electronics

MDPI

Article

MDD-DETR: Lightweight Detection Algorithm for Printed Circuit Board Minor Defects

Jinmin Peng [1,2], Weipeng Fan [1,2,*], Song Lan [3,*] and Dingran Wang [1,2]

[1] School of Mechanical and Automotive Engineering, Fujian University of Technology, Fuzhou 350118, China; pjm@fjut.edu.cn (J.P.); wdr@smail.fjut.edu.cn (D.W.)

[2] Fujian Key Laboratory of Intelligent Processing Technology and Equipment, Fujian University of Technology, Fuzhou 350118, China

[3] Fujian Vocational and Technical College of Water Resources and Electric Power, School of Automation Engineering, Sanming 366000, China

[*] Correspondence: fwp@smail.fjut.edu.cn (W.F.); lansong@126.com (S.L.)

Abstract: PCBs (printed circuit boards) are the core components of modern electronic devices, and inspecting them for defects will have a direct impact on the performance, reliability and cost of the product. However, the performance of current detection algorithms in identifying minor PCB defects (e.g., mouse bite and spur) still requires improvement. This paper presents the MDD-DETR algorithm for detecting minor defects in PCBs. The backbone network, MDDNet, is used to efficiently extract features while significantly reducing the number of parameters. Simultaneously, the HiLo attention mechanism captures both high- and low-frequency features, transmitting a broader range of gradient information to the neck. Additionally, the proposed SOEP neck network effectively fuses scale features, particularly those rich in small targets, while INM-IoU loss function optimization enables more effective distinction between defects and background, further improving detection accuracy. Experimental results on the PCB_DATASET show that MDD-DETR achieves a 99.3% mAP, outperforming RT-DETR by 2.0% and reducing parameters by 32.3%, thus effectively addressing the challenges of detecting minor PCB defects.

Keywords: PCB; defect detection; lightweight; minor defects; RT-DETR

check for updates

Citation: Peng, J.; Fan, W.; Lan, S.; Wang, D. MDD-DETR: Lightweight Detection Algorithm for Printed Circuit Board Minor Defects. *Electronics* **2024**, *13*, 4453. https://doi.org/10.3390/electronics13224453

Academic Editor: José Carlos Castillo

Received: 21 October 2024
Revised: 9 November 2024
Accepted: 12 November 2024
Published: 13 November 2024

1. Introduction

Advancing PCB (printed circuit board) production technology is crucial for the rapid development of electronic products and intelligent manufacturing [1]. As a physical carrier and electrical connection component in precision instruments, PCBs' surface quality and electrical properties are vital for performance in various fields, including the electronics, automotive, medical device, and manufacturing industries. The demand for high-quality PCBs is increasing with the shift towards intelligent, low-power, and high-precision equipment. The trend towards miniaturization, high density, and precision in PCB manufacturing also raises surface defects, complicating production and inspection [2]. Therefore, detecting PCB surface defects and improving production quality is essential for advancing electronic device development. Currently, the main challenges in PCB defect detection are the following:

(1) PCB defect detection is hard in burr and mouse bite defects as they are small targets, and local features are challenging to extract. PCB minor defects (mouse bite, spur) usually appear in a specific local area in the image (e.g., the link between the main road and the arterial road), and there is a slight difference with the surrounding background. (2) The extraction of crucial feature information of PCB minor defects could be better because PCB micro-defect images occupy only a small proportion of space (e.g., “mouse bite” and “spur” defects on a 640 × 640 pixel PCB image cover just 3 × 3 pixels) and have low contrast with the background, so the deep network struggles to effectively eliminate redundant

background information during feature extraction. This leads to reduced sensitivity of the network to micro-defect information. (3) Some background information is easy to mistake for important features, thus introducing unnecessary interference, leading to the severe loss of defective target semantic information in feature fusion.

To address the aforementioned challenges, researchers have applied Automatic Optical Inspection (AOI) techniques in PCB surface defect detection. For instance, Yang [3] proposed an AOI-based PCB defect detection system that successfully identifies the majority of defects. While AOI technology demonstrates effectiveness in defect detection, it falls short of meeting industrial demands due to its high cost and the necessity for adaptability in diverse manufacturing environments. In addition to AOI, some scholars have explored traditional image processing methods such as edge detection and threshold segmentation [4], in conjunction with conventional machine learning algorithms [5] like backpropagation (BP) neural networks and Support Vector Machines (SVMs), for PCB defect detection. However, the inherent difficulty in achieving perfect template matching in real-world images limits the effectiveness of these machine learning approaches, particularly when it comes to detecting minor defects on PCBs.

In recent years, with the extensive use of deep learning for surface defect detection, deep learning-based algorithms for PCB surface defect detection have evolved from two-stage approaches (e.g., Faster-RCNN [6] and DetectoRS [7]) to one-stage methods (e.g., SSD [8], EfficientDet [9] and YOLO [10]) and from anchor-based methods (e.g., RetinaNet [11] and YOLOv4 [12]) to anchor-free approaches (e.g., CenterNet [13], RepPoints [14], and YOLOX [15]). For example, Hu [16] proposed a PCB defect detection algorithm based on Faster-RCNN, aiming to address the high computational costs and susceptibility to noise associated with traditional machine learning algorithms. However, the detection accuracy achieved by this approach was only 94.3%, which proved insufficient for effectively identifying minor defects. Yuan et al. [17] proposed a novel YOLO-HMC algorithm based on YOLOv5, which incorporates a content-aware feature reassembly module to aggregate contextual semantic information from PCB images. This approach aims to identify small defects such as burrs and mouse bites. However, its limitations in deep feature extraction hinder its effectiveness, resulting in the ability to detect only minor defects that exhibit minimal differences from the background. Although CNNs have good local feature extraction capability, they may be limited by local features when dealing with complex backgrounds such as PCB images, resulting in low accuracy of tiny defect detection.

Vision Transformer (ViT) [18] has been proposed as a replacement for convolutional neural networks (CNNs) and is widely applied across various defect detection tasks. Detection Transformer (DETR) utilizes the Transformer architecture for object detection. Unlike CNN-based single-stage detectors [19], DETR removes the need for traditional techniques like a priori frames and Non-Maximum Suppression (NMS) [20]. DETR provides object detection and recognition in a fully end-to-end manner, making it a novel approach in the field of object detection. For example, Huang [21] proposed a new PCB detection algorithm, dab-deformable-DETR, which is effective in the detection of tiny PCB defects, but DETR is limited in industrial applications due to its low processing efficiency, long image processing time, and high computational cost.

In 2023, Lv introduced the Real-Time Detector Transformer (RT-DETR) [22], a real-time, end-to-end object detection model. RT-DETR features an efficient hybrid encoder that separates intra-scale interactions from cross-scale fusion, effectively managing multi-scale features. The model employs an IoU-aware query selection mechanism to enhance object query initialization and allows flexible inference speed adjustments by changing the number of decoder layers without retraining. Research on defect detection using the RT-DETR model is limited compared to the YOLO series: Yu et al. [23] utilized an enhanced RT-DETR for railway rutting defect detection; Liu et al. [24] proposed the Bearing–DETR lightweight deep learning model designed specifically for bearing defect detection. These improvements in the RT-DETR model for defect detection have achieved positive results, effectively demonstrating its advantages in defect detection. This paper proposes Minor Defect Detection—DETR (MDD-DETR). The contributions of this article are as follows:

(1) Aimed at the challenge of PCB defect detection, particularly for burr and mouse bite defects, which are characterized by small targets and difficult-to-extract local features, we redesign the backbone network and propose the Minor Defect Detection Network (MDDNet) to capture PCB defect features effectively.

(2) To address the issue of incomplete extraction of critical feature information for minor PCB defects, we introduce a novel module, AIFI-HiLo, designed to extract features at different underlying frequencies. This module helps eliminate confusing background information and enhances the extraction of essential feature information.

(3) The neck network is re-engineered to mitigate semantic information loss during feature fusion for defect targets. We implement the Small Object Enhancement Pyramid (SOEP) network to facilitate multi-level, multi-scale feature fusion, enabling the model to better comprehend and differentiate between various defect types.

(4) Finally, the INM-IoU loss function is applied to optimize the network, significantly improving the accuracy of PCB defect detection.

The remainder of this paper is organized as follows: Section 2 describes related work; Section 3 presents the method design; Section 4 provides the experimental analysis; and Section 5 concludes the paper and discusses future work.

2. Related Work

2.1. RT-DETR

In the detection of micro-defects on industrial components, particularly on printed circuit boards (PCBs), various deep learning algorithms have been employed. Notably, the two-stage object detection algorithm Faster R-CNN integrates a Region Proposal Network (RPN) with Fast -RCNN [25]. However, when faced with small target defects, the RPN may struggle to generate effective candidate bounding boxes that adequately cover these defects, leading to a decline in the model's ability to recognize small anomalies. On the other hand, the YOLO (You Only Look Once) series, which treats object detection as a regression problem, directly predicts bounding box coordinates and class probabilities through a neural network. While this approach shows some effectiveness in identifying small defects, it often requires additional post-processing steps to enhance precision and recall. Such steps can increase computational costs. DETR uses the global self-attention mechanism to solve the post-processing (NMS) operation problem and achieve true end-to-end training and reasoning. However, in practical detection situations, especially within the framework of assembly line applications, algorithms must ensure high accuracy to maintain detection quality and be capable of swiftly processing images to meet the demands of real-time operational efficiency. RT-DETR balances accuracy and real-time responsiveness, combining the feature extraction benefits of a visual Transformer with rapid computation speeds. Additionally, its end-to-end architecture enables swift deployment in industrial settings, making it an optimal choice for PCB defect detection.

