

A Method of Defect Detection for Focal Hard Samples PCB Based on Extended FPN Model

Cui-jin Li, Zhong Qu[✉], Shi-yan Wang, Kang-hua Bao[✉], and Sheng-ye Wang

Abstract—Suffering from the diversity, complexity, and miniaturization of printed circuit board (PCB) defects, traditional detection methods are difficult to detect. Despite object detection has made significant advances based on deep neural networks, it remains a challenge to focus on small objects. We address this challenge by allowing multiscale fusion. We introduce a PCB defect detection algorithm based on extended feature pyramid network model in this article. The backbone is constructed by part of ResNet-101, in order to accurately locate and identify small objects, this article constructs a feature layer, which integrates high-level semantic information and low-level geometric information. Based on feature pyramid networks (FPN) network structure, using 1×1 convolution lateral fusion of the previous semantic information, the fused features use 3×3 convolution to obtain the final feature layer. The problem that PCB defects are difficult to classify is considered, the focal loss function is introduced. To reduce over-fitting in the training process, the original data are enhanced using image clipping and rotation. Through the quantitative analysis on PCB defect dataset, these results are the best to be used in fused low-level feature layer for detection of the mean average precision (mAP). This is 96.2% on the public PCB dataset, which is surpassing the state-of-the-art methods.

Index Terms—Dataset enhancement, faster R-CNN, feature pyramid networks, focal hard samples.

I. INTRODUCTION

PRINTED circuit board (PCB) is the carrier for connecting various electronic components, which is responsible for providing circuit connection and hardware support for the equipment. The rationality of its design and standardization of

Manuscript received August 6, 2021; revised September 28, 2021 and November 27, 2021; accepted December 6, 2021. Date of publication December 20, 2021; date of current version February 23, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62176034 and Grant 61905033, in part by the National Natural Science Foundation of Chongqing under Grant cstc2021jcyj-msxmX0941, in part by the Scientific and Technological Research Program of Chongqing Municipal Education Commission under Grant KJQN202101907, and in part by the Doctoral Talent Training Project of Chongqing University of Posts and Telecommunications under Grant BYJS202007. Recommended for publication by Associate Editor X. Chen upon evaluation of reviewers' comments. (*Corresponding authors:* Zhong Qu; Shi-yan Wang.)

Cui-jin Li is with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the College of Electronic Information, Chongqing Institute of Engineering, Chongqing 400056, China.

Zhong Qu, Shi-yan Wang, and Sheng-ye Wang are with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

Kang-hua Bao is with the Institute of Information Technology, Southwest Computer Company Ltd., Chongqing 400056, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCMT.2021.3136823>.

Digital Object Identifier 10.1109/TCMT.2021.3136823

production play a vital role in the stable and reliable operation of the equipment [1]. However, due to the current production process and the complexity of integrated circuits [2], some PCBs have manufacturing defects, which makes the equipment unable to operate normally [3]. Therefore, PCB defect detection has become a research hotspot in the field of computer vision. Automatic optical inspection (AOI) is a noncontact detection method based on machine vision and image-processing algorithm [4], [5], which is widely used in PCB defect detection [6].

Ejiri *et al.* [7] proposed to use 2-D nonlinear logic filter to detect PCB defects, which can be regarded as the research form of early AOI technology. Moganti *et al.* [1] applied machine learning algorithm to PCB defect detection. Ibrahim *et al.* [8] proposes a method to eliminate, if possible, or to reduce as much as possible such noise during the computation of defect detection. Kaur *et al.* [9] proposed using image differential operation to realize effective detection of defects such as short, miss hole, spurious copper, spur, mouse bite, open circuit in PCB bare board. Kumar *et al.* [10] used image enhancement and standard database to generate standard template to detect PCB defects. With the development of deep learning image-detection technology [11], especially in automatic feature extraction and end-to-end detection, deep learning network can accurately and quickly recognize the target from the image [12], and has strong robustness. Deep learning-based image detection technology has been paid more and more attention in the field of PCB defect detection [13].

Faghil Roohi *et al.* [14] proposed a microdefect detection method based on deep learning; they designed three small deep convolutional neural network (DCNN) networks on the basis of neural network for micro feature extraction. Zhang [15] developed a structural damage detection method based on region-based convolutional neural network (Faster R-CNN) [16]. Ding *et al.* [17] proposed PCB defect detection network tiny defect detection network (TDD-Net) on the basis of Faster R-CNN; they used ResNet-101 [18] as the backbone network, carried out feature fusion through feature pyramid networks (FPNs), and designed anchors size($15^2, 25^2, 40^2, 60^2, 80^2$) for PCB defects. Finally, they achieved good detection results on public PCB defect dataset. Li [19], Li *et al.* [20] proposed an effective self-adaption method that collects “exception data” like the samples with which the artificial intelligent (AI) model made mistakes from the automated optical inspection inference edge to the training server, retraining with exceptions on the server and deploying

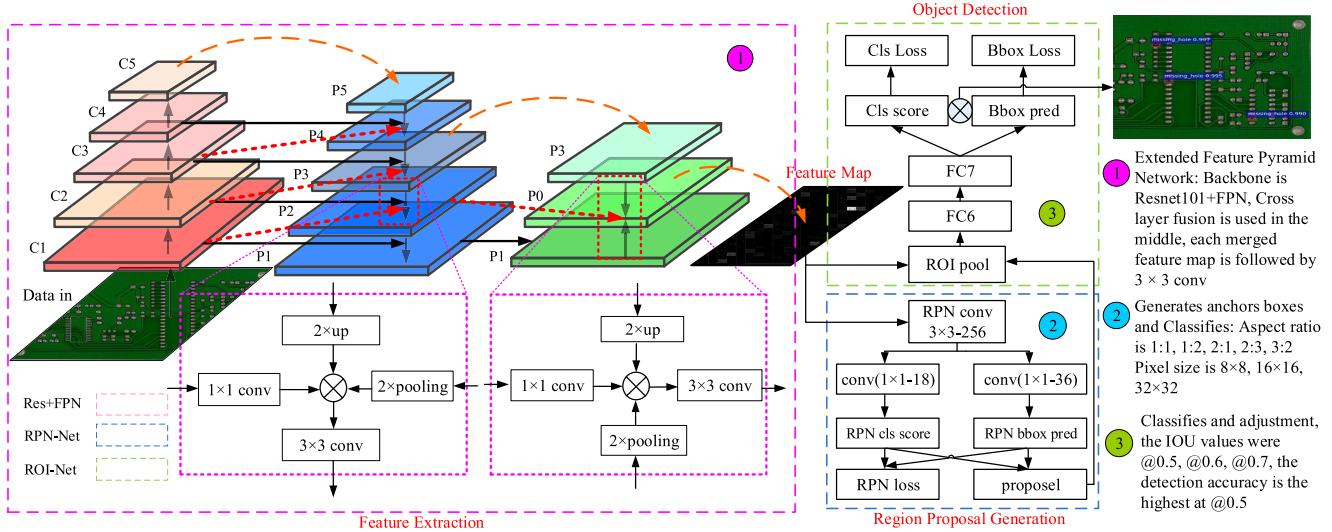


Fig. 1. Architecture of Extended Fpn. The five layers of ResNet-101 are used to build a bottom-up network, and the up-sampling feature map and the top-down path shallow feature map are merged.

back to the edge. Tao *et al.* [21] proposed A novel cascaded autoencoder (CASAE) architecture, which is designed for segmenting and localizing defects.

