ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

PCB Component Recognition based on Improved YOLOv8s for Lightweight Detection

Si Shen[a], Juan Wang[b], Fan Yang[c]

School of Engineering, Huzhou University, Huzhou, Zhejiang 313000, China

[a]2022388721@stu.zjhu.edu.cn, [b]01955@zjhu.edu.cn, [c]02567@zjhu.edu.cn

Abstract

To address the challenges posed by the diverse types of printed circuit board (PCB) components, the low detection accuracy for small components, and high model complexity, this paper proposes the GS-YOLOv8s algorithm. By introducing lightweight Ghost convolution, the computational costs are reduced. A dynamic lightweight C2f-GDC module is designed by incorporating the Ghost Module and DynamicConv into the C2f module of the Backbone, effectively minimizing redundant information within the model. In the Neck network, the SE module is integrated into the bottleneck layer of the C2f module, forming the C2f-SE module, which enhances the network's capability to extract features of small objects. Additionally, a multi-scale feature detection layer is incorporated, leveraging fine-grained information from shallow layers to substantially improve the detection accuracy of small components. Experimental validation conducted on a PCB component dataset demonstrates that GS-YOLOv8s achieves a detection accuracy of 95.6%, a recall rate of 94.6%, and an mAP@0.5 of 97.4%, outperforming YOLOv8s by 0.7%, 1.4%, and 0.5%, respectively. Notably, the model's parameter count is reduced by 30%. The results affirm that the proposed algorithm effectively enhances the detection capability for small objects in complex scenarios, while maintaining high detection accuracy and computational efficiency.

Keywords

YOLOv8s; Small Object Detection; Components; Lightweight; Multi-Scale Features.

1. Introduction

With the advent of Industry 4.0, the importance of electronic products in intelligent manufacturing has become increasingly prominent. As a core component of electronic products, the quality of printed circuit boards (PCBs) directly determines the overall performance and stability of devices. However, components on PCBs are often densely distributed with multi-scale and multi-modal characteristics, and the risk of missed detection is particularly high for small, precision components, which negatively impacts overall detection accuracy [1].

The development of deep learning technologies and significant advancements in GPU computing power have led to a surge in research on PCB component detection based on deep neural networks. Due to their exceptional performance and efficiency, single-stage object detection models have been widely applied in the PCB field. Karl et al. [2] proposed the PCB-METAL dataset, a high-resolution PCB image dataset covering four types of components: integrated circuits (ICs), capacitors, resistors, and inductors. This dataset has shown excellent performance in advanced deep learning object detection models such as YOLOv3 and Faster R-CNN. Sharma [3] optimized the performance of the YOLOv3 network by modifying its network modules and validated the model on the PCB-DSLR and domestic WPCB datasets, effectively addressing the challenges posed by complex scenarios in PCB

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

component recycling environments. Xu et al. [4] combined DyHead with advanced attention mechanisms based on the YOLOv7 model, introduced ACmix and AC-E-ELAN modules, and designed a novel WIoU-Soft-NMS bounding box loss regression method, collectively enhancing the ability to handle dense targets. Ling et al. [5] adopted a lightweight network model using Ghost convolutions and built the C2Focal module to address the scale variance problem in dense object detection. Additionally, they designed the Sig-IoU loss function to improve the accuracy of prediction regression and localization.

Despite these advancements, small components on PCBs often face detection challenges due to their small size, with their features being easily obscured by complex backgrounds, leading to reduced detection accuracy. Moreover, the visual similarity between different types of components further complicates the model's ability to distinguish between them. While improving detection accuracy generally comes at the cost of increased model complexity, which subsequently reduces inference speed, existing PCB component detection methods face a trade-off between accuracy and speed. This trade-off can result in detection delays or misdetections in practical production, limiting the system's usability and reliability. Therefore, optimizing models to balance detection accuracy and speed while improving small object detection capabilities is a critical challenge that needs to be addressed.

To tackle the aforementioned challenges, this study proposes an efficient lightweight object detection model, GS-YOLOv8s. The algorithm replaces standard convolutions with lightweight Ghost convolutions and designs a dynamic lightweight module, C2f-GDC, to further reduce computational redundancy. The C2f-SE module is integrated to enhance the feature representation ability of important channels. Finally, a small object detection layer is added to preserve more shallow-layer information. The proposed model significantly improves small object detection capabilities while reducing model parameters and computational load, providing crucial support for advancing automated PCB production processes.

2. YOLOv8s

The YOLOv1 algorithm initially proposed treating object detection as a regression problem. Following a series of iterative enhancements, YOLOv8 has significantly improved in overall performance, making it one of the most widely adopted single-stage object detection algorithms to date. The architecture of YOLOv8 is primarily divided into four components: Input, Backbone, Neck, and Head.

For the input, YOLOv8 utilizes images of size 640×640, applying preprocessing operations such as random flipping and scaling to augment data diversity. In the Backbone network, based on the CSPDarknet architecture, YOLOv8 integrates the advantages of the ELAN structure from YOLOv7 and introduces a more lightweight C2f module [6]. This module improves gradient flow by splitting the Bottleneck module and incorporating skip connections. The output feature maps are processed by an accelerated Spatial Pyramid Pooling (SPPF) module, which combines features from multiple scales through downsampling and multi-scale max-pooling operations, effectively aggregating information across different scales in the channel dimension.

