scientific reports

OPEN

An improved EAE-DETR model for defect detection of server motherboard

Jian Chi[1], Keming Zhang[1,3], Hongpu Zhang[1,3], Guowang Niu[1,3] & Zhihao Zheng[2,3]✉

This study addresses the challenges of missed and false detections in server motherboard defect identification, which arise from factors such as small target size, positional rotation deviations, and uneven scale distribution. To tackle these issues, we propose an enhanced detection model, EAE-DETR, which is based on an improved version of RT-DETR. Initially, we developed the CSP-EfficientVIM-CGLU module to enhance feature extraction capabilities while simultaneously reducing the model's parameter count through the implementation of dynamic gated convolution and global context modeling. Subsequently, we introduced the AIFI-ASSA module, designed to mitigate background noise interference and improve sensitivity to minor defects by employing an adaptive sparse self-attention mechanism. Lastly, we constructed the EUCB-SC upsampling module, which integrates depth convolution and channel shuffling strategies to enhance feature reconstruction efficiency. Experimental results on the PCBA-DET dataset indicate that EAE-DETR achieves a mean Average Precision (mAP) of 78.5% at IoU = 0.5 and 32.6% across IoU thresholds of 0.5 to 0.95, surpassing the baseline RT-DETR-R18 by 3.6% and 6.5%, respectively. Furthermore, the model demonstrates a reduction in parameter count by 21.7% and a decrease in computational load by 12.0%. On the PKU-Market-PCB dataset, the mAP50 reached 96.1%, and the mAP50:95 reached 65.1%.This model effectively facilitates high-precision and high-efficiency defect detection for server motherboards in complex industrial environments, thereby offering a robust solution for the intelligent manufacturing sector.

Keywords Server motherboard, Defect detection, Dynamic convolution, Sparse attention, Feature fusion

With the continuous improvement of the degree of integration of electronic equipment, the engineering design and production process of modern server motherboards are facing higher challenges. In this context, the quality control system of PCBA (printed circuit board components) has become a key link to ensure product reliability[1]. In the current industrial practice, intelligent screening technology represented by Automatic optical inspection (AOI) has been widely used in surface defect identification in the production process. However, the quality diagnosis after the terminal assembly link still mainly relies on manual visual inspection. This traditional quality inspection mode not only requires the allocation of a large number of professional and technical personnel but also has problems such as low detection efficiency and rising labor costs. Moreover, due to personnel fatigue or experience differences, the false detection rate remains high[2]. Given this, building an intelligent terminal quality inspection system has significant engineering application value for realizing accurate defect positioning, effectively controlling repair costs, and reducing resource losses[3].

At present, there are two main technical paths for defect screening of server motherboards in the industrial field. The first is the traditional quality inspection mode based on manual visual inspection, which relies on operators to identify defects through visual observation and experience accumulation. This model is subject to the difference between operator skill level and individual judgment, the prevalence of misjudgment, the slow speed of unit man-hour detection quantity, and other technical bottlenecks, so it is difficult to adapt to the demand for accurate defect identification in intelligent manufacturing scenarios[4]. The other is an intelligent solution that integrates deep learning algorithms and performs multi-dimensional feature analysis on high-resolution images through Convolutional Neural Network (CNN), YOLO, and other models to meet the industrial requirements of low cost, high precision, and high efficiency[5].

[1]Zhangjiakou Jingxi Cloud Computing Co. LTD, Zhangjiakou 075000, China. [2]School of Information Engineering, Hebei University of Architecture, Zhangjiakou 075000, China. [3]Keming Zhang, Hongpu Zhang, Guowang Niu and Zhihao Zheng contributed equally to this work. ✉email: zzh2023209135@hebiace.edu.cn

The current mainstream deep learning object detection methods are mainly divided into two technical routes. Among them, the two-stage detection framework realizes the step-by-step processing of target localization and recognition through the region generation mechanism: first, the potential candidate regions are generated through the region target network, and then these regions are refined classification and bounding box regression[6]. Typical representatives of this kind of method include R-CNN series models (region-based convolutional neural networks) and their improved versions, such as Fast R-CNN and Faster R-CNN, whose advantages are reflected in high detection accuracy and recall rate. However, the two-stage processing flow of region generation and feature re-calculation needs to be executed, which leads to high computational complexity and slow inference speed of the system, which is often difficult to meet the performance requirements in scenarios that require real-time processing[7].

Another class of single-stage object detection methods uses bounding box regression technology, which has significant advantages in computational efficiency compared with multi-stage detection models. Typical representatives include the YOLO series real-time detection framework SSD (Single Shot Multi-box detector) and other classical algorithms, whose core feature is that target localization and classification can be completed through a single forward propagation[8]. This high efficiency makes it a good application prospect in scenarios such as mobile terminals with limited computing resources. Based on YOLO series algorithms for PCB and motherboard defect detection, researchers have made many efforts and achieved many excellent results. However, the research and improvement of the RT-DETR target detection algorithm on the server motherboard is rarely seen in the current research community[9].

RT-DETR is the first real-time end-to-end detection model using Transformer architecture developed by the Baidu Research Institute team. The model innovatively integrates a lightweight Transformer module and efficient decoding structure, combines a multi-scale feature interaction strategy through the anchor-free mechanism, breaks through the performance bottleneck between real-time performance and detection accuracy of traditional detection models, and achieves a better balance between detection speed and recognition accuracy[10]. However, when facing the server motherboard detection task, due to the difficulty of feature extraction, it is easy to cause missed detection and false detection, and then the detection accuracy of the model is reduced. Therefore, in order to solve the above problems, this paper proposes a small target detection method EAE-DETR based on improved RT-DETR, which aims to reduce angular interference, achieve effective feature interaction, and achieve better target detection in complex scenes[11]. The main contributions are as follows:

·Using the EfficientViMBlock and Convolutional GLU module, combined with the CSP idea, the CSP-EfficientViM-CGlu module was creatively proposed to reduce the number of parameters and calculation of the DETR model, while maintaining the generalization ability of the model. The multi-stage hidden state fusion is used to further enhance the detection accuracy of the model for motherboard defects.

·Combined with an Efficient up-convolution block and Shift Channel Mix block, the EUCB-SC module is proposed to reduce the problem that it is difficult to extract target features due to shooting Angle when detecting motherboard defects and improve the detection accuracy.

·Adaptive Sparse Self-Attention is used to improve the AIFI module, and the features with low query-key matching scores are targeted to filter, so as to reduce the noise influence of irrelevant regions, maintain sparse characteristics, and retain more critical information to reduce missed detection and false detection of the target.

We will explain the effect of dynamic gated convolution and sparse self-attention mechanisms on enhancing small-scale defect detection of server motherboard images. It verifies the effect of the integrated EUCB-SC on improving the upsampling fidelity and positioning accuracy of the Transformer-based detection model. To compare the performance of the EAE-DETR model with standard YOLO and Faster R-CNN models in terms of accuracy and efficiency under industrial constraints. The comprehensive verification of our proposed model is carried out.

Related work

Research on two-stage algorithm

The PCB defect detection system developed by Geng et al. improved its performance through three stages: ResNet-50 was used to replace the original Faster R-CNN backbone network to enhance the feature representation ability; Feature Pyramid Network (FPN) was integrated to realize multi-scale defect feature fusion. The Focal Loss function was introduced to alleviate the problem of sample imbalance[12]. HU et al. proposed a lightweight improvement scheme and constructed a dual-path optimization mechanism in the Faster R-CNN framework. Firstly, they used Guided Anchoring RPN to improve the quality of anchor frame generation and combined the channel rearrangement module of ShuffleNetV2 to reduce the computational complexity[13]. According to the characteristics of integrated circuit packaging defects, Luo's team innovated to design a decoupled dual-branch architecture. Through the independent Spatial localization module (SLM) and semantic classification module (SCM), the bounding box regression and defect type recognition were optimized respectively. M.J. et al. constructed a thermal imaging defect analysis system, which combined a CNN feature extractor (ResNet-34) and Variational Autoencoder (VAE) to construct a dual feature space, and combined Regional saliency Analysis Module (RSAM) to locate abnormal heating regions[14].

One-stage algorithm study

Zhou et al. developed YOLOv7 enhanced architecture to improve performance through three technical innovations: constructing an SE attention module to optimize feature weight allocation; The Flexible Rectifier Unit (FReLU) was used to enhance the nonlinear representation ability. A cross-layer feature interaction path was designed to enhance semantic information fusion, and 2.3% mAP improvement was achieved on the COCO benchmark dataset[15]. Li et al. reconstructed the YOLOv3 feature extraction network, expanded the receptive field through the double residual module stacking strategy, and parallelized the CBAM hybrid attention mechanism, reaching 88.96% detection accuracy on the PCB dataset[16]. Wang Hengtao's team proposed a lightweight improvement scheme: a geometric pruning strategy was used to compress the model parameter scale (41% reduction), a Coordinate Attention (CA) module was embedded to enhance spatial positioning accuracy, and the genetic algorithm was used to optimize the anchor box clustering process so that the training convergence speed was increased by 35%[17]. B.A. et al. constructed a CNN-Transformer hybrid architecture and established a multi-scale defect feature expression system through the cascaded design of YOLOv5 local feature extraction and Transformer global context modeling[18]. Park et al. established a three-level classification system for PCB defects and innovatively adopted the ResNet50-YOLOv7 dual-model collaborative framework to realize the joint optimization of defect classification and localization[19].

RT-DETR

RT-DETR (Real-Time Detection Transformer) is an innovative object detection solution developed by the Baidu Feipaddle team. This model successfully applies the Transformer architecture to the field of real-time end-to-end detection for the first time. Aiming at the contradictory pain points of traditional detection models in speed and accuracy, the research team effectively broke through the performance bottleneck by integrating multi-dimensional innovative technologies: using a lightweight Transformer module to reconstruct the detection process, combining dynamic detection head optimization strategy and anchor removal design, and innovatively integrating multi-level feature fusion mechanism[20]. The architecture consists of three core components: the feature extraction backbone network is responsible for the basic semantic information extraction task, the efficient hybrid encoder realizes cross-scale feature interaction enhancement, and the improved Transformer decoder improves the detection confidence through the auxiliary prediction mechanism to form a complete real-time detection system (see Fig. 1 for the specific architecture). This technical solution significantly improves the accuracy of target recognition in complex scenes while maintaining real-time reasoning speed[21].

The visual feature processing system of the RT-DETR model adopts a hierarchical architecture. The backbone network hierarchically processes the input image through deep convolution and gradually extracts a multi-resolution feature map containing basic visual features, which provides a hierarchical representation basis for subsequent modules[22]. Its efficient hybrid encoder is composed of a dual-path feature enhancement mechanism: On the one hand, the attention-based Intra-Scale Feature Interaction Module (AIFI) focuses on high-level semantic features S5 and realizes the adaptive association of high-dimensional features through the lightweight Transformer encoder, which reduces the computational complexity and strengthens the expression ability of the target key features[23]. On the other hand, the Cross-scale Feature Fusion Module based on a convolutional neural network (CCFF) uses a hierarchical skip connection strategy to integrate features of adjacent scales gradually through multi-level fusion blocks, and finally generate composite feature sequences with rich spatial-semantic information through channel-weighted fusion. In particular, the AIFI module is designed for the strong semantic characteristics of high-level features, which avoids the invalid interaction problem caused by insufficient semantic information of low-level features, and effectively avoids the risk of redundant calculation and feature confusion[24].

At the decoding optimization level, the model innovatively introduces the dynamic Query screening mechanism (uncertain-minimal Query Selection) to solve the decoder initialization problem[25]. The strategy dynamically selects the feature vectors with high information density from the encoder output through confidence evaluation as the initial query and combines the multi-round refinement mechanism of the auxiliary prediction head to gradually optimize the spatial positioning and semantic classification results of the detection box. This two-stage optimization architecture not only ensures the quality of the initial query but also significantly improves the detection robustness of small and occluded objects through iterative feature alignment[26].

