

Article

PCES-YOLO: High-Precision PCB Detection via Pre-Convolution Receptive Field Enhancement and Geometry-Perception Feature Fusion

Heqi Yang ¹, Junming Dong ¹, Cancan Wang ¹, Zhida Lian ² and Hui Chang ^{1,*}

¹ School of Electrical and Automation Engineering, Shandong University of Science and Technology, Qingdao 266590, China; 202211010927@sdust.edu.cn (H.Y.); 202211100104@sdust.edu.cn (J.D.); 18297696053@163.com (C.W.)

² School of Electronic Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China; zhidalian@sdust.edu.cn

* Correspondence: changhui2000_mic@sdust.edu.cn

Abstract

Printed circuit board (PCB) defect detection faces challenges like small target feature loss and severe background interference. To address these issues, this paper proposes PCES-YOLO, an enhanced YOLOv11-based model. First, a developed Pre-convolution Receptive Field Enhancement (PRFE) module replaces C3k in the C3k2 module. The ConvNeXtBlock with inverted bottleneck is introduced in the P4 layer, greatly improving small-target feature capture and semantic understanding. The second key innovation lies in the creation of the Efficient Feature Fusion and Aggregation Network (EFAN), which integrates a lightweight Spatial-Channel Decoupled Downsampling (SCDown) module and three innovative fusion pathways. This achieves substantial parameter reduction while effectively integrating shallow detail features with deep semantic features, preserving critical defect information across different feature levels. Finally, the Shape-IoU loss function is incorporated, focusing on bounding box shape and scale for more accurate regression and enhanced defect localization precision. Experiments on the enhanced Peking University PCB defect dataset show that PCES-YOLO achieves a mAP50 of 97.3% and a mAP50–95 of 77.2%. Compared to YOLOv11n, it shows improvements of 3.6% in mAP50 and 15.2% in mAP50–95. When compared to YOLOv11s, it increases mAP50 by 1.0% and mAP50–95 by 5.6% while also significantly reducing the model parameters. The performance of PCES-YOLO is also evaluated against mainstream object detection algorithms, including Faster R-CNN, SSD, YOLOv8n, etc. These results indicate that PCES-YOLO outperforms these algorithms in terms of detection accuracy and efficiency, making it a promising high-precision and efficient solution for PCB defect detection in industrial settings.

Keywords: PCB defect detection; PCES-YOLO; Pre-convolution Receptive Field Enhancement; feature fusion; loss function



Received: 17 June 2025

Revised: 1 July 2025

Accepted: 2 July 2025

Published: 7 July 2025

Citation: Yang, H.; Dong, J.; Wang, C.; Lian, Z.; Chang, H. PCES-YOLO: High-Precision PCB Detection via Pre-Convolution Receptive Field Enhancement and Geometry-Perception Feature Fusion. *Appl. Sci.* **2025**, *15*, 7588. <https://doi.org/10.3390/app15137588>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Printed circuit boards (PCB) are the key carriers of electrical connections between electronic components, and their quality directly affects the performance and life of electronic products [1]. However, their manufacturing process is highly complex, which can easily result in defects on its surface, such as missing holes, mouse bites, open circuits, short circuits, burrs, miscellaneous copper, etc. [2]. Because of this, PCB defect detection has become an indispensable part of PCB industrial production [3].

Traditional PCB defect detection relies on manual visual inspection, which depends entirely on operator skill and personal judgment [4]. It has several drawbacks: low efficiency, human bias, and failure to meet industrial quality control requirements. With the increasing recognition of these limitations, machine vision has gained growing popularity in the semiconductor manufacturing industry. The industry now widely employs automated optical inspection (AOI) systems based on machine vision [5]. However, AOI systems face inherent challenges, such as high costs, suboptimal performance in complex environments, and difficulties detecting tiny defects [6]. In contrast, deep learning-based PCB defect detection offers a superior solution, characterized by lower costs, faster processing speed, and higher accuracy [7]. These advantages have established deep learning as the leading technology for PCB inspection in the current landscape.

Current deep learning-based object detection algorithms can be broadly categorized into two paradigms [8]. The first category comprises two-stage detection architectures, exemplified by the R-CNN [9] series, including Fast R-CNN [10] and Faster R-CNN [11]. These approaches employ a region proposal network to initially generate numerous candidate bounding boxes potentially containing target objects, followed by subsequent classification and regression operations performed on these proposals [12]. This hierarchical detection framework demonstrates particular suitability for PCB defect detection tasks, where precise localization and accurate classification of subtle defects are paramount. For PCB defect detection, Chen et al. [13] proposed an advanced PCB defect detection algorithm using MAFF-Net, an enhanced Faster R-CNN architecture with ResNeXt101 as the feature extraction backbone. Their approach integrates a normalization-based attention mechanism within the residual structure and an AFF module with a coordinate attention mechanism, collectively achieving visible improvements in detection accuracy. Xie et al. [14] developed a deep learning system based on the Faster R-CNN framework with ResNet101 as the backbone, incorporating a feature pyramid network (FPN) and replacing the ROI-Pooling algorithm with an ROI-Align approach utilizing bilinear interpolation. This modification mitigates feature map distortion caused by quantization during the interception period, yielding a high-performance detection network. Hu et al. [15] optimized the Faster R-CNN architecture by using ResNet50 with an FPN, GARPNet, and ShuffleNetV2's residual units. The experimental results demonstrate that the system is more suitable for practical production applications compared to alternative methods. The second paradigm consists of single-stage detection architectures, with YOLO (You Only Look Once) as the archetype. These end-to-end detectors exhibit superior computational efficiency and broader applicability than two-stage models, though with a slight trade-off in detection accuracy. Recent advancements in YOLO-based frameworks have significantly enhanced PCB defect detection through innovative modifications. Zhang et al. [16] designed SF-YOLO by miniaturizing the detection head of YOLOv8, obtaining 0.6% mAP50 and 1.6% mAP50–95 improvements alongside 17.2% parameter reductions. Yuan et al. [17] introduced YOLO-SSW, incorporating P2 high-resolution features, Conv SPD for fine-grained retention, and SimAM-enhanced backbone with Wise-IoU, achieving 98.4% accuracy (a 0.8% improvement). Zhou et al. [18] developed EPD-YOLO with a FIT structure and SF-Head, demonstrating 97.6% mAP at only 5.13M parameters and a 7.4ms inference time. Zhang et al. [19] proposed LPCB-YOLO, a lightweight and high-performance model for PCB defect detection, which integrates CSPELAN modules for feature extraction, a C-SPPF module for multi-scale feature fusion, and a C2f-GS module to enhance semantic and detail feature integration. LPCB-YOLO achieves a 24% reduction in model size while maintaining 97.0% precision and recall. Li et al. [20] developed YOLO-DFA, an enhanced YOLOv10-based PCB defect detection algorithm. It employs a dual-backbone architecture to reduce information loss, a fine-grained feature enhancement method with dynamic weighting, and

an adaptive scale-enhanced loss function. The experimental results show that the model attains a 96.4% mAP. Wang et al. [21] presented PCP-YOLO, a network that incorporates the lightweight PotentNet feature extraction module, the C2f_ParallelPolarized fusion module based on self-polarization attention, and the CARAFE up-sampling technique, demonstrating superior performance over existing methods. Wang et al. [22] proposed SSHP-YOLO, a high-precision PCB defect detector for small samples. It integrates ELAN-C and ASPPCSPC modules with SiOU loss, achieving 97.80% mAP and 11.84% higher recall than YOLOv7, effectively reducing missed/false detections. Yin et al. [23] proposed MAS-YOLO, a lightweight PCB defect detector based on YOLOv12. It introduces MECS attention for subtle features, AHFIN for adaptive fusion, and SAL loss for localization, greatly improving mAP and FPS for real-time industrial inspection.

A comprehensive review of the literature reveals that existing models for PCB defect detection have demonstrated high performance. Through the integration of advanced feature extraction modules, multi-scale feature fusion techniques, enhanced attention mechanisms, and optimized loss functions, these models have demonstrated high detection accuracy and efficiency. Nevertheless, they are inadequate for dealing with PCB background interference, and their preservation of small-target features is restricted to fusion within the backbone network, indicating that there is still room for further improvement in this field. To tackle the crucial problems, this paper proposes a high-precision PCB detection algorithm, termed PCES-YOLO. This novel framework brings together four key innovative elements: the custom-designed C3k2-Pre-convolution Receptive Field Enhancement (PRFE) module, the integration of ConvNeXtBlock, the self-constructed Efficient Feature Fusion and Aggregation Network (EFAN), and the Shape-IoU loss function. Collectively, these components substantially boost detection precision while reducing model parameters. The specific contributions of this paper are outlined as follows:

- (1) **Enhanced Feature Extraction:** The C3k2-PRFE module combines the C3k2 framework with the constructed PRFE module. In this design, a 3×3 convolution is placed before the Receptive Field Enhancement (RFE) module to perform initial local context fusion. This effectively reduces the pixel-level interference at the edges of small targets. The pre-processing enables cleaner and more semantically explicit feature processing in subsequent receptive field enhancement while preventing detail feature dilution caused by large receptive fields. Moreover, the ConvNeXtBlock modules are integrated into the backbone network to capture more extensive semantic information. The synergistic operation between C3k2-PRFE and ConvNeXtBlock evidently lessens the complex PCB background interference during the feature extraction process.
- (2) **Advanced Feature Fusion:** The innovative EFAN feature fusion network presents three new fusion paths. They can effectively combine the shallow, detail-rich backbone features with the deep-network information, thus preserving the critical PCB defect characteristics across multiple feature levels. To handle the increase in parameters resulting from these additional paths, the Spatial-Channel Decoupled Downsampling (SCDown) module is employed. This two-pronged approach resolves the issue of small-target feature loss and, at the same time, optimizes the efficiency of model parameters.
- (3) **Loss function optimization:** The introduced Shape-IoU loss function greatly improves bounding box regression accuracy by incorporating both shape and scale characteristics into its loss calculations, which endows the model with geometry-aware capability for diverse bounding boxes. Furthermore, comparative analyses with other loss functions demonstrate its superior performance for PCB defect detection tasks, particularly in precise defect localization.

