



A Lightweight One-Stage Defect Detection Network for Small Object Based on Dual Attention Mechanism and PAFPN

Yue Zhang¹, Fei Xie^{1*}, Lei Huang², Jianjun Shi^{3,4}, Jiale Yang¹ and Zongan Li¹

¹School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, China, ²School of Mechatronics Engineering, Nanjing Forestry University, Nanjing, China, ³Nanjing Zhongke Raycham Laser Technology Co. Ltd., Nanjing, China, ⁴School of Innovation and Entrepreneurship, Nanjing Institute of Technology Industrial Center, Nanjing, China

OPEN ACCESS

Edited by:

Lei Guo,
The University of Queensland,
Australia

Reviewed by:

Tuba Conka Yildiz,
Türkisch-Deutsche Universität, Turkey
Dongsheng Yu,
China University of Mining and
Technology, China
Haoqian Huang,
Hohai University, China

*Correspondence:

Fei Xie
xiefei@njnu.edu.cn

Specialty section:

This article was submitted to
Radiation Detectors and Imaging,
a section of the journal
Frontiers in Physics

Received: 11 May 2021

Accepted: 13 August 2021

Published: 06 October 2021

Citation:

Zhang Y, Xie F, Huang L, Shi J, Yang J
and Li Z (2021) A Lightweight One-
Stage Defect Detection Network for
Small Object Based on Dual Attention
Mechanism and PAFPN.
Front. Phys. 9:708097.
doi: 10.3389/fphy.2021.708097

Normally functioning and complete printed circuit board (PCB) can ensure the safety and reliability of electronic equipment. PCB defect detection is extremely important in the field of industrial inspection. For traditional methods of PCB inspection, such as contact detection, are likely to damage the PCB surface and have high rate of erroneous detection. In recent years, methods of detection through image processing and machine learning have gradually been put into use. However, PCB inspection is still an extremely challenging task due to the small defects and the complex background. To solve this problem, a lightweight one-stage defect detection network based on dual attention mechanism and Path Aggregation Feature Pyramid Network (PAFPN) has been proposed. At present, some methods of defect detection in industrial applications are often based on object detection algorithms in the field of deep learning. Through comparative experiments, compared with the Faster R-CNN and YOLO v3 which are usually used in the current industrial detection, the inference time of our method are reduced by 17.46 milliseconds (ms) and 4.75 ms, and the amount of model parameters is greatly reduced. It is only 4.42 M, which is more suitable for industrial fields and embedded development systems. Compared with the common one-stage object detection algorithm Fully Convolutional One-Stage Object Detection (FCOS), mean Average Precision (mAP) is increased by 9.1%, and the amount of model parameters has been reduced by 86.12%.

Keywords: defect detection, deep learning, dual attention mechanism, PAFPN, bounding box regression loss function

INTRODUCTION

As a carrier for connecting various electronic components, PCB is responsible for providing circuit connections and hardware support for the equipment. It is essential to detect defects on the surface of the PCB. In recent years, with the development of electronic products in the direction of light, thin and portable, PCBs have gradually developed in the orientation of high precision and high density, which has also posed a big challenge to detect defects of PCBs. Traditional PCB inspection generally uses methods such as manual inspection, electrical inspection, and optical inspection. Some of the inspection methods that make contact with the PCB surface [1] are likely to exert a bad effect on the surface components and their performance, while other inspection methods are highly dependent on

electrical and optical sensors, and existing problems have low efficiency of detection and high rate of erroneous detection. With the development of deep learning, object detection methods based on deep neural networks and computer vision are gradually applied to PCB defect detection [2]. In 2020, Saeed Khalilian proposed an approach based on denoising convolutional autoencoders to detect defective PCBs and determine the specific location. [3] Bing Hu [4] proposed a Faster R-CNN [5] detection algorithm based on ShuffleNetV2 [6] residual module and Guided Anchoring-Region Proposal Network (GA-RPN) optimization to detect several common types of PCB defects. However, Faster R-CNN has a large amount of model parameters and poor real-time performance due to its two-stage and anchor-based characteristics. In the same year, Ran Guangzai [7] et al. detected PCB defects based on the SSD [8] model, but the experiment only detected three types of defects and did not compare with other object detection methods based on deep neural networks. In 2021, Lan Zhuo [9] proposed a detection algorithm based on the YOLOv3 model, Li Yuting [10] proposed a detection algorithm based on the fusion of Hybrid-YOLOv2 and Faster R-CNN. Both methods have high detection accuracy, but they have not considered the memory consumption in actual applications.

In this regard, a lightweight one-stage defect detection network based on fusion attention mechanism and PAFPN [11] has been proposed.

In view of the actual problems in PCB defect detection, our method can realize real-time detection of common defects. Compared with the PCB defect detection algorithm based on deep learning proposed in the past, the model parameters and weight file size are greatly reduced, and the algorithm is more applicable for industrial production and actual deployment. The algorithm proposed in this paper has the following advantages:

1) First of all, a one-stage object detection model FCOS [11] has been used as the basic model. Compared with the two-stage object detection model, the one-stage object detection model reduces the proposal region detection module, the model structure is simplified, and the detection is more suitable to perform real-time detection, as shown in **Figure 1**. The overall

flow chart of our method proposed in this paper is shown in **Figure 2**. A lightweight Backbone neural network MobileNetV2 [12] has been applied to replace the commonly used Backbone: ResNet101 [13] in the FCOS, which greatly reduces model parameters and improves the real-time performance of the algorithm. At the same time, in order to ensure the feature extraction capabilities of the backbone neural network, dual attention mechanism module is added after the inverse residual module of MobileNetV2, by inferring the attention map in two different dimensions of channel and space, multiplying the attention map with the input feature map for adaptive feature optimization, thereby improving the feature extraction effect.