Although the above methods have achieved good results in PCB defect detection, there are still errors and omissions. Aiming at the diversity, complexity, and miniaturization of PCB defects, we propose a PCB defect-detection algorithm based on extended FPN model in this article, which focuses on hard samples to improve the target detection accuracy and network generalization ability. The overall structure of the model is shown in Fig. 1. The main contributions of this article are as follows.

1) ResNet-101 is used to build a Bottom-up network, and the upsampling feature map and the top-down path shallow feature map are merged. To reduce the aliasing effect caused by up sampling, each merged feature map is followed by a 3×3 convolution. By upsampling the P3 feature is rich in semantic information and by down-sampling the P1 feature is rich in geometric information. Finally, P1, P2, P3 is merged together by pixel-wise addition, and then a 3×3 convolution operation is followed to form the final output P0.

2) Considering the classification problem of hard samples [22] in PCB defect detection, we introduce the loss function, which focuses on hard samples; we increase the weight of hard samples in the loss function, and focus on learning the features of hard samples, so as to effectively improve the accuracy of image detection.

3) Considering the problem of over fitting in the training process, we use image clipping and rotation to enhance the original data, crop the input image into a uniform scale of 448×448 pixels, and rotate it by 180° , 90° , 60° , 45° , and 30° .

The rest of this article is arranged as follows. In Section II, the typical network structure of FPN is introduced. The extended FPN model and the loss function of focusing on hard samples are shown in Section III. Section IV introduces the data augmentation of PCB defect dataset. Section V introduces the overall training process and give the experimental results

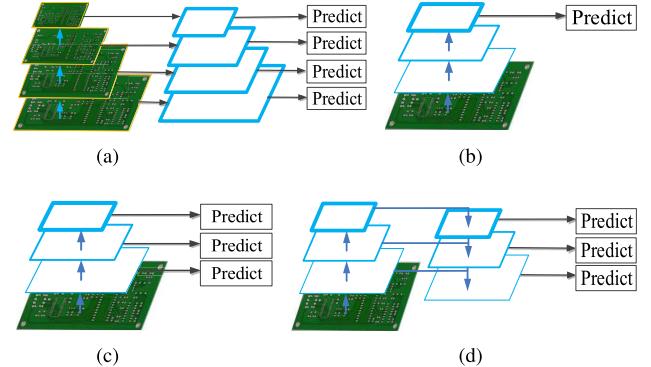


Fig. 2. Feature pyramid structure. (a) Featurized image pyramid. (b) Single feature map. (c) Pyramidal feature hierarchy. (d) Feature pyramid network.

to evaluate the proposed method. Summary and discussion are made in Section VI.

II. RELATED WORK

A. Feature Pyramid Networks

Aiming at the problem of multiscale target detection, a common scheme is to generate different size of feature maps according to the original image to construct image pyramid [23], and detect different sizes of targets on the corresponding feature maps. As shown in Fig. 2(a), the detection accuracy of this scheme is high, but it is not conducive to practical application due to the large amount of calculation [24]. Most convolutional neural networks use feature map in the last layer for detection, such as Fig. 2(b). This method is more efficient, but the resolution of the feature map in the last layer is low, the geometric features are missing seriously [25], and the detection performance for small targets is poor [26]. As shown in Fig. 2(c), the feature pyramid performs deep convolution on the original image, and uses the feature maps at different stages to detect targets of different sizes. This method is efficient and effective [27]. However, because features are

directly obtained from the shallow layer of the network, some semantic information are missing so the network is not robust enough.

FPN [28] uses the pyramid form of CNN hierarchical features [29] to generate feature pyramids with strong semantic information at all scales. FPN uses bottom-up path, top-down path, and lateral connection. In this way, a feature pyramid with strong semantic information at all scales can be quickly constructed from a single input image at a single scale, as shown in Fig. 2(d).

III. ALGORITHM BASED ON EXTENDED FPN MODEL

By comparing the performance of a variety of convolutional neural networks, it is found that ResNet-101 is more efficient under the same scale of parameters, so it is suitable for PCB defect detection with small dataset and high accuracy requirements. Therefore, ResNet-101 is selected as the backbone network in this article.

PCB images [17] are different from those in common open source object detection datasets [such as Common Objects in Context (COCO) [30]]. The resolution of PCB images captured by industrial cameras is usually very high (such as 2854×2357), and the defect area of the image only accounts for a small proportion of the whole image. When comparing different defect areas, the largest defect area may be several times of the smallest defect area. Therefore, PCB bare board defect detection is a problem that needs to detect a large number of small targets and detect a variety of different size targets at the same time. In this article, based on FPN, the spatial resolution is enlarged to twice of its size by two successive upsampling operation to construct the top-down feature map, and the upsampled map and the bottom-up feature map are combined according to the way of element-wise addition. Then, the output features of the last layer of feature extraction network are used to detect, and a feature layer integrating high-level semantic information and low-level geometric information is constructed, which is easier to be applied to detect PCB bare board defect target.

Considering the classification problem of hard samples in PCB defect detection, we introduce the focus loss function of difficult samples to increase the weight of difficult samples in the loss function, so as to focus on learning the characteristics of difficult samples and effectively improve the image detection accuracy.

A. Extended FPN

Because the PCB defect target is small, in order to improve the detection accuracy and reduce the amount of calculation and parameters, ResNet-101 is used to build the bottom-up network, as shown in Fig. 3. C5 generates the feature map P5 with the lowest resolution through a 1×1 convolution layer, and then up samples twice, and merges the Up-sample feature map with the Top-down path shallow feature map. In order to reduce the aliasing effect caused by up sampling, each merged feature graph generates the final feature graph P5, P4, P3, P2, P1 through a 3×3 convolution, because 3×3 convolution can increase the nonlinear expression ability of the network

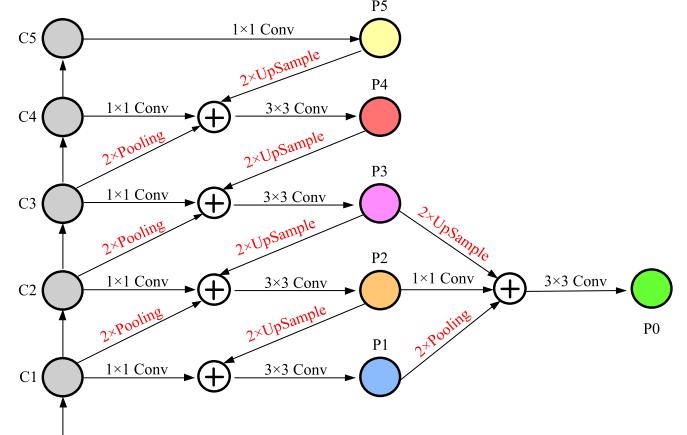


Fig. 3. Extended FPN structure, feature graph $\{P_5, P_4, P_3, P_2, P_1\}$ are corresponding to $\{C_5, C_4, C_3, C_2, C_1\}$.

and reduce parameters, corresponding to C_5, C_4, C_3, C_2, C_1 has the same space size. By sampling P_3 feature layer is rich in semantic information ($2 \times$ upsample) and P_1 feature layer is rich in geometric information ($2 \times$ pooling), it is fused with P_2 feature layer by pixel-wise addition. Finally, the final feature layer P_0 is obtained by a 3×3 convolution of the fused features. The image size of each layer of the network is shown in Fig. 4.