The remainder of this study is organized as follows. Section 2 introduces the related work of the YOLO algorithm, including receptive field enhancement, feature fusion, and loss function. Section 3 introduces the overall architecture of the PCES-YOLO model and each innovative module. Section 4 details the configuration of the experimental environment and training parameters, the processing of the experimental dataset, the experimental performance metrics, as well as the experimental results and analyses. Section 5 summarizes and presents the future work.

2. Related Work

2.1. Receptive Field Enhancement

In industrial defect detection, expanding the feature receptive field is critical for improving small target recognition capabilities. Traditional methods rely on stacking convolutional layers or downsampling operations to enlarge the receptive field, but the resulting resolution reduction and detail loss severely constrain small target localization accuracy, particularly under dense background interference. Dilated convolution offers a breakthrough solution. By inserting zero values to expand kernel coverage, it captures multi-scale contextual information and preserves both parameter efficiency and feature map resolution, significantly enhancing the semantic understanding of minute defects [24]. However, standard dilated convolutions still face challenges due to high-frequency noise interference and local feature dilution in complex industrial scenarios. To address this, a 3×3 convolutional layer before dilated convolution is embedded for local context fusion, effectively suppressing pixel-level noise at small target edges and preventing detail attenuation caused by large receptive fields. Simultaneously, the ConvNeXtBlock is introduced to further expand the receptive field and strengthen global semantic capture capability.

2.2. Feature Fusion

Efficiently fusing multi-scale features proves indispensable for improving the detection accuracy of small targets. YOLO architectures initially adopted Feature Pyramid Networks (FPN) [25] to propagate deep semantic information in a top-down manner, enhancing multi-scale representation but losing shallow details due to unidirectional flow. Later, Path Aggregation Networks (PANets) [26] added bottom-up paths to improve localization but still suffered from feature blurring from repeated sampling. Solutions like Bidirectional Feature Pyramid Network (BiFPN) [27] introduced weighted cross-scale fusion to balance feature contributions, but they incurred an increase in parameters through complex connections. To reduce the number of parameters while preserving more detailed features, we designed the EFAN feature fusion network, innovating with three dedicated fusion pathways. These pathways integrate high-resolution details from P2, P3, and P4 layers with deep semantics from P3, P4, and P5 layers, while lightweight SCDown modules control parameter growth. This achieves high-fidelity fusion of shallow and deep features, advancing PCB defect feature preservation.

2.3. Loss Function

Alongside model architecture advancements, significant progress has been made in designing loss functions for bounding box regression, aiming to improve intersection over union (IoU) optimization. Building upon the basic IoU loss, researchers developed GIoU [28] to address gradient vanishing in non-overlapping cases. Subsequently, DIoU [29] accelerates convergence through the centroid distance penalty, and CIoU [29] further optimizes bounding box fitting with aspect ratio constraints. However, these improvements remain limited to relative scale computations, proving inadequate for PCB defect detection with absolute size sensitivity. While SIoU [30] redefines directional penalties via vector

angles, its width–height difference modeling still relies on normalized relative values. WIoU [31] balances easy and hard samples through dynamic focusing but lacks explicit geometric prior modeling. The recently proposed Shape-IoU [32] represents a significant advance in geometric-aware loss design: it introduces size-sensitive weight coefficients to strengthen absolute scale perception and directly constrains physical dimension matching through a shape distance term. By integrating absolute geometric attributes into loss calculations, this approach significantly enhances localization stability for PCB defects, ensuring precise detection in high-density backgrounds.

3. Methodology

3.1. PCES-YOLO Network

The overall architecture of the PCES-YOLO model is illustrated in Figure 1. The model enhances PCB defect detection through three key innovations: feature extraction, feature fusion, and loss function optimization.

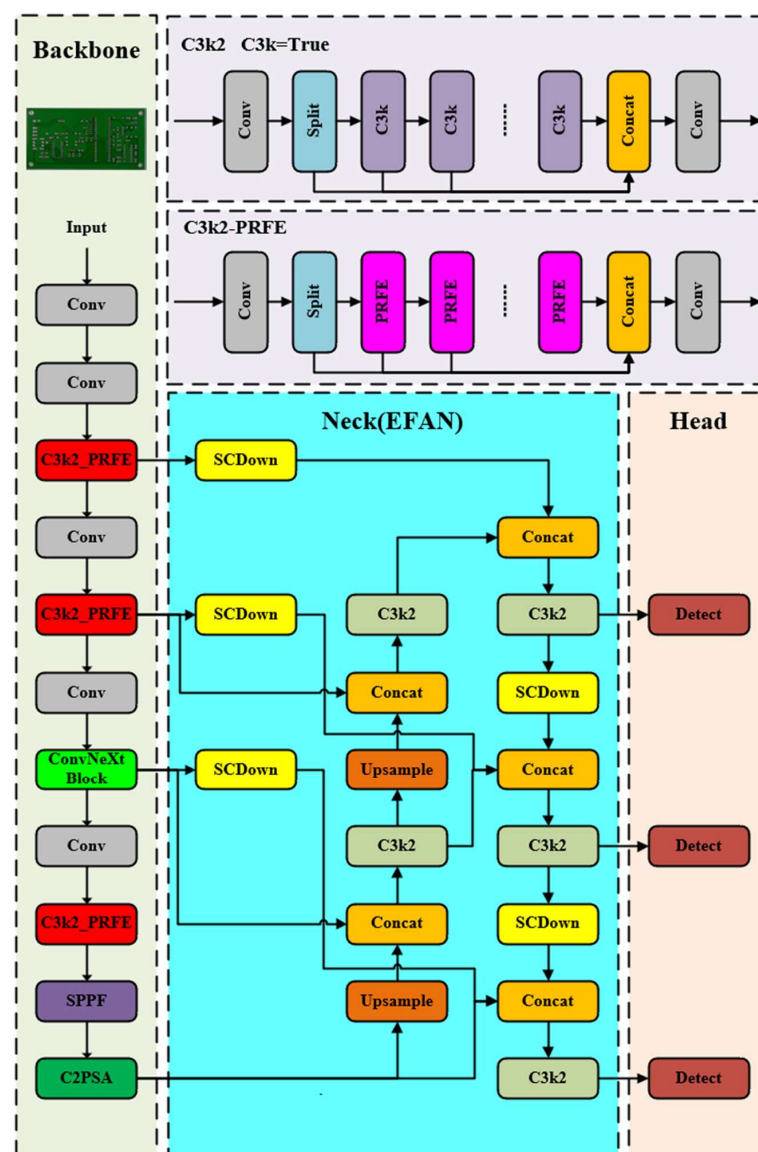


Figure 1. The overall architecture of the PCES-YOLO model consists of four modules: C3k2-PRFE module, ConvNeXtBlock module, EFAN module, and Shape-IoU loss function.

For feature extraction, the backbone network integrates the novel C3k2-PRFE module with ConvNeXt architectural elements. This design expands the feature map's receptive field, thereby deepening its understanding of input data and mitigating background interference. Employing lightweight SCDown modules, the EFAN is developed for feature fusion, which preserves critical detail information while reducing parameter quantity, effectively bridging low-level features from the backbone network and high-level semantic information from the neck network to improve small-target detection performance. In the loss function optimization aspect, a geometry-aware loss function named Shape-IoU is implemented. By considering the bounding box absolute dimensions, it enables more precise boundary regression for defect targets.

3.2. C3k2-PRFE Module

PCB images are characterized by densely packed components, intricate wiring, and diverse textures with substantial background interference. Expanding the receptive field allows the model to capture richer contextual information, thereby enhancing the defect-background discrimination and reducing false detections [33]. Inspired by TridentNet, the RFE [34] module employs four parallel dilated convolutional branches with varying dilation rates to extract multi-scale features. Each branch covers a distinct receptive field, which improves feature representation while minimizing parameters and overfitting risk. Building upon this, an improved receptive field enhancement module, named PRFE, is presented. By replacing the C3k module in C3k2 with PRFE, a novel feature extraction module termed C3k2-PRFE is constructed.