- 2) Secondly, the idea in Path Aggregation Network (PANet) has been applied to solve the problems caused by lightweight backbone. Feature fusion and enhancement applied in the Neck part of the overall model to further extract the features of smaller defects. Using PAFPN to replace the original Feature Pyramid Network (FPN) [14], shortening the information path and using low-level information to enhance FPN. The bottom-up feature enhancement is created, which can effectively enhance the feature and improve the feature extraction ability of the network.
- 3) In order to detect smaller size defect target accurately, the bounding box regression loss function in the existing algorithm has been optimized. The optimized intersection over union (IoU) function can consider the overlap rate, distance and ratio between the predicted box and the ground truth box, can directly minimize the distance, so that the convergence process is faster and the prediction bounding box regression becomes more stable.

PROPOSED METHOD

A Lightweight Feature Extraction Network Based on Dual Attention Mechanism

This paper propose an optimized lightweight neural network—MobileNetV2 as Backbone for feature extraction. As a lightweight Backbone, MobileNetV2 has a simpler network

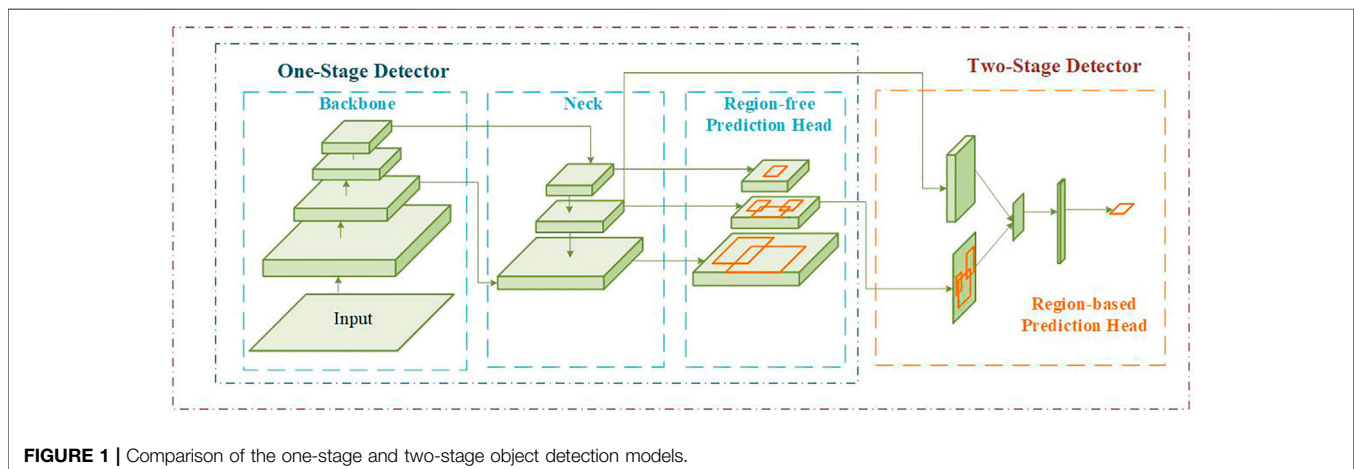


FIGURE 1 | Comparison of the one-stage and two-stage object detection models.

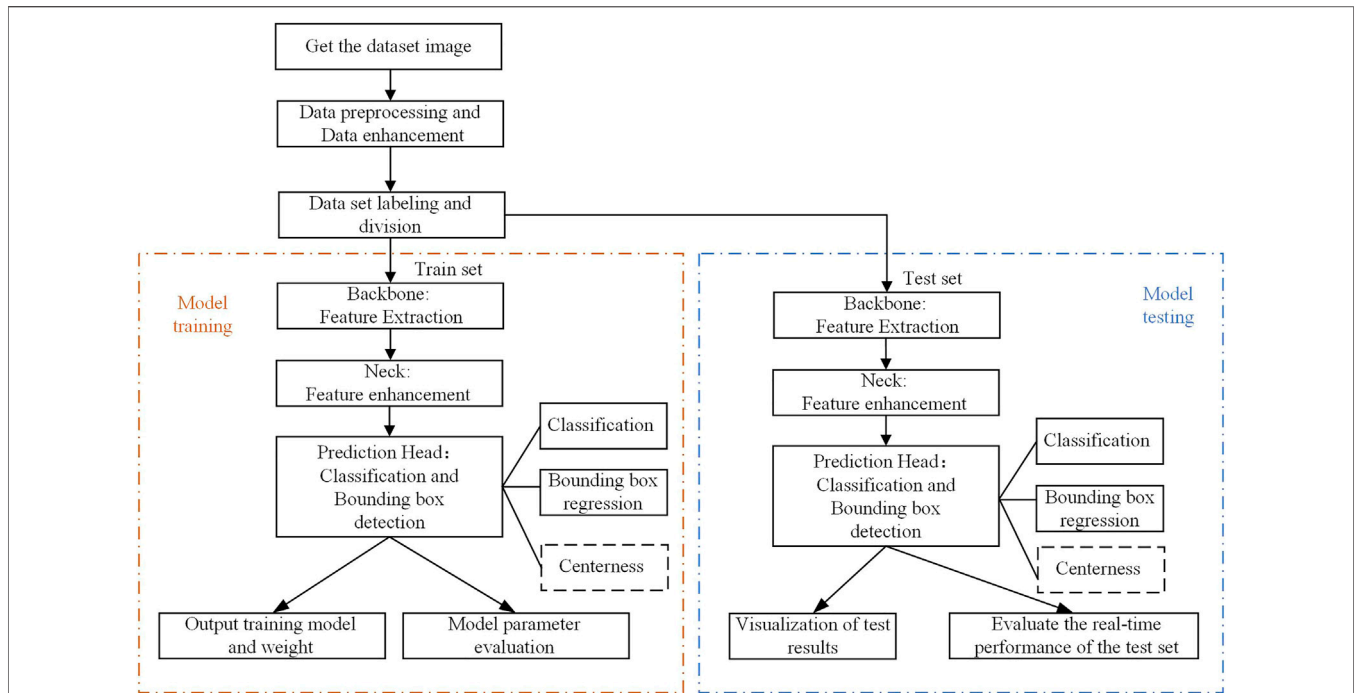


FIGURE 2 | The flow chart of the defect detection method proposed in this paper.

structure than conventional ResNet, which can effectively reduce the amount of parameters. Inverted residual module is proposed in MobileNetV2, which is the opposite of the classic residual module structure. First, the feature map channel is expanded by 1×1 convolution operation, and the number of features has been enriched to improve the accuracy. The specific structure of the inverted residual module in MobileNetV2 is shown in **Figure 3**.