The proposed method

EAE-DETR

Building upon the RT-DETR framework, the DETR algorithm has been optimized and enhanced, resulting in the development of a novel object detection model termed EAE-DETR. The network architecture of EAE-DETR is depicted in Fig. 2. In the design of the DETR backbone network, a phased stacking strategy is employed to integrate the CSP-EfficientVIM-CGLU module into the critical feature extraction layer. This integration is co-optimized with the global context modeling capabilities of ViM through a dynamic feature selection mechanism facilitated by gated convolution. To address the challenges associated with feature reconstruction efficiency and cross-channel information fusion during the upsampling process, this research introduces an innovative combination of the Efficient Up-Convolution Block (EUCB) and the Shift Channel Mix block (SCM). The proposed EUCB-SC composite upsampling unit, which incorporates a dynamic parameter-sharing mechanism, replaces the original upsampling module in DETR. Furthermore, to mitigate the susceptibility of the Adaptive Feature Interaction module (AIFI) in the RT-DETR detector to background noise when identifying defects on motherboards, an enhanced AIFI-ASSA module based on a dynamic sparse attention mechanism has been proposed as a replacement for the original AIFI module. These enhancements are specifically designed to address the challenges of low detection accuracy arising from factors such as target rotation, positional deviation, small target size, and significant scale differences in the context of motherboard image detection.

EfficientVIM-CGLU module

Object detection is a critical area within computer vision, particularly in the context of defect detection for industrial quality control. While the traditional RT-DETR architecture has demonstrated significant efficacy, it continues to encounter challenges related to inadequate feature extraction capabilities, elevated computational complexity, and limited detection accuracy for small objects, particularly in high-precision defect detection scenarios such as those involving server motherboards. This paper presents an enhanced RT-DETR methodology

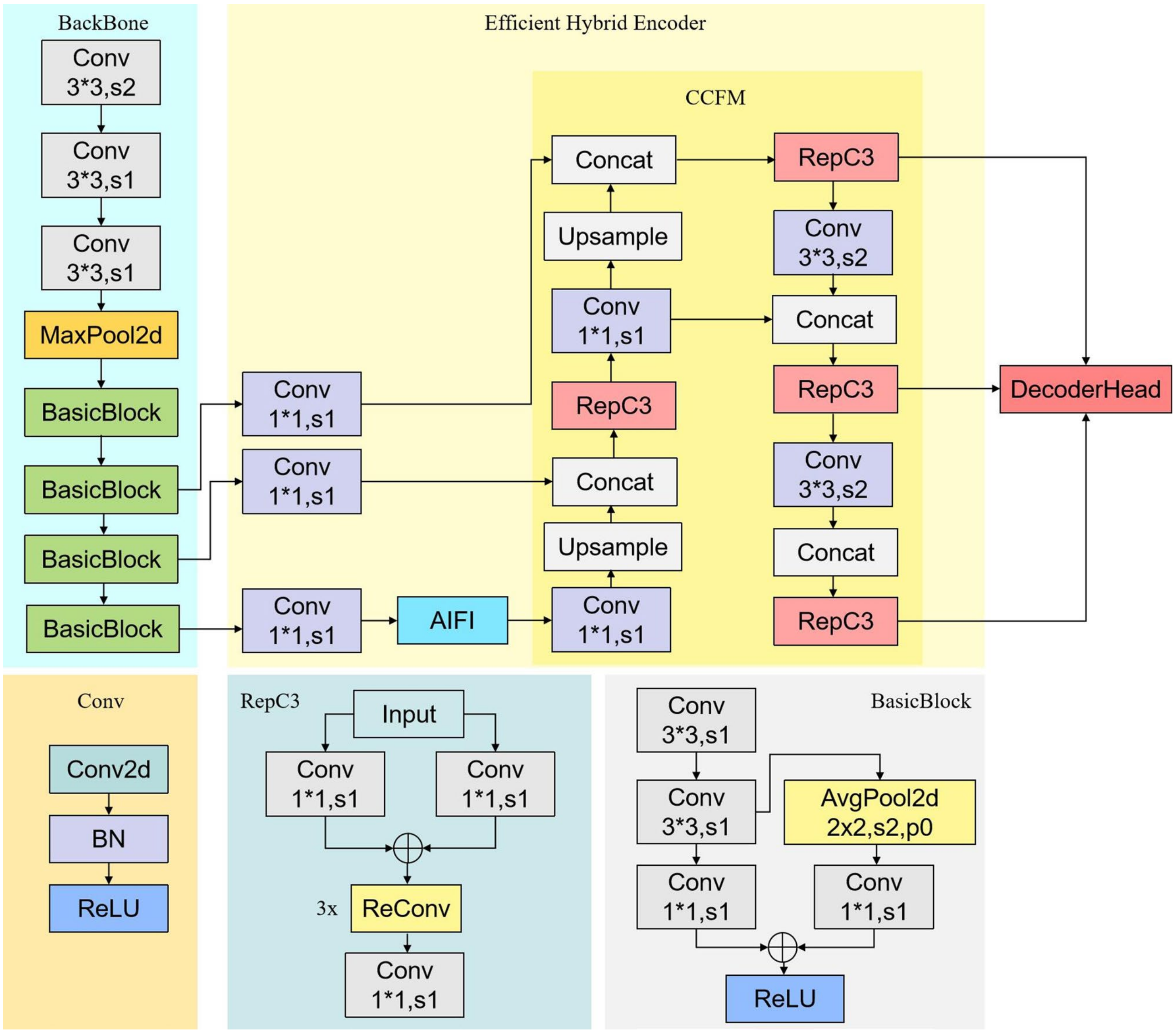

Fig. 1. The overall framework of the RT-DETR model.

that incorporates EfficientVIM-CGlu. This approach integrates the EfficientVIM architecture with the Convolutional Gated Linear Unit (CGLU) and introduces the concept of the Cross Stage Partial Network (CSP) to markedly enhance the model's performance in detecting defects on server motherboards. The structural representation of this model is illustrated in Fig. 3.

The conventional RT-DETR architecture employs BasicBlock as the fundamental component of the backbone network. However, BasicBlock presents several limitations, including restricted feature extraction capabilities, challenges in capturing intricate textures and subtle defect characteristics on server motherboards, as well as issues related to computational efficiency, parameter size, and overall computational complexity, all of which adversely impact real-time performance. To address these challenges, we propose the EfficientVIM-CGLU architecture. In this framework, the feedforward network (FFN) is substituted with a Convolutional Gated Linear Unit (CGLU). Additionally, the feature transfer pathway is optimized by integrating the concept of CSP, and a hierarchical feature fusion strategy is implemented to enhance the network's capacity to process multi-scale features, thereby improving the model's overall performance.

The EfficientVIM-CGLU module operates through a series of interconnected components that enhance feature extraction and integration capabilities. The workflow can be summarized as follows:

In the first stage, it is firstly depthwise separable convolution 1 (DWConv1) :

$$x_1 = (1 - \alpha_0) \times x + \alpha_0 \times DWConv_1(x) \quad (1)$$

Deep convolution is applied to the input feature map and then weighted fusion is performed with the original features. Where x is the input feature map with shape (B, C, H, W). x_1 outputs the feature map in the first

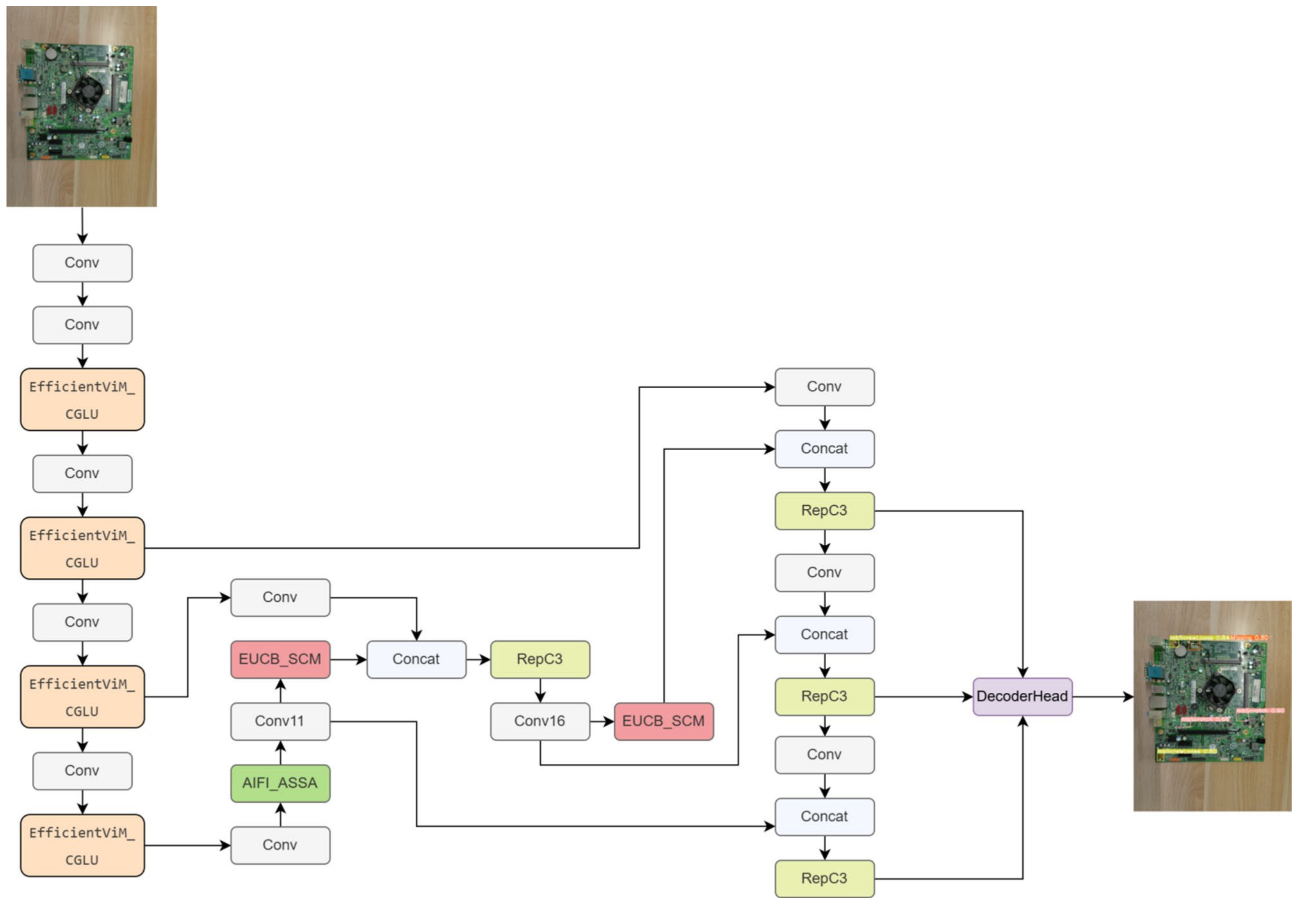

Fig. 2. The overall framework of the EAE-DETR model.