The PRFE module effectively enhances feature extraction through innovative pre-processing techniques. First, a 1×1 point-wise convolution expands input channels, projecting low-dimensional features into a high-dimensional space to enhance the non-linear expressive capacity. Subsequently, depthwise separable 3×3 convolutions are applied to extract spatial features, executing an initial fusion of local context features on the input. The step suppresses pixel-level interference at small target boundaries. The 3×3 convolution operation performs smoothing on input features through weighted averaging, effectively suppressing high-frequency interference, thereby reducing pixel-level noise at small target edges. Meanwhile, the output features of the 3×3 convolution contain local structural information, providing a clearer semantic foundation for subsequent multi-scale dilated convolution operations. The formula for calculating the output features of 3×3 convolution is as follows:

$$Z_{i,j,c} = \sum_{m=-1}^1 \sum_{n=-1}^1 w_{m,n} \cdot x_{i+m,j+n,c} + b_c, \quad (1)$$

where $Z_{i,j,c}$ represents the feature value of the output feature map at position (i,j) in channel c ; $x_{i+m,j+n,c}$ denotes the original value of the input feature map at position $(i+m, j+n)$ in channel c , which may contain pixel-level noise; $w_{m,n}$ denotes the weight parameters of the 3×3 convolution kernel at offset (m,n) ; and b_c denotes the bias term of channel c .

Then, subsequent receptive field enhancement operations are performed on cleaner, more semantically distinct features while also pre-augmenting local small-target characteristics, thus mitigating excessive detail attenuation caused by enlarged receptive fields. The preprocessed features are then split into three parallel branches, each of which undergoes dimensionality expansion via 1×1 convolutions. These branches employ 3×3 dilated convolutions with dilation rates $(d = 1, 2, 3)$, respectively, to capture multi-scale contextual information, forming hierarchical receptive field combinations. After the branch outputs are reduced in dimensionality through 1×1 convolutions, they are concatenated in a residual manner. The fused features are then summed with the original input to retain low-level de-

tails and mitigate gradient vanishing. During the fusion process, global average pooling is utilized to generate channel-wise weight coefficients, which are normalized and multiplied by branch features to dynamically prioritize channels. The weighted features are integrated across channels to produce multi-scale enhanced features. Finally, a 1×1 convolution adjusts the number of channels, which combines with secondary residual connections to ensure the retention of multi-scale critical information. The framework of the PRFE module is displayed in Figure 2.

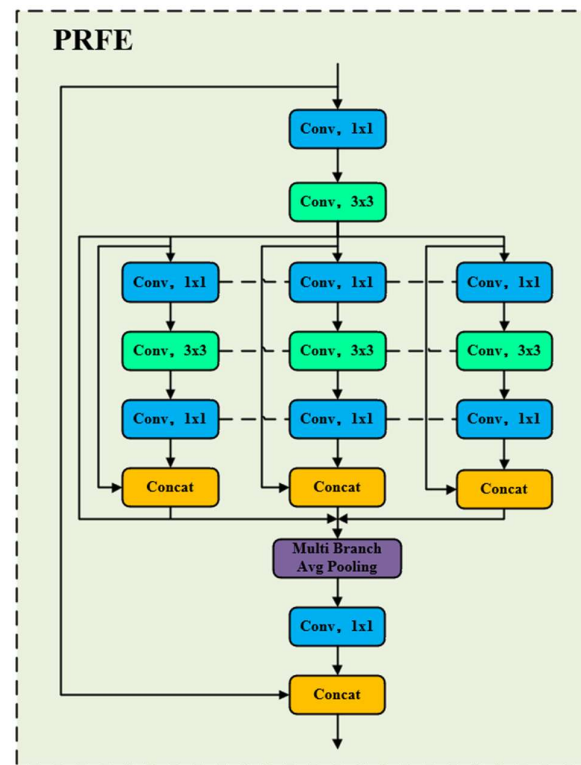


Figure 2. The framework of the PRFE module.

In summary, the PRFE module is constructed by incorporating a 3×3 convolution preprocessing with the RFE module for feature refinement. This stage extracts initial local features while suppressing high-frequency interference. The processed features then enter the receptive field enhancement stage, where parallel dilated convolutions progressively expand the receptive fields. Residual connections are used to preserve hierarchical information, and a feature reweighting mechanism dynamically recalibrates feature weights. This integrated design enables cross-channel feature fusion and dual residual calibration, overcoming fixed receptive field limitations in the conventional architectures. The preprocessing stage further prevents excessive detail attenuation. As a result, the framework greatly enhances feature representation for small-target detection in lightweight networks, yielding measurable performance gains.

3.3. ConvNeXtBlock Module

ConvNeXt [35], a purely convolutional architecture developed by Facebook AI Research and UC Berkeley, reimagines ResNet's structure using a Transformer-inspired design principles. It incorporates large-kernel (7×7) depthwise convolutions, inverted bottlenecks, and layer normalization while retaining the intrinsic translation equivariance and computational efficiency of convolutional networks.

The ConvNeXtBlock framework, as the core component of ConvNeXt, replaces the original C3k2 module at the P4 layer in this paper. The frame diagram of ConvNeXtBlock

module is shown in Figure 3. It employs 7×7 depthwise convolutional operations for spatial feature extraction, effectively expanding the receptive field to incorporate broader contextual information. This design helps reduce interference from PCB backgrounds during defect detection, thereby enhancing detection accuracy. Following the depthwise convolution, ConvNeXtBlock applies LayerNorm (LN) to normalize the channel dimensions of each sample, improving model stability during training. The formulas for calculating the normalized mean and variance are as follows:

$$\mu_n(x) = \frac{1}{CHW} \sum_{c=1}^C \sum_{h=1}^H \sum_{w=1}^W x_{n,c,h,w}, \quad (2)$$

$$\sigma_n(x)^2 = \frac{1}{CHW} \sum_{c=1}^C \sum_{h=1}^H \sum_{w=1}^W (x_{n,c,h,w} - \mu_n(x))^2, \quad (3)$$

where parameters C , H , and W denote the number of channels, height, and width of the input feature map, respectively. The total number of learnable parameter groups is $C \times H \times W$.

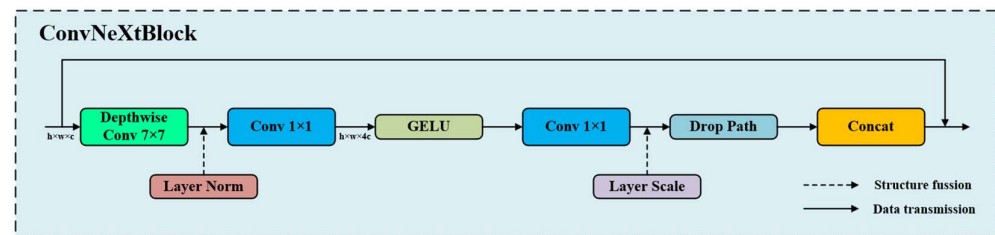


Figure 3. The framework diagram of ConvNeXtBlock module.

Subsequently, ConvNeXtBlock adopts an inverted bottleneck structure. It utilizes two consecutive 1×1 convolutional layers to adjust channel dimensions, extending channel dimensions before feature compression. The combined effect of these two convolutional layers enhances the feature representation for PCB defects. Specifically, the first 1×1 convolution allows the model to learn more complex nonlinear mappings, while the second 1×1 convolution reduces computational complexity and preserves key features. Moreover, the ConvNeXtBlock exploits Gaussian Error Linear Unit (GELU) as its sole activation function, which amplifies PCB defect feature representation capabilities through stochastic regularization properties. Its calculation formula is defined as follows:

$$GELU(x) = 0.5 \times x \times \left(1 - \tanh \left[\sqrt{\frac{2}{\pi}} \times \left(x + 0.044715x^3 \right) \right] \right) \quad (4)$$

The ConvNeXtBlock demonstrates exceptional effectiveness in capturing edge features and fine-grained patterns characteristic in high-density PCB defects, such as short circuits, broken traces, and missing pads. Its expansive receptive field offers distinct advantages for detecting minute defects against complex backgrounds, markedly reducing the likelihood of oversight compared to conventional approaches. Additionally, its inverted bottleneck structure and GELU activation function enhance the feature representation of PCB defects, decreasing the likelihood of false detections. These designs enable the model to better distinguish subtle defect patterns from normal circuit structures, further improving the reliability of high-density PCB inspection.

3.4. EFAN Neck Network

The YOLOv11 neck architecture integrates FPN and PANet for multi-scale feature fusion. As shown in Figure 4a, FPN propagates semantic information in a top-down manner to enrich multi-scale representations, while PANet reinforces localization capabilities by bottom-up aggregation. Although this structure improves detection accuracy, it suffers

from shallow information loss and feature blurring due to repeated sampling. Shallow features are crucial for small-target detection, as their high-resolution edge and texture details compensate for the positional inaccuracies of deep but low-resolution semantics. To address this, an enhanced feature fusion network termed as EFAN is proposed. As depicted in Figure 4b, EFAN incorporates lightweight SCDowN [36] convolutions and three novel feature fusion pathways to optimize the integration of shallow-deep features.