In the Neck, YOLOv8 adopts the PAN-FPN structure, where the Feature Pyramid Network (FPN) combines high-level semantic information with low-level spatial details in a top-down fashion, while the Path Aggregation Network (PAN) propagates localization information in a bottom-up manner. This architecture enhances the information flow and the overall representational capacity of the feature maps. In the Head, YOLOv8 employs a decoupled head structure, splitting the classification and bounding box regression tasks into two distinct branches. Binary Cross-Entropy (BCE) loss is used to predict classification scores, while Distribution Focal Loss (DFL) and CIOU loss are applied for bounding box regression [7].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

3. Improved YOLOv8 Network Architecture

PCB components exhibit multi-scale characteristics and varying shapes, which present significant challenges for detection. While the YOLOv8s algorithm is effective, it struggles with detecting multi-scale categories and small objects, resulting in suboptimal performance in these areas. To address these limitations, we propose a novel algorithm, GS-YOLOv8s, which aims to reduce missed detections, improve the detection accuracy of small objects, and simultaneously reduce model parameters. The architecture of GS-YOLOv8s is shown in Figure 1.

First, some standard convolutions in the network are replaced with lightweight Ghost convolutions, significantly reducing the model's parameter size. Furthermore, DynamicConv is incorporated into the Ghost module and integrated with the C2f module to form the C2f-GDC module, which replaces the original C2f module in the Backbone network. In the Neck, we design a C2f-SE module, where the SE attention mechanism is applied to the bottleneck of the C2f module. This modification enhances the network's feature representation capability, thereby improving the detection accuracy for small PCB components. Finally, we introduce a small-object detection scale of 160×160 pixels, which increases the receptive field. This new scale, combined with the original three detection layers, forms a multi-scale structure that more effectively handles components with varying sizes in the dataset, enhancing the network's ability to extract features from small objects.

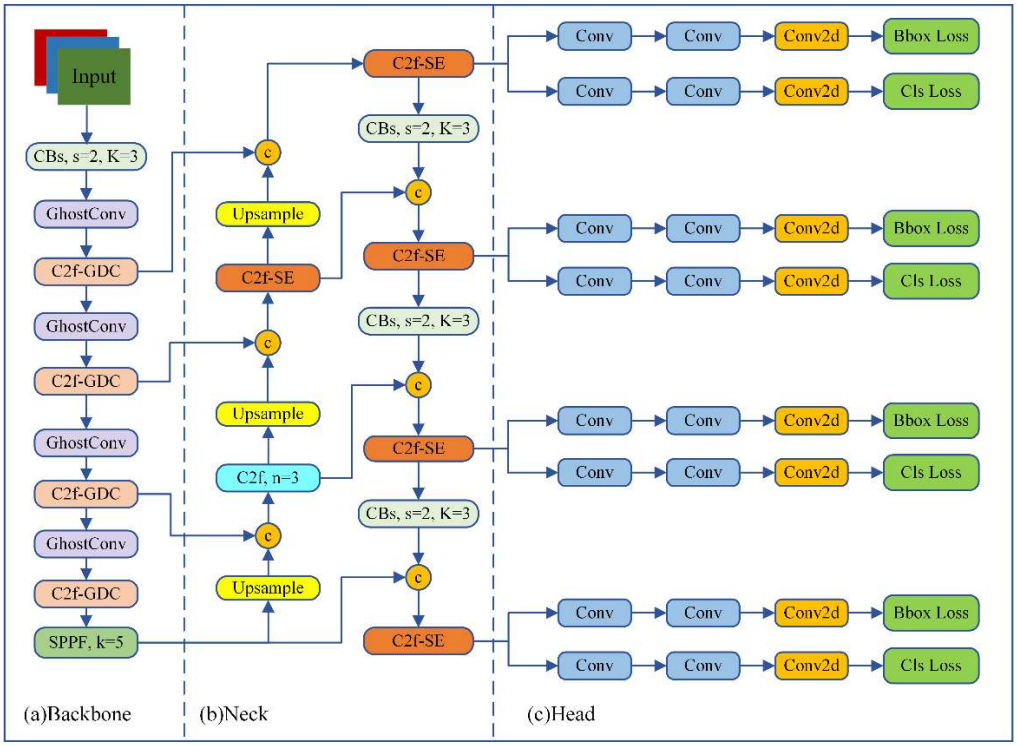


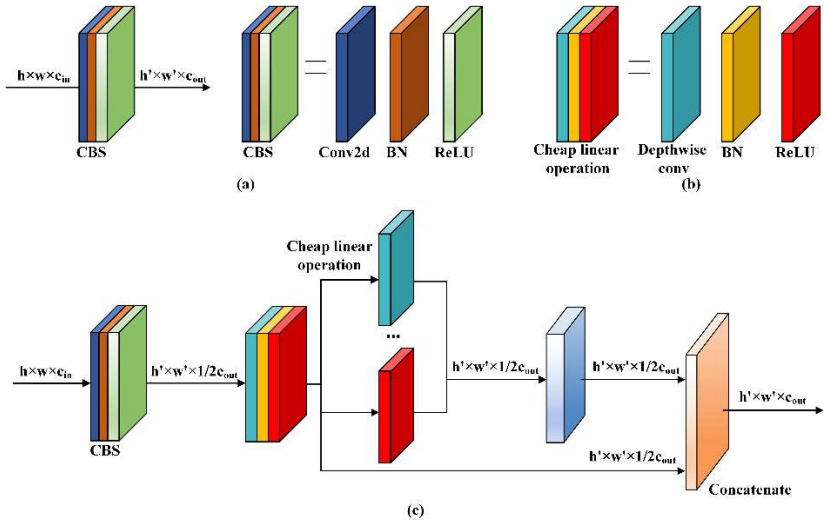
Figure 1. Improved YOLOv8s network architecture