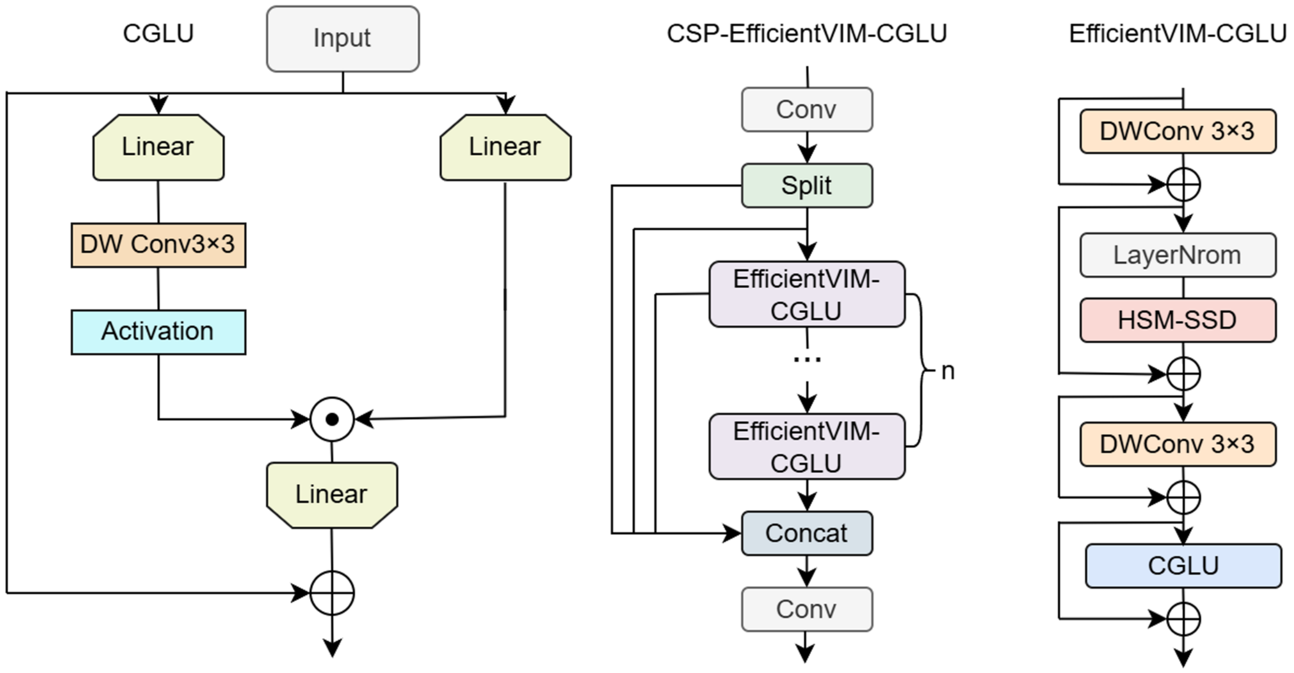

Fig. 3. EfficientVIM-CGLU module diagram.

stage, α_0 is the weight coefficient of the first stage, and $DWConv_1\left(\right)$ is the depth-wise separable convolution operation.

Where the depth convolution is defined as follows:

$$DWConv_1\left(x\right)=Conv2D\left(x,kernel=3\times\ 3,groups=dim,padding=1\right) \tag{2}$$

Each input channel performs a 3×3 convolution operation independently to capture local spatial features without increasing the number of parameters. $kernel=3\times\ 3$ represents the size of the convolution kernel is 3×3, $groups=dim$ represents the number of groups is equal to the number of channels, to achieve depthwise separable convolution, $padding=1$ means that the fill is 1, keeping the size of the feature map unchanged.

The second stage is HSM-SSD(hidden state mixer-based state space duality).This is the core component of the module and contains sophisticated state-space modeling.The first is input preprocessing:

$$x_{flat}=LayerNorm1D\left(flatten\left(x_1,dims=\left[H,W\right]\right)\right) \tag{3}$$

The 2D feature maps are converted to 1D sequences and normalized to stabilize training. x_1 is the first-stage output with shape (B, C, H, W). $flatten\left(\right)$ means flattening the spatial dimensions, (B, C, H, W) → (B, C, H×W). $LayerNorm1D\left(\right)$ represents layer normalization in the channel dimension. x_{flat} is the normalized flat feature of shape (B, C, L), where L = H×W.

This is followed by state-space parameter generation:

$$BCdt=DWConv2D\left(Conv1D\left(x_{flat},out=3\times\ state\_dim\right)\right) \tag{4}$$

The three key parameter matrices of the state space model are generated, and the spatial correlation of the parameters is enhanced by convolution. $Conv1D\left(\right)$ stands for 1 × 1 convolution and maps the c-dimension to the 3×state_dim dimension. $state\_dim$ represents the state dimension (64 by default). $DWConv2D\left(\right)$ is a 3×3 depth convolution to enhance spatial information interaction. $BCdt$ contains tensors with three parameters B, C, and dt of shape (B, 3×state_dim, H, W).

$$B,C,dt=split\left(flatten\left(BCdt\right),dim=state\_dim\right) \tag{5}$$

The merged parameter tensor is separated into three independent state-space parameters.Next is the adaptive state transition matrix:

$$A=softmax\left(dt+A_{param}\right) \tag{6}$$

The learnable parameters and the dynamic time step are combined to generate a normalized state transition probability distribution. Where A_{param} represents the learnable state transition parameter of shape (state_dim,). dt is the dynamic time step of shape (B, state_dim, L). A is the adaptive state transition matrix of shape (B, state_dim, L).The next step is to implement the gating mechanism:

$$\mathrm{h}_{proj},z=split\left(Conv1D\left(\mathrm{h},out=2\times\ d\_inner\right),dim=d\_inner\right) \tag{7}$$

The state representation is projected to higher dimensions and divided into two branches in preparation for the gating mechanism. h_{proj} is the main branch feature of shape (B, d_inner, state_dim). z is the gated branch feature of shape (B, d_inner, state_dim).

$$\mathrm{h}_{out}=Conv1D\left(\mathrm{h}_{proj}\odot\ SiLU\left(z\right)+\mathrm{h}_{proj}\odot\ D\right) \tag{8}$$

The gating mechanism is applied to enhance the feature expression ability by combining the activated gating signals and direct connections. D stands for the learnable direct connection parameter of shape 1. h_{out} is the gated output of shape (B, d_inner, state_dim).The third stage is depthwise separable convolution 2 (DWConv2):

$$x_3=\left(1-\alpha_2\right)\times\ x_2+\alpha_2\times\ DWConv_2\left(x_2\right) \tag{9}$$

Deep convolution is applied again for spatial feature extraction and residual fusion with the HSM-SSD output. x_2 is the output feature map of HSM-SSD in the second stage. x_3 is the third stage output feature map. The α_2 is the weight coefficient of the third stage. $DWConv_2\left(\right)$ is the second depthwise separable convolution with parameters independent of $DWConv_1\left(\right)$.

The last stage is ConvolutionalGLU, which starts with feature projection:

$$x_{expand},v=split\left(Conv2D\left(x_3,out=2\times\ hidden\_features\right),dim=hidden\_features\right) \tag{10}$$

The input features are expanded and divided into two parallel branches, one for feature transformation and one for gating. $hidden\_features$ is used to hide the layer dimension, x_{expand} is the main branch feature with shape (B, $hidden\_features$, H, W), and v is the gated branch feature with shape (B, hidden_features, H, W).

The next step is deep convolutional activation:

$$x_{dw}=GELU\left(DWConv2D\left(x_{expand},kernel=3\times\ 3\right)\right) \tag{11}$$

Deep convolution and GELU activation are applied to the main branch to enhance the spatial feature expression. $GELU\left(\right)$ is the Gaussian error linear unit activation function. x_{dw} is the activated features with shape (B, $hidden\_features$, H, W).

Finally, it is the output projection and residual connection:

$$x_{ffn} = Dropout\left(Conv2D\left(Dropout\left(x_{gated}\right), out = dim\right)\right) \tag{12}$$

The features are compressed back to their original dimensions by projected convolution and Dropout.

$$x_4 = \left(1 - \alpha_3\right) \times x_3 + \alpha_3 \times \left(x_3 + x_{ffn}\right) \tag{13}$$

x_4 is the final output feature map.

AIFI-ASSA

The Real-time Detection transformer (RT-DETR) has proven to be highly effective in general object detection tasks. However, our experimental analysis indicates that the Attention to Feature Integration (AIFI) module within the conventional RT-DETR architecture exhibits several limitations when specifically applied to the domain of server motherboard defect detection. Firstly, the AIFI module demonstrates constrained feature representation capabilities, as it struggles to capture intricate texture anomalies and subtle structural defects that are characteristic of server motherboards. This limitation leads to diminished detection accuracy for small-scale defects, such as irregularities in solder joints and minor misalignments of components. Secondly, the AIFI module employs a uniform attention distribution across the entire feature space, which fails to adaptively concentrate on regions of the server motherboard that have a high likelihood of containing defects. These defects often display spatial sparsity, yet they are critical to the overall functionality of the motherboard. To mitigate these limitations, we propose the IAFI-ASSA, an innovative augmentation method that substitutes the traditional AIFI module with an adaptive sparse self-attention mechanism. This approach significantly enhances the detection of subtle defects in server motherboards while preserving the computational efficiency required for real-time applications.

In accordance with the principle of adaptive sparse attention, the AIFI-ASSA model is designed to concentrate computational resources on feature regions that are deemed most pertinent to the identification of potential defects. The architecture of the proposed module is depicted in Fig. 4.

The fundamental element of AIFI-ASSA is the Adaptive Sparse Self-Attention (ASSA) module, which adjusts the attention sparsity in accordance with the significance of the features. For the input feature map $\mathrm{X} \in \mathrm{R}^{\mathrm{B} \times \mathrm{C} \times \mathrm{H} \times \mathrm{W}}$, where B denotes the batch size, C represents the channel dimension, and H and W indicate the spatial dimensions, the ASSA operation can be delineated through the subsequent stages.First, we reshape the input feature map into a sequence representation:

$$X_{seq} = \mathrm{Reshape}\left(X\right) \in \mathrm{R}^{B \times (H \times W) \times C} \tag{14}$$

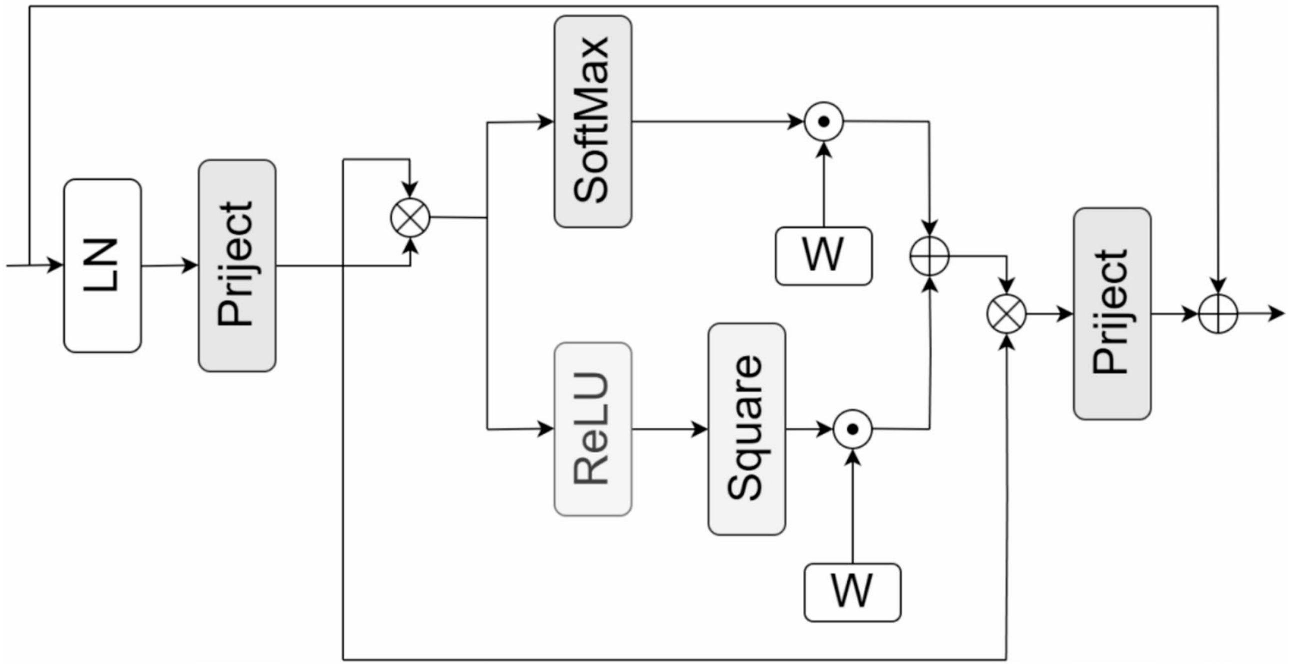

Fig. 4. Diagram of the module structure of AIFI-ASSA.