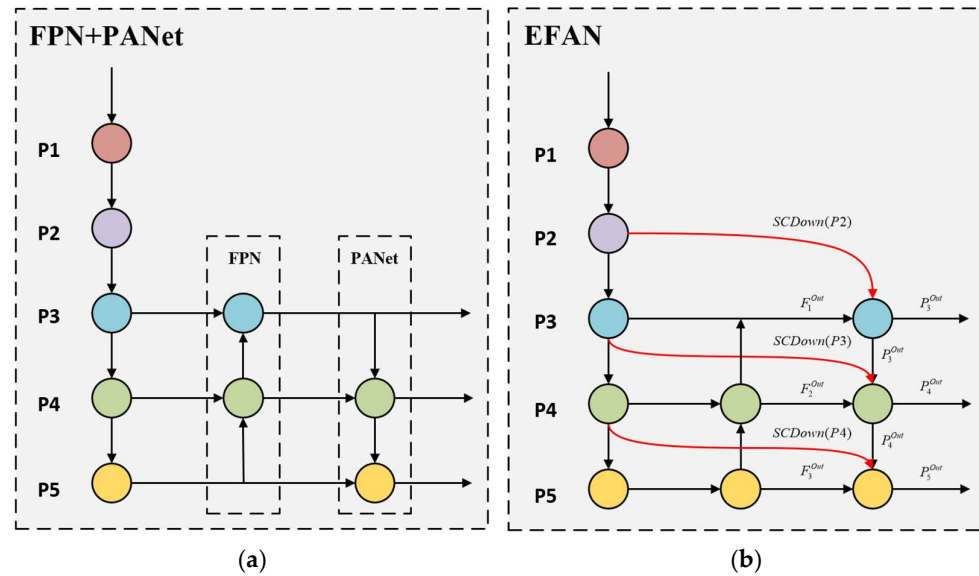


Figure 4. Comparison of feature fusion structure before and after improvement. (a) Original network (FPN + PANet); (b) improved network (EFAN).

To address the loss of feature map details during repeated sampling and fusion, three novel feature fusion pathways are designed. These pathways downsample features from the P2, P3, and P4 layers, then, respectively integrate them with the final outputs of P3, P4, and P5. By directly injecting low-level edge and texture details into high-level features, this design preserves fine-grained information critical for small-target detection. However, the added pathways carry the risk of excessive parameter expansion, which may dilute subtle but essential signals—particularly detrimental for PCB defect detection, which relies on high-resolution features. To mitigate this, SCDowN, a lightweight downsampling module, is introduced to replace standard convolutions. SCDowN employs point-wise convolution for channel adjustment followed by depthwise separable convolution for spatial reduction, significantly reducing computational costs while retaining downsampling fidelity. The calculation of the improved feature fusion is as follows.

$$P_3^{\text{out}} = F_1^{\text{Out}} \oplus F_2^{\text{Out}} \oplus \text{SCDowN}(P_2), \quad (5)$$

$$P_4^{\text{Out}} = F_2^{\text{Out}} \oplus P_3^{\text{out}} \oplus \text{SCDowN}(P_3), \quad (6)$$

$$P_5^{\text{Out}} = F_3^{\text{Out}} \oplus P_4^{\text{out}} \oplus \text{SCDowN}(P_4), \quad (7)$$

where F_1^{Out} denotes the output of the backbone network P3 layer; F_2^{Out} and F_3^{Out} respectively, represent the output of feature fusions along the top-down pathway in EFAN; $\text{SCDowN}(P_2)$, $\text{SCDowN}(P_3)$ and $\text{SCDowN}(P_4)$ respectively, denote the results of downsampling output of layers P2, P3, and P4 using SCDowN; P_3^{Out} , P_4^{Out} and P_5^{Out} signify the output of the feature fusion network; and $\oplus$ represents the feature fusion operation.

In short, EFAN enhances shallow detail retention through supplementary fusion routes and counters parameter inflation via SCDowN, thereby boosting model efficacy.

3.5. Shape-IoU Loss Function

YOLOv11 utilizes the CIoU loss function for localization, which improves upon GIoU by minimizing the normalized distance between bounding box centroids, rather than merely maximizing the overlap area. CIoU also contains an aspect ratio penalty term to improve detection accuracy [29]. However, it only focuses on the relative positional and shape relationships between boxes, ignoring intrinsic geometric properties like absolute size and shape, which are crucial for small-target detection [37]. In PCB defect analysis, where targets are often minute, this can lead to small features being suppressed within overlap regions. To address this, the Shape-IoU loss function was adopted [32], which explicitly incorporates target frame geometry into IoU calculations. This enables the geometric awareness of bounding boxes and stabilizing bounding box regression. Consequently, by introducing a geometry-sensitive loss computation mechanism, the model achieves heightened sensitivity to absolute physical size deviations. This effectively addresses the core issue of minuscule defect sizes being neglected during training, making Shape-IoU particularly suitable for PCB defect detection and other scenarios involving minute object recognition. Shape-IoU is calculated in the following section.

First, the calculation of the intersection over union (IoU) is calculated as follows:

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|}, \quad (8)$$

where IoU measures the overlap between the predicted box (B) and the ground truth box (B^{gt}) as the ratio of their intersection area to their union area.

The weight coefficients in the horizontal (ww) and vertical (hh) directions are defined, respectively, as follows:

$$ww = \frac{2 \times (w^{gt})^{scale}}{(w^{gt})^{scale} + (h^{gt})^{scale}}, \quad (9)$$

$$hh = \frac{2 \times (h^{gt})^{scale}}{(w^{gt})^{scale} + (h^{gt})^{scale}}, \quad (10)$$

where h^{gt} and w^{gt} represent the height and width of the ground truth box, respectively. $scale$ is the scale factor, which is related to the scale of the target in the dataset. The calculation of shape distance is defined as

$$distance^{shape} = hh \times \frac{(x_c - x_c^{gt})^2}{c^2} + ww \times \frac{(y_c - y_c^{gt})^2}{c^2}, \quad (11)$$

where (x_c, y_c) and (x_c^{gt}, y_c^{gt}) represent the center coordinates of the predicted box and the ground truth box, respectively. By accumulating weighted differences in width and height, the shape consistency term is calculated as follows:

$$\Omega^{shape} = \sum_{t=w,h} (1 - e^{-\omega t})^\theta, \theta = 4 \quad (12)$$

The proportions of weighted differences in width (ω_w) and height (ω_h) are, respectively defined as

$$\begin{cases} \omega_w = hh \times \frac{|w - w^{gt}|}{\max(w, w^{gt})}, \\ \omega_h = ww \times \frac{|h - h^{gt}|}{\max(h, h^{gt})}, \end{cases} \quad (13)$$

where w and h represent the height and width of the predicted box, respectively. Shape-IoU is defined as

$$L = 1 - IoU + distance^{shape} + 0.5 \times \Omega^{shape} \quad (14)$$

The diagram of the Shape-IoU prediction frame and real frame is displayed in Figure 5.

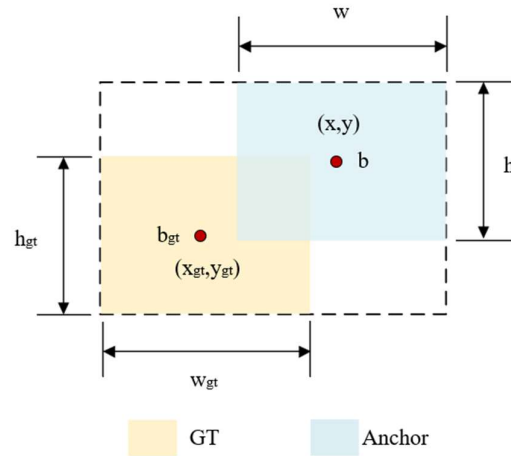


Figure 5. The diagram of Shape-IoU prediction frame and real frame.

4. Experiments and Analysis of Results

4.1. Experimental Environment

The training configuration comprises 640×640 input images with a total of 300 epochs, a batch size of 32, and four parallel processes. The model parameters are optimized using SGD with an initial learning rate of 0.01. The experimental configuration details are listed in Table 1.

Table 1. Experimental environment configuration.

Environment	Configuration
Operating system	Ubuntu 18.04.5 LTS
CPU	Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz $\times$ 4
GPU	Tesla V100S-PCIE-32 GB $\times$ 1
Programming environment	Python 3.8.5
Deep learning framework	Pytorch 1.10.0
CUDA	11.0
RAM	32 GB

4.2. Experiment Dataset

This study utilizes the publicly available PCB defect detection dataset from Open Lab on Human Robot Interaction, Peking University. The dataset comprises 693 images featuring six distinct defect categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper. To mitigate potential overfitting caused by limited sample size, the original dataset was augmented to 10,395 images through cropping, rotation, noise addition, and brightness adjustment. Subsequently, the enhanced dataset was randomly partitioned into training, validation, and test sets in a ratio of 8:1:1 through random sampling. The dataset processing images are shown in Figure 6.