In **Figure 3**, $Dwise3 \times 3$ represents Depthwise Convolution with a convolution kernel size of 3×3 . Each convolution kernel in Depthwise Convolution is responsible for one channel. After this convolution operation, the number of channels in the output feature map is exactly as same as the number of input channels. Compared to conventional convolution, Depthwise Convolution greatly reduces the amount of parameters and operation cost. The specific network structure of MobileNetV2 is shown in **Table 1**.

On the basis of MobileNetV2, a dual attention mechanism: Convolutional Block Attention Module (CBAM) has been used for optimization. The optimization scheme is shown in **Figure 4**. CBAM combines the spatial attention and channel attention mechanism, it can obtain better feature extraction results than the attention mechanism SENet (Squeeze and Excitation Networks) [15] which only focuses on the channel. Using the avg-pooling and max-pooling operations to process the feature map F , aggregate the spatial information of F , and generate two different spatial context descriptors: F_{avg}^c and F_{max}^c . MLP means a multi-layer perceptron which used as the shared network with one hidden layer.

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F))) \quad (1)$$

$$= \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c)))$$

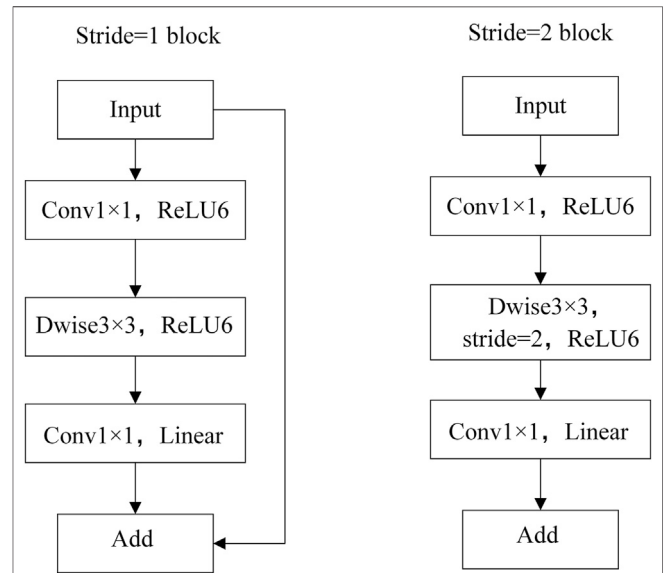


FIGURE 3 | The architecture of inverted residual module in MobileNetV2.

In **Eq. 1**, σ means sigmoid function, W_0 and W_1 denotes different shared weights.

$$M_s(F) = \sigma(\text{Conv}^{7 \times 7}([\text{AvgPool}(F); \text{Max Pool}(F)])) \quad (2)$$

$$= \sigma(\text{Conv}^{7 \times 7}([F_{avg}^s; F_{max}^s]))$$

In order to calculate spatial attention, avg-pooling and max-pooling have been operated along the channel axis and generate

TABLE 1 | The network structure of MobileNetV2.

Input	Operator	Expansion factor	Output channel	Number of blocks	Stride
$224^2 \times 3$	Conv2d	—	32	1	2
$112^2 \times 32$	Bottleneck	1	16	1	1
$112^2 \times 16$	Bottleneck	6	24	2	2
$56^2 \times 24$	Bottleneck	6	32	3	2
$28^2 \times 32$	Bottleneck	6	64	4	2
$14^2 \times 64$	Bottleneck	6	96	3	1
$14^2 \times 96$	Bottleneck	6	160	3	2
$7^2 \times 160$	Bottleneck	6	320	1	1
$7^2 \times 320$	Conv 1×1	—	1,280	1	1
$7^2 \times 1,280$	Avgpool 7×7	—	—	1	—
$1 \times 1 \times 1,280$	Conv 1×1	—	k	—	—

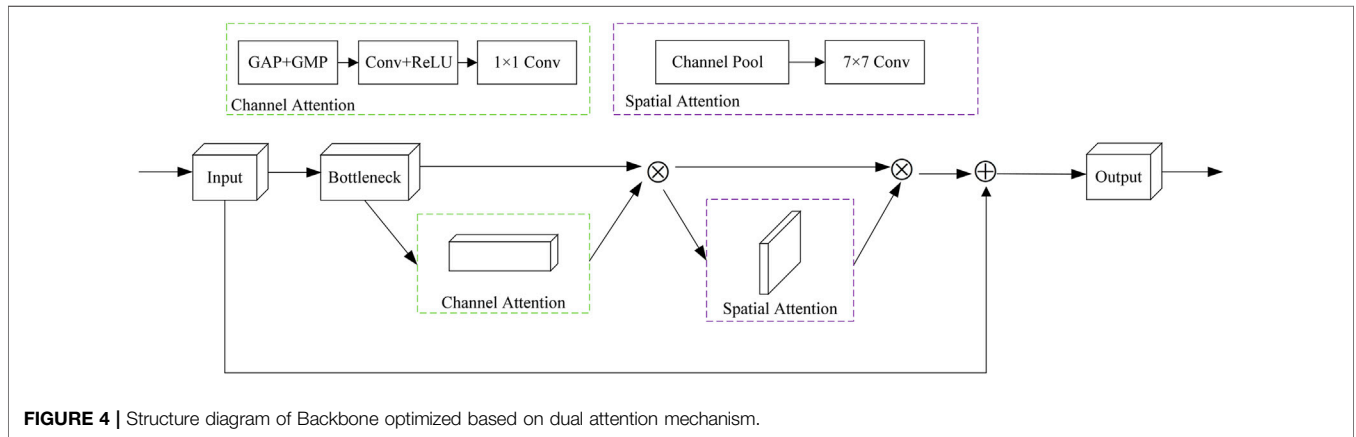


FIGURE 4 | Structure diagram of Backbone optimized based on dual attention mechanism.

two 2D maps: F_{avg}^s and F_{max}^s . Then concatenated the two feature maps together through a standard 7×7 convolution operation. In Eq. 2, $Conv^{7 \times 7}$ represents a convolution operation with the kernel size of 7×7 .