Then, the adaptive sparse attention mechanism computes:

$$\mathrm{ASSA}\left(X\right)=\mathrm{SoftMax}\left(\frac{QK^{T}}{\sqrt{d_k}}\odot\ M_{sparse}\right)V \tag{15}$$

Where $\mathrm{Q}=\mathrm{X_{seq}}W_{\mathrm{Q}}$, $\mathrm{K}=\mathrm{X_{seq}}W_{\mathrm{K}}$ and $\mathrm{V}=\mathrm{X_{seq}}W_{\mathrm{V}}$ is the query projection, key projection and value projection, respectively. $W_Q,W_K,W_V\in\ R^{C\times\ d}$ is the learnable parameter matrix and d_k is the dimension of the key vector.

Sparse attention mask M_{sparse} is adaptively generated based on query key similarity distribution:

$$\begin{gathered}S=\frac{QK^{T}}{\sqrt{d_k}}\\ M_{sparse}=\mathrm{TopK}\left(S,k\right)\end{gathered} \tag{16}$$

Where $TopK(S,k)$ keeps the highest k value in each row of S and masks the other values to 0. The sparsity parameter k is adaptively determined as follows:

$$k=\max\left(\min\left(\lceil\ \alpha\ \cdot\ (H\ \times\ W)\rceil\ ,k_{max}\right),k_{min}\right) \tag{17}$$

Where α is a learnable parameter representing the density ratio, and k_{min} and k_{max} are hyperparameters constraining the minimum and maximum number of posts attended.

After the ASSA module, the Location Feedforward Network (FFN) processes the features of participation:

$$\mathrm{FFN}\left(X\right)=\mathrm{Conv}_{1\times\ 1}\left(\mathrm{Dropout}\left(\mathrm{GELU}\left(\mathrm{Conv}_{1\times\ 1}\left(X\right)\right)\right)\right) \tag{18}$$

The complete forward process of AIFI-ASSA can be expressed as follows.

$$X'\ =\mathrm{Norm}_1\left(X+\mathrm{Dropout}\left(\mathrm{ASSA}\left(X\right)\right)\right),X''\ =\mathrm{Norm}_2\left(X'\ +\mathrm{Dropout}\left(\mathrm{FFN}\left(X'\ \right)\right)\right) \tag{19}$$

In this context, Norm_1 and Norm_2 denote operations related to layer normalization. The model put forth is capable of preserving both local contextual information, which is critical for identifying defect boundaries, and a comprehensive understanding of global structures, which is vital for accurately modeling the relationships among components during the inspection of server motherboards.

The AIFI-ASSA module successfully overcomes the shortcomings of the original AIFI methodology by employing an adaptive sparse attention mechanism. This advancement renders it particularly suitable for high-precision defect detection in the manufacturing of server motherboards, where both accuracy and efficiency are of paramount importance.

EUCB-SC

The traditional RT-DETR framework employs a straightforward nearest neighbor upsampling technique within its Feature Pyramid Network (FPN) to enhance the spatial resolution across various feature layers. While this approach is computationally efficient, it presents notable drawbacks when utilized in the demanding context of defect detection in server motherboards. Standard upsampling modules are inadequate in effectively capturing the fine-grained defect features that are essential for the identification of subtle manufacturing imperfections, such as loose screws, loose fan wiring, and scratches on the motherboard. Furthermore, the absence of spatial information exchange during the upsampling process constrains the network's capacity to comprehend the contextual nuances of defects within intricate printed circuit board (PCB) layouts.

In order to overcome the identified limitations, this study introduces the Enhanced Upsampling Channel Block with Shift Channel Mixing (EUCB-SC) module. This novel approach substitutes the conventional nearest neighbor upsampling technique utilized in the RT-DETR architecture at critical nodes, particularly within layers 11 and 16 of the detection head. The EUCB-SC module facilitates improved feature propagation through four primary mechanisms: (1) regulated recovery of spatial resolution, (2) depth convolution to maintain spatial context, (3) channel shuffling for the redistribution of information, and (4) targeted feature transfer to enhance boundary detection. The structural representation of the EUCB-SC module is illustrated in Fig. 5.

The EUCB-SC module consists of three main components working in sequence:

1. Upsampling with depth convolutions: Initially, the input feature map $\mathrm{X}\in\ \mathrm{R}^{\mathrm{B}\times\ \mathrm{C}\times\ \mathrm{H}\times\ \mathrm{W}}$ is bilinear upsampled, followed by deep convolution:

$$X_{up}=\mathrm{DWConv}\left(\mathrm{Upsample}\left(X\right)\right) \tag{20}$$

Among them, DWConv applies spatial filtering while preserving channel independence, which achieves more efficient computation while preserving spatial context compared to standard convolution.

2. Channel shuffling: The upsampled features are channel-shuffled to facilitate information exchange across channels:

$$X_{shuffle}=\mathrm{ChannelShuffle}\left(X_{up},g\right) \tag{21}$$

Where g denotes the number of groups (equal to the number of input channels C). The mathematical expression for this operation is:

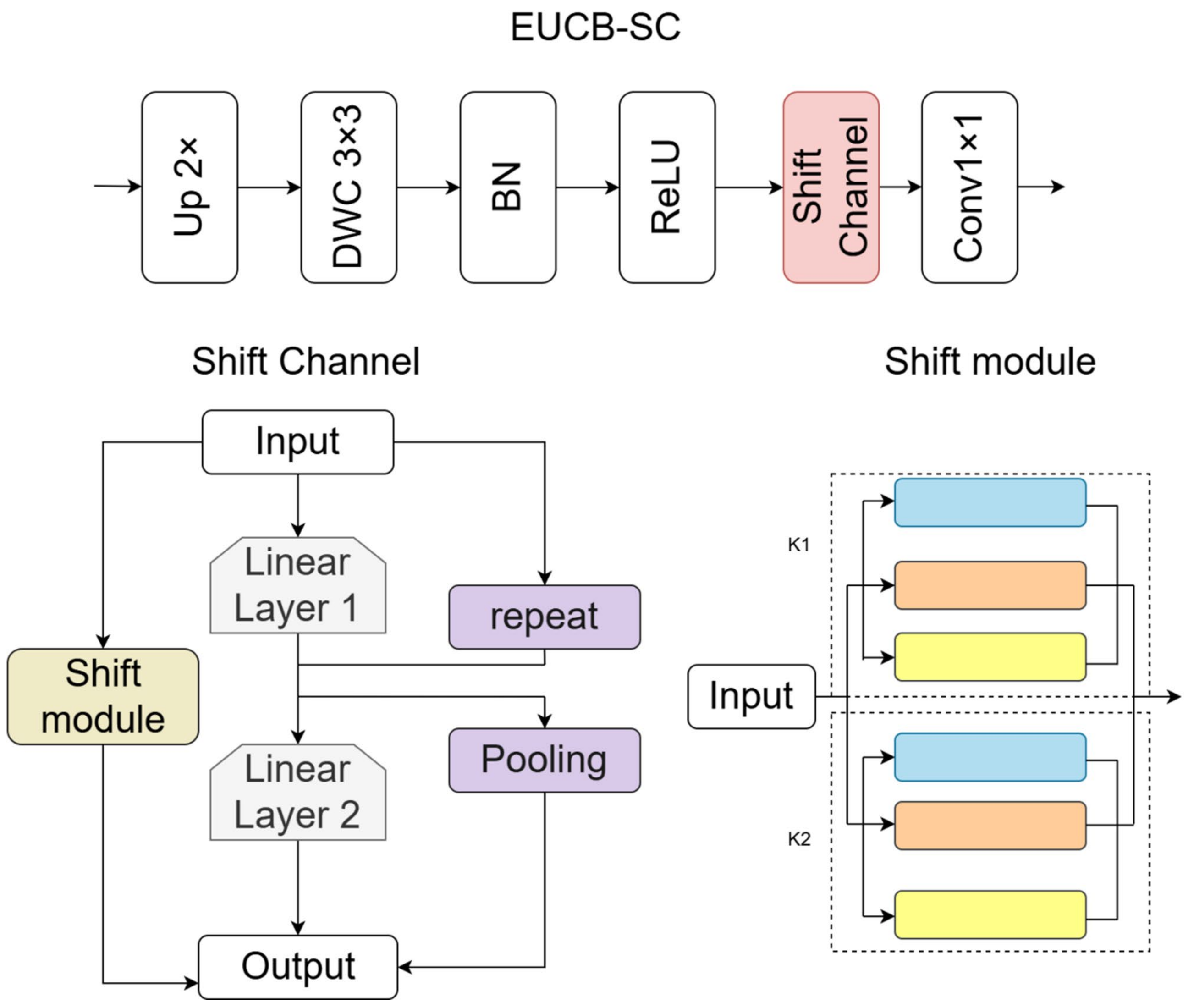

Fig. 5. The EUCB-SC module.

$$X_{shuffle} = \text{Reshape}\left(\text{Transpose}\left(\text{Reshape}\left(X_{up}, \left[B, g, C/g, H', W'\right]\right), 1, 2\right), \left[B, C, H', W'\right]\right) \quad (22)$$

3. Shifted channel mixing: The core innovation of the proposed method is the shifted channel mixing operation, which divides the feature channels into four equal groups and applies a directional shift to each group:

$$\begin{gathered} X_1, X_2, X_3, X_4 = \text{Split}\left(X_{shuffle}, 4, \text{dim} = 1\right) \\ X_1' = \text{Roll}\left(X_1, s, \text{dims} = 2\right) \\ X_2' = \text{Roll}\left(X_2, -s, \text{dims} = 2\right) \\ X_3' = \text{Roll}\left(X_3, s, \text{dims} = 3\right) \\ X_4' = \text{Roll}\left(X_4, -s, \text{dims} = 3\right) \\ X_{shift} = \text{Concat}\left(\left[X_1', X_2', X_3', X_4'\right], \text{dim} = 1\right) \end{gathered} \quad (23)$$

Where s denotes the displacement magnitude (set to 1 in our implementation). This operation shifts the features in horizontal and vertical directions, which enhances the sensitivity of the network to defect boundary and edge features.