3.1 GhostConv Module

Traditional feature extraction methods typically employ multiple convolutional kernels to perform convolutional mapping across all channels of the input feature map, thereby generating output feature maps. While stacked convolutional layers in deep networks can capture rich feature information, they often produce a large amount of redundant, similar information, leading to unnecessary computational resource consumption. Additionally, the feature maps generated by these convolutions often exhibit overlapping or similar activation responses, failing to provide sufficiently distinct feature representations [8]. To address this issue, Han et al. [9] introduced an efficient lightweight convolution called GhostConv. This method generates additional "ghost features" through inexpensive linear transformations, which are then combined with the primary feature maps generated

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

by conventional convolutions. The ghost features share similar representational capabilities with the primary features while significantly reducing the parameter count and computational complexity. This approach effectively enhances the overall efficiency of the network. The structure of the GhostConv module is shown in Figure 2.



a) Standard Convolution; b) Cheap Operations; c) GhostConv

Figure 2. GhostConv module structure

Given an input feature map $X_{in} \epsilon R^{C_{in} \times h \times w}$, where w and h represent the width and height of the input features, and C_{in} is the number of input channels, the output feature map for a standard convolution is calculated as follows:

$$Y_{c_{out}} = X_{in} * f + b \tag{1}$$

In the equation, $*$ represents the convolution operation, and b is the bias term. The output feature map $Y_{out} \in R^{c_{out} \times h\prime \times w\prime}$, where $w'$and h' represent the width and height of the output features, and c_{out} is the number of output channels.

In contrast, GhostConv optimizes the convolution process. First, a 1×1 convolution is applied to the input feature map to reduce its dimensionality, generating $1/2\, c_{out}$ primary features. Then, these primary features undergo a linear transformation through 3×3 depthwise convolutions, generating $1/2\, c_{out}$ redundant features. Finally, the primary and redundant features are concatenated along the channel dimension to form the complete output feature map.

The floating-point operations (FLOPs) of a convolution are computed based on the element-wise multiplication and accumulation between each element of the convolution kernel and all channels of the input feature map. Therefore, the FLOPs for standard convolution are given by $c_{out} \times h' \times w' \times c_{in} \times k \times k$, with a parameter count of $c_{out} \times c_{in} \times k \times k$. For GhostConv, the FLOPs and parameter count are $1/2\, c_{out} \times h' \times w' \times c_{in} \times k \times k$ and $1/2\, c_{out} \times c_{in} \times k \times k$, respectively. The FLOPs for depthwise convolution are $1/2\, c_{out} \times h' \times w' \times d \times d$, with a parameter count of $1/2\, c_{out} \times d \times d$.

Thus, the computational cost ratio between standard convolution and GhostConv is given in Equation (2), while the parameter count ratio is provided in Equation (3).

$$r_f = \frac{c_{out} \times h\prime \times w\prime \times c_{in} \times k \times k}{1/2 c_{out} \times h\prime \times w\prime \times c_{in} \times k \times k + 1/2 c_{out} \times h\prime \times w\prime \times d \times d} \approx 2 \tag{2}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

$$r_p = \frac{c_{out} \times c_{in} \times k \times k}{1/2c_{out} \times c_{in} \times k \times k + 1/2c_{out} \times d \times d} \approx 2 \quad (3)$$

Here, $k \times k$ and $d \times d$ represent the kernel sizes for standard convolution and depthwise convolution, respectively, with $d \times d \approx k \times k$.

As can be seen from the above equations, the computational cost and parameter count of GhostConv are reduced to half of that of standard convolution [10]. This approach effectively eliminates redundant calculations, significantly reducing computational overhead while maintaining performance in lightweight models.

3.2 C2f-GDC Module

PCB components are typically densely distributed, with complex shapes and diverse sizes. Particularly, small-sized components are often difficult to capture adequately under a fixed receptive field, negatively affecting detection accuracy. The fixed receptive field of standard convolution is limited by a rectangular range, which restricts the flexible extraction of multi-scale information. For components of varying sizes, standard convolution struggles to adapt to the feature requirements of multiple scales simultaneously. Moreover, in regions with minimal feature variation or where spatial information is not prominent, standard convolution continues to perform the same computations, leading to unnecessary information redundancy, which increases computational costs and hinders real-time detection capabilities.

Dynamic Convolution [11] introduces an adaptive feature extraction approach through conditional convolution (CondConv) and dynamic routing (Routing), enabling an "on-demand computation" mechanism. Specifically, DynamicConv first uses global average pooling to extract the global spatial information of the input features, then employs a fully connected layer to generate normalized attention weights. These weights are subsequently used to select different convolution kernels with weighted contributions, optimizing the feature representation. Finally, the output features are generated through the conditional convolution operation, as shown in Figure 3.

Compared to fixed convolution kernels, DynamicConv can dynamically select the optimal combination of kernels based on the input features, thereby more efficiently extracting features in complex scenes, especially in small-object detection tasks. This module not only enables quick focus on the most relevant features but also significantly reduces unnecessary computational burdens.