B. Improvement of Focal Loss

In the process of sample training, we divide the difficult samples of PCB defect detection into difficult classification samples and difficult regression samples, which adopts the idea of focusing on the loss function, and increases the weight of the difficult samples in the loss function to focus on learning the characteristics of the difficult samples [31], [32]. As shown in (1), the total loss is the weighted sum of classified loss and regression loss that mainly composed of four parts: 1) binary classification loss of region proposal network (RPN), whether anchor is foreground or background; 2) position regression loss of RPN, anchor position fine tuning; 3) classification loss of region of interest (ROI), ROI belongs category; and 4) position regression loss of ROI, continue to fine tuning the ROI position. The total loss is obtained after N iterations, and the pseudo code is shown in Algorithm 1

$$\begin{aligned} L_{\text{loss}} = & \lambda_1(L_{\text{cls}}(\text{RPN}) + L_{\text{cls}}(\text{ROI})) \\ & + \lambda_2(L_{\text{reg}}(\text{RPN}) + L_{\text{reg}}(\text{ROI})) \end{aligned} \quad (1)$$

where λ_1, λ_2 are the weights of classification and regression loss function, respectively, the value is $[0,1]$, The experiments show that the best results are $\lambda_1 = 0.95$ and $\lambda_2 = 0.95$.

In order to solve the problem of class imbalance between positive and negative samples, a weighting factor α is introduced, as shown in (2). The positive and negative samples are balanced through α , but it does not differentiate between easy sample and hard sample. Therefore, this article uses the Focal-loss function to reduce the loss of easy samples and increase the weight of hard samples in the loss function. The

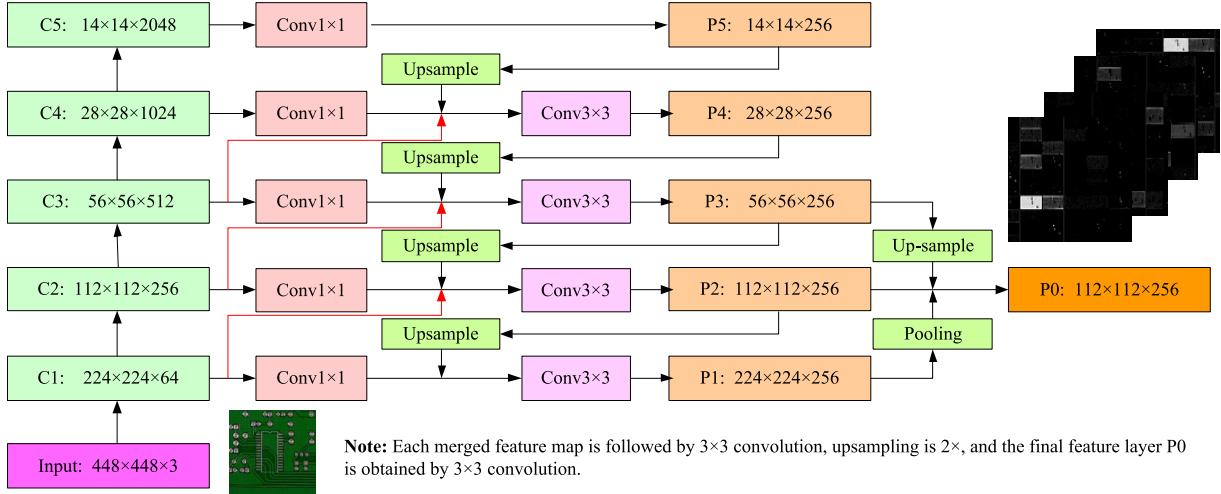


Fig. 4. Pixel values of extended FPN structure. The input image is cropped into a uniform scale of 448×448 pixels, Fixed number of channels is 256, and the fused feature pixel is 112×112 .

Algorithm 1 Calculate Total Losses

Require: RPN and ROI location loss, RPN and ROI class loss
Ensure: Total loss

- 1: $L_{loss} = \lambda_1(L_{cls}(RPN) + L_{cls}(ROI)) + \lambda_2(L_{reg}(RPN)$
- 2: $+ L_{reg}(ROI));$
- 3: Training N iterations
- 4: **for** i in N **do**
- 5: Total loss = Losses+[sum(Losses)];
- 6: **end for**
- 7: return Total loss(*losses);

Focal-loss function is shown in (3), and the pseudo code is shown in Algorithm 2

$$CE = \begin{cases} -\alpha \log(p), & \text{if } y = 1 \\ -(1 - \alpha) \log(1 - p), & \text{otherwise} \end{cases} \quad (2)$$

where α is the positive/negative sample balance factor, y represents label, $y \in (-1, 1)$, p is the probability of $y = 1$, $p \in (0, 1)$.

$$FL = \begin{cases} -\alpha(1 - P)^\gamma \log(p), & \text{if } y = 1 \\ -(1 - \alpha)P^\gamma \log(1 - p), & \text{otherwise} \end{cases} \quad (3)$$

where γ is the sample balance factor of hard/easy samples, p is the probability that the prediction sample belongs to 1, y is label, and the value is $(-1, +1)$, P is the probability that the sample belongs to true class, α is the hyperparametric($\alpha = 0.25$). Based on the above two classification loss function, the PCB defect multi-classification sample loss function is obtained and shown in the following:

$$L_{cls} = \sum_i (\alpha_i(1 - p_i)^\gamma \log(p_i) + (1 - \alpha_i)p_i^\gamma \log(1 - p_i)) \quad (4)$$

where p_i is the probability that the target is in the candidate box, $p_i \in [0, 1]$, γ is the classification difficulty factor ($\gamma > 0$), the experiments show that $\gamma = 2$ is the best. The classification difficulty becomes greater if the positive

Algorithm 2 Focal Loss Function Classification

Require: pred, object, weight, γ , α
Ensure: $Loss_{cls}$;

- 1: def $\gamma = 2.0$, $\alpha = 0.25$, Gradient descent function='mean',
- 2: def sigmoid_focal_loss(pred, object, weight)=None,
- 3: **while** $N \neq 0$ **do**
- 4: pred_sigmoid = pred. sigmoid()
- 5: object = object. type_as(pred)
- 6: $p_t = (1 - pred_sigmoid) * object + pred_sigmoid * (1 - object)$
- 7: focal_weight = $(\alpha * object + (1 - \alpha) * (1 - object))$
- 8: pt.pow(γ)
- 9: loss = binary_cross_entropy(pred, object, reduction='none' * focal_weight)
- 10: loss = loss(loss, weight, reduction, avg factor)
- 11: **end while**
- 12: return Focal loss
- 13: /*pred is estimate, object is class, weight is weight function*/

sample threshold is close 0.7, and $(1 - p_i)^\gamma$ becomes greater, which lead p_i is closer 1. $(1 - p_i)^\gamma$ is close to 0 when the classification difficulty becomes small.