4. Pointwise convolution: Finally, a 1×1 convolution integrates information from all channels:

$$Y = \text{PWConv}\left(X_{shift}\right) \quad (24)$$

The complete EUCB-SC transform can be expressed as follows.

$$Y = \text{PWConv}\left(\text{ShiftChannelMix}\left(\text{ChannelShuffle}\left(\text{DWConv}\left(\text{Upsample}\left(\text{X}\right)\right)\right)\right)\right) \quad (25)$$

Experiments and results

Data description

In the context of quality inspection within the circuit board assembly process, the PCBA defect detection system is required to accurately identify anomalies in component positioning and detect surface damage on components. To enhance the applicability of the dataset for practical use, the PCBA-DET detection database utilized in this research is constructed based on the characteristics inherent to industrial detection environments and employs a multi-dimensional data acquisition strategy. Initially, twelve prevalent models of computer motherboards are selected as representative samples, and the image acquisition environment is simulated using multi-angle industrial cameras at the production line detection station. This process culminates in the creation of a benchmark dataset comprising 4,000 high-resolution images. The dataset encompasses two primary detection dimensions: connector anomalies, which include six distinct defect types (such as loose or missing screws and loose or missing wiring), and surface damage, which consists of two defect types (specifically scratches on the motherboard and fan). In total, 19,240 defect areas have been meticulously labeled. Notably, all samples have undergone double labeling and verification in accordance with established industrial quality inspection standards, thereby ensuring the reliability of the labeling outcomes. The dataset is partitioned such that the training set contains 2,800 images, the validation set comprises 800 images, and the test set includes 400 images, adhering to a ratio of 7:2:1. Figure 6 illustrates the morphological characteristics of eight typical defect types: (1) loose fan screws, (2) missing fan screws, (3) loose motherboard screws, (4) missing motherboard screws, (5) loose fan wiring, (6) missing fan wiring, (7) scratches on the fan, and (8) scratches on the motherboard. Figure 7 depicts the distribution of label counts for various targets within the dataset, along with the corresponding target sizes. Figure 8 presents several images from the dataset, showcasing motherboards captured from diverse perspectives, models, and environments, thereby encompassing a wide range of motherboard detection scenarios encountered in actual industrial settings.

To fully evaluate the generalization ability of the proposed model, we conducted a comparative experiment using the PCB defect detection dataset publicly released by the Intelligent Robot Open Laboratory of Peking University. We will refer to this dataset below as PKU-Market-PCB.This dataset initially contained a total of 693 PCB defect images, covering six typical defect categories: incomplete welding, burrs, open circuits, short circuits, notches, and excess copper. Considering that the original data sample size was relatively limited and might not be sufficient for comprehensive training and validation of the model's generalization performance, we implemented a systematic data augmentation strategy. By comprehensively applying various image processing techniques such as image translation, rotation, adding noise, and adjusting brightness, we successfully expanded the dataset size significantly to 4158 images. We divided the enhanced complete dataset into training set, validation set, and test set in a ratio of 8:1:1.The presentation of the dataset is shown in Fig. 9, which includes the situations of Missing_hole, Mouse_bite, Open_circuit, Short, Spur and Spurious_copper.

Evaluation metrics and experimental configurations

This research establishes a deep learning experimental framework utilizing the Windows 11 × 64 platform (version 22H2). The hardware configuration comprises an NVIDIA GeForce RTX 3080 Ti graphics processor, which is equipped with 12GB of video memory. The software environment is based on the Python 3.10 interpreter and the PyTorch 2.1 framework, supplemented by the CUDA 11.8 acceleration library to facilitate model training. The RT-DETR-R18 network architecture has been selected as the baseline model for this experiment, adhering strictly to its official parameter configuration scheme.

During the data preprocessing phase, input images are uniformly resized to a resolution of 640 × 640 pixels, while maintaining the original aspect ratio through intelligent filling techniques to enhance training efficiency. The training rounds for PCBA-DET are 300, and for PKU-Market-PCB, they are 100., with the learning rate

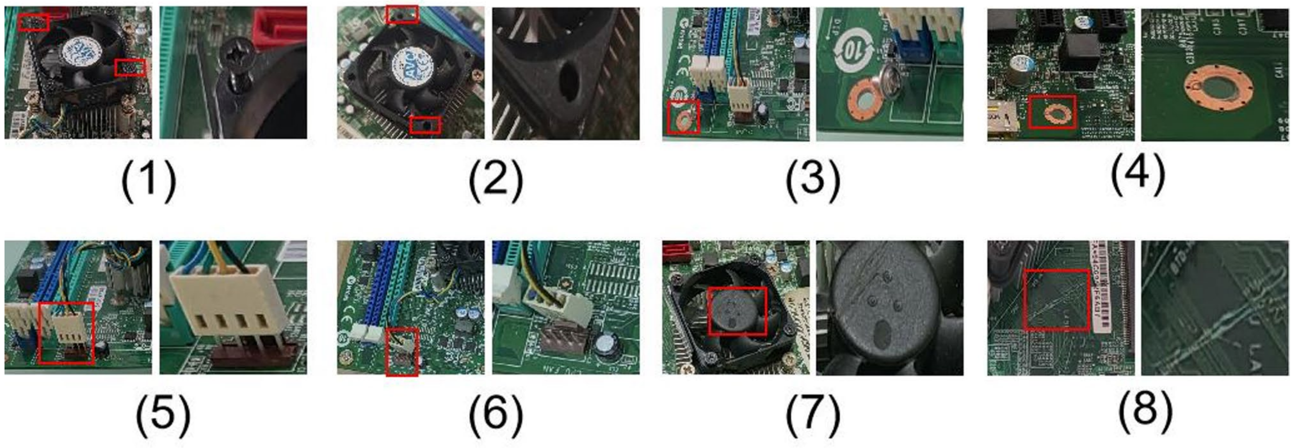

Fig. 6. Eight different kinds of defects for PCBA-DET.

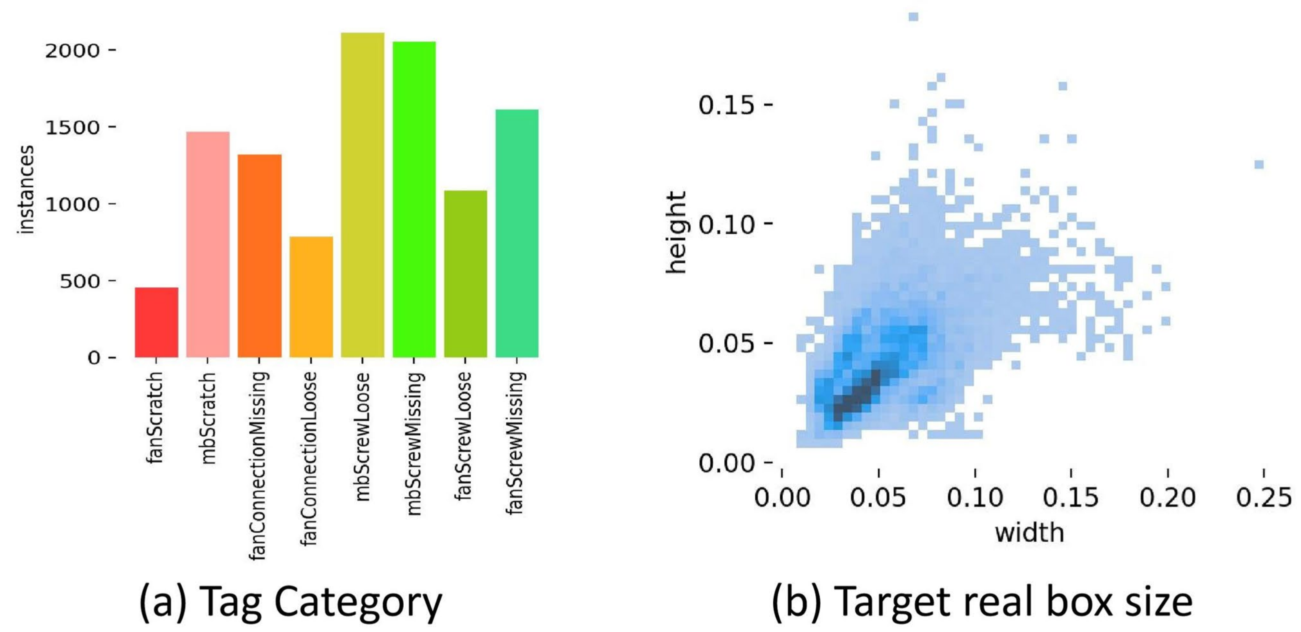

Fig. 7. Signature class information.

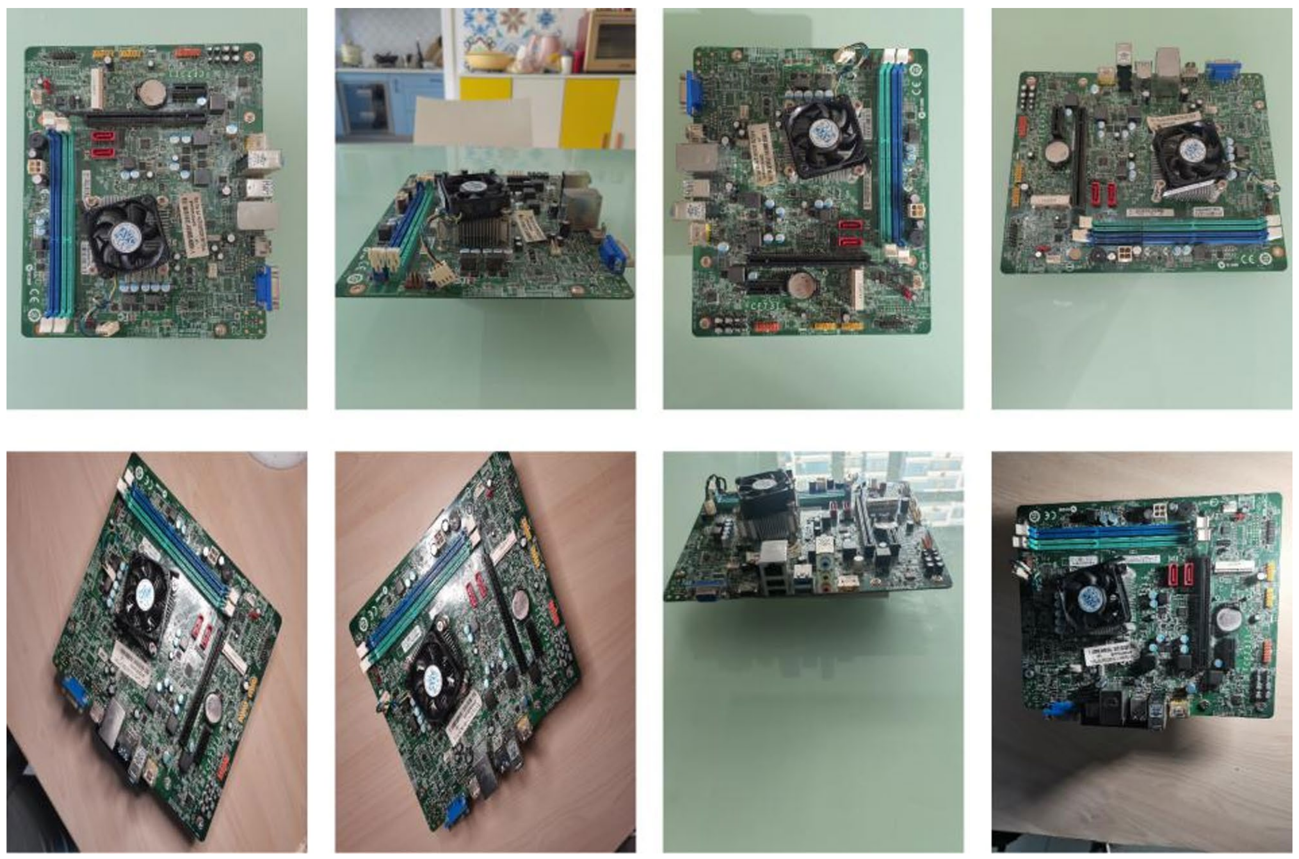

Fig. 8. PCBA-DET dataset display graph.