The RT-DETR architecture, illustrated in Figure 1, is composed of three key components: the backbone network, the hybrid encoder, and the decoder. During training, images are fed into the backbone network to extract features. It utilizes the multi-scale feature outputs from its final three stages as inputs to the hybrid encoder. The hybrid encoder extracts information within and across different scale features, followed by bidirectional fusion to produce fused, concatenated features. Query features are selected from these concatenated features to serve as the target query vectors for the decoder. The decoder then produces the final predicted objects and their corresponding bounding boxes, which are used for counting purposes. When dealing with PCB Minor Defect Detection, RT-DETR's detection performance is insufficient. Therefore, this paper presents a novel PCB defect detection algorithm, MDD-DETR, for more efficient defect detection.

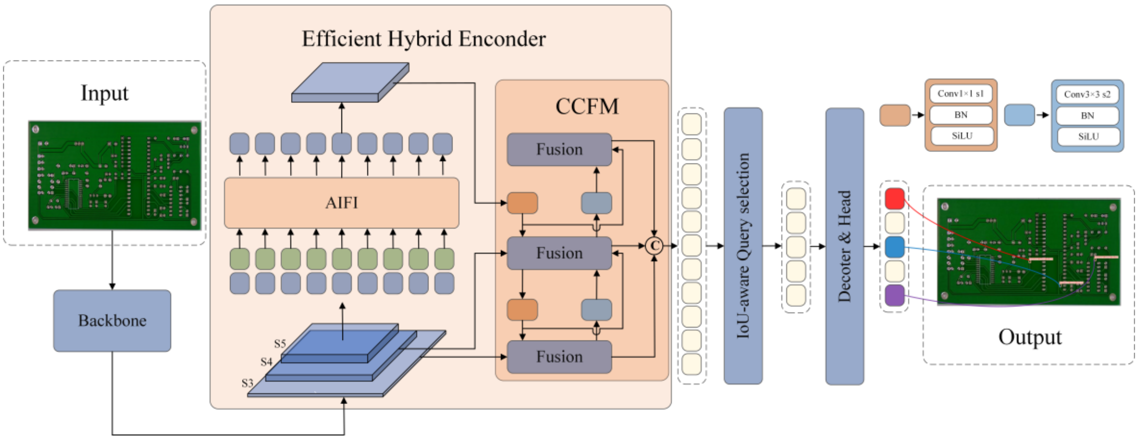


Figure 1. RT-DETR network structure.

2.2. PCB Defect Detection

Conventional PCB surface defect detection techniques, such as manual visual inspection, electrical testing, and infrared scanning, often suffer from omissions, false detections, and low efficiency. As computer vision technology progresses, various methods for detecting defects in printed circuit boards based on visual analysis have been investigated, significantly improving the precision and effectiveness of defect identification while reducing labor expenses.

For example, Niu [26] and some researchers have introduced a PCB defect detection algorithm based on an enhanced Faster R-CNN, utilizing deep convolution in place of standard convolution to decrease the number of parameters and the improved k-means++ algorithm to obtain anchor points suitable for micro-defect target detection. Zeng [27] et al. proposed a novel enhanced multi-scale feature pyramid network to merge feature maps, employing dilated convolution operators with varying expansion rates to fully leverage contextual information, thereby improving the detection performance for minor PCB target defects. Tang [28] et al. enhanced YOLOv5 by adding a small target detection layer to focus on small targets, improving detection accuracy. They also used deep separable convolution to reduce model size and the EIoU loss function to optimize regression, enhancing overall performance. Liu [29] and colleagues opted to adopt YOLOv4 as their foundational framework, introducing an innovative loss function grounded in GIoU regression. Despite improving the recall rate of target detection performance, the original YOLOv4 algorithm still needs a redundant structure and high computational complexity. Lan [30] employs the Cross-Scale Fusion Module (CFM) to connect the YOLOv8 backbone network with the neck network, enhancing the feature fusion and interaction capabilities of the neck network.

In recent research, Qin [31] proposed a novel PCB defect detection algorithm called SDD-Net. This algorithm enhances multi-level feature fusion and representation capabilities by integrating a residual feature pyramid network with a hybrid attention mechanism. While it has demonstrated strong performance across multiple datasets, its detection capabilities in complex backgrounds still need improvement. Additionally, Zhang [32] incorporated a lightweight backbone network (LFEN) into a new PCB defect detector, effectively balancing accuracy, computational cost, and detection speed. However, the detection accuracy is only 95.90%, which poses challenges when addressing minor defects.

However, these studies focus on regular large-size defects, neglecting the challenges of detecting defects in small or complex backgrounds. When faced with minor defects like mouse bites and spur in PCBs, traditional detection models encounter difficulties.

3. Methodology

3.1. MDD-DETR Network Model

As shown in Figure 2, the model utilizes MDDNet as the backbone for image feature extraction. For the processing of deep network features, we introduced the AIFI-HiLo module, which integrates the HiLo noticing mechanism and is able to extract features at different low-level frequencies to optimize the extraction of critical micro-defect features. We design a neck network called the SOEP. In the first bottom-up final path, the SPDConv module is responsible for extracting small target features from AIFI-HiLo and fusing them at the shallow level of the neck. Meanwhile, CREC collects gradient information from each layer through more efficient connections in the second top-down path. To enhance the focus on PCB micro-defect regions, we utilize an INM-IoU optimized loss function to help the model learn micro-defect features more efficiently. Ultimately, by using a decoder with additional prediction headers, we iteratively refine the object query to generate bounding boxes and confidence scores.

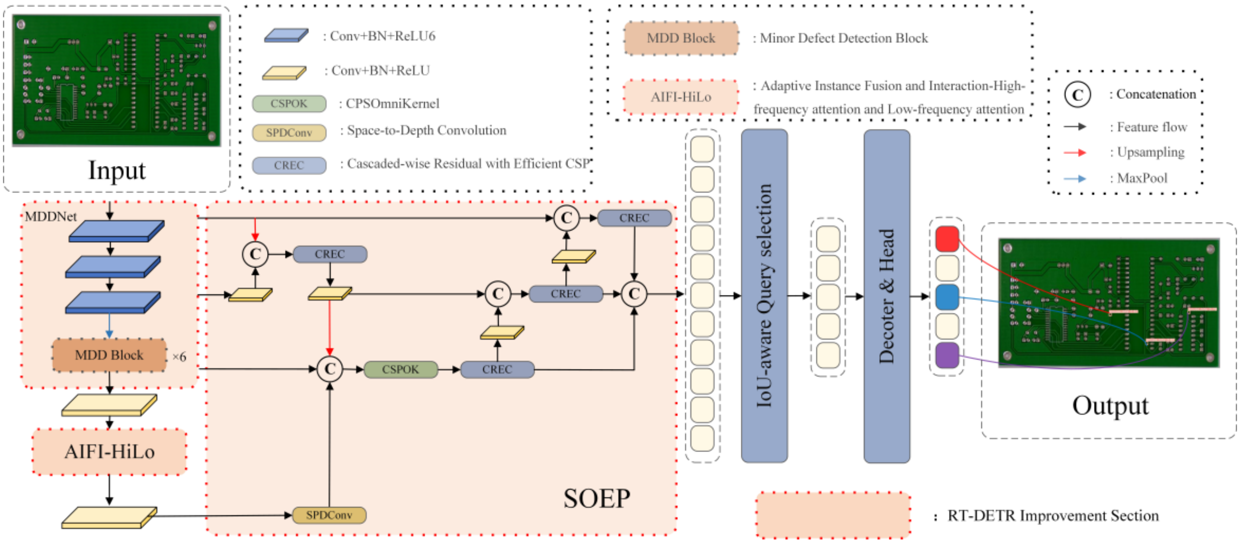

Input
: Conv+BN+ReLU6
: Conv+BN+ReLU
CSPOK : CPSOmniKernel
SPDConv : Space-to-Depth Convolution
CREC : Cascaded-wise Residual with Efficient CSP
MDD Block : Minor Defect Detection Block
AIFI-HiLo : Adaptive Instance Fusion and Interaction-High-frequency attention and Low-frequency attention
C : Concatenation
: Feature flow
: Upsampling
: MaxPool
MDDNet
MDD Block ×6
AIFI-HiLo
CREC
CSPOK
SPDConv
SOEP
IoU-aware Query selection
Decoter & Head
Output
: RT-DETR Improvement Section

Figure 2. MDD-DETR network structure.