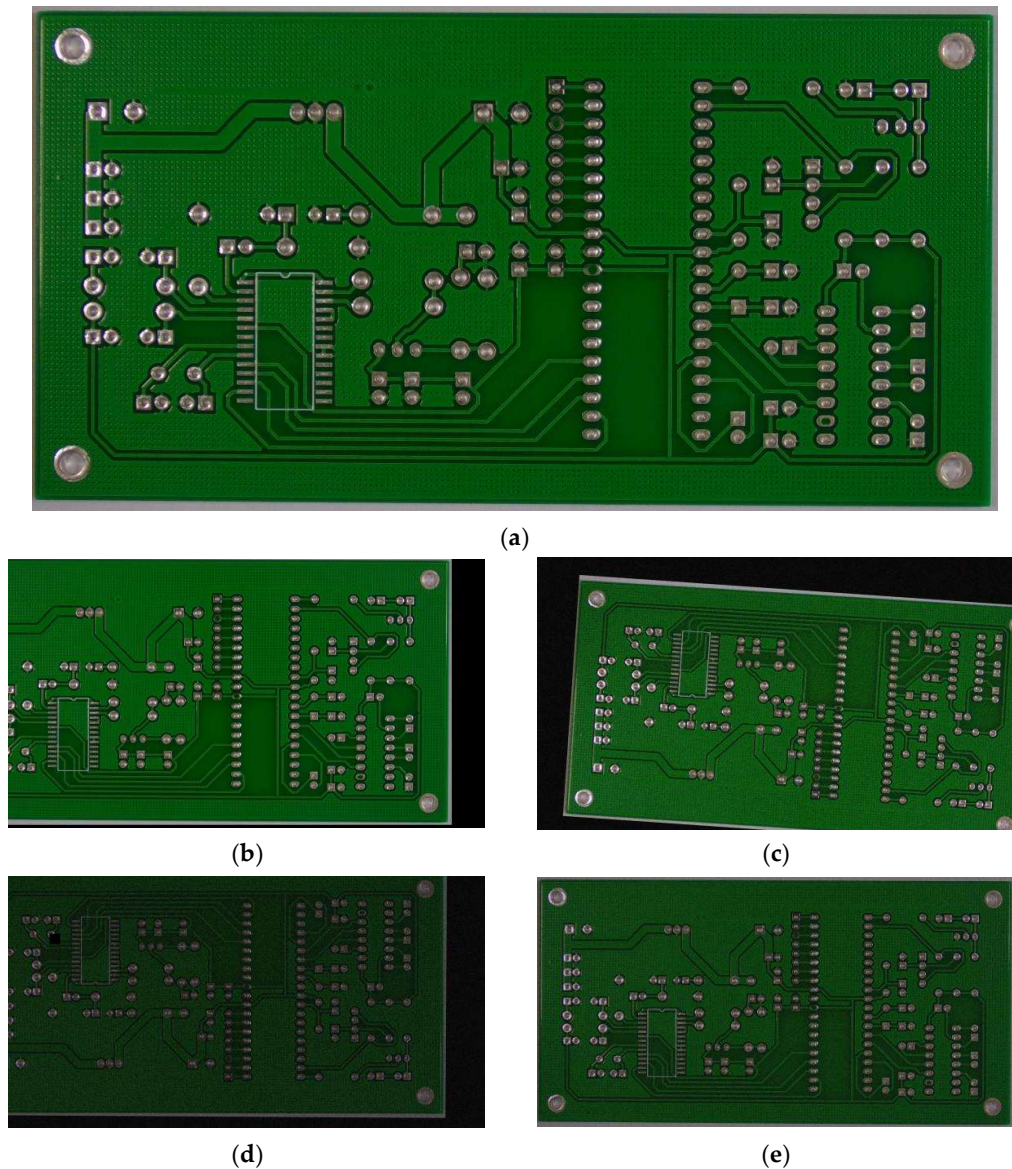


Figure 6. The dataset processing images. (a) Original image; (b) cropped image; (c) rotated image; (d) noise-added image; (e) brightness-adjusted image.

4.3. Evaluation Metrics

In this subsection, precision (P), recall (R), and the mean average precision (mAP) were adopted as the evaluation metrics of the model.

Precision (P) represents the proportion of predicted positive samples that are actually positive and is calculated as follows:

$$P = \frac{TP}{TP + FP} \times 100\%, \quad (15)$$

where TP denotes the number of actual positive samples correctly predicted as positive, and FP represents the number of actual negative samples incorrectly predicted as positive.

Recall (R) represents the proportion of all actual positive samples predicted to be positive. Its calculation is given by the following:

$$R = \frac{TP}{TP + FN} \times 100\%, \quad (16)$$

where FN is the number of actual positive samples incorrectly predicted as negative.

Average precision (AP) quantifies the recognition performance for a specific category by computing the area enclosed by the precision–recall (PR) curve. The calculation is defined as follows:

$$AP = \int_0^1 P(R) dR, \quad (17)$$

where $P(R)$ denotes precision as a function of recall. The mean average precision (mAP) is an important evaluation metric for object detection models, which is computed as the mean of AP values across all categories. It is defined as follows:

$$mAP = \frac{\sum_{i=1}^N AP_i}{N}, \quad (18)$$

where N denotes the total number of categories, and AP_i represents the AP for the i -th category.

4.4. Ablation Experiments

To validate the individual contributions of each model component in PCES-YOLO, a comprehensive ablation study with eight experimental configurations was performed. Initial baselines were established using the standard YOLOv11n architecture, followed by the sequential integration of four novel modules. The quantitative experimental results are presented in Table 2.

Integration of the C3k2-PRFE module yields performance gains, with mAP_{50} increasing by 0.9% and mAP_{50-95} increasing by 2.8%. These enhancements stem from expanded receptive fields and strengthened feature extraction in the backbone network, which reduces background interference and false detections. Subsequent incorporation of the ConvNeXtBlock module raises mAP_{50} by 0.6% and mAP_{50-95} by 2.3%. Implementation of the EFAN feature fusion mechanism achieves measurable benefits, with mAP_{50} increasing by 0.8%, mAP_{50-95} by 2.6%, and recall by 1.9%, indicating its role in preserving shallow feature details and reducing missed detections. Adoption of the Shape-IoU loss function raises mAP_{50} by 1.2%, mAP_{50-95} by 0.4%, and recall by 2.5% through optimized boundary-aware loss computation, which helps to address the challenge of bounding box regression instability.

Progressive integration results in the PCES-YOLO architecture, delivering comprehensive performance gains of 2.5% higher precision, 4.9% better recall, 3.6% greater mAP_{50} , and 15.2% higher mAP_{50-95} than the baseline model.

Table 2. Results of ablation experiments.

Added Modules				P/%	R/%	mAP50/%	mAP50–95/%
C3k2-PRFE	ConvNeXtBlock	EFAN	Shape-IoU				
				96.1	89.7	93.7	62.0
✓				96.8	90.9	94.6	64.8
	✓			97.2	89.9	94.3	64.3
		✓		96.2	91.6	94.5	64.6
			✓	97.0	92.2	94.9	62.4
✓	✓			97.6	91.2	95.1	67.0
✓	✓	✓		98.5	93.0	96.3	72.5
✓	✓	✓	✓	98.6	94.6	97.3	77.2

4.5. Comparison Experiments

4.5.1. Comparison of Different Receptive Field Enhancement Modules

To evaluate the improved receptive field enhancement module, the unimproved RFE module was fused with the C3k2 module to form the C3k2-RFE module, whose performance was compared with that of the C3k2-PRFE module through an experiment. The results are shown in Table 3.

Table 3. Comparative experimental results for different receptive field enhancement modules.

Models	P/%	R/%	mAP50/%	mAP50–95/%
YOLOv11n	96.1	89.7	93.7	62.0
YOLOv11n + C3k2-RFE	96.4	88.2	93.2	61.9
YOLOv11n + C3k2-PRFE	96.8	90.9	94.6	64.8

It can be seen in the data in Table 3 that YOLOv11n with the C3k2-RFE module shows a 0.3% precision improvement, along with a 1.5% recall reduction, a 0.5% decrease in mAP50, and a 0.1% decline in mAP50–95. In contrast, the model with the C3k2-PRFE module demonstrates consistent improvements: a 0.7% increase in precision, a 1.2% recall increase, a 0.9% gain in mAP50, and a 2.8% enhancement in mAP50–95. The results confirm the effectiveness of the improved receptive field enhancement module.

4.5.2. Comparison of Different Loss Functions

To evaluate the effectiveness of Shape-IoU for PCB defect detection, comparative experiments were conducted by replacing the original CIoU loss function in YOLOv11n with three alternatives: SIoU, WIoU, and Shape-IoU. The performance comparison results are presented in Table 4.

Table 4. Comparative experimental results of different loss functions.

Loss Function	P/%	R/%	mAP50/%	mAP50–95/%
CIoU	96.1	89.7	93.7	62.0
SIoU	95.7	89.4	93.5	61.8
WIoU	96.9	91.0	94.6	65.1
Shape-IoU	97.0	92.2	94.9	62.4

It can be seen in the data in Table 4 that the SIoU loss function shows lower performance across all metrics compared to the baseline YOLOv11n with CIoU. In contrast, WIoU achieves measurable improvements, with precision increasing by 0.8%, recall by 1.3%, mAP50 by 0.9%, and mAP50–95 by 3.1%. Shape-IoU outperforms all comparison methods, achieving the highest scores in precision, recall, and mAP50, thereby confirming its suitability for PCB defect detection applications.