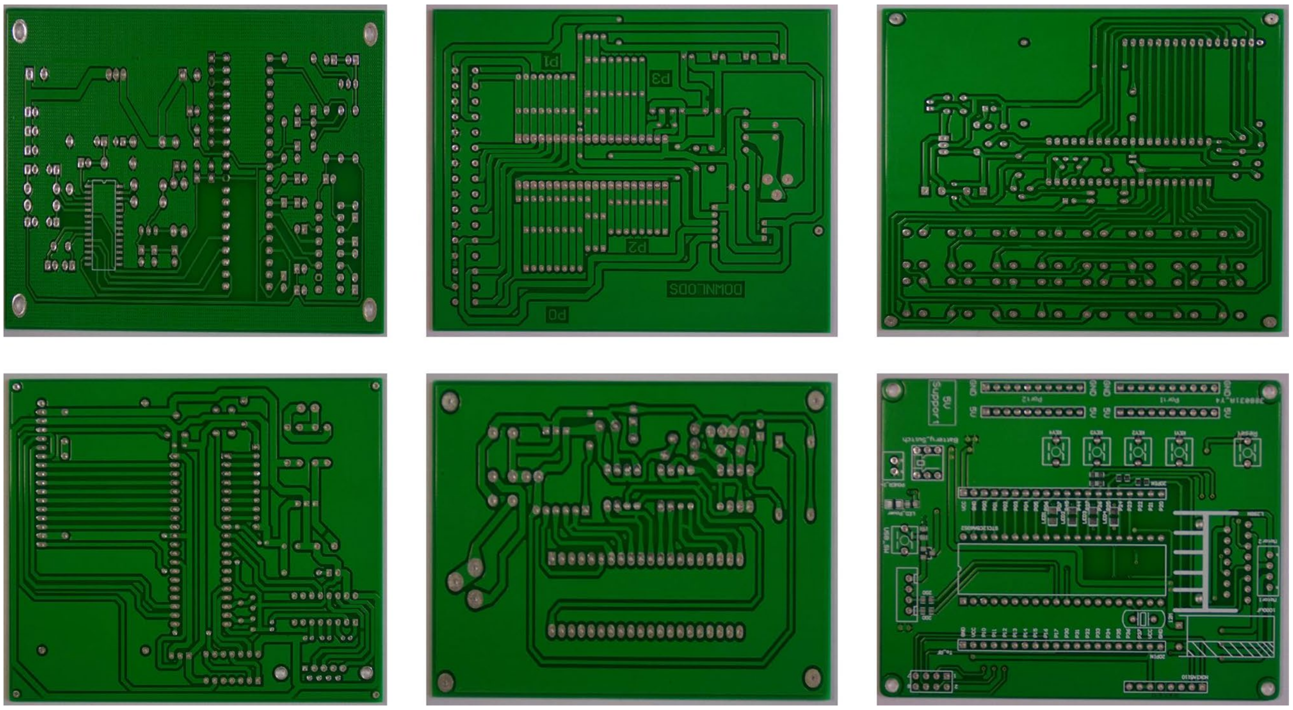

Fig. 9. PKU-Market-PCB dataset display graph.

Methods	mAP50 (%)	mAP50:95(%)	FPS	Parameters	GFLOPs
RT-DETR-R18	75.8	30.6	68	19.88 M	57.0
DETR + EfficientVIM-CGLU	76.7	31.2	77	14.58 M	48.3
DETR + AIFI-ASSA	76.2	31.1	69	14.78 M	47.3
DETR + EUCB-SC	75.9	30.8	72	14.28 M	46.8
DETR + EfficientVIM-CGLU + EUCB-SC	77.2	31.8	71	15.21 M	48.8
DETR + EUCB-SC + AIFI-ASSA	77.5	31.7	75	15.34 M	49.7
DETR + EfficientVIM-CGLU + AIFI-ASSA	77.6	31.9	76	15.42 M	49.1
DETR + EfficientVIM-CGLU + AIFI-ASSA + EUCB-SC	78.5	32.6	74	15.56 M	50.2

Table 1. Ablation experiments based on the PCBA-DET dataset.

being dynamically adjusted from an initial value of 0.01 to a minimum threshold of 0.0001, employing a cosine annealing strategy. All models are trained from scratch without using pre-trained models.

The performance evaluation framework encompasses two dimensions: core indicators and auxiliary indicators. The primary accuracy evaluation metrics employed in the domain of object detection are mAP50 (average precision at an intersection over a union threshold of 0.5) and mAP50-95 (average precision across intersection over union thresholds ranging from 0.5 to 0.95, with a step size of 0.05). Precision is defined as the ratio of true positive instances among the samples predicted as belonging to the positive class, while Recall indicates the proportion of actual positive class samples that are accurately identified. The auxiliary evaluation dimension incorporates computational efficiency metrics, including theoretical floating-point operations (FLOPs) to denote the model's computational complexity, trainable parameters to indicate the model's size, and frame rate (FPS) to assess inference speed during actual deployment. Notably, mAP50-95 provides a more comprehensive evaluation of the model's overall performance across varying positioning accuracy requirements by calculating the mean average precision at different IoU thresholds.

Comparative experiment

In order to comprehensively evaluate the contribution of each module to the whole network, the impact of each module on the network performance is analyzed by ablation experiments. We evaluate the effectiveness of each module by comparing mAP50, MAP50:95, floating-point computations (FLOPs), parameters, and FPS. The detailed experimental results are shown in Table 1.

Ablation studies indicate that the EAE-DETR model significantly enhances average detection accuracy while simultaneously reducing both the number of parameters and GFLOPs in comparison to the baseline model. The incorporation of three additional modules progressively enhances the performance of the RT-DETR model in the context of defect detection on server motherboards. Initially, the integration of the EfficientVIM-CGLU module resulted in an increase in mAP50 from 75.8 to 76.7% and an improvement in mAP50:95 from 30.6 to 31.2%. This modification leads to a reduction in parameters by 26.7% (from 19.88 M to 14.58 M) and a decrease in computational load by 15.3% (from 57.0 GFLOPs to 48.3 GFLOPs). The module enhances feature extraction capabilities through efficient channel interaction and the design of a Gated Linear Unit (GLU), thereby minimizing redundant parameters and improving computational efficiency.

Subsequently, the addition of the AIFI-ASSA module further elevates mAP50 from 76.7 to 77.6% and mAP50:95 from 31.2 to 31.9%. Although there is a slight increase in parameters (from 14.58 M to 15.42 M) and a marginal rise in computational demand (from 48.3 to 49.1 GFLOPs), this module enhances the model's capacity to capture local details of motherboard defects by dynamically allocating spatial and channel attention weights, thereby improving detection accuracy.

Finally, the incorporation of the EUCB-SC module results in an increase in mAP50 from 77.6 to 78.5% and mAP50:95 from 31.9 to 32.6%. The increase in parameters and computational requirements is minimal (from 15.42 M to 15.56 M and from 49.1 to 50.2 GFLOPs, respectively). This module optimizes defect localization in complex environments by integrating multi-scale contextual information, such as global defect distribution and background interference suppression, thereby enhancing robustness.

The cumulative effect of these three modules results in an improvement of mAP50 from 75.8 to 78.5% (+3.6%) and mAP50:95 from 30.6 to 32.6% (+6.5%). This enhancement is particularly advantageous for defect detection in server motherboard scenarios, effectively addressing challenges associated with small targets, intricate backgrounds, and subtle defects.

Comparative experiment

We compare EAE-DETR with other models on the PCBA-DET dataset, as shown in Table 2.It includes the current mainstream detection models YOLO family (YOLO11n, YOLO11s) and some of their variants (MELF-YOLOv5s). Some excellent two-stage models are also included.Among them, the improved algorithms PCBA-YOLO and YOLO-MBBi have high recognition and representativeness in the field of PCB defect detection.

Based on the comprehensive evaluation presented in Table 2, our proposed model demonstrates significant superiority over state-of-the-art object detection methods for server motherboard defect detection across critical performance metrics, while maintaining a balanced computational profile. Our model achieves a leading mAP50 of 78.5%, surpassing the closest competitor (YOLOv8s at 74.1%) by a substantial 4.4% point. This significant margin highlights its superior ability to correctly localize and identify defects at the standard IoU threshold. Demonstrating robust localization precision across stricter IoU thresholds, our model attains the highest mAP50:95 of 32.6%. This outperforms the next best model (YOLOv9s at 26.5%) by a notable 6.1% points, confirming its exceptional precision in bounding box regression critical for detailed defect analysis. Our model excels in both reducing false positives and minimizing false negatives. With a Precision of 76.8% (highest) and Recall of 75.4% (highest), it significantly outperforms alternatives like YOLOv8s (72.2% P, 71.6% R) and YOLO11s (72.3% P, 71.4% R). This dual strength ensures highly reliable defect identification with minimal misses.

In comparison with the newer family of DETR (RTDETRV2-R18) or with the latest DETR variants (D-Fine, DEIM)SOTA models. Our model also has significant advantages. Firstly, compared with D-Fine-S and DEIM-S,

Methods	mAP50 (%)	mAP50:95(%)	Precision(%)	Recall(%)	Parameters	GFLOPs
YOLOv8s	74.1	47.2	72.2	71.6	11.27 M	28.8
Faster-RCNN[27]	70.1	23.5	64.2	63.1	41.3 M	206
RTMDet-Tiny[28]	69.8	23.2	64.1	62.2	4.86 M	8.13
RetinaNet[29]	64.2	21.5	58.8	58.3	36.6 M	93.7
YOLOv5s	71.2	25.4	69.5	68.1	7.3 M	16.2
YOLOv10n	70.1	25.6	68.8	68.1	2.23 M	6.6
SSD[30]	50.2	17.2	40.6	38.2	26.21 M	85.3
YOLOv6m	68.3	24.7	62.7	60.6	34.5 M	82.2
YOLOX-Tiny[31]	66.2	23.1	61.9	60.8	5.01 M	7.57
YOLOv9s	71.2	26.5	70.2	68.4	9.6 M	25.3
MELF-YOLOv5s[32]	62.6	22.1	55.1	53.8	3.5 M	8.1
YOLO11n	70.8	25.3	68.3	66.4	2.62	6.4
YOLO11s	74.7	28.7	72.3	71.4	9.43 M	21.5
D-Fine-S	71.2	25.1	69.1	67.4	10.18 M	24.85
DEIM-S	70.5	24.3	68.4	67.1	10.26 M	24.68
RTDETRV2-R18	73.3	27.5	71.2	68.8	20 M	60
PCBA-YOLO[33]	72.4	26.3	70.1	68.2	12.4 M	20.8
YOLO-MBBi[34]	70.2	25.5	69.4	66.1	6.2 M	12.8
Ours	78.5	32.6	76.8	75.4	15.56 M	50.2

Table 2. The EAE-DETR dataset is compared with various other excellent methods in in PCBA-DET.

our model is slightly inferior to these two models in the number of parameters and calculations, but it is significantly ahead in mAP, Precision, and Recall. Our model improves the mAP50 by 7.3% points compared to D-Fine-S and 8% points compared to DEIM-S. Compared with RTDETRV2-R18, our model has achieved a lead in all aspects, not only the number of parameters and calculations are better than RTDETRV2-R18, but also the mAP50 is increased by 5.2% points, and the MAP50:95 is increased by 5.1% points.

While our model utilizes 15.56 M parameters and 50.2 GFLOPs, this computational footprint is strategically justified by its unparalleled accuracy gains. Crucially, it remains markedly more efficient than larger, less accurate models like Easter-RCNN (41.3 M params, 206 GFLOPs, 70.1% mAP50) and RetinaNet (36.6 M params, 93.7 GFLOPs, 64.2% mAP50). It also offers a compelling accuracy/efficiency trade-off compared to YOLOv6m (34.5 M params, 82.2 GFLOPs, 68.3% mAP50). Compared to models with similar or slightly lower computational costs like YOLOv8s (11.27 M params, 28.8 GFLOPs) and YOLOv9s (9.6 M params, 25.3 GFLOPs), our model delivers a dramatic accuracy boost (e.g., + 4.4% mAP50, + 6.1% mAP50:95 over YOLOv9s) for a moderate increase in parameters and FLOPs.Our model significantly outperforms highly efficient models designed for edge deployment (e.g., YOLOv10n: 2.23 M params, 6.6 GFLOPs, 70.1% mAP50; YOLO1ln: 2.62 M params, 6.4 GFLOPs, 70.8% mAP50; RTMDet-Tiny: 4.86 M params, 8.13 GFLOPs, 69.8% mAP50; MELF-YOLOv5s: 3.5 M params, 8.1 GFLOPs, 62.6% mAP50) by large margins (e.g., + 8.4% mAP50 over YOLOv10n). This establishes its position as the preferred solution where detection accuracy is paramount over minimal resource consumption.