3.2. MDDNet

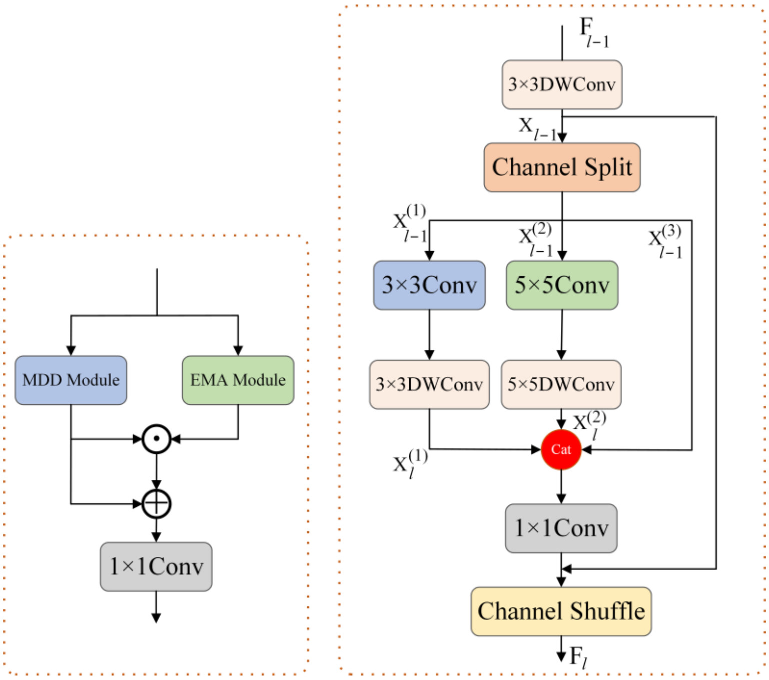
The contrast between PCB minor defects and the background is minimal, and the extraction effect of minor defect features is poor. To capture the features of these minor defects, MDDNet first extracts the features through three CBHS modules, which use a convolutional layer with a scale of 3 × 3. Under the premise of guaranteeing the complete extraction of the features of the minor defects, the number of module references is effectively controlled, and no additional parameters are added; in order to balance the computational accuracy, execution efficiency, and portability, with h-swish instead of swish as the activation function, h-swish is denoted as

$$h - swish[x] = x \cdot [RELU\ 6(x+3)/6] \tag{1}$$

where x is the input vector, and $RELU\ 6$ is the activation function.

Unlike the swish function, the h-swish function differentiates itself using the RELU6 activation function instead of the sigmoid function. RELU6 is used as the approximation function to approximate the swish; this function can be integrated into nearly all software and hardware frameworks, effectively eliminating the accuracy loss associated with using a linear function in place of the sigmoid function. Subsequently, features are extracted through a maximum pooling layer, simplifying the model, improving its robustness, and significantly reducing the computational burden and the number of parameters. Finally, a series of MDD blocks are used for multi-channel output.

As depicted in Figure 3a, within the MDD block, the MDD module extracts local features. It incorporates the Efficient Multi-scale Attention (EMA) [33] mechanism to enhance the focusing ability of target features and aid the model in better comprehending the relationship between minor defects and their surrounding areas. These two components collaboratively enhance the performance of MDDNet in PCB defect detection.



(a)MDD Block (b)MDD Module

Figure 3. MDD block.

MDD Module

As illustrated in Figure 3b, in the MDD module, firstly, the maximum pooling layer output is used as input for $F_{l-1} \in R^{C_{l-1}\times H_{l-1}\times W_{l-1}}$ using a 3 × 3 deep convolution to capture the contextual information. After the initial processing, it is obtained that X_{l-1} is divided into thirds by channel slicing and is provided to the three paths:

$$X_{l-1} = Split(DWConv_{3\times3}(F_{l-1})) \in R^{C_l\times H_l\times W_l}, \tag{2}$$

$$X^{(1)}_{l-1} = X_{l-1}[\frac{1}{4}C_l,\cdots], X^{(2)}_{l-1} = X_{l-1}[\frac{1}{4}C_l,\cdots], X^{(3)}_{l-1} = X_{l-1}[\frac{1}{2}C_l,\cdots], \tag{3}$$

The first two branches are split into convolutional layers with different receptive fields to cover the possible dimensions of defective features and to extract more comprehensive defect-related information. Subsequently, depthwise convolution is used to conduct separate operations for each input channel, resulting in significantly fewer parameters than traditional convolution operations. The three processed feature maps are concatenated to achieve multidimensional differential feature fusion. The fused features undergo pointwise convolution and residual fusion with the initial input features. Finally, a channel blending operation is applied to the output feature map, enabling the model to capture complex features more effectively and enhancing its generalization capability. This process can be represented by Equations (4) and (5).

$$X^{(1)}_{l} = DWConv_{3\times3}(Conv_{3\times3}(X^{(1)}_{l-1})), X^{(2)}_{l} = DWConv_{5\times5}(Conv_{5\times5}(X^{(2)}_{l-1})), \tag{4}$$

$$F_l = Shuffle(Conv_{1\times1}(Concat(X^{(1)}_{l}, X^{(2)}_{l}, X^{(3)}_{l-1})), F_{l-1}) \in R^{C_l\times H_l\times W_l} \tag{5}$$

where $Split$ denotes the channel slicing operation, $Shuffle$ denotes the channel mixing operation, and $Concat$ denotes the connecting operation.

3.3. AIFI-HiLo

The multihead self-attention mechanism in AIFI employs uniform global attention across all image blocks, which fails to account for the distinct features associated with various underlying frequencies. This limitation hampers the model's capacity to effectively capture crucial details of small PCB defects. To address this issue, AIFI-HiLo is proposed. HiLo (high-frequency attention and low-frequency attention) [34] captures local details through high frequencies in the image, and low frequencies focus on the global structure. To address this issue, the attention layer can decouple high- and low-frequency modes by dividing the heads into two groups, where one group focuses on capturing high-frequency information through self-attention within each local window.

The other model's low-frequency information is captured by applying attention across local windows. In the HiLo mechanism, the attention layer differentiates high and low frequencies by assigning distinct roles to different heads. As illustrated in Figure 4, in the upper pathway, specific heads are allocated for high-frequency attention (Hi-Fi) through local window self-attention (for instance, 2 × 2 windows), enabling efficient extraction of detailed high-frequency features compared to the standard MSA. Low-frequency attention (Lo-Fi) is established in the lower pathway by using average pooling within each window to extract low-frequency signals. The focus of the remaining attention head is on Lo-Fi, which captures the global relationships between each query position in the input feature map and the average-pooled low-frequency key for each window. This approach greatly reduces Lo-Fi complexity by using shorter key and value lengths. Finally, the refined high- and low-frequency features are concatenated and passed to subsequent layers for further processing.

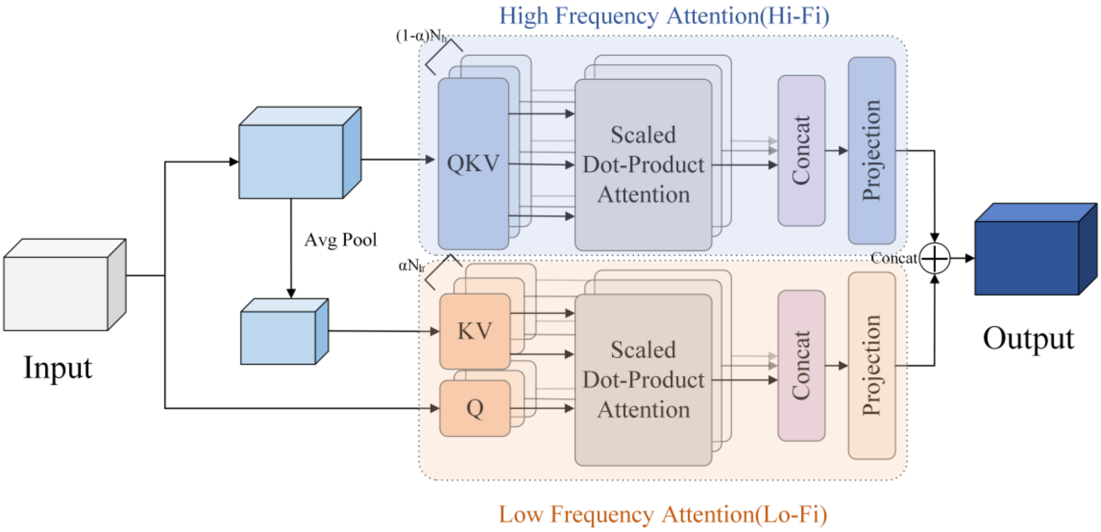

High Frequency Attention(Hi-Fi)
(1-α)N$_h$
QKV
Scaled Dot-Product Attention
Concat
Projection
Avg Pool
αN$_h$
KV
Q
Scaled Dot-Product Attention
Concat
Projection
Concat
Input
Output
Low Frequency Attention(Lo-Fi)

Figure 4. HiLo attention framework.

3.4. Loss Function Optimizsation

To place a stronger focus on the minute defect areas on printed circuit boards, the model can be directed to more effectively learn the characteristics of these defects. This enables it to differentiate more clearly between the defects and the background, ultimately improving the precision of PCB defect detection; so, INM-IoU is proposed. This approach integrates the auxiliary bounding box concept of inner-IoU [35], the NWD (Normalized Wasserstein Distance) [36] probability distribution computational system, and the bounding box alignment accuracy principle of MPDIoU [37] to optimize the GIoU loss function of RT-DETR. The rationale behind the creation of INM-IoU is as follows:

Intersection over Union (IoU), a crucial element in existing mainstream bounding box regression loss functions, is defined as follows. Figure 5 illustrates its calculation, while the corresponding formula is given in Equation (6):

$$IoU = \frac{B_{gt} \cap B_{prd}}{B_{gt} \cup B_{prd}} \tag{6}$$

where B_{prd} and B_{gt} represent the predicted box and the GT box, respectively.