Optimized Feature Enhancement Module Based on Path Aggregation Feature Pyramid Network

In order to extract small features effectively, the Neck part of the model has been optimized. Common Neck module includes: FPN. FPN adds a top-down path for feature fusion on the basis of Backbone. FPN uses the high-resolution information of low-level features and high-level features information, achieves the prediction effect by fusing the features of these different layers. Drawing lessons from the ideas in PANet and add a top-down path on the basis of the FPN to enhance the feature information of the image, so that the overall network can obtain better detection results. After Backbone processing, output feature layer : F_1, F_2, F_4, F_6 can be obtained. First, the intermediate feature layer: P_1, P_2, P_4, P_6 are generated through conventional FPN processing. At the same time, the middle feature layer obtains high-resolution feature maps: $F_i, i \in \{1, 2, 4, 6\}$ through lateral connection, each feature layer F_i reduces the space size through a 3×3

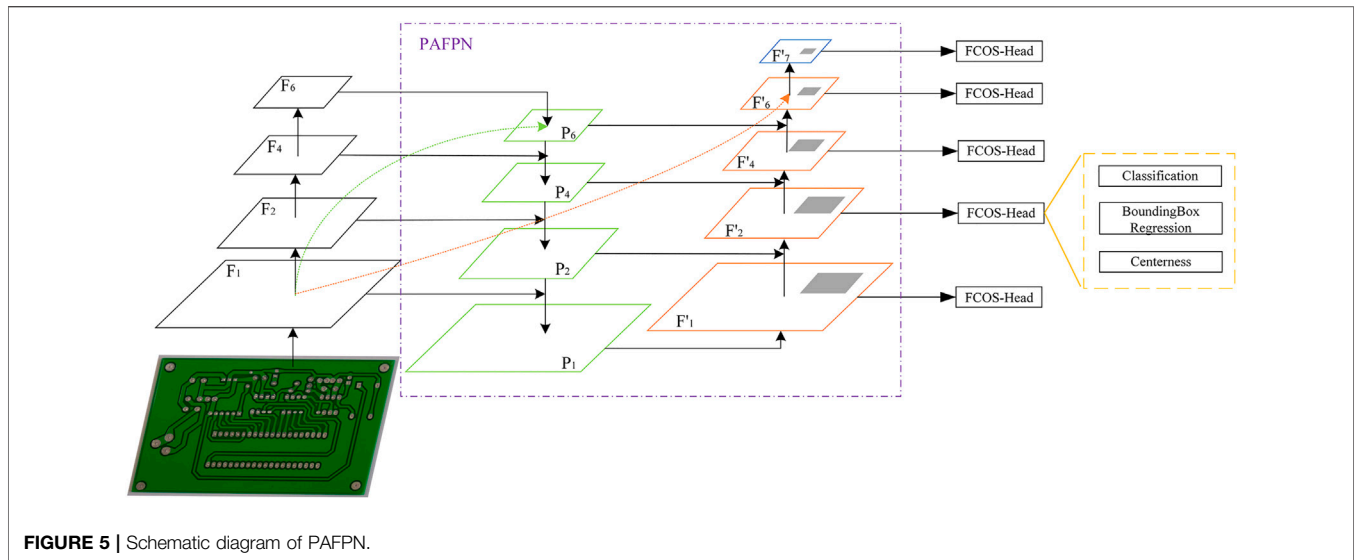
convolution with stride = 2, Then it is lateral connected with each element sum of the corresponding upper feature layer P_i to generate a new feature layer F' . The structure of PAFPN is shown in Figure 5.

On this basis, referring to the original structure of the FCOS object detection model. After the feature fusion of Neck in FCOS, five feature layers are sent to the FCOS detection head.

As shown in Figure 5, a feature layer F'_7 can be obtained after F'_6 by additional extraction through convolution, and added a ReLU operation before this convolution, which can effectively improve the detection effect.

One-Stage Object Detector Head Based on Optimized IoU Function

Common object detection algorithms are divided into two types, one-stage and two-stage, the specific comparison is shown in Figure 1. The one-stage object detection algorithm obtains the prediction result directly from the feature map after feature extraction and feature enhancement. The two-stage object detection algorithm additionally generates a proposal region and makes predictions based on this region. Two-stage object detection algorithms, such as Fast R-CNN, Faster R-CNN, etc., often have better detection accuracy, but their model complexity is higher, and the detection speed is slow.



FCOS is a one-stage anchor-free object detector. Compared with other object detectors, FCOS has a clear structure and fewer model parameters, which is convenient for optimization. The FCOS detection head predicts the bounding box by obtaining a 4D vector on the feature map. The 4D vector feature includes the horizontal distance from the center point on the feature map to the four sides of the ground truth bounding box. The FCOS detection head includes three branches: bounding box regression, classification and centerness branch. As shown in Eq. 3, L_{bbox} represents the loss of bounding box, L_{cls} means the classification branch, which adopts the Focal loss, and the centerness loss $L_{centerness}$ adopts the cross-entropy loss function.

The loss function L of the optimization algorithm proposed in this paper is:

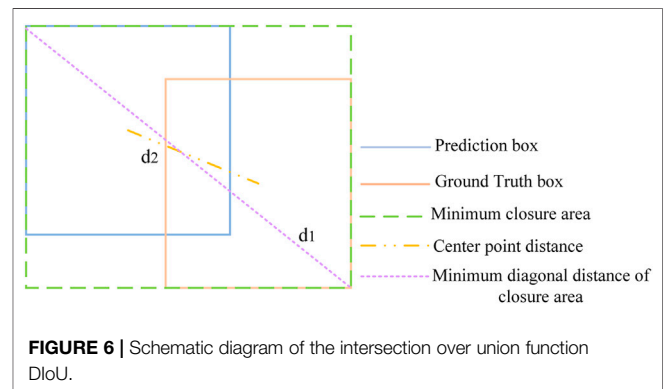
$$L = L_{cls} + L_{centerness} + L_{bbox} \tag{3}$$

On this basis, the bounding box regression loss function has been optimized and a Distance Intersection over Union (DIoU) [16] has been adopted. Compared with the currently widely used IoU function, DIoU takes the overlap rate and scale into account. Through the comparison of the previous experiments on the public COCO data set, although some researchers have proposed that the Complete Intersection over Union (CIoU) loss function which add penalty items on the basis of DIoU loss function is better, the performance improvement of CIoU for small defect detection is not as good as the DIoU loss function, the effect is better only in the detection of medium and large objects. In view of the fact that there are many small objects on the PCB, the DIoU loss function has been used. The schematic diagram of DIoU is shown in Figure 6.