4.5.3. Comparison of Different Detector Models

To comprehensively evaluate PCES-YOLO, comparative experiments were conducted against nine different detector models: Faster R-CNN (FR-CNN), SSD [38], YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n, YOLOv12n, and YOLOv11s. Consistent datasets and parameter configurations were maintained across all experiments. As shown in Table 5, PCES-YOLO achieves a mAP50 of 97.3%, significantly outperforming comparative models. Specifically, it surpasses Faster R-CNN by 11.6%, SSD by 21.0%, YOLOv5n by 3.9%, YOLOv8n by 3.8%, YOLOv9t by 5.2%, YOLOv10n by 5.6%, YOLOv11n by 3.6%, and YOLOv12n by 7.2%. Further, it also obtains consistent accuracy gains across all defect categories and maintains parameter efficiency with a reduced model size relative to YOLOv11n.

Among benchmarked models, only YOLOv9t and YOLOv12n have fewer parameters. However, this reduction corresponds to substantially lower accuracy. Furthermore, YOLOv11n with higher detection accuracy is more suitable as the baseline than YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv12n. Compared to YOLOv11s, PCES-YOLO achieves a 1.0% improvement in mAP50, a 5.6% enhancement in mAP50–95, a 0.9% increase in precision, and a 1.1% improvement in recall, thereby reducing false positives and missed detections, despite significant parameter reduction. Consistent accuracy improvements are observed across all defect categories.

Table 5. Comparative experimental results of different detector models.

Algorithm	AP50/%						P/%	R/%	mAP50 /%	mAP50–95/%	Params /M
	Missing Hole	Mouse Bite	Open Circuit	Short	Spur	Spurious Copper					
FR-CNN	93.3	81.3	82.8	91.7	80.2	85.1	88.4	82.6	85.7	47.3	136.5
SSD	85.5	71.2	75.4	81.4	65.4	79.0	79.4	71.8	76.3	39.7	24.28
YOLOv5n	99.3	90.7	91.4	96.6	85.7	96.9	95.8	87.9	93.4	60.6	2.65
YOLOv8n	99.4	92.3	90.6	97.8	84.0	96.6	95.7	88.1	93.5	61.4	3.16
YOLOv9t	99.3	89.7	87.7	97.0	84.2	94.6	95.4	86.9	92.1	58.6	2.13
YOLOv10n	99.3	89.4	87.1	96.5	83.4	94.7	94.4	86.7	91.7	60.0	2.78
YOLOv11n	99.4	91.7	91.1	98.1	84.5	97.5	96.1	89.7	93.7	62.0	2.62
YOLOv12n	98.8	87.8	88.8	95.7	80.4	89.3	94.3	84.6	90.1	54.5	2.55
YOLOv11s	99.5	95.5	96.1	99.3	88.2	99.0	97.7	93.5	96.3	71.6	9.46
PCES-YOLO	99.5	96.7	98.2	99.4	90.7	99.4	98.6	94.6	97.3	77.2	2.60

Tables 6 and 7, and Figure 7a present the training dynamics of PCES-YOLO compared to YOLOv11n, with faster and smoother loss convergence, a much lower loss value, strong training stability, and higher detection accuracy. Figure 7b–d indicate that PCES-YOLO outperforms YOLOv11n in precision, recall, and mAP50. During the initial 0–10 training epochs, PCES-YOLO exhibited a rapid decrease in loss, while its precision, recall, and mAP50 metrics increased rapidly. Its convergence speed was significantly faster than that of YOLOv11n. Between epochs 10 and 290, the performance metrics of PCES-YOLO stabilized, with its stable values surpassing those of YOLOv11n. In the 290–300 epoch phase, both PCES-YOLO and YOLOv11n showed an increase in loss, indicating overfitting to some extent. This phenomenon occurs because the models learn irrelevant features, such as noise, present in the training data. However, as demonstrated by the data in Tables 6 and 7, the severity of overfitting in PCES-YOLO was far lower than in YOLOv11n, demonstrating that it significantly mitigates overfitting risks and effectively suppresses loss fluctuations. The proposed model exhibits faster convergence and fewer training oscillations.

Table 6. Performance of PCES-YOLO during the training process.

Performance	50 Epochs	100 Epochs	150 Epochs	200 Epochs	250 Epochs	300 Epochs
Loss	0.096	0.083	0.075	0.070	0.062	0.086
Precision/%	96.06	98.04	98.23	98.62	98.55	98.61
Recall/%	91.01	93.22	94.29	94.30	94.42	94.57
mAP50/%	94.69	96.44	96.87	97.04	97.16	97.34

Table 7. Performance of YOLOv11n during the training process.

Performance	50 Epochs	100 Epochs	150 Epochs	200 Epochs	250 Epochs	300 Epochs
Loss	0.731	0.634	0.563	0.502	0.448	1.087
Precision/%	91.01	94.14	95.45	96.33	96.44	96.09
Recall/%	80.24	85.51	87.07	87.41	88.44	89.69
mAP50/%	86.17	90.76	92.09	92.62	93.28	93.72

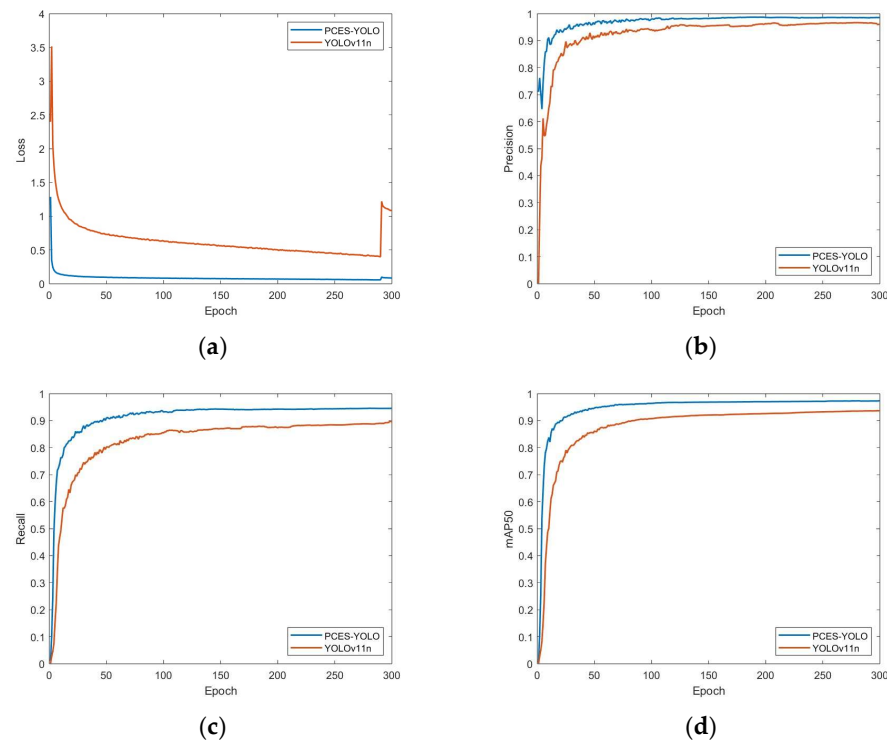


Figure 7. Performance curves: before and after improvement. (a) loss curve; (b) precision curve; (c) recall curve; (d) mAP50 curve.

To verify the performance of the PCES-YOLO model across various defect detection tasks, the confusion matrices for YOLOv11n and PCES-YOLO were examined, as illustrated in Figure 8. PCES-YOLO demonstrates higher correct detection rates and lower missed detection rates for all defect categories compared to YOLOv11n. However, the probability of misclassifying background regions as spur and spurious_copper shows a slight increase relative to YOLOv11n. This occurs because the SCDown lightweight downsampling module adopted in the neck network, while reducing parameter count, may compromise some spatial information, leading to boundary ambiguity between background features and the textures characteristic of spur and spurious_copper defects. In contrast, the probability of misdetection for other defect types decreases due to EFAN's enhanced preservation of detailed features. The experiment shows that the overall performance of the model is good, mainly manifested in the improvement in detection accuracy for all defect categories.

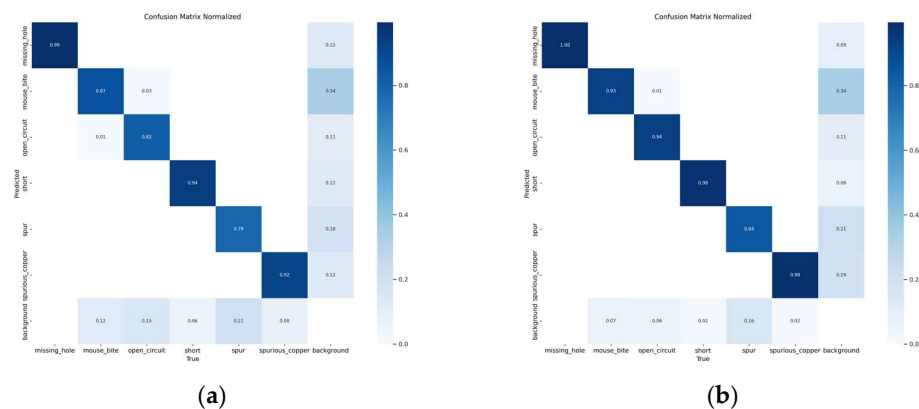


Figure 8. Confusion matrix of YOLOv11n and PCES-YOLO. (a) YOLOv11n; (b) PCES-YOLO.