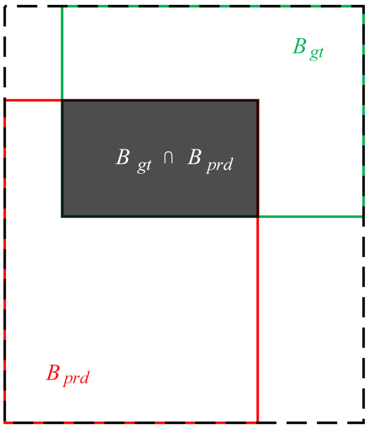


Figure 5. The IoU calculation factor.

The GIoU in RT-DETR is improved based on the IoU to solve the problem of the gradient disappearing when the un-overlapped area is 0. The GIoU loss function is mainly based on the overlapped area between the predicted and GT boxes to optimize the model. However, regarding PCB defect detection, relying only on the overlapped area may only partially reflect the geometric alignment between the boxes. In this situation, MPDIoU compensates for this by introducing an additional distance metric, i.e., the minimum perpendicular distance between the vertices of the predicted box and the actual box, so that the loss function can pay more attention to the precise alignment of the bounding box during the optimization process. The schematic diagram of the MPDIoU loss function is shown in Figure 6. The expression of MPDIoU is

$$L_{MPDIoU} = 1 - IoU + \frac{d_1^2}{h^2 + w^2} + \frac{d_2^2}{h^2 + w^2} \tag{7}$$

$$\begin{aligned} d_1^2 &= \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2, \\ d_2^2 &= \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2 \end{aligned} \tag{8}$$

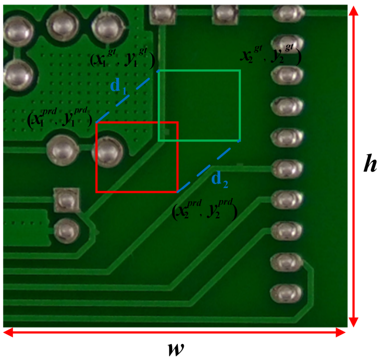


Figure 6. The MPDIoU calculation factor.

MPDIoU takes into account not only the overlap degree of the bounding boxes but also the relative positions and shapes of the boxes. This enhances the accuracy of bounding box alignment while preserving the benefits of the IoU loss function. For small targets, the MPDIoU loss function can reduce the impact of defects being indistinguishable from the background, as the bounding boxes for small targets are better optimized through multi-scale regression.

The inner-IoU loss function builds on MPDIoU, enhancing its performance by calculating IoU loss through an auxiliary bounding box. The key concept of this approach is to use auxiliary bounding boxes of different scales for samples with high and low IoU values, respectively, to calculate the loss. This strategy effectively accelerates the bounding box regression process while improving the model's generalization capability and localization accuracy.

$$inter = (min(b^r_{gt}, b^r_{prd}) - max(b^l_{gt}, b^l_{prd})) * (min(b^b_{gt}, b^b_{prd}) - max(b^t_{gt}, b^t_{prd})) \tag{9}$$

$$union = (w^{gt} * h^{gt}) * (ratio)^2 + (w^{prd} * h^{prd}) * (ratio)^2 - inter \tag{10}$$

$$IoU_{inner} = \frac{inter}{union} \tag{11}$$

$$L_{Inner-IoU} = 1 - IoU_{inner} \tag{12}$$

$$L_{Inner-MPDIoU} = 1 + \frac{d_1^2}{h^2 + w^2} + \frac{d_2^2}{h^2 + w^2} - IoU_{inner} \tag{13}$$

where b^r_{gt}, b^l_{gt}, b^b_{gt}, b^t_{gt} denote the right, left, lower and upper boundaries of the GT box, respectively, b^r_{prd}, b^l_{prd}, b^b_{prd}, b^t_{prd} denote the right, left, lower and upper boundaries of the predicted box (or the auxiliary bounding box), respectively, (w^{prd}, h^{prd}) and (w^{gt}, h^{gt}) denote the width and height of the predicted box and the GT box, respectively, and *ratio* is the scaling factor ratio, which is used to resize the auxiliary bounding box.

If the predicted box and the ground truth (GT) box do not overlap, the inner-MPDIoU loss function significantly hinders the model's ability to update parameters during backpropagation. Additionally, it becomes challenging to compute the similarity between predicted and GT boxes when they fully contain each other. At this time, the NWD loss function calculation similarity is introduced to improve the detection efficiency.

NWD represents the bounding box as a two-dimensional Gaussian distribution and measures the similarity between the predicted and GT boxes based on their respective Gaussian distributions. The calculation formulas are given in Equation (14).

$$W_2^2(N_{B_{prd}}, N_{B_{gt}}) = \left\| \left(\left[cx^{prd}, cy^{prd}, \frac{w^{prd}}{2}, \frac{h^{prd}}{2} \right]^T, \left[cx^{gt}, cy^{gt}, \frac{w^{gt}}{2}, \frac{h^{gt}}{2} \right]^T \right) \right\|_2^2 \tag{14}$$

where (cx^{prd}, cy^{prd}) and $(cx^{\mathrm{gt}}, cy^{\mathrm{gt}})$ are the center coordinates of predicted box and the GT box, respectively. $W_2^2(N_{B_{prd}}, N_{B_{gt}})$ is a distance measure and cannot be directly used as a similarity measure. Therefore, normalize its exponential form to obtain a new metric NWD; the calculation formula is as Equation (15).

$$NWD(N_{B_{prd}}, N_{B_{gt}}) = \exp\left(-\frac{\sqrt{w_2^2(N_{B_{prd}}, N_{B_{gt}})}}{C} \right) \tag{15}$$

where C is a constant.

The new loss function, INM-IoU, is created by proportionally combining inner-MPDIoU with NWD, as presented in Equation (16).

$$L_{INM-IoU} = \alpha L_{Inner-MPDIoU} + (1 - \alpha) NWD(N_{B_{prd}}, N_{B_{gt}}) \tag{16}$$

where α is the scale factor, $0 \leq \alpha \leq 1$.

3.5. SOEP

Traditional detection layers like P3, P4, and P5 face challenges in effectively addressing these small-scale issues in PCB micro-defect detection. A common approach to enhancing the detection capability of micro-defects is to introduce a P2 detection layer. However, this introduces a series of issues, such as increased computational load and prolonged post-processing times. Hence, there is an urgent need to develop novel feature pyramid structures to address the challenges of micro-defect detection effectively. We analyzed and designed a novel neck network called SOEP to address this.

In contrast to traditional methods of adding P2 detection layers, we utilize P2 feature layers processed through SPDConv [38], which contain abundant information about small targets, and integrate them with the P3 layer. Subsequently, drawing inspiration from the CSP structure and OmniKernel-based [39] approaches, we propose CSP-OmniKernel for feature integration. Furthermore, we developed a CSP module, known as the CREC module, which incorporates cascaded residuals into the SOEP network. This design aims to strengthen the fusion capabilities of micro-defect features, ultimately enhancing the performance of micro-defect detection.

3.5.1. SPDConv

The Space to Depth Convolution module (SPDConv) comprises a Space to Depth (SPD) layer and a non-stepped convolution (Conv) layer. Firstly, the SPD layer cuts the intermediate feature input X of size $S \times S \times C_1$ into a series of sub-feature maps $f_{x,y}$, all of size $(S/scale, S/scale, C_1)$, which are down samples of the original feature input X. Next, all sub-feature maps are concatenated along the channel dimension to create feature map $X^{'}(S/scale, S/scale, scale2C_1)$. After the feature map X' is fed into a non-stepped convolutional layer (N-S Conv) containing C_2 filters, the output feature map $X^{''}(S/scale, S/scale, C_2)$ is obtained. The non-stepped convolutional layer preserves as much discriminative information as possible.

$$\begin{cases} f_{0,0} = X[0 : S : scale, 0 : S : scale], \cdots, \\ f_{scale-1,0} = X[scale-1 : S : scale, 0 : S : scale] \\ f_{0,1} = X[0 : S : scale, 1 : S : scale], \cdots, \\ f_{scale-1,1} = X[scale-1 : S : scale, 1 : S : scale] \\ \quad \vdots \\ f_{0,scale-1} = X[0 : S : scale, scale-1 : S : scale], \cdots, \\ f_{scale-1,scale-1} = X[scale-1 : S : scale, scale-1 : S : scale] \end{cases} \tag{17}$$

In the formula, $f_{x,y}$ consists of all $i + x$ and $i + y$ in $x(i, j)$ that can be divided proportionally by *scale*.

To avoid feature loss caused by small defect networks, such as mouse bites and spikes, due to stride convolution, an SPDConv layer with a scale of 2 is selected to collect small target information from the P2 layer and send it to the P3 layer for fusion.

3.5.2. CSP-OmniKernel

As shown in Figure 7a, the CSP OmniKernel (CSPOK) module follows the idea of the CSP structure, which can enhance the information flow of the network, promote the transmission and learning of features, enhance computational efficiency, reduce module complexity and inference time, and make the model more lightweight and practical. We generate four slices of the input channels, only one of which is processed by the OmniKernel module. In Figure 7b, the OmniKernel (OKM) is shown with the given input feature $X \in R^{C \times H \times W}$. Following 1×1 convolution processing, the features are directed into three branches—local, large, and global—allowing for effective learning of the feature representation of PCB defects across scales from global to local. The outputs from these branches are subsequently merged using addition and refined with an additional 1 × 1 convolution. Parallel deep convolutions of 1 × K and K × 1 are used in a large branch to obtain bar-shaped contextual information. To mitigate the substantial computational burden associated with large kernel size convolutions, we set the kernel size K to 31. In the inference phase of the global branch, the input image is much larger than the training image. Therefore, a 31 × 31 kernel cannot cover the global domain. To alleviate this issue, dual-domain processing was adopted, which added global modeling capability to the global branch. Specifically, the global branch comprises a dual domain channel attention module (DCAM) and a frequency-based spatial attention module (FSAM), as shown in Figure 7c,d.