The calculation process of DIoU is as follows:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{d_1^2} \tag{4}$$

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{5}$$



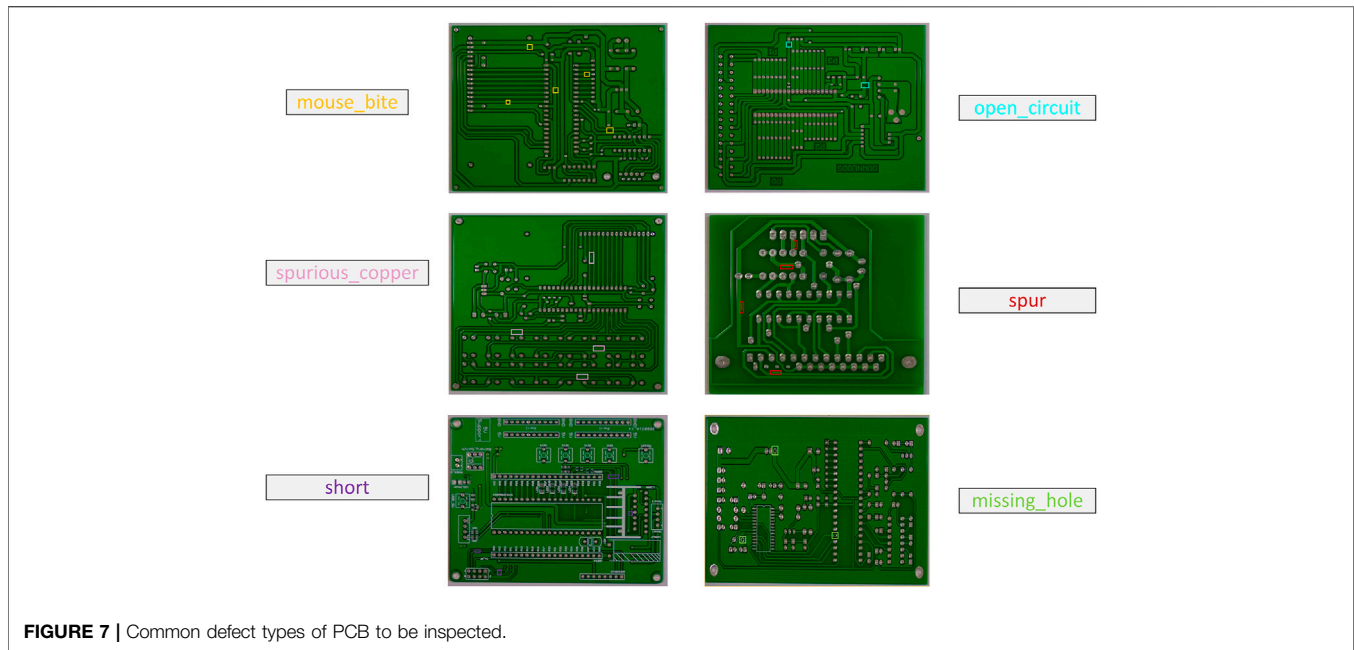
In this formula, B represents the prediction bounding box, B^{gt} represents the ground-truth bounding box, b and b^{gt} represent the center point positions of the prediction bounding box and the ground-truth bounding box, d_1 represents the diagonal distance of the minimum closure area that contains both the prediction bounding box and ground-truth bounding box, and d_2 represents the calculation of the Euclidean distance between these two center points, $d_2 = \rho^2(b, b^{gt})$, as shown in Figure 6.

EXPERIMENTS AND ANALYSIS

Dataset Processing and Training

Due to the limitation of open access data sets for PCB defects, a PCB data set with six common types of defects has been selected. The six types of defects are: missing hole, mouse bite, open circuit, short, spurious copper, and spur, as shown in Figure 7.

In order to enhance the detection effect, data enhancement processing has been applied, by changing the illumination and contrast of the same image to simulate the complex environment, and finally generate a data set, train set and test set have been divided according to the ratio of 7:3.



Finally, it contains 1,455 images in the train set and 624 images in the test set. Using the image labeling tool LabelMe to label the images according to the format of the COCO data set, and generate the corresponding JSON file.

Due to the modification of model structure and the lack of a corresponding pre-training model, the model trained and tested on the existing PCB data set, without using transfer learning method. Through experiments, it has confirmed that the method proposed in this paper also has a better improvement in detection accuracy and real-time performance compared with the classic algorithm that uses the pre-training model.

The neural network models proposed in this paper trained and tested based on MMDetection. The relevant hardware configuration is as follows:

The experimental platform of this paper is built under Ubuntu 18.04 system. The experimental environment configuration is: Python3.7 + PyTorch1.5.1-GPU + CUDA 10.1 + CUDNN + mmcv 1.2.4 + mmdet 2.8.0.