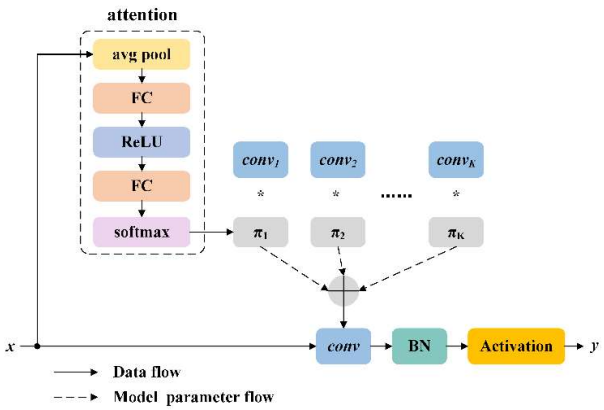


Figure 3. Dynamic convolution structure

In the C2f module of YOLOv8s, the Bottleneck structure employs two standard convolutions for feature extraction. While this structure excels at capturing multi-scale features, its fixed convolutional operations lead to an accumulation of redundant information, resulting in suboptimal computational efficiency. Moreover, the fixed receptive field of standard convolution struggles to adapt to varying

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

feature scales, further constraining the model's performance. To address these limitations, this paper proposes a dynamic lightweight module, referred to as Ghost Dynamic Convolution (GDC). The structure of GDC is illustrated in Figure 4.

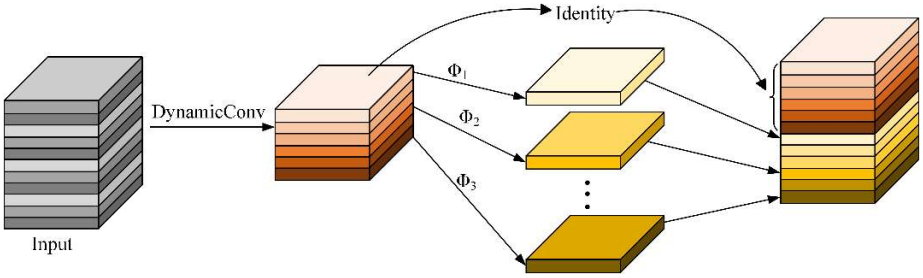


Figure 4. GhostDynamicConv(GDC) structure

The GDC module replaces the standard convolutions in the Ghost module with DynamicConv, enabling more efficient feature extraction. Additionally, in this study, the GDC module is incorporated into the Bottleneck of the C2f module to construct the enhanced dynamic lightweight module, C2f-GDC [12], as shown in Figure 5. The C2f-GDC module effectively reduces information redundancy during the feature extraction phase, significantly optimizing computational efficiency. Furthermore, this module enhances the model's adaptability and feature representation capabilities, providing a superior solution for detecting densely distributed, complex-shaped, and size-diverse PCB components.

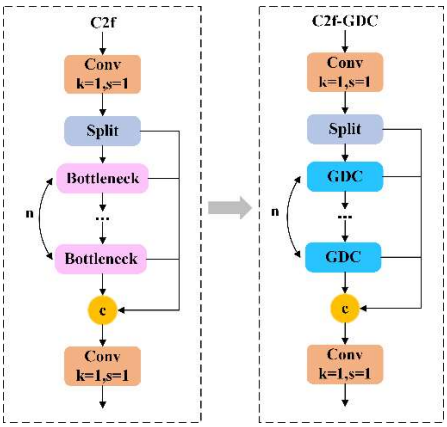


Figure 5. C2f-GDC structure

3.3 C2f-SE Module

To further enhance the model's feature selection and representation capabilities, this study proposes the C2f-SE module (shown in Figure 6). The C2f module in YOLOv8 utilizes a dual convolution bottleneck connection across stages. Initially, the input feature map is processed through convolution and then split into two parts: one part undergoes multiple bottleneck modules, while the other part is retained. The deep features processed by the bottleneck are concatenated with the shallow features that are retained, thereby integrating multi-level features and enhancing the network's semantic representation capability. The bottleneck module consists of two convolution layers, which efficiently extract features and reduce model size. However, after passing through several bottleneck layers, the feature dimensions are compressed, which may lead to the weakening of small or detailed features, thus diminishing the model's ability to detect complex and small targets.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

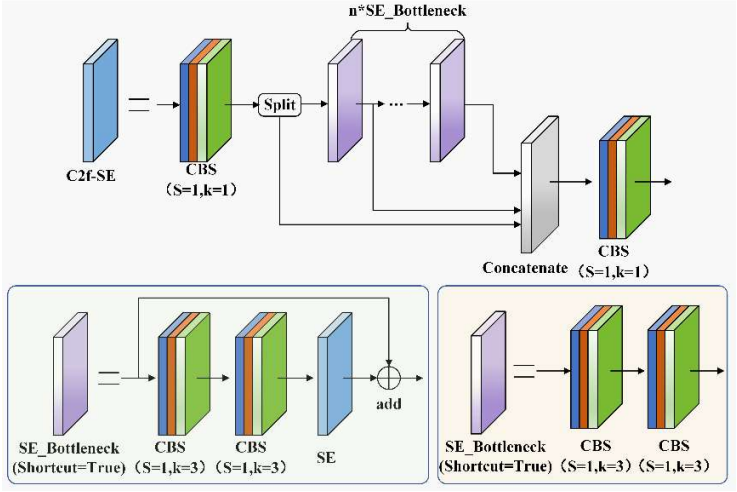


Figure 6. C2f-SE structure

The SE (Squeeze and Excitation) channel attention mechanism [13] consists of three core steps: Squeeze, Excitation, and Scale. Squeeze: The input feature map $X[H' \times W' \times C']$ undergoes a standard convolution operation to generate C feature maps, forming the feature map $U[H \times W \times C]$. Excitation: Global average pooling is applied to compress the spatial information of each channel $(H * W)$ into a single value, reducing the feature map from $[H \times W \times C]$ to $[1 \times 1 \times C]$. This is followed by two fully connected layers, both with a kernel size of 1×1. The first layer reduces the number of channels to lower computational complexity, followed by a ReLU activation. The second layer restores the original number of channels C, using the Sigmoid activation function to generate the channel weights. This entire process transforms the feature map as $[1 \times 1 \times C] \rightarrow [1 \times 1 \times C/r] \rightarrow [1 \times 1 \times C]$. Scale: The obtained channel weights are multiplied with each channel of the input feature map, i.e., $[H \times W \times C] * [1 \times 1 \times C] \rightarrow [H \times W \times C]$[14]. The structure of the SE module is shown in Figure 7. This module dynamically adjusts the importance of each channel in the feature map by adaptively assigning weights to each channel, thereby enhancing the network's ability to capture critical features.