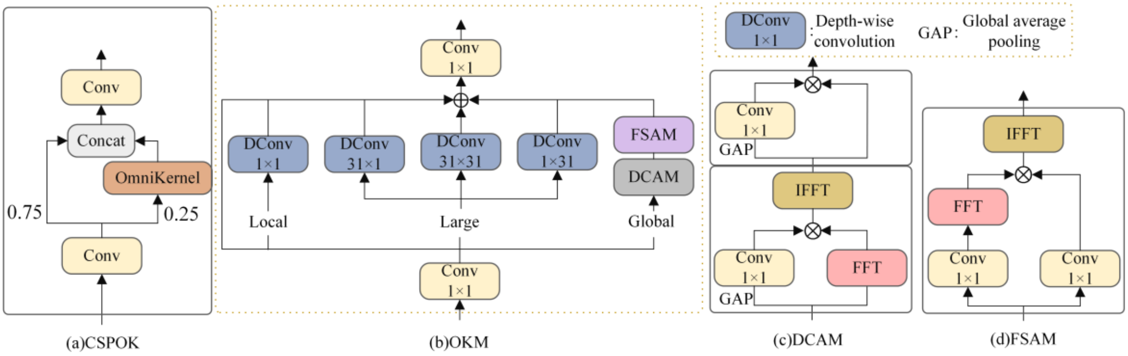


Figure 7. The architecture of CSPOK. FFT and IFFT represent the fast Fourier transform and its inverse operation, respectively.

3.5.3. CREC Module

The CREC module is illustrated in Figure 8.

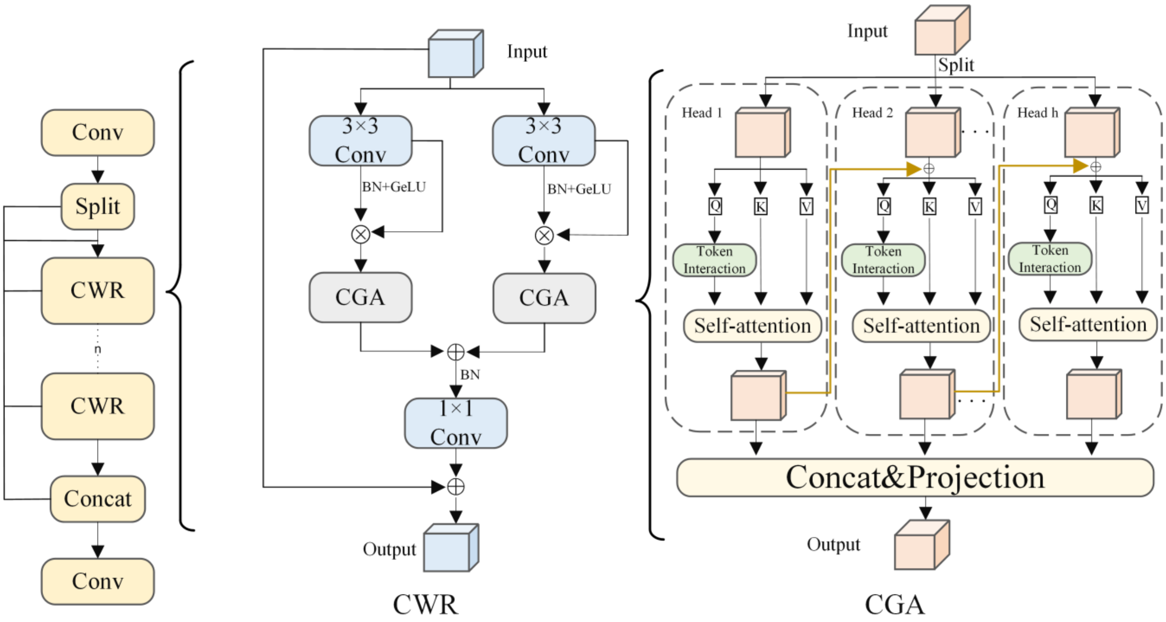


Figure 8. CREC module.

It comprises two convolutional blocks, Batch Normalization (BN) and SiLU activation functions, along with 'n' CWR residual modules. However, after slicing operations and residual connections on input tensors $X \in R^{n \times c \times h \times w}$, the output undergoes convolution through convolutional layers. This process is depicted by Equation (18).

$$\begin{aligned}&Y_1 = CWR_n(X_1),\\&Y_2 = CWR_n(X_2) + \cdots + CWR_2(X_2) + CWR_1(X_2) + X_2,\\&Y = [Concat(Y_1, Y_2)] + X_2\end{aligned} \tag{18}$$

$CWR_{\mathrm{n}}(\cdot)$ represents the nth CWR module, $Concat(\cdot)$ represents features that connect two branches, and $[\cdot]$ represents 1 × 1 convolution operation, +: residual connection.

The proposed residual structure design method ensures a light weight while acquiring richer gradient flow information, thereby capturing more local details and enhancing feature extraction for minor defects in PCBs.

The core element of this module is CWR. It employs a two-step strategy to gather multi-scale contextual information effectively and subsequently merges the feature maps acquired from different receptive fields.

Step 1: Generate corresponding residual features from the input features, termed regional residualization. The image can be segmented into distinct regions by utilizing regional residuals, allowing separate extraction of features from each region, thus facilitating more precise identification of minor defects.

Step 2: Employ cascaded attention to cascade and iterate multi-level attention, directing focus to feature regions of varying sizes, gradually focusing on local information of minor PCB defects, and capturing more spatial relationships. Each channel applies only a single expected receptive field to avoid redundant receptive fields.

Subsequently, the BN layer aggregates multi-scale contextual information obtained from these two steps through pointwise convolution.

CGA (Cascaded Group Attention) [40] level group attention is expressed as

$$\begin{aligned}&\widetilde{X}_{ij} = Attn(X_{ij}W_{ij}^{Q}, X_{ij}W_{ij}^{K}, X_{ij}W_{ij}^{V}),\\&\widetilde{X}_{i+1} = Concat[\widetilde{X}_{ij}]_{j=1:h}W_{i}^{P}\end{aligned} \tag{19}$$

The jth head calculates self attention on X_{ij}, which is the jth segmentation of input feature X_i, i.e., $X_i = [X_{i1}, X_{i2}, \cdots, X_{ih}]$, and $1 \leq j \leq h$. h is the total number of heads. W_{ij}^{Q}, W_{ij}^{K} and W_{ij}^{V} are projection layers that map the input features into distinct subspaces, while W_{i}^{P} is a linear layer that projects the concatenated output features back to the same dimension as the input.

Integrating the CREC module within the SOEP network enhances the accurate capture of local features, thereby improving the distinction between PCB micro-defects and their background.

4. Experiments

4.1. Experimental Description

4.1.1. Experimental Environment and Parameter Settings

Tables 1 and 2 show the experimental environment and configuration parameters, respectively. No pre-training weights are set to ensure the experiment's fairness.

Table 1. Experimental environment.

Configure	Setting
CPU	i7-10700F 2.90GHz
GPU	NVIDIA GeForce RTX 4090
Operating systems	Windows 11
Deployment environment	Python 3.10.11
Deep learning framework	PyTorch 2.0.0
Accelerated computing framework	CUDA 11.7
Optimizer	SGD

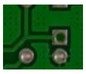
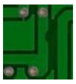
Table 2. Configuration of experimental parameter.

Parameters	Setting
Input image size	640 × 640
Epoch	300
Parameter learning rate	0.001 (First 200epoch), 0.0001 (Post 100epoch)
Batch size	8

4.1.2. PCB Defect Image Dataset

The defect dataset used in the study is provided by the Intelligent Robotics Open Laboratory at Peking University (http://robotics.pkusz.edu.cn/resources/dataset/, accessed on 16 January 2023.) and contains six defect types: missing hole, spurious copper, short circuit, mouse bite, open circuit, and spur. The original dataset consists of 693 images, and data augmentation techniques are used to increase the dataset size to 9009 images. Data augmentation techniques include blurring, brightness adjustment, random cropping, rotation, translation, and mirroring. The augmented dataset is divided into a training set and a test set in an 8:2 ratio. The distribution of each defect type after data augmentation is presented in Table 3.

Table 3. Dataset details.