The regression loss is the frame loss, which refers to the error of calculating the geometric parameter offset of the candidate box relative to the label box. The regression loss function L_{reg} is shown in (5), and the pseudo code is shown in Algorithm 3

$$L_{reg}(v, v^*) = \sum_{i \in \{x, y, w, h\}} p_i * \text{smooth}_{L_1}(v_i^* - v_i) \quad (5)$$

where x, y, w, h is the geometric parameter of the candidate box, p_i^* is the ground-truth label of the labeled sample, where the positive sample is 1, and the negative sample is 0, that is, only the positive sample is used for position regression learning, smooth_{L_1} is the smooth loss function, as shown in (6), v_i^* and v_i are the set of geometric parameter offsets of the

Algorithm 3 Focal Loss Function of Regression

```

Require: N, array
Ensure: loss;
1: def yhat = np. array([0.1, 0.2, 0.3, 0.4, 0.5])
2: def y = np. array([1, 1, 0, 1, 1])
3: import N as np
4: for (i from 0 to N) do
5:   Smooth_L1 loss;
6:   loss = np. sum(np. abs(y - yhat))
7:   Smooth_L2 loss;
8:   loss = np. sum(np. power((y - yhat), 2))
9: end for
10: return loss

```

label box and the candidate box of the feature map, as shown in (7) and (8)

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5\sigma^2, & \text{if } |x| \leq 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (6)$$

$$v_x = \frac{x - x_a}{w_a}, \quad v_y = \frac{y - y_a}{h_a}$$

$$v_w = \lg \frac{w}{w_a}, \quad v_h = \lg \frac{h}{h_a}, \quad (7)$$

$$v_{x^*} = \frac{x^* - x_a}{w_a}, \quad v_{y^*} = \frac{y^* - y_a}{h_a},$$

$$v_{w^*} = \lg \frac{w^*}{w_a}, \quad v_{h^*} = \lg \frac{h^*}{h_a} \quad (8)$$

where (x, y, w, h) is the geometric parameters of the candidate box, x, y is the coordinate, w is the width and h is the height, (x^*, y^*, w^*, h^*) is the geometric parameters of the real annotation box, x^*, y^* is the coordinate, w^* is the width and h^* is the height, (x_a, y_a, w_a, h_a) is the geometric parameter set of anchor annotation box, x_a, y_a is the coordinate, w_a is the width and h_a is the height, $(v_x, v_y, v_w, v_h \in v)$, $(v_{x^*}, v_{y^*}, v_{w^*}, v_{h^*}) \in v^*$.

IV. PCB DEFECT DATASET ENHANCEMENT

In this article, the PCB defect dataset published by Ding *et al.* [17], which is used as the training set and test set of the model. The original image pixel is 3034×1586 and 2282×2248 . The dataset contains 693 PCB defect images and defect type annotation files. The dataset are divided into six categories, representing six types of faults, i.e., short, miss hole, spurious copper, spur, mouse bite, open circuit, as shown in Fig. 5. Through the statistical study of PCB bare board defect dataset, it is found that 77% of the defects in the original figure account for proportion 1/64 to 1/256 (pixel is 48×24 to 12×6 or 36×36 to 9×9), and only 23% of the defects in the original figure account for Proportion more than 1/64 or less than 1/256. In the training process of deep learning model, many parameters in the model need to be calculated and updated, but the size of the dataset is not large enough, resulting in the inaccuracy of the detection. Therefore, this article uses image clipping and rotation to enhance the original data. The input image is cropped into a uniform scale of 448×448 pixels and rotated by 180° , 90° , 60° , 45° , and

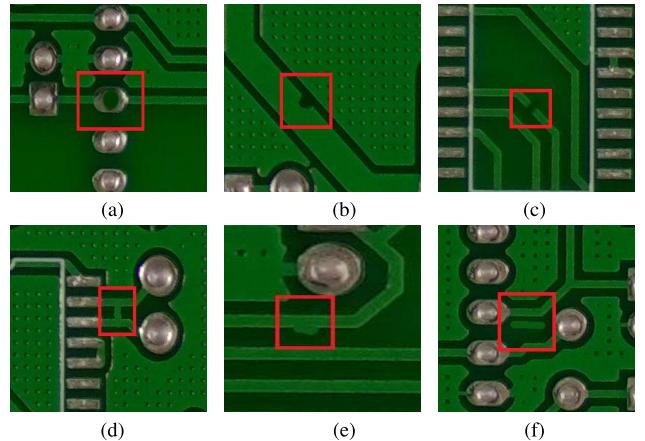


Fig. 5. PCB defect fault diagram. The defect classes are divided into six categories, representing six types of faults, i.e., Short, Miss hole, Spurious copper, Spur, Mouse bite, Open circuit. (a) Miss hole. (b) Mouse bite. (c) Open circuit. (d) Short. (e) Spur. (f) Spurious copper.

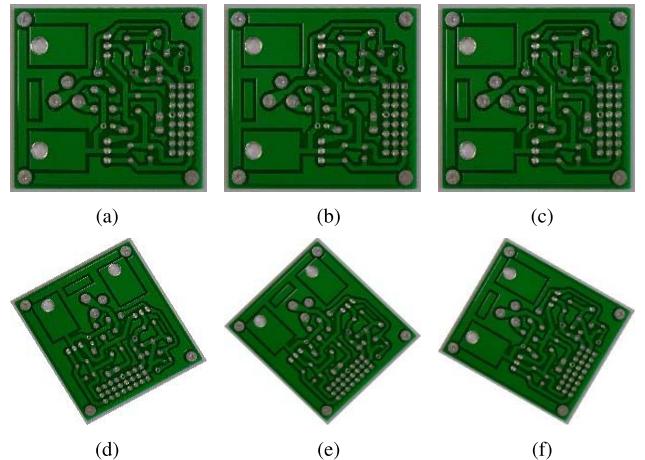


Fig. 6. Data enhancement of PCB defect. The input images are rotated by 180° , 90° , 60° , 45° , and 30° . (a) Original. (b) 180° . (c) 90° . (d) 60° . (e) 45° . (f) 30° .

TABLE I
TDD PCB DEFECT DATASET

Defect type	Image Numbers	Defect Numbers
Miss hole	1906	6958
Mouse bite	1906	6888
Open circuit	1922	7988
Short	1906	8052
Spur	1906	6888
Spurious copper	1922	8255
total	11484	46242

30° as shown in Fig. 6. The total number of images after data enhancement is 11484. Before the model training, the data are divided into training set and test set with a ratio of 4:1. The training set contains 9197 images and the test set contains 2287 images. The specific distribution of faults is summarized in Table I.

V. EXPERIMENTAL RESULTS**A. Experimental Platform**

All the experiments in this article are carried out in the fast feature embedded cafe software environment under the