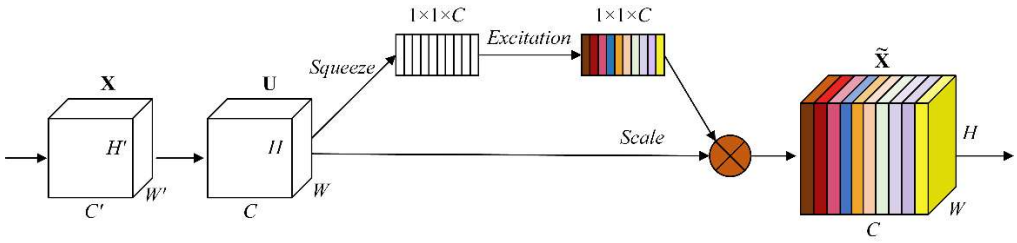


Figure 7. SE structure

The C2f-SE module integrates the SE attention mechanism into the Bottleneck structure of C2f, dynamically reweighting feature maps channel-wise after two consecutive convolution operations. When shortcut=True, the reweighted feature map is added to the original input feature map, forming a residual connection and creating the SE_Bottleneck module. This module emphasizes important feature channels during model training, effectively mitigating information loss in C2f. By replacing the Bottleneck structure in the cross-stage feature fusion module of C2f with SE_Bottleneck, the model enhances the representation of critical features in the convolved feature maps, thereby improving the detection capability for small-sized PCB components.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

3.4 Addition of a Small Object Detection Layer

PCB images contain various components of different sizes, including a significant number of extremely small instances. The YOLOv8 model employs three detection branches at different resolutions (i.e., P3, P4, and P5) to predict small, medium, and large-scale objects, respectively. However, experimental results indicate that these detection branches exhibit limitations in small object detection, leading to a higher risk of missed detections.

To address this issue, this study introduces a multi-scale feature fusion branch by incorporating an additional 160×160 detection head (P2) into the original architecture. This modification integrates fine-grained information from shallower network layers, facilitating a more effective fusion of shallow and deep features. As a result, the proposed approach significantly enhances the detection performance for small PCB components, ensuring improved accuracy in complex industrial scenarios [15].

The multi-scale feature fusion branch is designed based on the PAN-FPN structure, as illustrated in Figure 8. An upsampling module is introduced after the 15th layer to enlarge the 80×80 feature map to 160×160. Subsequently, the 160×160 feature map undergoes a concatenation (Concat) operation, integrating shallow and deep features to enhance multi-scale detection.

Furthermore, since upsampling increases the feature map size, a downsampling GhostConv module is incorporated after the 18th layer to restore the feature map resolution to 80×80. This refined feature map is then fused with the original 80×80 feature map from the 15th layer, ensuring consistency in feature alignment for subsequent detection.

By implementing this multi-scale feature fusion branch, the final detection layers consist of P2, P3, P4, and P5, each responsible for detecting objects at different scales. This multi-scale detection strategy significantly enhances the model's robustness across various object sizes, particularly improving small object detection and precise localization. Additionally, the integration of high-resolution feature maps provides richer shallow-layer information, aiding in the capture of fine details in small objects and reducing missed detections.

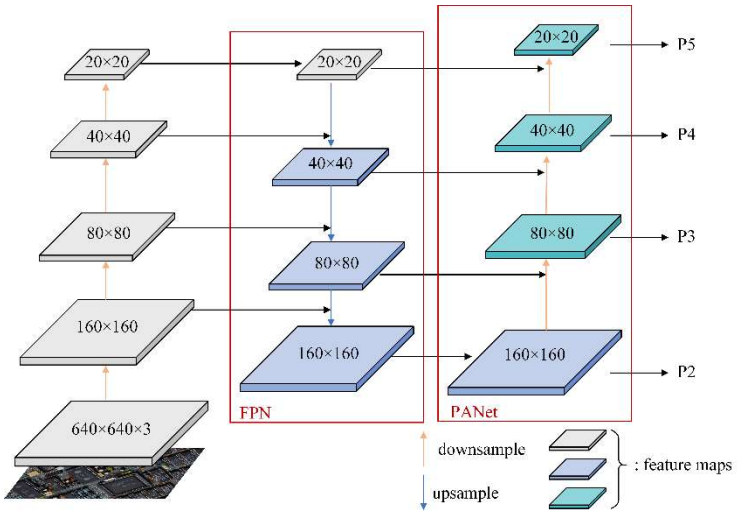


Figure 8. Multi-scale detection diagram

4. Experimental Results and Analysis

4.1 Dataset

Due to the limited availability of publicly accessible datasets for electronic components, this study utilizes a combination of self-collected and online-sourced data, comprising a total of 5,803 images. The dataset is partitioned into training, validation, and test sets in an 8:1:1 ratio. It includes twelve common types of electronic components, such as buttons and capacitors, as detailed in Table 1.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

Table 1. Component categories and types in the dataset

Types	resistor	capacitor	diode	triode
Number	4116	1185	1206	1117
Types	mos	led	crystal	digital tube
Number	932	4087	1131	676
Types	button	stm32	buzzer	photoresistor
Number	2369	416	1016	1070

4.2 Experimental Environment and Parameter Configuration

The hardware for this experiment is based on the Ubuntu 20.04 operating system, with the system specifications including an Intel Core i9-10900KF CPU, an NVIDIA GeForce RTX 3090 GPU, and 24 GB of RAM. The software platform utilizes Python 3.9.0, built on the PyTorch 2.3.0 deep learning framework. To better investigate the proposed model, YOLOv8s is chosen as the baseline model. Key parameters used during the training process are listed in Table 2.