Defect Type	Example of Defects	Number of Original Images	Number of Images After Expansion
missing_hole		115	1495
spurious_copper		116	1508
short		116	1508
mouse_bite		115	1495
open_circuit		116	1508
spur		115	1495
Total number	-	693	9009

4.1.3. Performance Evaluation and Indicators

To assess the algorithm's effectiveness in detecting PCB defects, evaluation indexes such as precision (P), recall (R), mean average precision (mAP), mouse bite mean average precision (mAP_{mb}), spur mean average precision (mAP_{spur}), and floating-point operation (Flops) were used in this study.

$$\begin{aligned} P &= \frac{T_P}{T_P+F_P} \times 100\%, \\ R &= \frac{T_P}{T_P+F_N} \times 100\% \end{aligned} \tag{20}$$

$$\begin{aligned} AP &= \int_0^1 P(R)dR, \\ mAP &= \frac{1}{n}\sum_{i=1}^{n} AP(i) \end{aligned} \tag{21}$$

where T_P represents the count of correctly predicted positive samples, F_P denotes the number of incorrectly predicted negative samples, and F_N indicates the number of incorrectly predicted positive samples.

4.2. Experimental Analysis

4.2.1. Ablation Study on MDD-DETR

To evaluate the effects of the proposed modules on the PCB defect detection model, this study designs an ablation experiment based on the original RT-DETR. The MDD-DETR contains an SOEP module, INM-IoU loss function, MDDNet module, and AIFI-HiLo module. The ablation experiment results are shown in Table 4. We first added the SOEP module, and with only 0.3M parametric quantities added, the mAP was improved by 1.2%, while both spur and mouse bite were substantially improved. Then, the loss function was optimized using INM-IoU to improve the target recall R by 6.4% with a small increase in mAP. Then, the lightweight MDDNet network was added to greatly reduce the number of 5.3M senators with a 0.1% increase in mAP, and finally the HiLo attention mechanism was used to increase the detection accuracy to 99.3% with controlled parameter costs.

Table 4. Ablation study on MDD-DETR.

SOEP	INM-IoU	MDDNet	AIFI-HiLo	*P*	*R*	*mAP*	*mAP*$_{mb}$	*mAP*$_{spur}$	Params	FLOPs	F1
				98.2	88.4	97.3	94.3	90.2	19.8 M	57.3 G	93.0
√				98.6	89.8	98.5	97.5	94.7	20.1 M	58.2 G	94.0
√	√			98.8	96.2	99.0	98.8	95.3	20.1 M	58.2 G	97.5
√	√	√		99.3	97.0	99.1	99.2	96.6	14.8 M	38.6 G	98.1
√	√	√	√	99.9	97.9	99.3	99.3	98.2	13.4 M	36.4 G	98.9

4.2.2. MDDNet Analysis

In order to verify the effectiveness of the MDDNet design, comparison experiments are conducted with the current popular backbone network. This experiment is carried out on the basis of RT-DETR. Among backbones, MDDNet (without EMA) is the backbone that lacks the EMA mechanism. As shown in Table 5, MDDNet has the best results in detecting mouse bite and spur defects, with an improvement of 4.5% mAP and 6.2% mAP, respectively, compared to the baseline backbone ResNet18. Meanwhile, with the addition of the EMA mechanism, MDDNet has a 0.4% mAP improvement with an increase of only 0.4M parameters.

Table 5. Comparison results of different backbones.

Backbone	*P*	*R*	*mAP*	*mAP*$_{mb}$	*mAP*$_{spur}$	Params	FLOPs
ResNet18 [41]	98.2	88.4	97.3	94.3	90.2	19.8 M	57.0 G
PKINet [42]	97.5	86.8	96.6	93.1	84.5	12.8 M	45.4 G
CSwinTramsformer [43]	96.2	86.3	95.8	91.6	82.6	30.5 M	90.2 G
EfficientFormerv2 [44]	98.2	89.1	97.5	93.9	89.8	11.9 M	29.8 G
EfficientViT [40]	98.2	89.2	97.5	94.5	90.4	**10.8 M**	**27.6 G**
LSKNet [45]	97.9	85.7	96.8	93.5	81.7	12.6 M	37.9 G
RepViT [46]	97.5	89.5	97.3	94.5	88.7	13.4 M	36.7 G
RMT [47]	98.4	91.5	97.6	94.9	91.8	21.4 M	61.5 G
SwinTransformer [48]	97.3	86.5	96.5	93.1	81.6	36.4 M	97.3 G
UniRepLKNet [49]	97.8	86.8	96.7	93.8	86.5	12.8 M	33.7 G
VanillaNet [50]	96.8	85.9	96.1	89.8	80.6	21.8 M	110.5 G
MDDNet (without EMA)	98.6	92.2	97.8	98.2	96.2	14.1 M	37.2 G
MDDNet	**98.9**	**94.3**	**98.2**	**98.8**	**96.4**	14.5 M	37.9 G

4.2.3. INM-IoU Analysis

We performed ablation experiments on INM-IoU, using the RT-DETR model as the baseline. As shown in Table 6, we first compared INM-IoU with other conventional loss functions. Then, we conducted ablation experiments using the three components of INM-IoU: MPDIoU, inner-IoU, and NWD. Each component contributed to an improvement in mAP, and when combined, they produced the best overall performance.

Table 6. Comparison results of different loss functions.

Model	Loss	*P*	*R*	*mAP*
RT-DETR	GIoU	98.2	88.4	97.3
	CIoU	97.6	86.5	97.1
	SIoU [51]	97.5	86.2	96.9
	Shape-IoU [52]	97.1	87.9	96.9
	MPDIoU	98.8	91.5	97.6
	inner-IoU	98.6	91.2	97.4
	NWD	99.0	92.8	97.6
	MPDIoU + inner-IoU	99.0	94.3	97.7
	INM-IoU	99.1	94.8	97.9

4.2.4. SOEP Analysis

To verify the generality of SOEP, we replaced the neck structure with different algorithms and applied them to various models. The results, shown in Table 7, demonstrate that replacing PAFPN with SOEP in YOLOv8n resulted in a 0.4% mAP improvement, with fewer training epochs and a slight increase in the number of parameters. Additionally, we validated the effectiveness of SOEP using the latest YOLO detector, YOLOv10n.

Table 7. Comparison of SOEP's effectiveness in other algorithms.

Model	Neck	*mAP*	Params	FLOPs	Epochs
MDD-DETR	CCFM	98.8	13.1 M	35.5 G	300
	BiFPN	99.1	13.7 M	42.6 G	300
	PAFPN	98.7	13.1 M	35.4 G	300
	GLSA [53]	99.0	15.3 M	42.5 G	300
	SOEP	99.3	13.4 M	36.4 G	300
YOLOv8n	PAFPN	94.3	3.1 M	8.7 G	300
	SOEP	94.7	3.4 M	9.7 G	200
YOLOv10n	PAFPN	95.3	2.3 M	6.7 G	200
	SOEP	95.4	2.7 M	8.2 G	150

4.2.5. AIFI-HiLo Analysis

To highlight the effectiveness of the HiLo attention mechanism in the AIFI module, we combine HiLo with multihead self-attention (MHSA) [54], Cascaded Group Attention (CGA) [40], DAttention [55], Efficient Additive Attention (EAA) [56], the multiscale multihead self-attention (M2SA) in CMTFNet [57], Histoformer Dynamic-range Histogram Self-Attention (DHSA) [58], and the recently popular Transformer attention. This experiment was performed based on the RT-DETR model. The experimental results are shown in Table 8. HiLo attention has the best effect. Compared with other Transformer attentions, HiLo captures local details through the high frequency in the image, while the low frequency focuses on the global structure, which can better extract small defect features.

Table 8. Contrast of the different Transformer attentions.

Model	Attention	*P*	*R*	*mAP*
RT-DETR	MHSA	98.2	88.4	97.3
	CGA	98.5	89.9	97.6
	DAttention	97.4	84.7	96.8
	EAA	96.5	81.9	95.9
	M2SA	98.9	91.8	98.2
	DHSA	98.9	91.5	98.2
	HiLo	99.3	94.2	98.9

4.3. Comparison of PCB Defect Detection Algorithms

As seen in Table 9, the performance of the improved network is compared with several established models, including SSD, Faster R-CNN, YOLOv3 [59], YOLOv8, DETR [60], and YOLOv9 [61]. Additionally, the experiment includes comparisons with current popular detection algorithms such as Huawei's Gold-YOLO and YOLOv10 from 2024. The algorithm proposed in this paper outperforms most network models, with a minimum mean average precision (mAP) improvement of 2.1% over other networks. Furthermore, the number of parameters is kept relatively low at 14.12 M.

Table 9. Comparison of the results with the latest algorithm.