The experimental results unequivocally demonstrate that our proposed model establishes a new state-of-the-art for server motherboard defect detection. It delivers statistically significant improvements in all core accuracy metrics (mAP50, mAP50:95, Precision, Recall) compared to a diverse set of leading models, including various YOLO versions (v5s, v6m, v8s, v9s, v10n, v11s), Easter-RCNN, RTMDet-Tiny, RetinaNet, SSD, YOLOX-Tiny, and MELF-YOLOv5s. While demanding more computational resources than the most compact models, the substantial gains in detection performance achieved by our model represent a highly favorable and effective trade-off, particularly for critical industrial inspection tasks requiring maximum reliability. The efficiency remains competitive and often superior when compared to other models achieving lower accuracy levels.

As demonstrated in Table 2, our proposed model achieves a landmark mAP50 of 78.5% and mAP50:95 of 32.6%, surpassing PCBA-YOLO (72.4% mAP50, 26.3% mAP50:95) by + 6.1%/+6.3% and YOLO-MBBi (70.2% mAP50, 25.5% mAP50:95) by + 8.3%/+7.1%. While requiring moderately higher computational cost (50.2 GFLOPs) than PCBA-YOLO (20.8 GFLOPs), our model establishes an optimal accuracy-efficiency trade-off for high-stakes PCB inspection scenarios where missed defects are intolerable.

Based on the comprehensive experimental results comparing state-of-the-art object detection models on the PKU-Market-PCB dataset, our proposed model demonstrates significant superiority, particularly in achieving the highest detection accuracy while maintaining competitive computational efficiency. The test results are shown in Table 3.

Our model achieves unprecedented accuracy on both primary evaluation metrics. It attains a mAP50 of 96.1%, surpassing the closest competitor (RTDETR-R18 at 93.4%) by a substantial margin of 2.7% points. Furthermore, it achieves a mAP50:95 of 65.1%, outperforming the previous best result (RTDETR-R18 at 63.2%) by 1.9% points. This significant lead in both precision (mAP50) and overall localization-quality-aware precision (mAP50:95) underscores the model's exceptional ability to precisely identify and localize subtle PCB defects, a critical requirement for automated inspection systems.Our model significantly outperforms all listed YOLO architectures, which represent highly efficient and popular object detection families. It exceeds the accuracy of the very competitive YOLOv8s (mAP50: 92.3%, mAP50:95: 61.3%) by 3.8% and 3.8% in mAP50 and mAP50:95, respectively. It also shows clear advantages over YOLOv5s (91.2%, 59.2%), YOLOv7 (85.2%, 56.1%), and the tiny versions (YOLOv3s-Tiny, YOLOv4s-Tiny). Compared to other models designed with efficiency in mind, such as D-Fine-S (91.8% mAP50, 61.7% mAP50:95, 10.18 M Params, 24.85 GFLOPs), our model delivers significantly higher accuracy (+ 4.3% mAP50, + 3.4% mAP50:95) while utilizing a moderately increased parameter count (15.56 M vs. 10.18 M) and computational cost (50.2 vs. 24.85 GFLOPs). This demonstrates that the additional complexity of our model is highly effective in extracting features crucial for accurate PCB defect detection. Our model also surpasses the transformer-based RTDETR-R18 (93.4% mAP50, 63.2% mAP50:95), which itself outperforms all convolutional models except ours.

Methods	mAP50 (%)	mAP50:95(%)	Parameters	GFLOPs
Faster R-CNN	80.1	49.2	41.3 M	206.0
SSD	79.2	48.8	62.8 M	26.2
YOLOv3s-Tiny	84.2	52.8	12.7 M	19.0
YOLOv4s-Tiny	85.2	54.3	6.06 M	16.5
YOLOv5s	91.2	59.2	7.02 M	15.8
YOLOv7	85.2	56.1	36.48 M	103.2
YOLOv8s	92.3	61.3	11.27 M	28.8
D-Fine-S	91.8	61.7	10.18 M	24.85
RTDETR-R18	93.4	63.2	19.8 M	53.1
Ours	96.1	65.1	15.56 M	50.2

Table 3. The EAE-DETR dataset is compared with various other excellent methods in PKU-Market-PCB.

Comparison of experimental results

To further demonstrate the superiority of the proposed algorithm, we selected four groups of images from the detection outcomes of the RT-DETR-R18 and EAE-DETR models for comparative analysis. The chosen images are highly representative, encompassing a variety of defect issues across different motherboard models and viewing angles. As shown in Fig. 10, images (a), (b), (c), and (d) depict the detection results of the RT-DETR model, while images (a'), (b'), (c'), and (d') illustrate the detection results of the EAE-DETR model. The findings reveal that the RT-DETR model exhibits a significant number of missed detections and fails to accurately identify all defects present on the motherboard, which is unacceptable in industrial production contexts. In contrast, the detection results from the EAE-DETR model encompass nearly all correct targets, demonstrating the capability to accurately identify defects even at the edges of the images, thereby fulfilling the requirements of industrial production and inspection.

We present a selection of images from the test set in which defects were accurately detected, as illustrated in Fig. 11. In this figure, images (a), (b), (c), and (d) represent the detection results of the RT-DETR model, while images (a'), (b'), (c'), and (d') correspond to the detection results of the EAE-DETR model. Although both models are capable of correctly identifying defects on the motherboard, a comparison of the detection results indicates that the detection accuracy of the proposed EAE-DETR model is significantly superior to that of the RT-DETR model. This finding substantiates the effectiveness of our model enhancements and demonstrates its greater generalization and rigor in industrial applications.

Despite advancements in algorithmic performance, it is important to note that achieving 100% defect detection in industrial applications remains unattainable. Figure 12 presents a comparative analysis of the RT-DETR model and the EAE DETR model, specifically focusing on instances of partial errors and missed detections. The results for the RT-DETR model are depicted in panels (a), (b), (c), and (d), while panels (a'), (b'), (c'), and (d') illustrate the outcomes for the EAE-DETR model. In the figure, areas where the model failed to detect defects are highlighted with red boxes. In the detection results shown in (a) and (a'), a fan screw was overlooked; however, the detection accuracy of (a') significantly surpassed that of (a). Similarly, in the comparison between (b) and (b'), our model demonstrated superior accuracy over the original benchmark model in the presence of missed detections. In panels (c) and (c'), both instances involved missing fan screws, albeit in different locations—one positioned above and the other below. Notably, the motherboard screw was undetected in (c) but was accurately identified in (c'), with the detection accuracy for other defects also exceeding that of (c). Lastly, in (d) and (d'), both the motherboard screws and fan wiring were subject to missed detections; nevertheless, the detection accuracy for other defects in (d') was higher than in (d).

Conclusion

This study addresses the challenges of missed and false detections in the defect identification of server motherboards by proposing the EAE-DETR detection model, which is based on an enhanced version of RT-DETR. The incorporation of the CSP-EfficientVIM-CGLU module allows for a reduction in the number of

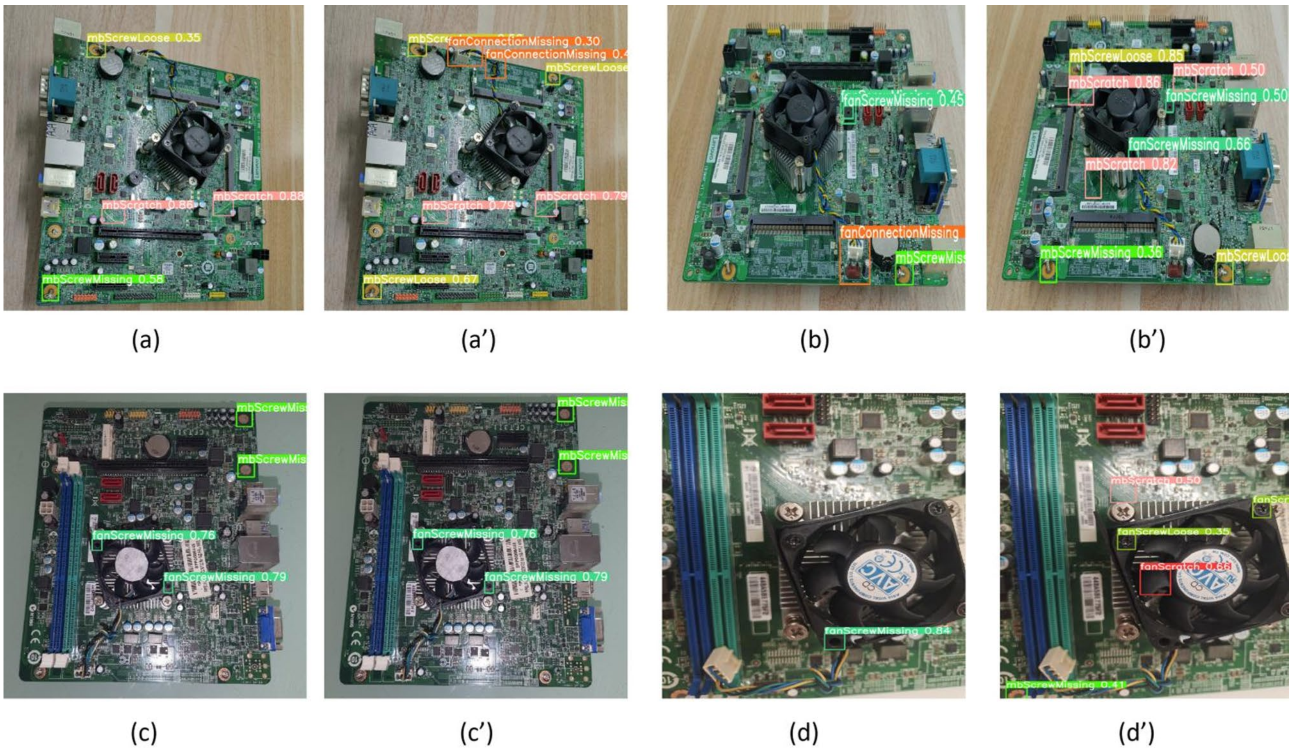

Fig. 10. Comparison of detection results between RT-DETR-R18 and EAE-DETR.

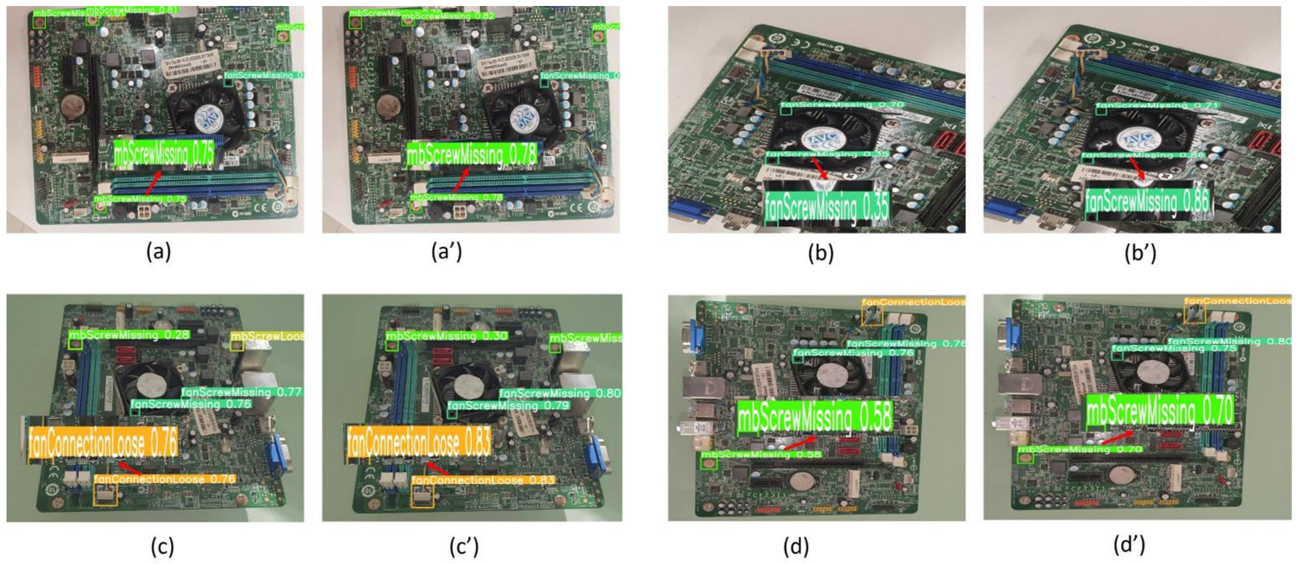

Fig. 11. Comparison of correct detection results between RT-DETR-R18 and EAE-DETR.