Table 2. The hyperparameters of the improved YOLOv8 model

Hyperparameters	Parameter	Hyperparameters	Parameter
epochs	100	final learning rate	0.0001
batchsize	32	momentum	0.937
initial learning rate	0.01	optimize	SGD

4.3 Evaluation Index

In order to fairly evaluate the detection performance of the improved model, this study uses Precision (P), Recall (R), mean Average Precision (mAP), model parameters, model size, and Frames Per Second (FPS) as evaluation metrics.

The calculation formulas for Precision and Recall are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Precision P represents the ratio of correctly detected positive samples to the total number of positive samples predicted by the model. Recall R denotes the ratio of correctly predicted positive samples to the total actual positive samples. In the formulas, TP (True Positives) refers to the number of actual positive samples correctly classified and predicted as positive, FP (False Positives) represents the number of actual negative samples misclassified and predicted as positive, and FN (False Negatives) indicates the number of actual positive samples misclassified as negative and predicted as negative.

The formulas for calculating Average Precision (AP) and Mean Average Precision (mAP) are as follows:

$$AP = \int_0^1 P(R)d(R) \tag{6}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

$$mAP@0.5 = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{7}$$

Average Precision (AP) is the area under the precision-recall (P-R) curve, reflecting the average accuracy at various recall levels for each class. Mean Average Precision (mAP) provides an overall evaluation of the model's performance by averaging the AP across all classes. In this study, NN represents the number of classes, and since twelve types of PCB components are considered, N = 12.

$$FPS = \frac{1}{t_{preprocess} + t_{inference} + t_{NMS}} \tag{8}$$

Frames Per Second (FPS) is calculated by dividing the number of frames processed by the algorithm by the total time spent on preprocessing, inference, and non-maximum suppression (NMS). The number of parameters refers to the total count of learnable weights and biases in the neural network.

4.4 Ablation Experiment of the GS-YOLOv8s Network

To validate the effectiveness of each improvement module, we conducted ablation experiments on the baseline model using the same dataset, and the experimental results are shown in Table 3. Specifically, M1 represents replacing some of the Conv modules in the network with GhostConv modules, M2 indicates replacing the C2f module in the backbone network with the C2f-GDC module, M3 refers to replacing the C2f module in the neck with the C2f-SE module, and M4 denotes the addition of a small-object detection layer. **A checkmark (√)** indicates that the current network adopts this improvement strategy.

Table 3. Ablation experiment comparison of different improvement modules

Module	M1	M2	M3	M4	P%	R%	mAP@0.5%	Size(MB)	FPS	Parameter (MB)
YOLOv8s	×	×	×	×	94.9	93.2	96.9	22.5	256	11.1
	√	×	×	×	94.5	92.6	96.7	19.3	323	10.0
	√	√	×	×	93	93	96.5	16.9	204	8.3
	√	√	√	×	94.2	93	97.1	17.6	312	8.7
	√	√	√	√	95.6	94.6	97.4	16.0	244	7.8

As shown in Table 3, the GhostConv module reduces the overall network parameters and computational load, significantly improving inference speed. Although the detection accuracy decreased by 0.4%, the mean average precision (mAP) remained almost unchanged, demonstrating that the lightweight GhostConv convolution significantly optimized network efficiency while maintaining model accuracy. The C2f-GDC module excels in reducing both the model parameters and model size, with reductions of 1.7MB and 2.4MB, respectively, indicating that C2f-GDC effectively reduces redundant calculations and optimizes the model structure. Although there is a slight decrease in mAP@0.5, the detection accuracy is still high, and the recall is improved, demonstrating that the improved module enhances the overall feature extraction capability for object detection. The C2f-SE module improved detection accuracy and mAP@0.5 by 1.2% and 0.6%, respectively, with FPS reaching 312 frames per second. This performance improvement is attributed to the channel attention mechanism of the SE module, which enables the network to focus more effectively on key features in the target area. The integrated C2f-SE module shows significant effectiveness in optimizing feature representation and reducing computational complexity. After

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

adding the small target detection layer, detection accuracy, recall, and mAP@0.5 increased by 1.4%, 1.6%, and 0.3%, respectively. The additional detection layer enables the model to capture finer details of small targets at shallower layers, enhancing its perception of small targets. Furthermore, this adjustment effectively reduced redundant parameters, resulting in a decrease in the overall model size.

4.5 Comparison of Different Attention Mechanisms

To validate the effectiveness of integrating the SE and C2f modules, this experiment compares the performance of different attention mechanisms, including SE [16], CA [17], ECA [18], and CBAM [19], when fused with the C2f module. The experimental results are summarized in Table 4.