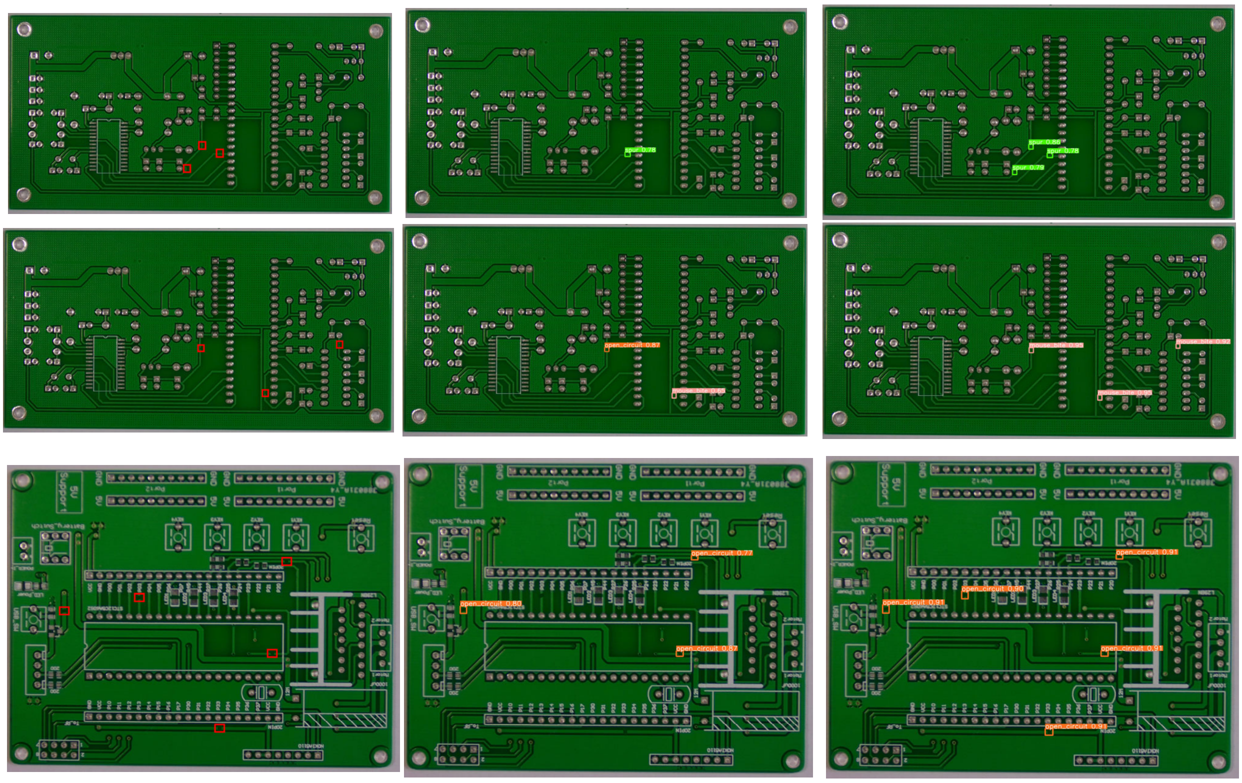
Model	***mAP***	mAP_{mb}	mAP_{spur}	**Params**	**FLOPs**	**FPS (bs = 8)**
SSD	64.5	43.2	35.4	150.2 M	320.5 G	81
Faster-RCNN	72.2	57.8	50.6	40.6 M	89.2 G	168
YOLOv3-tiny	91.5	82.5	75.6	12.1 M	24.9 G	179
YOLOv8n	94.3	91.2	86.4	3.1 M	8.7 G	188
DETR	93.8	88.7	84.6	41.6 M	100.5 G	148
YOLOv9s	94.3	88.6	83.8	7.2 M	26.7 G	186
Gold-YOLOn	95.2	92.1	89.5	5.6 M	12.1 G	190
YOLOv10n	95.3	93.5	89.2	**2.3** M	**6.7** G	189
RT-DETR-MobileNetV4	97.1	96.2	**93.5**	11.4 M	48.8 G	201
RT-DETR-StarNet	**97.2**	**96.4**	93.1	11.5 M	48.5 G	**205**
YOLOv8s	94.3	91.1	86.4	**11.2 M**	**28.6 G**	184
YOLOv9m	97.1	93.5	90.8	20.1 M	76.8 G	172
Gold-YOLOs	95.9	92.8	88.5	21.5 M	46.1 G	175
YOLOv10m	97.2	94.1	91.8	15.4 M	59.1 G	186
MDD-DETR	**99.3**	**99.3**	**98.2**	13.4 M	36.4 G	**198**

RT-DETR-MobileNetV4 [62] and RT-DETR-StarNet [63] modified the RT-DETR backbone to be more lightweight. While these adjustments reduced the number of parameters and computations, they also resulted in lower accuracy, highlighting the superior performance of MDDNet. Additionally, compared to MDD-DETR, MDD-DETR maintains ultra-high accuracy with approximately the same number of parameters, further emphasizing its effectiveness.

4.4. Visualization Experiments

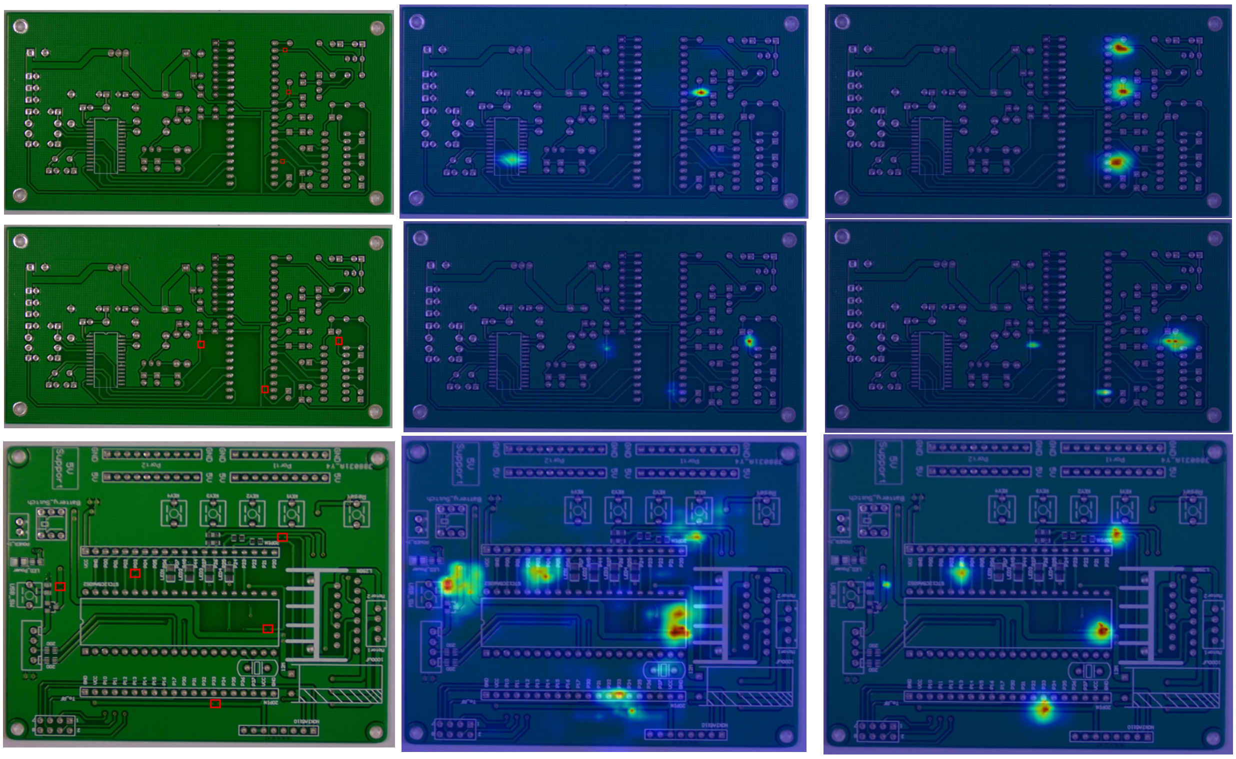
As shown in Figure 9, the detection results for three challenging defect types—spur, mouse bite, and open circuit—are presented. The results indicate that our algorithm not only accurately identifies all defects but also achieves higher detection scores compared to the baseline. Furthermore, our algorithm correctly classifies these defects. The results show that the baseline algorithm can only recognize a single defect when detecting small spur defects, while our algorithm can recognize all defects, and the baseline algorithm can only correctly recognize one defect when detecting mouse bite defects and recognize another mouse bite defect as a very similar open circuit defect, while our algorithm not only recognizes all defects but also correctly categorizes the defects. In the detection of open circuit defects (complex backgrounds), it is comprehensive and achieves higher detection scores than the baseline.

Figure 10 presents the heat map of PCB detection performance. As shown in the figure, compared to the original model, our model is able to accurately and effectively focus on PCB defects.



(**a**) (**b**) (**c**)

Figure 9. Visualization comparison: (**a**) original; (**b**) baseline; (**c**) MDD-DETR.



(**a**) (**b**) (**c**)

Figure 10. Comparison of heat maps for detection performance: (**a**) original; (**b**) baseline; (**c**) MDD-DETR.

4.5. The Generalization of the Experiment

To demonstrate the general applicability of the algorithm presented in this paper, we compared MDD-DETR and RT-DETR across three datasets: the ceramic tile surface defect dataset, the aluminum surface defect dataset [64], and the strip surface defect dataset (NEU-DET). The ceramic tile dataset includes six types of defects, such as white and black spots, which are small targets with features resembling subtle background differences and detection challenges similar to those in PCB datasets. Both the NEU-DET and aluminum surface defect datasets display a long-tail distribution with numerous small target defects, further increasing detection difficulty. These datasets were chosen to evaluate the generality of the proposed algorithm. As shown in Table 10, in all three datasets, our algorithm achieved better detection performance compared to the baseline algorithm.

Table 10. The generalization of the experiment.