To further assess detection performance, comparative testing was conducted on randomly selected defective images from six categories, with results in Figure 9. Both models

avoid missed and false detections, but PCES-YOLO demonstrates higher detection accuracy across all defect types compared to the baseline YOLOv11n.

In summary, PCES-YOLO outperforms mainstream detection algorithms, including Faster R-CNN, SSD, and YOLOv8n, in overall performance metrics.

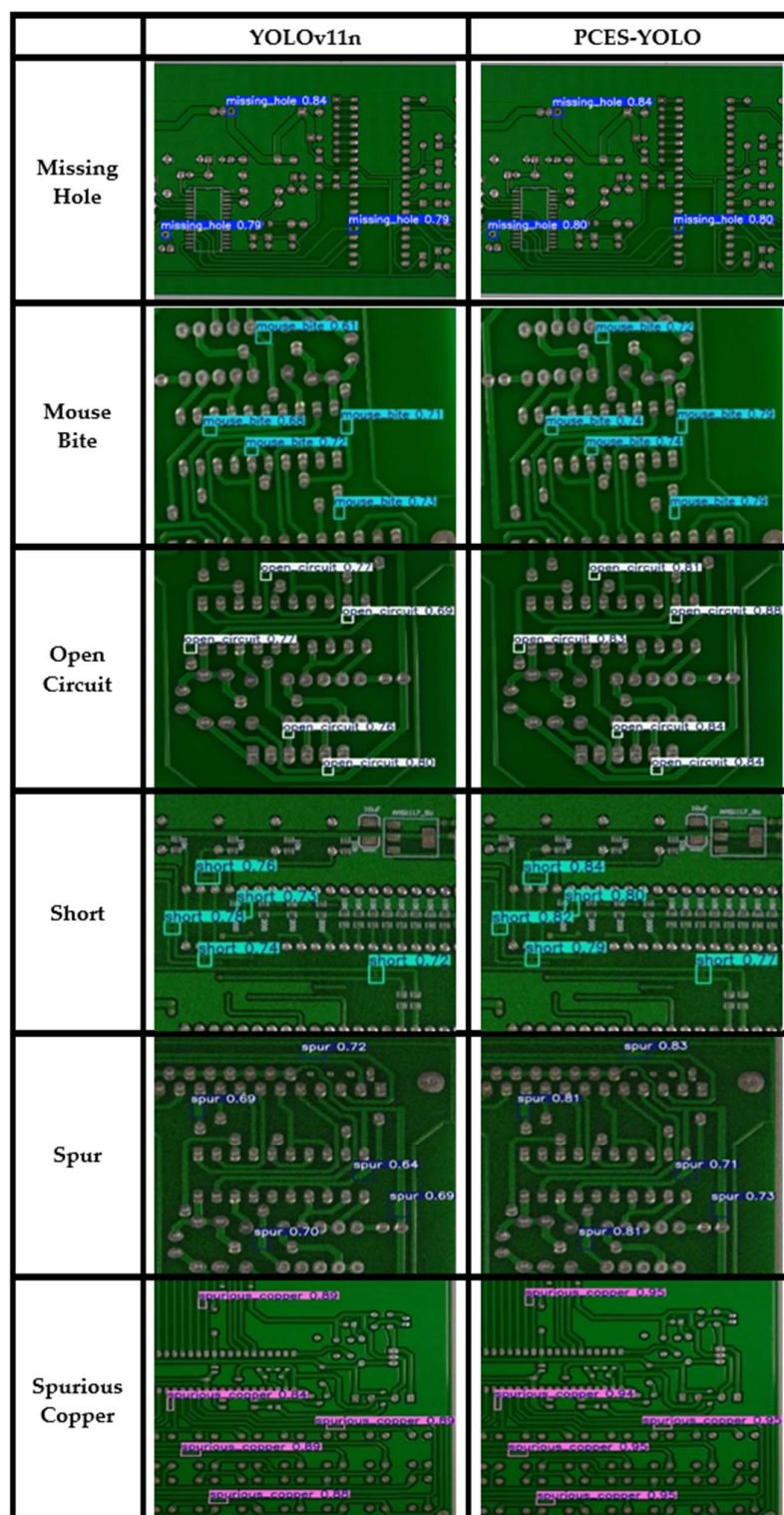


Figure 9. Comparison results of visualization experiments on self-enhanced dataset.

4.6. Experiments Based on the DeepPCB Dataset

To rigorously assess PCES-YOLO's performance on PCB defect detection, comparative experiments were conducted using the DeepPCB dataset [39]. This specialized dataset comprises 1500 image pairs, each containing a defect-free reference image and its corresponding annotated test image. The images were captured using a linear scanning CCD device with a native resolution of approximately 48 pixels/mm and were subsequently processed into aligned 640×640 pixel sub-images. The dataset provides comprehensive annotations for six prevalent PCB defect categories: open, short, mousebite, spur, copper, and pin-hole. The experimental results are detailed in Table 8. The visualization experiment results are shown in Figure 10.

Table 8. Experimental comparison results on the DeepPCB dataset.

Algorithm	P/%	R/%	mAP50/%	mAP50–95/%
YOLOv9t	97.0	94.8	98.3	74.2
YOLOv10n	94.9	93.9	98.1	76.9
YOLOv11n	97.3	96.3	98.7	77.1
PCES-YOLO	98.0	96.9	98.9	77.4

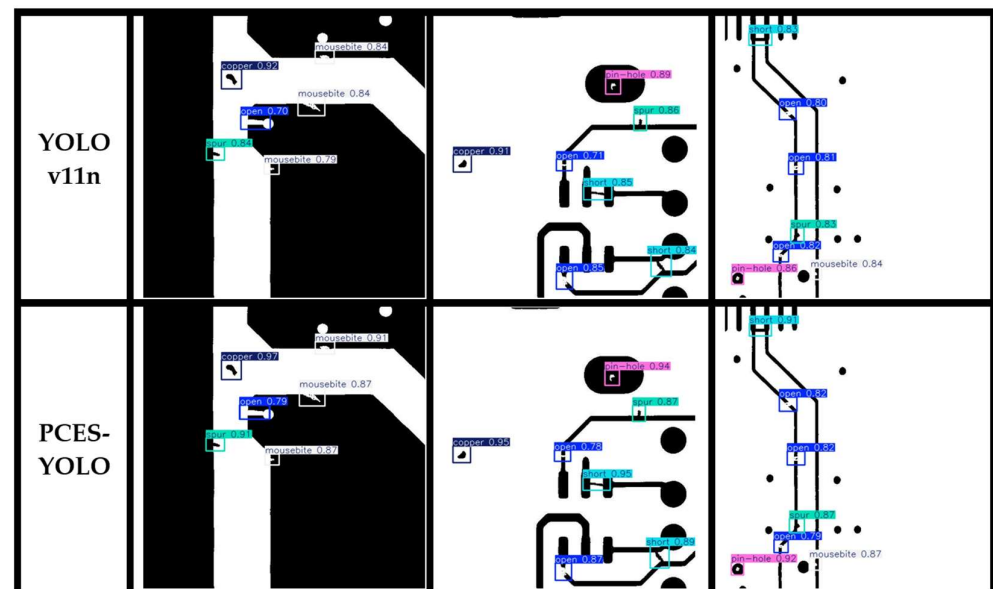


Figure 10. Comparison results of visualization experiments on the DeepPCB dataset.

Table 8 and Figure 10 demonstrate PCES-YOLO's superior performance on the DeepPCB dataset. It consistently outperforms YOLOv9t, YOLOv10n, and YOLOv11n in mAP50, mAP50–95, precision, and recall. The model achieves higher accuracy than YOLOv11n across all defect categories, confirming its exceptional capability in small target detection with low false and missed detection rates. These results demonstrate the excellent performance of the model on different PCB defect datasets and validate its strong generalization performance.

4.7. Limitation of the PCES-YOLO

As demonstrated in the preceding research, PCES-YOLO achieves significant improvements in PCB defect detection accuracy while concurrently reducing model parameters. However, the integration of large-kernel convolutions within the ConvNeXtBlock module introduces a marginal increase in computational load compared to YOLOv11n, resulting in a slight reduction in inference frame rate, as listed in Table 9. It is expected that employing

updated versions of PyTorch, optimizing batch configurations, and improving the acceleration method can effectively mitigate this frame rate decrease. Consequently, the impact on practical deployment scenarios is anticipated to be minimal. Moreover, Table 9 shows that PCES-YOLO's GPU memory usage is lower than that of YOLOv11n and YOLOv11s, making it suitable for deployment on devices with limited resources.