Evaluation Standards

In order to detect the effect of model, mean Average Precision (mAP) has been used as a performance evaluation index. mAP can fully express the classifier and detection performance of the defect detection model. The calculation of average precision included two indicators: accuracy and recall. The accuracy and recall can be expressed by Eq. 6 and Eq. 7:

$$p = \frac{TP}{TP + FP} \quad (6)$$

$$r = \frac{TP}{TP + FN} \quad (7)$$

In these formulas, p represents the accuracy rate, r represents the recall rate, and TP represents the number of

correctly divided positive samples; FP represents the number of wrongly divided positive samples; FN represents the number of wrongly divided negative samples, AP is the average precision. AP_s , AP_m , AP_l represents the average precision of three different sizes targets: small, medium and large. The average precision can be expressed in Eq. 8. In general, the higher the average accuracy value, the better the classifier performance. Classes represents the types of all detected objects, $Num(Classes)$ indicates the number of categories, the formula of mAP is as follows. mAP is the different objects' AP sum divided by the number of object categories.

$$AP = \int_0^1 p(r)dr \quad (8)$$

$$mAP = \frac{\sum_{Classes} AP}{Num(Classes)} \quad (9)$$

Tests and Results

The selection of the training parameters will affect the model performance. The model structure proposed in this paper is improved based on FCOS, and the overall model is constructed based on MMDetection. Therefore, the training parameters has been modified on the basis of FCOS and MMDetection. The parameter data selection is shown in Table 2.

In order to verify the effectiveness of the model proposed in this paper, a set of ablation experiments are established to compare the effects of different common optimization schemes on the average precision value. At the same time, the Adam optimizer has been selected to replace the default SGD optimizer and add the GradNorm module. The gradient equalization operation has a good effect on the improvement

TABLE 2 | The selection of parameter data.

Parameter name	Parameter Value
NMS iou_threshold	0.5
Loss_cls	FocalLoss
Loss_centerness	CrossEntropyLoss
FCOS Inference Threshold	0.003
NMS Threshold	0.4
Basic learning rate	0.0025
Weight Decay	0.0001
Warmup_iters	500
Warmup_ratio	0.33
Max Iter	50,000

of the average accuracy value. The specific test results are shown in **Table 3** and **Figure 8**.

FCOS often uses ResNet50 and ResNet101 as the feature extraction network. Compared with the lightweight neural

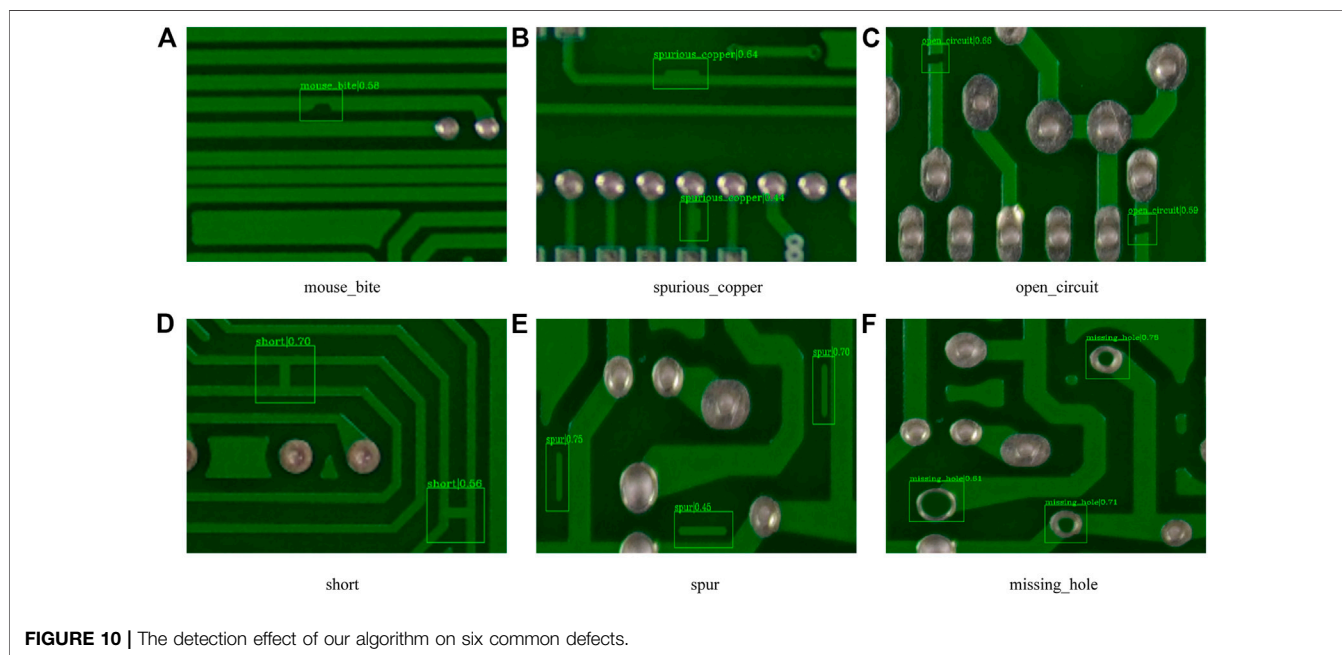
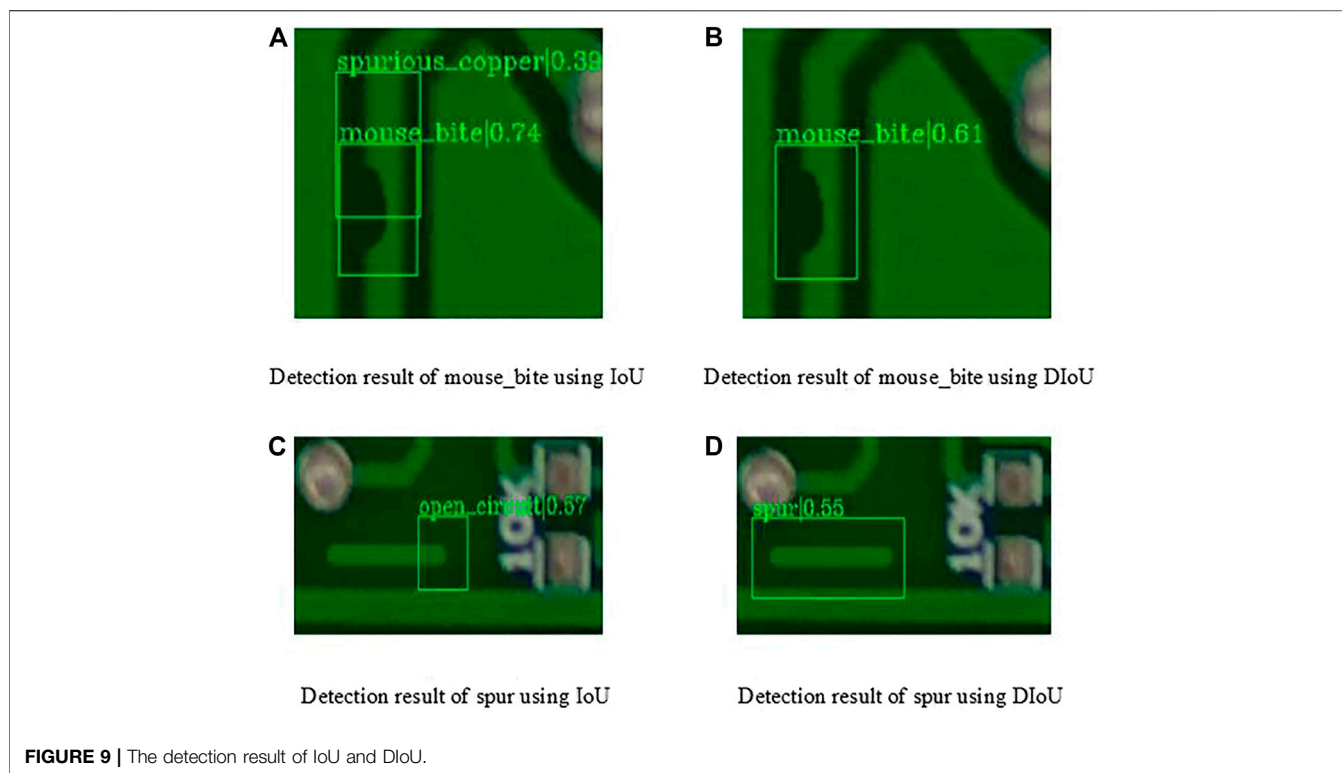
network MobileNetV2, ResNet has a better feature extraction effect, but the neural network is complex, the process of training takes up more memory and time. Through comparison in **Figure 8** and **Table 3**, it can be found that the test mAP result of ResNet50 and MobileNetV2 are almost unanimous due to the lack of pre-training model. While the model using ResNet101 as Backbone has a higher mAP, but the deep network structure also means that the generated weight models and parameter models has a larger memory footprint. Using the dual attention mechanism CBAM to optimize MobileNetV2 in order to achieve the same effect as ResNet101. After replacing the traditional FPN module with PAFPN, the feature extraction effect is further enhanced. Compared with the original model using MobileNetV2 as Backbone, mAP increased by 1.2%.