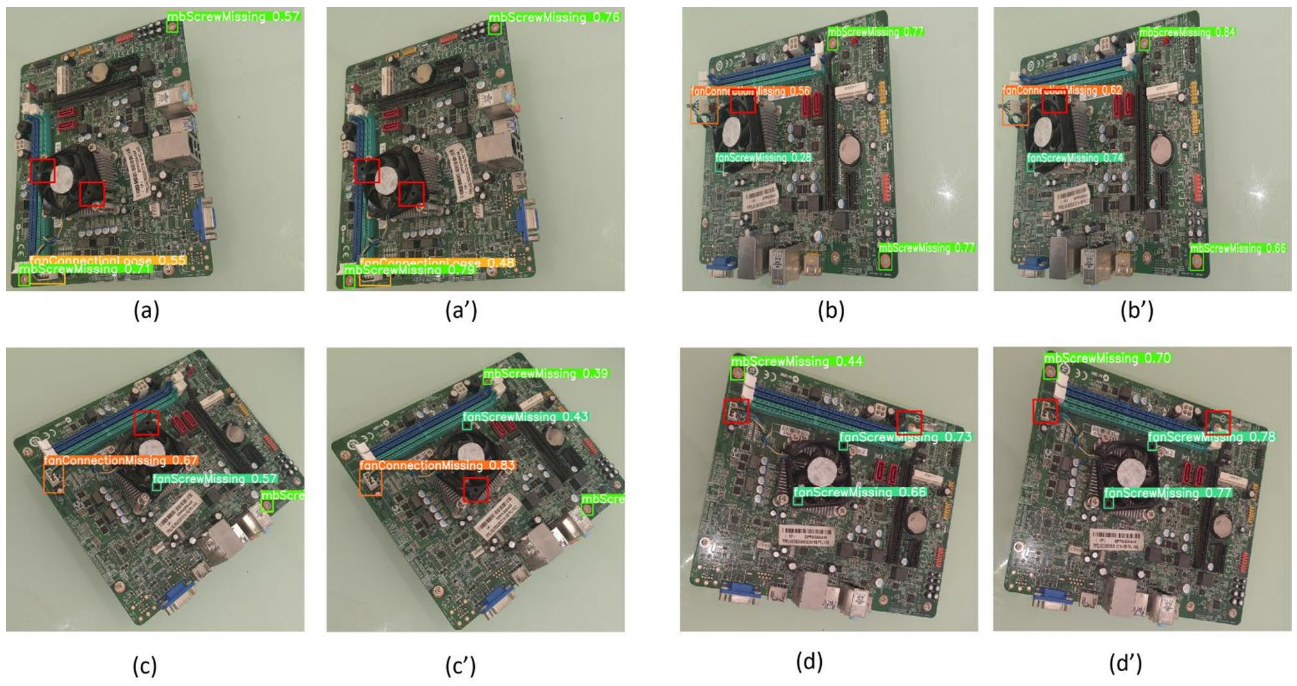

Fig. 12. Comparison of error detection results between RT-DETR-R18 and EAE-DETR.

parameters while simultaneously improving the model's ability to express subtle defects. Additionally, the AIFI-ASSA module employs a dynamic sparse attention mechanism that effectively mitigates background noise, thereby enhancing detection robustness in complex environments. The EUCB-SC module further optimizes multi-scale feature fusion through cross-channel information interaction. Experimental results indicate that EAE-DETR achieves a mean average precision (MAP) of 78.5% at 50% Intersection over Union (IoU) and 32.6% MAP at 50:95 IoU on the PCBA-DET dataset.On the PKU-Market-PCB dataset, the mAP50 reached 96.1%, and the mAP50:95 reached 65.1%,demonstrating a significant improvement over existing mainstream detection models while balancing computational efficiency and accuracy. Future research will concentrate on the lightweight design of the model and its adaptive optimization under extreme lighting conditions to facilitate broader applications in the domain of industrial quality inspection.In the future, it is planned to deploy lightweight models on terminal devices for embedded reasoning. The proposed optimization method will be applied to the actual industrial scenario of server motherboard defect detection to improve detection efficiency and deployment flexibility.

Data availability

The PCBA-DET dataset can be accessed at https://github.com/ismh16/PCBA-Dataset.

Received: 14 May 2025; Accepted: 4 August 2025
Published online: 08 August 2025

References

1. Liu, Z. & Qu, B. Machine vision based online detection of PCB defect. *Microprocess. Microsyst.* **82**, 103807 (2021).
2. Tang, J. et al. PCB-YOLO: an improved detection algorithm of PCB surface defects based on YOLOv5. *Sustainability* **15**(7), 5963 (2023).
3. Zhou, Y. et al. Review of vision-based defect detection research and its perspectives for printed circuit board. *J. Manuf. Syst.* **70**, 557–578 (2023).
4. Jiang, W. et al. PCB defects target detection combining multi-scale and attention mechanism. *Eng. Appl. Artif. Intell.* **123**, 106359 (2023).
5. Chen, M. et al. A comprehensive review of deep learning-based PCB defect detection. *IEEE Access.* **11**, 139017–139038 (2023).
6. Liu, H. I. et al. A denoising FPN with transformer R-CNN for tiny object detection. *IEEE Trans. Geosci. Remote Sens.* (2024).
7. Jawaharlalnehru, A. et al. Target object detection from unmanned aerial vehicle (UAV) images based on improved YOLO algorithm. *Electronics* **11**, 2343 (2022).
8. Khan, A. A. et al. Dual-3DM 3 AD: mixed transformer based semantic segmentation and triplet pre-processing for early multi-class alzheimer's diagnosis. *IEEE Trans. Neural Syst. Rehabil. Eng.* **32**, 696–707 (2024).
9. Khan, A. A. et al. D2PAM: epileptic seizures prediction using adversarial deep dual patch attention mechanism. *CAAI Trans. Intell. Technol.* **8**(3), 755–769 (2023).
10. Zhou, Y. A YOLO-NL object detector for real-time detection. *Expert Syst. Appl.* **238**, 122256 (2024).
11. Yang, Y. & Kang, H. An enhanced detection method of PCB defect based on improved YOLOv7. *Electronics* **12**(9), 2120 (2023).
12. Yuan, M. et al. YOLO-HMC: an improved method for PCB surface defect detection. *IEEE Trans. Instrum. Meas.* **73**, 1–11 (2024).
13. Zhao, W. et al. PCB component detection using computer vision for hardware assurance. *Big Data Cogn. Comput.* **6**(2), 39 (2022).
14. Liao, Z. et al. YOLOv4-MN3 for PCB surface defect detection. *Appl. Sci.* **11**(24), 11701 (2021).
15. Santoso, A. et al. Development of Pcb defect detection system using image processing with Yolo Cnn method. *Int. J. Artif. Intell. Res.* **6**(1),1 (2024).
16. Ling, Z. et al. Deep Siamese semantic segmentation network for PCB welding defect detection. *IEEE Trans. Instrum. Meas.* **71**, 1–11 (2022).
17. Ling, Q. & Nor Ashidi Mat Isa. Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey. *IEEE Access.* **11**, 15921–15944 (2023).
18. An, K. & Zhang, Y. LPViT: A transformer based model for PCB image classification and defect detection. *Ieee Access.* **10**, 42542–42553 (2022).
19. Zhou, G. et al. Lightweight PCB defect detection algorithm based on MSD-YOLO. *Cluster Comput.* **27**(3), 3559–3573 (2024).
20. Li, C. et al. A method of defect detection for focal hard samples PCB based on extended FPN model. *IEEE Trans. Compon. Packag. Manuf. Technol.* **12**(2), 217–227 (2021).
21. Xuan, W. et al. A lightweight modified YOLOX network using coordinate attention mechanism for PCB surface defect detection. *IEEE Sens. J.* **22**(21), 20910–20920 (2022).
22. Chen, B. & Dang, Z. Fast PCB defect detection method based on FasterNet backbone network and CBAM attention mechanism integrated with feature fusion module in improved YOLOv7. *Ieee Access.* **11**, 95092–95103 (2023).
23. Wang, H. et al. Few-shot PCB surface defect detection based on feature enhancement and multi-scale fusion. *IEEE Access.* **10**, 129911–129924 (2022).
24. Wang, F. et al. Laser-induced thermography: an effective detection approach for multiple-type defects of printed circuit boards (PCBs) multilayer complex structure. *Measurement* **206**, 112307 (2023).
25. Du, B. et al. YOLO-MBBi: PCB surface defect detection method based on enhanced YOLOv5. *Electronics* **12**(13), 2821 (2023).
26. Onshaunjit, J. & Srinonchat, J. Algorithmic scheme for concurrent detection and classification of printed circuit board defects. *Comput. Mater. Continua* **71**, 1 (2022).
27. Chen, W. et al. PCB defect detection method based on transformer-YOLO. *IEEE Access.* **10**, 129480–129489 (2022).
28. Khan, A. A. et al. Privacy preserved and decentralized smartphone recommendation system. *IEEE Trans. Consum. Electron.* **70**(1), 4617–4624 (2023).
29. Zhang, H., Jiang, L. & Li, C. CS-ResNet: Cost-sensitive residual convolutional neural network for PCB cosmetic defect detection. *Expert Syst. Appl.* **185**, 115673 (2021).
30. Liu et al. Gaussian-IoU loss: better learning for bounding box regression on PCB component detection. *Expert Syst. Appl.* **190**, 116178 (2022).
31. Kujur, A. et al. Data complexity based evaluation of the model dependence of brain MRI images for classification of brain tumor and alzheimer's disease. *IEEE Access.* **10**, 112117–112133 (2022).
32. Bhattacharya, A. & Sylvain, G. Cloutier. End-to-end deep learning framework for printed circuit board manufacturing defect classification. *Sci. Rep.* **12**(1), 12559 (2022).
33. Shen, M. et al. Defect detection of printed circuit board assembly based on YOLOv5. *Sci. Rep.* **14**, 19287. https://doi.org/10.1038/s41598-024-70176-1 (2024).
34. Du, B. et al. YOLO-MBBi: PCB surface defect detection method based on enhanced YOLOv5. *Electronics* **12**, 2821. https://doi.org/10.3390/electronics12132821 (2023).

Author contributions

Conceptualization, K.Z. and H.Z; methodology, J.C.; validation, G.N. and Z.Z.; writing—original draft preparation, J.C.; writing—review and editing, Z.Z.and J.C. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the Zhangjiakou Basic Research Project: Research on Pre-critical Technology of Track Safety Intelligent Monitoring Based on Deep Learning(grant number: 2311010 A).

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to Z.Z.