Table 9. Comparative experimental results for frame rate and GPU memory usage.

Algorithm	FPS	GPU Memory Usage/%
YOLOv11n	139	18
YOLOv11s	115	32
PCES-YOLO	133	16

5. Conclusions

This paper presents PCES-YOLO, a high-precision PCB detection model based on YOLOv11n. A 3×3 convolution is added before the RFE module to form the PRFE module, which reduces high-frequency interference and prevents excessive dilution of detailed features by large receptive fields. The PRFE module is then integrated with C3k2 to create the C3k2-PRFE feature extraction module. Furthermore, the ConvNeXtBlock module replaces C3k2 in the P4 layer. These modifications enhance the model's ability to capture a wide range of semantic information, deepen its understanding of input data, and effectively reduce small target feature loss. The innovative EFAN achieves two main goals: reducing the parameter count while effectively integrating shallow detail features with deep semantic features, ensuring the retention of critical defect information across different feature levels. During training, PCES-YOLO employs the Shape-IoU loss function, which emphasizes the geometric attributes of bounding boxes during loss computation, thereby improving regression stability. The experimental results indicate that the proposed model outperforms mainstream detection models in comparison, with improvements of 2.5% in precision, 4.9% in recall, 3.6% in mAP50, and 15.2% in mAP50–95 over YOLOv11n, and effectively addresses the issues of small target feature loss and severe background interference. For industrial deployment, future work will focus on model lightweighting to reduce computational costs without sacrificing accuracy, enabling embedded device deployment. Further optimization will involve real production line testing to enhance applicability in PCB manufacturing.

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

Funding: This research was funded by the Shandong Provincial Natural Science Foundation (No. ZR202103010716), the National Natural Science Foundation of China (No. 62473238), the Shandong Province Major Science and Technology Innovation Project (No. 2022CXGC020204), and the Science and Technology Support Plan for Youth Innovation of Colleges and Universities of Shandong Province of China (No. 2023KG304).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors gratefully acknowledge the reviewers' constructive suggestions and insightful comments that have helped strengthen this research.

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

References

1. Ding, R.; Dai, L.; Li, G.; Liu, H. TDD-net: A tiny defect detection network for printed circuit boards. *CAAI Trans. Intell. Technol.* **2019**, *4*, 110–116. [\[CrossRef\]](#)
2. Shi, P.; Zhang, Y.; Cao, Y.; Sun, J.; Chen, D.; Kuang, L. DVCW-YOLO for printed circuit board surface defect detection. *Appl. Sci.* **2024**, *15*, 327. [\[CrossRef\]](#)
3. Zhang, T.; Zhang, J.; Pan, P.; Zhang, X. YOLO-RRL: A lightweight algorithm for PCB surface defect detection. *Appl. Sci.* **2024**, *14*, 7460. [\[CrossRef\]](#)
4. Yuan, Z.; Tang, X.; Ning, H.; Yang, Z. LW-YOLO: Lightweight Deep Learning Model for Fast and Precise Defect Detection in Printed Circuit Boards. *Symmetry* **2024**, *16*, 418. [\[CrossRef\]](#)
5. Chin, R.T.; Harlow, C.A. Automated visual inspection: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **1982**, *4*, 557–573. [\[CrossRef\]](#)
6. Deng, Y.S.; Luo, A.C.; Dai, M.J. Building an automatic defect verification system using deep neural network for pcb defect classification. In Proceedings of the 4th International Conference on Frontiers of Signal Processing (ICFSP), Poitiers, France, 24–27 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 145–149.
7. Chen, X.; Wu, Y.; He, X.; Ming, W. A comprehensive review of deep learning-based PCB defect detection. *IEEE Access* **2023**, *11*, 139017–139038. [\[CrossRef\]](#)
8. Ma, Y.; Yin, J.; Huang, F.; Li, Q. Surface defect inspection of industrial products with object detection deep networks: A systematic review. *Artif. Intell. Rev.* **2024**, *57*, 333. [\[CrossRef\]](#)
9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 580–587.
10. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1440–1448.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 8–13 December 2014; ACM: New York, NY, USA; pp. 91–99.
12. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 6154–6162.
13. Chen, J.; Wang, S.; Chen, Z.; Huang, W.; Lin, L.; Hong, H.; Chen, G.; Jiang, Z. MAFF-Net: PCB defect detection via feature-enhancement fusion. *Meas. Sci. Technol.* **2025**, *36*, 026137. [\[CrossRef\]](#)
14. Xie, Y.; Dong, Y.; He, H. Defect detection of printed circuit board based on small target recognition network. In Proceedings of the 2022 23rd International Conference on Electronic Packaging Technology (ICEPT), Dalian, China, 10–13 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–5.
15. Hu, B.; Wang, J. Detection of PCB surface defects with improved faster-RCNN and feature pyramid network. *IEEE Access* **2020**, *8*, 108335–108345. [\[CrossRef\]](#)
16. Zhang, K.; Wang, Y.; Shi, S.; Wang, Q.; Chen, X.; Zhang, Z.; Wang, C.; Liu, S. SF-YOLO: Designed based on tiny feature for PCB surface defect detection and deployment in embedded systems. *Signal Image Video Proc.* **2025**, *19*, 263. [\[CrossRef\]](#)
17. Yuan, T.; Jiao, Z.; Diao, N. YOLO-SSW: An improved detection method for printed circuit board surface defects. *Mathematics* **2025**, *13*, 435. [\[CrossRef\]](#)
18. Zhou, Y.; Zheng, X.; Ma, Z.; Huang, K.; Xie, H. Efficient detection method for tiny defects on printed circuit boards based on focus information transfer and structurally flexible head. *Meas. Sci. Technol.* **2024**, *36*, 016004. [\[CrossRef\]](#)
19. Zhang, H.; Li, Y.; Sharid Kayes, D.M.; Song, Z.; Wang, Y. Research on PCB defect detection algorithm based on LPCB-YOLO. *Front. Phys.* **2025**, *12*, 1472584. [\[CrossRef\]](#)
20. Li, K.; Zhong, X.; Han, Y. A high-performance small target defect detection method for PCB boards based on a novel YOLO-DFA algorithm. *IEEE Trans. Instru. Meas.* **2025**, *74*, 2008312. [\[CrossRef\]](#)
21. Wang, Z.; Yuan, H.; Lv, J.; Liu, Y.; Xu, H.; Zhang, Y. PCB fault identification with multiscale feature fusion and heat source attention based on multimodal infrared thermal imaging. *IEEE Trans. Instru. Meas.* **2025**, *74*, 3515613. [\[CrossRef\]](#)
22. Wang, J.; Ma, L.; Li, Z.; Cao, Y.; Zhang, H. SSHP-YOLO: A High Precision Printed Circuit Board (PCB) Defect Detection Algorithm with a Small Sample. *Electronics* **2025**, *14*, 217. [\[CrossRef\]](#)

23. Yin, X.; Zhao, Z.; Weng, L. MAS-YOLO: A Lightweight Detection Algorithm for PCB Defect Detection Based on Improved YOLOv12. *Appl. Sci.* **2025**, *15*, 6238. [\[CrossRef\]](#)
24. Wang, Z.; Ji, S. Smoothed dilated convolutions for improved dense prediction. *Data Min. Knowl. Discov.* **2021**, *35*, 1470–1496. [\[CrossRef\]](#)
25. Tang, Q.; Cao, G.; Jo, K.H. Integrated feature pyramid network with feature aggregation for traffic sign detection. *IEEE Access* **2021**, *9*, 117784–117794. [\[CrossRef\]](#)
26. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 8759–8768.
27. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *arXiv* **2022**, arXiv:1911.09070. [\[CrossRef\]](#)
28. Rezaatofghi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 2767–2778.
29. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IOU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; AAAI: Palo Alto, CA, USA; pp. 12993–13000.
30. Gevorgyan, Z. SIOU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740. [\[CrossRef\]](#)
31. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IOU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051v1. [\[CrossRef\]](#)
32. Zhang, H.; Zhang, S. Shape-iou: More accurate metric considering bounding box shape and scale. *arXiv* **2023**, arXiv:2312.17663. [\[CrossRef\]](#)
33. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Cai, W.; Yang, N.; Wang, B. Multi-scale receptive fields: Graph attention neural network for hyperspectral image classification. *Exp. Sys. Appl.* **2023**, *223*, 119858. [\[CrossRef\]](#)
34. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. Yolo-facev2: A scale and occlusion aware face detector. *Pattern Recogn.* **2024**, *155*, 110714. [\[CrossRef\]](#)
35. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 11976–11986.
36. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458. [\[CrossRef\]](#)
37. Lu, Z.; Zhu, X.; Guo, H.; Xie, X.; Chen, X.; Quan, X. FishFocusNet: An improved method based on YOLOv8 for underwater tropical fish identification. *IET Image Proc.* **2024**, *18*, 3634–3649. [\[CrossRef\]](#)
38. 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, Proceedings, Part I 14, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
39. Tang, S.; He, F.; Huang, X.; Yang, J. Online PCB defect detector on a new PCB defect dataset. *arXiv* **2019**, arXiv:1902.06197. [\[CrossRef\]](#)

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.