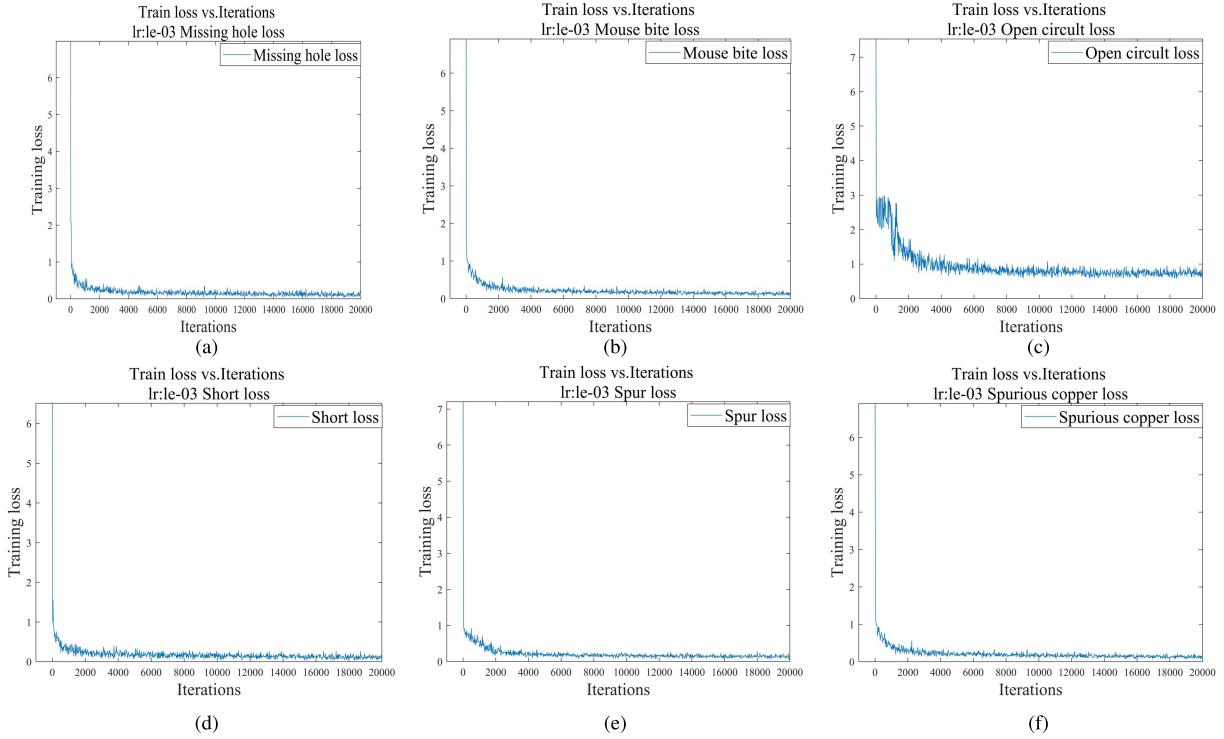


Fig. 7. In order to analyze the difficult and easy samples, we train each type of defect object. The figure shows the initial loss value of the PCB bare board detection algorithm based on the extended FPN model is 6.8 during the training process. With the increase of the number of iterations, the loss value drops sharply by about 0.9. When the loss cost slowly approaches 0.2 (Due to the high similarity between open and background, the training loss function is relatively poor, Open circuit cost slowly approaches 0.8), the curve is relatively stable in the whole process. (a) Miss hole. (b) Mouse bite. (c) Open circuit. (d) Short. (e) Spur. (f) Spurious copper.

Algorithm 4 Extended FPN Model Training Process

- 1: Pre-train Extended FPN CNN model.
- 2: Initialize the RPN network with the model in Step1, and then train the RPN.
- 3: Initialize the ROI network with the model in Step1, then use the trained RPN to calculate the proposal and give the proposal to the ROI network.
- 4: Fine-tune the RPN network generation sharing Feature layers trained in Step 2.
- 5: Fine-tune the ROI network for object detection using region proposals obtained from Step 4.
- 6: Output a unified network trained in Step 4 and Step 5 as the final model.

condition of Ubuntu 18.04 and convolutional architecture, including NVIDIA-SMI 440.33.01, cuda10.2, opencv3.2.0. The hardware environment is i7 8700k, and the GPU is GTX 1070ti 8G memory.

B. Training Process

In order to verify the influence of extended FPN model, focusing on hard sample in loss function and PCB defect data enhancement on the model detection performance, the training process of this article is divided into the following six steps. Algorithm 4 shows the pseudocode of training.

- 1) Prepare the training dataset and test dataset after image enhancement.

2) The Caffe framework is used to define the training and testing model, and modify the loss function and feature extraction network.

3) When the loss reaches the maximum number of iterations or the loss value is less than the threshold, the training is stopped and the script file of Caffe model is obtained.

4) The Caffe model is used to initialize the parameters of the test model, and the labeled test dataset is used to obtain the final output image.

5) The evaluation model is used to obtain the output image from the corresponding dataset, and obtains some mapping tables, average accuracy indexes of various categories, and the evaluation result chart of intersection over union (IoU) confidence score covering the object positioning frame.

C. Loss Function

As shown in Fig. 7, the total initial loss value of the PCB bare board detection algorithm based on the extended FPN model is 6.8 during the training process. As the number of iterations goes up, the loss value drops sharply by about 0.9. When the loss cost slowly approaches to 0.2, the flattens in the whole process. Therefore, the loss value of the whole training process is decreasing, and there is no adverse trend of up and down vibration in the whole process, so it is best for the hyper-parameter setting and it is fixed in the extended FPN network structure.

Fig. 8 compares the detection accuracy based on three loss functions, CE loss, Focal loss and improved Focal loss.

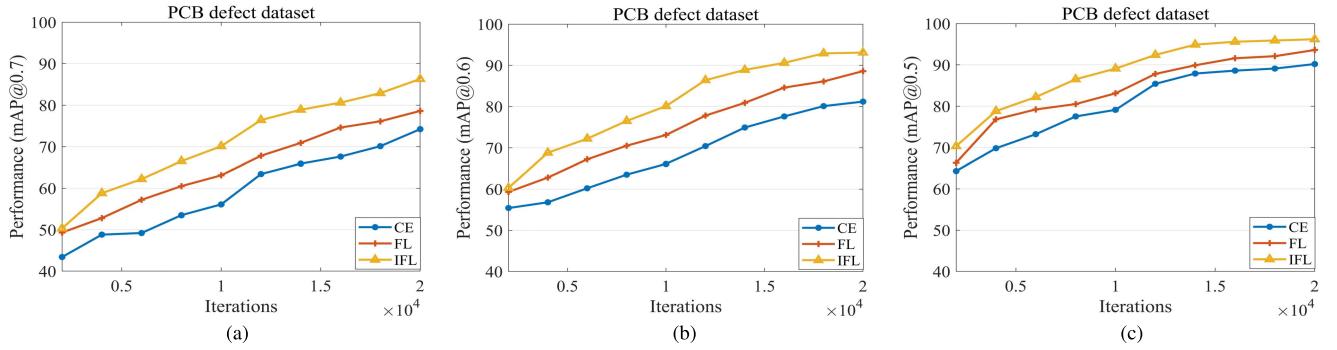


Fig. 8. Denotes the detection performance as iterations progress, improvement of focal loss (IFL) compared to standard cross entropy (CE), and focal loss (FL). (a) IOU = 0.7. (b) IOU = 0.6. (c) IOU = 0.5.

Because the weight of difficult samples has increased on improved Focal loss, with the increase of iteration times, performance (*mAP*) has greatly improved.

D. Feature Map Extraction

Fig. 9 shows the outputs of P1, P3, P5, and P0 convolution layer features of the extended FPN model on the PCB defect dataset. Fig. 9(b) shows P1 layer 224×224 pixels feature map, which is convolved by 64 1×1 convolution kernels, Fig. 9(c) shows P2 layer 112×112 pixels feature map, which is convolved by 256 1×1 convolution kernels, Fig. 9(d) P5 layer shows 14×14 pixels feature map, which is convolved by 2048 1×1 convolution kernels, It can be seen from Fig. 9(e) that after the fusion of high-level semantic information and low-level geometric information, the feature information of layer P0 is richer for PCB defect detection.

E. Results on the PCB Defect Dataset

Fig. 10 shows the output effect of target detection and classification on PCB defect dataset based on extended FPN network. From the results, it can be seen that the IoU confidence of each type of target reach more than 0.9, and the object can be accurately classified.