Table 4. Ablation experiment comparison of different improvement modules

Module	P%	R%	mAP@0.5%	Size(MB)	FPS
C2f-CA	94.4	94.8	97.2	16.0	244
C2f-SE	95.6	94.6	97.4	16.0	244
C2f-ECA	95.9	93.6	97.2	16.0	250
C2f-CBAM	94.4	94.1	97	16.1	250

Based on the data presented in Table 4, the fusion of the SE module with the C2f module yields the best overall network performance. The C2f-SE module achieves a detection precision of 95.6%, representing an improvement of 1.2% compared to C2f-CA and C2f-CBAM, and closely approaching the 95.9% of C2f-ECA. The recall rate reaches 94.6%, surpassing both C2f-ECA and C2f-CBAM while being slightly lower than the 94.8% of C2f-CA. In terms of mAP@0.5, C2f-SE achieves the highest mean average precision of 97.4%, outperforming C2f-CA and C2f-ECA by 0.2% and C2f-CBAM by 0.4%. These results demonstrate that the C2f-SE module effectively balances recall while significantly enhancing detection precision and mAP, thereby exhibiting the best overall performance.

4.6 Multi-Scale Detection Analysis

To validate the effectiveness of the P2 detection layer for small object detection, we conducted replacement experiments by substituting the three detection heads (P3, P4, and P5) in the baseline model with the P2 detection layer. As shown in Table 5, when P2 was used to replace P3, P4, and P5, the model's mAP@0.5 increased by 0.4%, 0.4%, and 0.1%, respectively, while the recall improved by 1.7%, 1.3%, and 1.5%. These results indicate that the introduction of the P2 detection layer provides richer shallow feature information, contributing to a steady improvement in average precision. With the incorporation of the P2 detection layer, the four-scale detection structure achieved the highest detection accuracy of 95.6%, a recall improvement of 1.4%, and an mAP@0.5 increase of 0.5%. The proposed multi-scale structure effectively integrates shallow and deep features through the collaborative operation of P2, P3, P4, and P5, significantly enhancing the overall detection performance of the network.

Table 5. Performance comparison under different detection heads

Module	P%	R%	mAP@0.5%	Size(MB)	FPS	Parameters(M)
P3+P4+P5	94.9	93.2	96.9	22.5	256	11.1
P2+P4+P5	94.8	94.9	97.3	15.5	240	7.6
P2+P3+P5	94.5	94.5	97.3	15.2	263	7.4
P2+P3+P4	94.8	94.7	97	10.1	263	4.8
P2+P3+P4+P5	95.6	94.6	97.4	16.0	244	7.8

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

4.7 Experimental Comparison with State-of-the-Art Object Detection Algorithms

To further validate the superiority of the proposed algorithm, the improved model is compared with several state-of-the-art object detection methods, including Faster-RCNN [20], SSD [21], RetinaNet [22], YOLOv3-tiny [23], YOLOv5s, YOLOv7-tiny [24], and YOLOv8s. The experimental results are presented in Table 6.

Table 6. Performance comparison of different detection models

Module	P%	R%	mAP@0.5%	Size(MB)	FPS	Parameters(M)
RetinaNet	74.2	55.2	50.6	139.9	12.2	36.7
Faster-RCNN	87.6	60.9	56.1	108.7	11.3	28.48
SSD	76.2	56.8	52.1	97.2	29.7	25.5
YOLOv3-tiny	89.6	83.5	89.9	17.5	166.7	8.7
YOLOv5s	95	93.4	96.7	14.4	277.8	7.06
YOLOv7-tiny	91	93.7	95.9	12.3	285.7	6.06
YOLOv8s	94.9	93.2	96.9	22.5	256.4	11.13
Ours	95.6	94.6	97.4	16.0	243.9	7.79
RetinaNet	74.2	55.2	50.6	139.9	12.2	36.7

Based on the experimental results, the improved GS-YOLOv8s model demonstrates superior performance across multiple metrics, particularly achieving a remarkable balance between detection accuracy and model complexity. Compared to the classic two-stage detection algorithm, Faster-RCNN, GS-YOLOv8s shows improvements of 21.4%, 39.4%, and 46.8% in precision, recall, and mAP@0.5, respectively, while its model size is only 11.4% of Faster-RCNN. In single-stage detection algorithms, GS-YOLOv8s outperforms RetinaNet and SSD not only in accuracy but also in computational efficiency. In terms of mAP@0.5, GS-YOLOv8s achieves a 0.5% increase over YOLOv8s and improves by 7.5%, 0.7%, and 1.5% compared to YOLOv3-tiny, YOLOv5s, and YOLOv7-tiny, respectively. Furthermore, compared to YOLOv8s, GS-YOLOv8s reduces the number of parameters by 30%, further lowering computational resource consumption. GS-YOLOv8s significantly outperforms other advanced models in both efficiency and accuracy, demonstrating its excellent effectiveness in detecting small targets in dense components.