In the bounding box regression branch of the FCOS detection head, the original IoU Loss has been replaced with DIoU to better detect objects with smaller sizes, and mAP reached 39.3%. The

TABLE 3 | Defect detection results of different methods based on FCOS model.

Methods	mAP (bbox)/%	AP _s (bbox)/%	AP _m (bbox)/%	AP _l (bbox)/%
FCOS + ResNet50	35.2	20.9	35.8	30.9
FCOS + ResNet101	36.0	20.5	36.6	36.0
FCOS + MobileNetV2	35.3	22.4	35.7	26.8
FCOS + CBAMMobileNetV2	36.0	25.5	36.6	27.9
FCOS + CBAMMobileNetV2+PAN	36.4	26.9	36.6	28.1
FCOS + CBAMMobileNetV2+PAN + DIoU	39.3	27.2	39.4	30.3
FCOS + CBAMMobileNetV2+PAN + DIoU + Adam (Our Method)	44.3	28.3	42.5	33.3





comparison of visualization detection results is shown in **Figure 9**, Using the DIoU loss function can better mark the position of the detection bounding box, avoiding problems such as false detection and overlapping bounding boxes.

After using the Adam optimizer to replace the SGD optimizer, the detection effect has further improvement, mAP is 44.3%. Compared with the original FCOS model using ResNet50, mAP is

increased by 9.1%. The detection results of six common defects are shown in **Figure 10**.

In **Table 4**, comparing our method with the object detection algorithms which commonly used in the industry: Faster R-CNN, YOLO v3 [17] and YOLO v3-Tiny [18]. Compared with Faster R-CNN, our method has a lower mAP, but it can maintain the detection speed while having better accuracy. Compared with

TABLE 4 | Comparison of different defect detection models.

Methods	mAP (bbox)/%	Inference time (ms)	Model parameters (MB)	Size of weights file (MB)
Faster R-CNN	49.50	57.14	60.14	460.00
YOLO v3	42.50	44.43	61.55	124.30
YOLO v3-Tiny	31.50	37.60	8.80	17.80
Our Method	44.30	39.68	4.42	34.00

YOLO v3, the method proposed in this manuscript has a better average precision, mAP is increased by 1.8%, and the model parameter is only about one-fourteenth of YOLO v3. Compared with YOLO v3-Tiny, the mAP of our method is increased by 12.8%. Although the weight file of YOLO v3-Tiny is smaller and the inference time is only 2.08 milliseconds, faster than the method proposed in this paper, but the model parameter size of our algorithm in this paper is only half of YOLO v3-Tiny. Compared with these common models, the method proposed in this paper has fewer model parameters, be more suitable for industrial applications and more convenient for porting to embedded development equipment.

CONCLUSION

This paper propose a lightweight defect detection network based on dual attention mechanism and PAFPN optimization. On the basis of keeping the network model's low memory usage and strong real-time performance, it has improved its ability to detect small-size defects.

Compared original FCOS model, mAP of the model proposed in this paper is greatly improved, and it is also increased by 1.8% compared with the commonly used YOLO V3 model in industrial scenarios. The model parameters are about only one-fifteenth of those traditional methods, which is more suitable for application to actual PCB defect detection.

The inference time of our method still has space for improvement when compared with YOLOv3-Tiny. The subsequent work will optimize the feature enhancement module on the basis of maintaining the detection accuracy and streamline the model structure, thus reduce the detection time.