Small target acquisition can get better effect by focusing on hard sample in the loss function, compared with the original Faster R-CNN network structure. The output results in RPN network make full use of the feature information and the threshold of IoU is 0.5, 0.6, 0.7, as summarized in Table II. The output contains rich positive sample information and eliminates a lot of non-object information, so the image target classification is more clear and accurate. Compared with the output object of the original Faster R-CNN, the output of the extended FPN network is closer to the ground truth.

As summarized in Table III, the improved network structure uses six kinds of targets on PCB defect dataset for training. Performance is measured by class average precision (AP) and speed on the test set. The maximum *mAP* of the extended FPN network is 96.2%, which is 5.4% higher than the original Faster R-CNN and 0.6% higher than UF-Net. This is because the extended FPN network integrates multilayer feature map, which fuse low-level and high-level image feature information together, and use the loss function that focus on hard sample

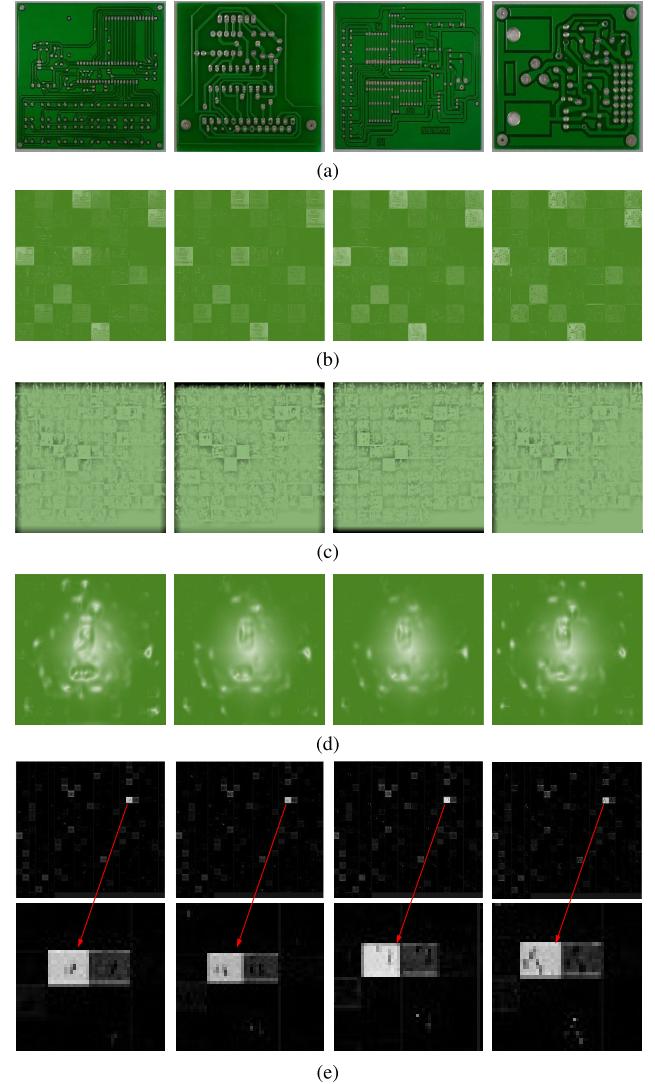


Fig. 9. Middle layer output results of Extended FPN model, the output results of P1, P2, P5 and P0 convolution layer features of the extended FPN model on the PCB defect dataset. (a) Original image. (b) Feature map of P1 layer. (c) Feature map of P2 layer. (d) Feature map of P5 layer. (e) Feature map of P0 layer.

to improve the detection accuracy of small target and hard sample. The extended FPN network integrates the semantic information of C1 and C2 layers, which increased the network

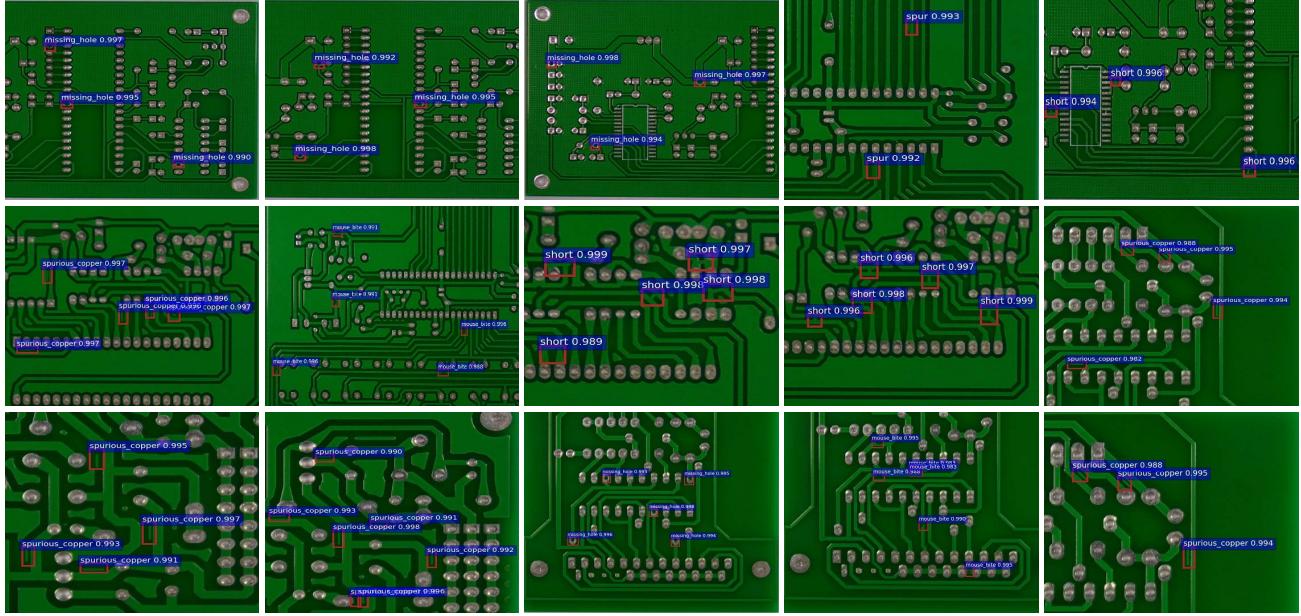


Fig. 10. Output images show the output effect of target detection and classification on PCB defect dataset based on extended FPN network. From the results, it can be seen that the IoU confidence of each type of target can reach more than 0.9, and the object can be accurately classified.