4.8 Analysis of PCB Component Detection Visualization Results

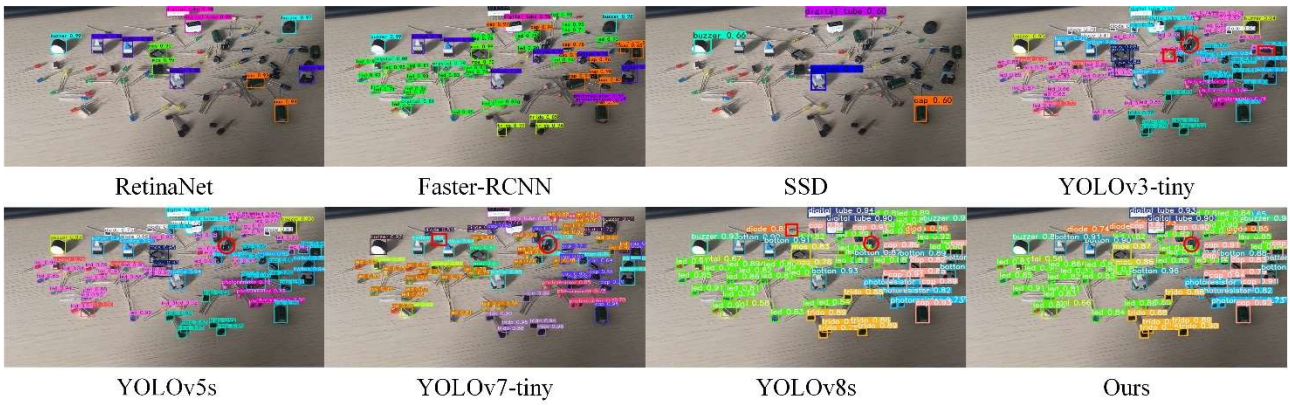


Figure 9. Comparison of detection performance under different models

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

To further validate the feasibility of the proposed algorithm, a PCB image was randomly selected from the test set, and the GS-YOLOv8s algorithm was compared with current state-of-the-art object detection algorithms. The visual results of the comparison are shown in Figure 9.

Analysis of the detection results in Figure 9 shows that the RetinaNet and SSD models failed to effectively identify most components, exhibiting a high miss detection rate. In contrast, the Faster-RCNN model demonstrated improved performance but still exhibited noticeable miss detections. Regarding the detection of overlapping components, the YOLOv3-tiny, YOLOv5s, YOLOv7-tiny, and YOLOv8s models were unable to accurately identify the component categories. Among them, YOLOv3-tiny showed weak detection capabilities for small targets, resulting in significant miss detections. The YOLOv5s model had slightly higher confidence than YOLOv7-tiny, but YOLOv7-tiny was only able to detect one target as a whole in overlapping areas, failing to distinguish components accurately. The GS-YOLOv8s model outperformed YOLOv8s in terms of miss detection rate and demonstrated superior detection accuracy. These results validate the effectiveness of the proposed algorithm, particularly in tasks involving densely distributed and complexly featured component detection, where it exhibits significant advantages.

5. Conclusion

This paper addresses the issues of low small-object detection accuracy, high miss detection rates, and large model parameters by proposing a multi-scale lightweight model—GS-YOLOv8s. The proposed model demonstrates higher detection accuracy while maintaining lightweight characteristics, with significant advantages in small-object detection and dense component recognition, significantly reducing miss detection rates. Building upon the YOLOv8s model, the lightweight Ghost convolution is introduced to significantly reduce computational costs. The C2f-GDC module is employed to reduce redundant information in the network structure and optimize computational efficiency. Additionally, the C2f-SE structure is designed to enhance the model's ability to extract small-object features. Finally, the introduction of a small-object detection layer enables the fusion of shallow and deep features, further improving small-object detection accuracy. Experimental results show that the improved GS-YOLOv8s model achieves a detection accuracy of 95.6%, a recall rate of 94.6%, and a mAP@0.5 of 97.4% in PCB component detection, demonstrating superior efficiency and accuracy compared to other advanced models.

Acknowledgments

The National Natural Science Foundation of China (No. 52305151).

References

[1] Chen J, Bao E, Pan J. Classification and positioning of circuit board components based on improved YOLOv5[J]. Procedia Computer Science, 2022, 208: 613-626.

[2] Mahalingam G, Gay K M, Ricanek K. Pcb-metal: A pcb image dataset for advanced computer vision machine learning component analysis[C]//2019 16th International Conference on Machine Vision Applications (MVA). IEEE, 2019: 1-5.

[3] Sharma H, Kumar H. A computer vision-based system for real-time component identification from waste printed circuit boards[J]. Journal of Environmental Management, 2024, 351: 119779.

[4] Luo S, Wan F, Lei G, et al. EC-YOLO: Improved YOLOv7 Model for PCB Electronic Component Detection[J]. Sensors, 2024, 24(13): 4363.

[5] Ling Q, Isa N A M, Asaari M S M. Precise detection for dense PCB components based on modified YOLOv8[J]. IEEE Access, 2023.

[6] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202504_11(4).0002

[7] Wang G, Chen Y, An P, et al. UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios[J]. Sensors, 2023, 23(16): 7190.

[8] Zhu Hongyan, Li Zeping, Zhao Yong, et al. PCB defect detection algorithm based on multi-scale fusion and deformable convolution[J]. Computer Engineering and Design, 2022, 43(08): 2188-2196.

[9] Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.

[10] Wu L, Zhang L, Zhou Q. Printed circuit board quality detection method integrating lightweight network and dual attention mechanism[J]. IEEE Access, 2022, 10: 87617-87629.

[11] Chen Y, Dai X, Liu M, et al. Dynamic convolution: Attention over convolution kernels[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11030-11039.

[12] Wang B, Hua J, Xia L, et al. A defect detection method for Akidzuki pears based on computer vision and deep learning[J]. Postharvest Biology and Technology, 2024, 218: 113157.

[13] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[14] Li H, Gu Z, He D, et al. A lightweight improved YOLOv5s model and its deployment for detecting pitaya fruits in daytime and nighttime light-supplement environments[J]. Computers and Electronics in Agriculture, 2024, 220: 108914.

[15] Gui Xiangquan, Qing Qingsong, Kong Lingwang. Small object detection algorithm based on improved YOLOv5s[J]. Computer Engineering and Design, 2024,45(04):1134-1140.

[16] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[17] Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13713-13722.

[18] Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11534-11542.

[19] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[20] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.

[21] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[22] Ross T Y, Dollár G. Focal loss for dense object detection[C]//proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2980-2988.

[23] Adarsh P, Rathi P, Kumar M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model[C]//2020 6th international conference on advanced computing and communication systems (ICACCS). IEEE, 2020: 687-694.

[24] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.