H., Feng, X. F., Li, F., Xian, Q. L., Jia, Z. H., & Liu, J. L. (2025). An improved YOLOv5n algorithm for detecting surface defects in industrial components. Scientific Reports, 15(1), 9756. doi:10.1038/s41598-025-94109-8.

[10] Ge, Y., Li, Z., & Meng, L. (2025). YOLO-MSD: a robust industrial surface defect detection model via multi-scale feature fusion. Applied Intelligence, 55(12), 1–18. doi:10.1007/s10489-025-06739-0.

[11] Ma, R., Chen, J., Feng, Y., Zhou, Z., & Xie, J. (2025). ELA-YOLO: An efficient method with linear attention for steel surface defect detection during manufacturing. Advanced Engineering Informatics, 65, 103377. doi:10.1016/j.aei.2025.103377.

[12] Shunmugam, R., Yogarayan, S., Abdul Razak, S. F., & Sayeed, M. S. (2025). IMpc-PyrYOLO: Hybrid YOLO Based Feature Pyramidal Network for Pest Detection in Rice Leaves. Emerging Science Journal, 9(3), 1731–1748. doi:10.28991/ESJ-2025-09-03-029.

[13] Khan, A., Rauf, Z., Sohail, A., Khan, A. R., Asif, H., Asif, A., & Farooq, U. (2023). A survey of the vision transformers and their CNN-transformer based variants. Artificial Intelligence Review, 56(Suppl 3), 2917–2970. doi:10.1007/s10462-023-10595-0.

[14] Yao, W., Bai, J., Liao, W., Chen, Y., Liu, M., & Xie, Y. (2024). From CNN to Transformer: A Review of Medical Image Segmentation Models. Journal of Imaging Informatics in Medicine, 37(4), 1529–1547. doi:10.1007/s10278-024-00981-7.

[15] Yao, T., Li, Y., Pan, Y., Wang, Y., Zhang, X. P., & Mei, T. (2023). Dual Vision Transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9), 10870–10882. doi:10.1109/TPAMI.2023.3268446.

[16] Zhang, T., Xu, W., Luo, B., & Wang, G. (2025). Depth-Wise Convolutions in Vision Transformers for efficient training on small datasets. Neurocomputing, 617, 128998. doi:10.1016/j.neucom.2024.128998.

[17] Somvanshi, S., Javed, S. A., Islam, M. M., Pandit, D., & Das, S. (2025). A survey on kolmogorov-arnold network. ACM Computing Surveys, 58(2), 1-35. doi:10.1145/3743128.

[18] Hussain, M. (2024). YOLOv1 to v8: Unveiling Each Variant-A Comprehensive Review of YOLO. IEEE Access, 12, 42816–42833. doi:10.1109/ACCESS.2024.3378568.

[19] Xu, J., Sun, X., Zhang, Z., Zhao, G., & Lin, J. (2019). Understanding and improving layer normalization. Advances in Neural Information Processing Systems, 32.

[20] Frick, T., Rigotti, M., Antognini, D. M., Giurgiu, I., & Malossi, A. C. I. (2025). Layer normalization for calibrated uncertainty in deep learning (U.S. Patent Application Publication No. US20250094784A1). U.S. Patent and Trademark Office, Washington, D.C., United States.

[21] Zhu, J., Chen, X., He, K., LeCun, Y., & Liu, Z. (2025). Transformers without Normalization. Proceedings of the Computer Vision and Pattern Recognition Conference 2025, 14901–14911. doi:10.1109/cvpr52734.2025.01388.

[22] Talha, S., Akhssas, A., Aarab, A., Aabi, A., Berkat, B., & Amouch, S. (2025). Robust Ensemble Machine Learning for Flash Flood Susceptibility Mapping Across Semiarid Regions. Civil Engineering Journal, 11(12), 4926–4959. doi:10.28991/CEJ-2025-011-12-02.

[23] Dey, S., Goswami, M., Sethi, J., & Pattnaik, P. K. (2025). Hyb-KAN ViT: Hybrid Kolmogorov-Arnold Networks Augmented Vision Transformer. arXiv preprint arXiv:2505.04740. doi:10.48550/arXiv.2505.04740.

[24] Kosson, A., & Jaggi, M. (2023). Multiplication-Free Transformer Training via Piecewise Affine Operations. Advances in Neural Information Processing Systems, 36.

[25] Guo, Y., Chen, Y., Liu, X., Peng, W., Zhang, Y., Huang, X., & Ma, Z. (2024). Ternary Spike: Learning Ternary Spikes for Spiking Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 38(11), 12244–12252. doi:10.1609/aaai.v38i11.29114.

[26] Zhu, R. J., Zhang, Y., Abreu, S., Sifferman, E., Sheaves, T., Wang, Y., ... & Eshraghian, J. K. (2024). Scalable matmul-free language modeling. arXiv preprint arXiv:2406.02528. doi:10.48550/arXiv.2406.02528.

[27] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716. doi:10.3390/make5040083.

[28] Xiao, R., Wang, H., Wang, L., & Yuan, H. (2025). C3Ghost and C3k2: performance study of feature extraction module for small target detection in YOLOv11 remote sensing images. Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024, 13550, 139. doi:10.1117/12.3059792.

[29] Lv, M., & Su, W. H. (2023). YOLOV5-CBAM-C3TR: an optimized model based on transformer module and attention mechanism for apple leaf disease detection. Frontiers in Plant Science, 14, 1323301. doi:10.3389/fpls.2023.1323301.

[30] Agac, S., & Durmaz Incel, O. (2023). On the Use of a Convolutional Block Attention Module in Deep Learning-Based Human Activity Recognition with Motion Sensors. Diagnostics, 13(11), 1861. doi:10.3390/diagnostics13111861.

[31] Huang, W., Wei, P., Zhang, M., & Liu, H. (2020). HRIPCB: a challenging dataset for PCB defects detection and classification. The Journal of Engineering, 2020(13), 303–309. doi:10.1049/joe.2019.1183.

[32] Ke, H., Li, H., Wang, B., Tang, Q., Lee, Y. H., & Yang, C. F. (2024). Integrations of LabelImg, You Only Look Once (YOLO), and Open Source Computer Vision Library (OpenCV) for Chicken Open Mouth Detection. Sensors & Materials, 36(11), 4903-4913.

[33] Niu, Y., & Yin, J. (2024). PA-Net: Trustworthy weakly supervised point cloud semantic segmentation with primary–auxiliary structure. Computers and Electrical Engineering, 119, 109555. doi:10.1016/j.compeleceng.2024.109555.

[34] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics [Computer software]. GitHub. Available online: https://github.com/ultralytics/ultralytics (accessed on May 2026).

[35] Wang, S., & Zhang, Y. (2023). Grad-CAM: Understanding AI Models. Computers, Materials and Continua, 76(2), 1321–1324. doi:10.32604/cmc.2023.041419.