TABLE II
DETECTION $mAP(\%)$ FOR DIFFERENT BACKBONES AND THE ACCURACY OF EACH CATEGORY $AP(\%)$

Backbone	Miss_hole	Mouse bite	Open circuit	Short	Spur	Spurious copper	$mAP(\%)$	Runtime(s/img)
VGG16	80.5	76.2	70.3	81.3	79.3	78.6	80.2	0.26
ResNet-50	86.7	81.8	80.1	87.9	84.9	83.2	81.6	0.31
ResNet-50+FPN	94.2	90.8	89.4	95.6	91.3	91.6	91.5	0.32
ResNet-101	90.7	86.2	82.7	89.4	86.9	87.3	86.8	0.41
ResNet-101+FPN	97.1	95.2	93.2	98.4	95.6	96.3	96.2	0.46

TABLE III
TEST RESULTS UNDER DIFFERENT IoU VALUES COMPARISON ON THE PCB DEFECT DATASET

Method	Backbone	Input size	FPS	$mAP(\%)@0.5$	$mAP(\%)@0.6$	$mAP(\%)@0.7$	Model size
<i>Two-stage:</i>							
Faster R-CNN	ResNet-50	600×600	26.5	81.2	74.2	63.2	108M
Faster R-CNN	ResNet-101	600×600	39.7	86.4	78.7	68.5	125M
R-FCN	ResNet-50	600×600	36.6	87.1	81.3	71.9	131M
R-FCN	ResNet-101	600×600	30.2	90.8	85.2	72.4	139M
TDD-Net	ResNet-101	448×448	26.7	94.2	88.3	75.1	127M
UF-Net	ResNet-101	448×448	33.5	95.6	90.1	80.3	143M
<i>One-stage:</i>							
YOLOv3	Darknet-53	320×320	19	75.8	61.3	45.1	235MB
YOLOv4	CSPDarknet-53	512×512	11	79.6	66.8	51.3	289MB
SSD	VGG-16	300×300	16.7	80.3	70.3	61.9	127M
CornerNet	Hourglass-104	511×511	31	85.6	74.6	64.3	101MB
CenterNet-Triplets	Hourglass-104	511×511	29	87.2	75.2	65.1	140MB
ExtremeNet	Hourglass-104	511×511	17	89.3	71.4	62.8	133M
CenterNet	Resnet-50	384×384	34	90.3	76.3	68.3	117MB
CenterNet	ResNet-101	384×384	42	91.5	78.2	71.5	189MB
<i>Ours:</i>							
Extended FPN	ResNet-50	448×448	7.6	93.4	90.5	84.9	128M
Extended FPN	ResNet-101	448×448	13.5	96.2	93.1	86.3	137M

running time, the FPS is 13.5 When ResNet101 + FPN is used as the backbone network. Table II summarizes the detection accuracy of each class of defect under the different

backbone network, because the Short category are easy to detect, the accuracy of the bus category is up to 98.4%, while the Open_circuit category features are not easy to capture,

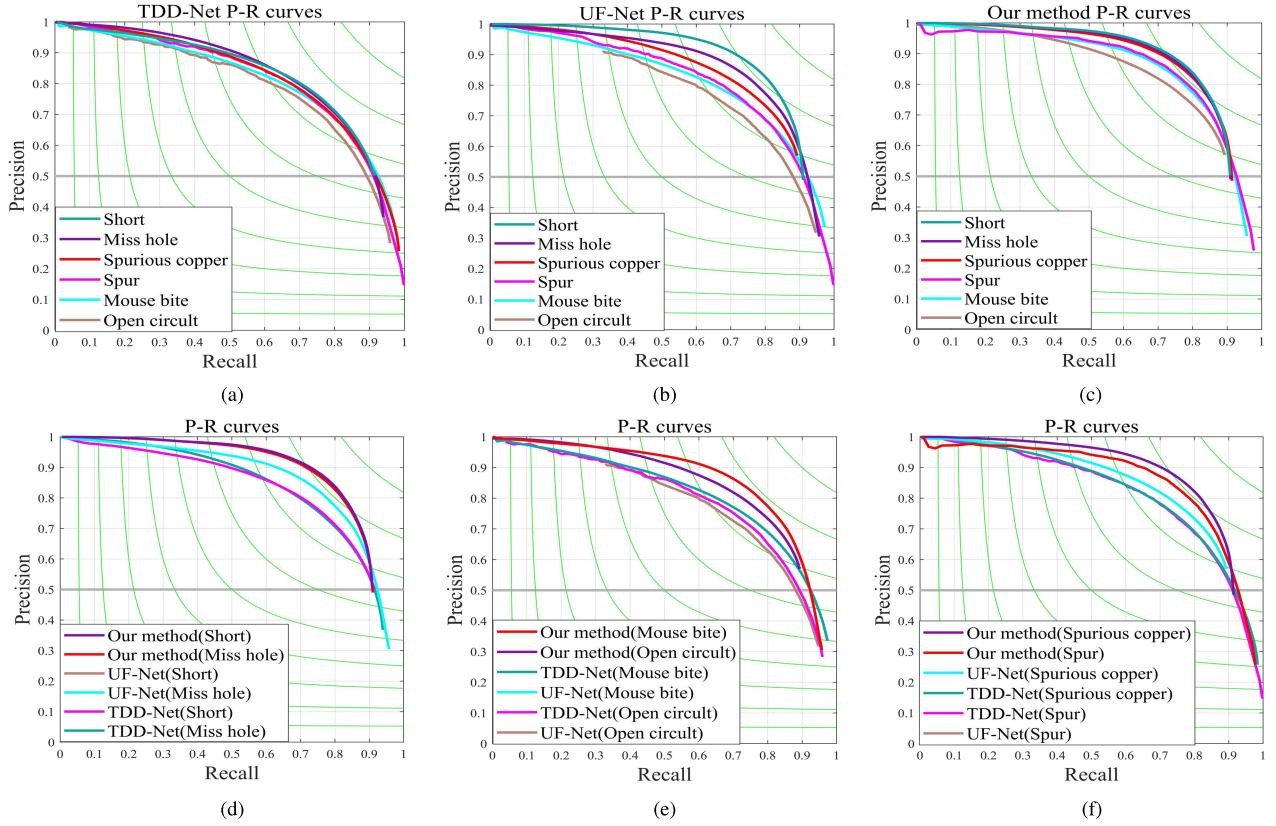


Fig. 11. P-R curves of TDD-Net, UF-Net, and our method, the network architecture of the three algorithms are used ResNet-101, (a)–(c) P-R curve of each defect category on TDD-Net, UF-Net and our method, (d)–(f) comparison of the three algorithms under different defects. (a) TDD-Net. (b) UF-Net. (c) Our method. (d) Short and Miss hole. (e) Mouse bite and Open circuit. (f) Spurious copper and Spur.

TABLE IV
TEST RECALL UNDER DIFFERENT METHOD COMPARISON
ON THE PCB DEFECT DATASET

Method	Backbone	<i>mAP</i> (%)	Recall(%)
Faster R-CNN	VGG16	81.2	72.6
	ResNet-50	86.4	73.8
	ResNet-101	87.1	78.5
	ResNet-101+FPN	90.8	79.2
TDD-Net	ResNet-50	86.5	76.3
	ResNet-101	89.7	78.8
	ResNet-101+FPN	94.2	72.9
UF-Net	ResNet-101	95.6	82.5
Our method	ResNet-50+FPN	93.4	83.7
	ResNet-101+FPN	96.2	85.6

there are fewer features, so the accuracy of the Open_circuit category is at least 93.2%. As summarized in Table IV, recall are compared under different methods and backbones on the PCB defect dataset, and the best is presented approach under ResNet101 + FPN backbone.

F. P-R Curves

The *mAP* is defined as the AP for all the object categories, and AP is the area under the precision and recall (P-R) curve.

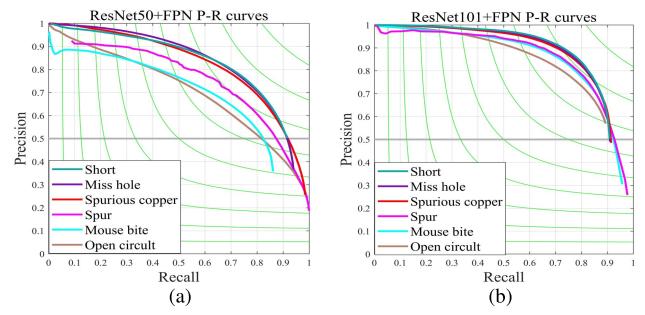


Fig. 12. P-R curves comparison of different main network structures, the results show that the performance of ResNet101 + FPN structure is better than ResNet50 + FPN structure. (a) ResNet50 + FPN. (b) ResNet101 + FPN.