Dataset	Model	*P*	*R*	*mAP*
Ceramic Tiles	RT-DETR	72.8	68.6	72.5
	MDD-DETR	80.6	73.8	77.9
NEU-DET	RT-DETR	76.8	72.7	75.6
	MDD-DETR	83.3	77.8	81.7
Aluminum Product	RT-DETR	90.2	84.2	89.5
	MDD-DETR	90.9	84.6	90.2

5. Conclusions

To address the challenges posed by minimal differences between target features and the background, as well as the high proportion of small targets in PCB defect detection tasks, this study proposes an enhanced RT-DETR detection model, MDD-DETR. The model improves the extraction of small defect features through the novel backbone network, MDDNet, and enhances the ability to capture critical features related to minor PCB defects by incorporating a high- and low-frequency feature extraction attention mechanism. Additionally, INM-IoU is used to optimize the loss function, allowing the model to better focus on the features of minor defects and distinguish them from the background. The reconstructed neck further improves feature fusion and information transfer efficiency. Experimental validation on a PCB defect dataset demonstrates the algorithm's effectiveness in defect detection tasks. This model is still lacking in terms of the edge device deployment performance. Future work will focus on optimizing the model to reduce parameters and computational requirements, while also enhancing its robustness for deployment on edge devices.

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

Funding: This work was supported by the Fuxiaquan National Independent Innovation Demonstration Zone High end Flexible Intelligent Packaging Equipment Collaborative Innovation Platform Project (2023-P-006) and the 2022 Fujian Provincial Key Project for Science and Technology Innovation (2022G02007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy concerns.

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

References

1. Annaby, M.H.; Fouda, Y.M.; Rushdi, M.A. Improved Normalized Cross-Correlation for Defect Detection in Printed-Circuit Boards. *IEEE Trans. Semicond. Manuf.* **2019**, *32*, 199–211. [CrossRef]
2. Tian, X.; Zhao, L.; Dong, H. Application of image processing in the detection of printed circuit board. In Proceedings of the 2014 IEEE Workshop on Electronics, Computer and Applications, Ottawa, ON, Canada, 8–9 May 2014; pp. 157–159.
3. Yang, Q.; Li, Z. Software Design for PCB defects detection system based on AOI technology. *Information* **2011**, *14*, 4041.
4. Ma, J.; Cheng, X. Fast segmentation algorithm of PCB image using 2D OTSU improved by adaptive genetic algorithm and integral image. *J. Real-Time Image Process.* **2023**, *20*, 10. [CrossRef]
5. Cheng, Y.; Wang, S.; Chen, B.; Mei, G.; Zhang, W.; Peng, H.; Tian, G. An improved envelope spectrum via candidate fault frequency optimization-gram for bearing fault diagnosis. *J. Sound Vib.* **2022**, *523*, 116746. [CrossRef]
6. Sun, X.; Wu, P.; Hoi, S.C. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing* **2018**, *299*, 42–50. [CrossRef]
7. Qiao, S.; Chen, L.-C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 21–25 June 2021; pp. 10213–10224.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016.
9. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
10. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
11. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
12. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
13. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October—2 November 2019.
14. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October—2 November 2019.
15. Ge, Z. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
16. Hu, B.; Wang, J. Detection of PCB Surface Defects with Improved Faster-RCNN and Feature Pyramid Network. *IEEE Access* **2020**, *8*, 108335–108345. [CrossRef]
17. Yuan, M.; Zhou, Y.; Ren, X.; Zhi, H.; Zhang, J.; Chen, H. YOLO-HMC: An Improved Method for PCB Surface Defect Detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2001611. [CrossRef]
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
19. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef] [PubMed]
20. Salscheider, N.O. Featurenms: Non-maximum suppression by learning feature embeddings. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021.
21. Huang, J. PCB defect detection based on an enhanced dab-deformable-DETR. In Proceedings of the Ninth International Symposium on Advances in Electrical, Electronics, and Computer Engineering (ISAEECE 2024), Changchun, China, 1–3 March 2024.
22. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
23. Yu, C.; Chen, X. Railway rutting defects detection based on improved RT-DETR. *J. Real-Time Image Process.* **2024**, *21*, 146. [CrossRef]
24. Liu, M.; Wang, H.; Du, L.; Ji, F.; Zhang, M. Bearing-DETR: A Lightweight Deep Learning Model for Bearing Defect Detection Based on RT-DETR. *Sensors* **2024**, *24*, 4262. [CrossRef] [PubMed]
25. Girshick, R. Fast r-cnn. *arXiv* **2015**, arXiv:1504.08083.
26. Niu, J.; Huang, J.; Cui, L.; Zhang, B.; Zhu, A. A PCB Defect Detection Algorithm with Improved Faster R-CNN. In Proceedings of the ICBASE, Online, 21–23 October 2022.
27. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach with Application to Defect Detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3507014. [CrossRef]
28. Tang, J.; Liu, S.; Zhao, D.; Tang, L.; Zou, W.; Zheng, B. PCB-YOLO: An Improved Detection Algorithm of PCB Surface Defects Based on YOLOv5. *Sustainability* **2023**, *15*, 5963. [CrossRef]
29. Liu, X.; Hu, J.; Wang, H.; Zhang, Z.; Lu, X.; Sheng, C.; Song, S.; Nie, J. Gaussian-IoU loss: Better learning for bounding box regression on PCB component detection. *Expert Syst. Appl.* **2022**, *190*, 116178. [CrossRef]

30. Lan, H.; Zhu, H.; Luo, R.; Ren, Q.; Chen, C. PCB defect detection algorithm of improved YOLOv8. In Proceedings of the 2023 8th International Conference on Image, Vision and Computing (ICIVC), Dalian, China, 27–29 July 2023.
31. Ling, Q.; Isa, N.A.M.; Asaari, M.S.M. SDD-Net: Soldering defect detection network for printed circuit boards. *Neurocomputing* **2024**, *610*, 128575. [CrossRef]
32. Zhang, L.; Chen, J.; Chen, J.; Wen, Z.; Zhou, X. LDD-Net: Lightweight printed circuit board defect detection network fusing multi-scale features. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107628. [CrossRef]
33. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023.
34. Pan, Z.; Cai, J.; Zhuang, B. Fast vision transformers with hilo attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 14541–14554.
35. Zhang, H.; Xu, C.; Zhang, S. Inner-IoU: More effective intersection over union loss with auxiliary bounding box. *arXiv* **2023**, arXiv:2311.02877.
36. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.
37. Ma, S.; Xu, Y. Mpdiou: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv:2307.07662.
38. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Grenoble, France, 19–23 September 2022.
39. Cui, Y.; Ren, W.; Knoll, A. Omni-Kernel Network for Image Restoration. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024.
40. Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; Yuan, Y. Efficientvit: Memory efficient vision transformer with cascaded group attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
42. Cai, X.; Lai, Q.; Wang, Y.; Wang, W.; Sun, Z.; Yao, Y. Poly kernel inception network for remote sensing detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
43. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
44. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking vision transformers for mobilenet size and speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
45. Li, Y.; Li, X.; Dai, Y.; Hou, Q.; Liu, L.; Liu, Y.; Cheng, M.-M.; Yang, J. LSKNet: A Foundation Lightweight Backbone for Remote Sensing. *Int. J. Comput. Vis.* **2024**, 1–22. [CrossRef]
46. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. Repvit: Revisiting mobile cnn from vit perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
47. Fan, Q.; Huang, H.; Chen, M.; Liu, H.; He, R. Rmt: Retentive networks meet vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 17–21 June 2024.
48. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
49. Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; Shan, Y. UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 17–21 June 2024.
50. Chen, H.; Wang, Y.; Guo, J.; Tao, D. Vanillanet: The power of minimalism in deep learning. *Adv. Neural Inf. Process. Syst.* **2024**, *36*. [CrossRef]
51. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
52. Zhang, H.; Zhang, S. Shape-iou: More accurate metric considering bounding box shape and scale. *arXiv* **2023**, arXiv:2312.17663.
53. Tang, F.; Xu, Z.; Huang, Q.; Wang, J.; Hou, X.; Su, J.; Liu, J. DuAT: Dual-aggregation transformer network for medical image segmentation. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Xiamen, China, 13–15 October 2023.
54. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
55. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision transformer with deformable attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
56. Shaker, A.; Maaz, M.; Rasheed, H.; Khan, S.; Yang, M.-H.; Khan, F.S. SwiftFormer: Efficient Additive Attention for Transformer-based Real-time Mobile Vision Applications. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023.

57. Wu, H.; Huang, P.; Zhang, M.; Tang, W.; Yu, X. CMTFNet: CNN and Multiscale Transformer Fusion Network for Remote-Sensing Image Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 2004612. [CrossRef]
58. Sun, S.; Ren, W.; Gao, X.; Wang, R.; Cao, X. Restoring images in adverse weather conditions via histogram transformer. In Proceedings of the European Conference on Computer Vision, London, UK, 15–16 January 2025.
59. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
60. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020.
61. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024.
62. Qin, D.; Leichner, C.; Delakis, M.; Fornoni, M.; Luo, S.; Yang, F.; Wang, W.; Banbury, C.; Ye, C.; Akin, B. MobileNetV4-Universal Models for the Mobile Ecosystem. *arXiv* **2024**, arXiv:2404.10518.
63. Ngiam, J.; Caine, B.; Han, W.; Yang, B.; Chai, Y.; Sun, P.; Zhou, Y.; Yi, X.; Alsharif, O.; Nguyen, P. Starnet: Targeted computation for object detection in point clouds. *arXiv* **2019**, arXiv:1908.11069.
64. Rezazadeh, N.; Perfetto, D.; Polverino, A.; De Luca, A.; Lamanna, G. Guided wave-driven machine learning for damage classification with limited dataset in aluminum panel. *Struct. Health Monit.* **2024**, 14759217241268394. [CrossRef]

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