REFERENCES

- Raj A, and Sajeena A. *Defects Detection in PCB Using Image Processing for Industrial Applications*. In: 2018 Second International Conference on Inventive Communication and Computational Technologies. Hyderabad: ICICCT (2018) p. 1077–9. doi:10.1109/ICICCT.2018.8473285
- Ma J. *Defect detection and recognition of bare PCB based on computer vision, 2017 36th Chinese Control Conference*. Dalian: IEEE (2017) p. 11023–8. doi:10.23919/ChiCC.2017.8029117
- Khalilian S, Hallaj Y, Balouchestani A, Karshenas H, and Mohammadi A. *PCB Defect Detection Using Denoising Convolutional Autoencoders*. 2020 International Conference on Machine Vision and Image Processing (MVIP); 2020 Feb 18–20. Iran: IEEE (2020) 2020:1–5. doi:10.1109/MVIP49855.2020.9187485
- Hu B, and Wang J. *Detection of PCB Surface Defects with Improved Faster-RCNN and Feature Pyramid Network*. *IEEE Access* (2020) 8:108335–45. doi:10.1109/ACCESS.2020.3001349
- Ren S, He K, Girshick R, and Sun J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. *IEEE Trans Pattern Anal Mach Intell* (2017) 39(6):1137–49. doi:10.1109/TPAMI.2016.2577031
- Ma N, Zhang X, Zheng HT, and Sun J. *ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design*. In: V Ferrari, M Hebert, C Sminchisescu, and Y Weiss, editors. *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, Vol. 11218. Cham: Springer (2018) doi:10.1007/978-3-030-01264-9_8
- Ran G, Lei X, Li D, and Guo Z. *Research on PCB Defect Detection Using Deep Convolutional Neural Network*. In: 2020 5th International Conference on Mechanical, Control and Computer Engineering

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

Conceptualization, YZ and FX; methodology, YZ; software, FX and JS; validation, YZ, FX, and JS; formal analysis, YZ, FX, and JS; investigation, YZ and FX; resources, FX, JS, and TZ; data curation, YZ and JY; writing–original draft preparation, YZ; writing–review and editing, YZ, FX, and JY; visualization, YZ; supervision, FX and ZL; All authors have read and agreed to the published version of the manuscript.

FUNDING

This work was partially supported by the National Key Research and Development Program of China (Grant No. 2017YFB1103200), the Scientific and technological achievements transformation project of Jiangsu Province (BA2020004), the National Natural Science Foundation of China (Grant Nos. 41974033 and 61803208), 2020 Industrial Transformation and Upgrading Project of Industry and Information Technology Department of Jiangsu Province, Postgraduate Research and Practice Innovation Program of Jiangsu Province (SJCX21_0578), Bidding project for breakthroughs in key technologies of advantageous industries in Nanjing (2018003).

- (ICMCCE); 2020 Dec 25–27. Harbin: IEEE (2020). 1310–4. doi:10.1109/ICMCCE51767.2020.00287
8. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, and Fu C-Y. SSD: single shot multibox detector. In: 2016 14th European Conference on Computer Vision (ECCV). Amsterdam: Springer (2016) 9905:21–37. doi:10.1007/978-3-319-46448-0_2
 9. Lan Z, Hong Y, and Li Y. An improved YOLOv3 method for PCB surface defect detection. In: 2021 IEEE International Conference on Power Electronics Computer Applications (ICPECA). Shenyang: IEEE (2021) 2021:1009–15. doi:10.1109/ICPECA51329.2021.9362675
 10. Li Y-T, Kuo P, and Guo J-I. Automatic Industry PCB Board DIP Process Defect Detection System Based on Deep Ensemble Self-Adaption Method. *IEEE Trans Compon., Packag Manuf Technol* (2021) 11(2):312–23. Feb. 2021. doi:10.1109/TCPMT.2020.3047089
 11. Liu S, Qi L, Qin H-F, Shi J-P, and Jia J-Y. Path Aggregation Network for Instance Segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; Salt Lake City, UT; June 18–23, 2018. United States: IEEE (2018). 8759–8768. doi:10.1109/CVPR.2018.00913
 12. Tian Z, Shen C, and Chen H. FCOS: Fully Convolutional One-Stage Object Detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE (2019) p. 9626–35. doi:10.1109/iccv.2019.009722019
 13. Sandler M, Howard A, Zhu M, Zhmoginov A, and Chen L-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE/CVF Conf Comp Vis Pattern Recognition* (2018) 2018:4510–20. doi:10.1109/CVPR.2018.00474
 14. Ji L-P, Fu C-Q, and Sun W-Q. Soft Fault Diagnosis of Analog Circuits Based on a ResNet With Circuit Spectrum Map. *IEEE Transactions on Circuits and Systems I* (2021) 68p. 2841–9. doi:10.1109/TCSI.2021.3076282
 15. Zhao B-J, Zhao B-Y, Tang L-B, Wang W-Z, and Wu C. Multi-scale object detection by top-down and bottom-up feature pyramid network. *Journal of Systems Engineering and Electronics* (2019) 30 1–12. doi:10.1109/cvpr.2017.1062017
 16. Hu J, Shen L, and Albanie S. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020) 42: 2011–23. doi:10.1109/TPAMI.2019.2913372
 17. Zheng Z-H, Wang P, and Liu W. *Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression* (2019) Network in network. arXiv preprint arXiv: 1911.08287. Available at: <https://arxiv.org/abs/1911.08287>.
 18. Kong W-Z, Hong J-C, Jia M-Y, Yao J-L, Cong W-H, and Hu H. YOLOv3-DPPIN: A Dual-Path Feature Fusion Neural Network for Robust Real-Time Sonar Target Detection. *IEEE Sens* (2020) 20:3745–3756. doi:10.1109/JSEN.2019.2960796
 19. Adarsh P, Rathi P, and Kumar M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS); Coimbatore, India; 2020 Mar 6-7: IEEE (2020):687–94. doi:10.1109/ICACCS48705.2020.9074315

Conflict of Interest: JS was employed by Nanjing Zhongke Raycham Laser Technology Co. Ltd.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Zhang, Xie, Huang, Shi, Yang and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.