The calculation equation of the precision and recall is shown as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (10)$$

where TP is the number of positive samples correctly divided into positive examples, FP is the number of negative samples incorrectly divided into positive examples, and FN is the number of positive samples incorrectly divided into negative examples. The *mAP* represents the average AP value of all

categories. The equations are expressed in the following:

$$AP = \int_0^1 P(R)dR \quad (11)$$

$$mAP = \sum_{i=1}^N \frac{AP(i)}{N}. \quad (12)$$

We compare the P-R with TDD-Net and UF-Net method, which are the state-of-the-art methods. Fig. 11 shows the P-R curves. In the experiment, the network architecture of the three algorithms both use ResNet-101, the three figures show that for all P-R curve of different defect category, our algorithm outperforms TDD net and UF-Net. The three figures show that the recall rate of short defect category is the highest, which is 98.4% for our algorithm, and open defect category is the lowest, which is 93.2% for our algorithm.

The P-R curve comparison of different main network structures are shown in Fig. 12. The experimental results show that the performance of ResNet101 + FPN structure is better than ResNet50 + FPN structure.

VI. CONCLUSION

In view of the limitations of the detection range and feature extraction of microdefects in PCB images, we propose a PCB defect detection algorithm based on extended FPN model. The bottom-up network is constructed by ResNet-101, and the up sampling feature map is combined with the shallow feature map of the top-down path. In order to reduce the aliasing effect caused by up sampling, each merged feature map is divided into two parts. And then the feature map is followed by a 3×3 convolution. The P3 feature layer that is rich in semantic information is sampled up, the P1 feature layer that is rich in geometric information is sampled down, after that, P1, P2, P3 are fused together by pixel-wise addition. Finally, a 3×3 convolution operation is done to form P0. Aiming at the hard samples classification in PCB defect detection, the focal loss function is introduced to increase the weight of hard samples in the loss function, so as to focus on learning the features of hard samples, so as to effectively improve the image detection accuracy. Aiming at the overfitting in the training process, this article uses image clipping and rotation to enhance the original data, crops the input image into a uniform scale of 448×448 pixels and rotates it by 180° , 90° , 60° , 45° , and 30° . Through the quantitative analysis of PCB defect dataset, the results show that the method has good portability, and the accuracy on the public PCB dataset reaches 96.2%.

The average accuracy of the extended FPN model in the manually labeled PCB defect dataset has been improved, but it is not the best network model for the actual industrial production (noise interference, exposure) and so on [33], [34]. In the future, we will further study the generation of countermeasure network (GAN) to solve the PCB defect target detection and recognition in more complex environment.

REFERENCES

- [1] M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa, "Automatic PCB inspection algorithms: A survey," *Comput. Vis. Image Understand.*, vol. 63, no. 2, pp. 287–313, Mar. 1996.
- [2] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 2, pp. 3465–3471, Jul. 2018.
- [3] W.-Y. Wu, M.-J.-J. Wang, and C.-M. Liu, "Automated inspection of printed circuit boards through machine vision," *Comput. Ind.*, vol. 28, no. 2, pp. 103–111, May 1996.
- [4] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.
- [5] J. Chen, Z. Liu, H. Wang, A. Núñez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 257–269, Feb. 2018.
- [6] M. Moganti, F. Ercal, and C. Dagli, "Automatic PCB inspection algorithms: A survey," *Comput. Vis. Image Understand.*, vol. 37, no. 9, pp. 1904–1916, Aug. 2014.
- [7] M. Ejiri, T. Uno, M. Mese, and S. Ikeda, "A process for detecting defects in complicated patterns," *Comput. Graph. Image Process.*, vol. 2, nos. 3–4, pp. 326–339, Dec. 1973.
- [8] Z. Ibrahim, N. K. Khalid, and I. Ibrahim, "A noise elimination procedure for printed circuit board inspection system," in *Proc. Eng. Comput. Sci.*, Las Vegas, NV, USA, May 2008, pp. 1063–6919.
- [9] B. Kaur, G. Kaur, and A. Kaur, "Detection and classification of printed circuit board defects using image subtraction method," in *Proc. Eng. Comput. Sci.*, Chandigarh, India, Mar. 2014, pp. 1–5.
- [10] M. Kumar, N. K. Singh, M. Kumar, and A. K. Vishwakarma, "A novel approach of standard data base generation for defect detection in bare PCB," in *Proc. Eng. Comput. Sci.*, Greater Noida, India, May 2015, pp. 11–15.
- [11] Y. Li *et al.*, "Convolutional neural network-based block up-sampling for intra frame coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2316–2330, Sep. 2018.
- [12] M. Jubran, A. Abbas, A. Chadha, and Y. Andreopoulos, "Rate-accuracy trade-off in video classification with deep convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 1, pp. 145–154, Jan. 2020.
- [13] P. Wei, C. Liu, M. Liu, Y. Gao, and H. Liu, "CNN-based reference comparison method for classifying bare PCB defects," *J. Eng.*, vol. 2018, no. 16, pp. 1528–1533, Nov. 2018.
- [14] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 2584–2589.
- [15] C. Zhang, W. Shi, X. Li, H. Zhang, and H. Liu, "Improved bare PCB defect detection approach based on deep feature learning," *J. Eng.*, vol. 16, no. 1, pp. 1415–1420, Jun. 2018.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [17] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-Net: A tiny defect detection network for printed circuit boards," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 110–116, Jun. 2019.
- [18] X. Xu, Y. Lei, and F. Yang, "Railway subgrade defect automatic recognition method based on improved faster R-CNN," *Sci. Program.*, vol. 24, no. 12, pp. 1–3, Jun. 2018.
- [19] Y.-T. Li, P. Kuo, and J.-I. Guo, "Automatic industry PCB board DIP process defect detection system based on deep ensemble self-adaption method," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 2, pp. 312–323, Feb. 2021.
- [20] Y.-T. Li, P. Kuo, and J.-I. Guo, "Automatic industry PCB board DIP process defect detection with deep ensemble method," in *Proc. IEEE 29th Int. Symp. Ind. Electron. (ISIE)*, Delft, The Netherlands, Jun. 2020, pp. 453–459.
- [21] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.*, vol. 9, no. 8, pp. 1–15, Jun. 2018.
- [22] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 761–769.
- [23] B. Yu and D. Tao, "Anchor cascade for efficient face detection," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2490–2501, May 2019.
- [24] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Aug. 2019.

- [25] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 850–859.
- [26] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 6568–6577.
- [27] Y. Liang *et al.*, "TFPN: Twin feature pyramid networks for object detection," in *Proc. IEEE 31st Int. Conf. Tools Artif. Intell. (ICTAI)*, Portland, OR, USA, Nov. 2019, pp. 1702–1707.
- [28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.
- [29] O. Ghahabi and J. Hernando, "Restricted Boltzmann machines for vector representation of speech in speaker recognition," *Comput. Speech Lang.*, vol. 47, pp. 16–29, Jan. 2018.
- [30] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8693, Zürich, Switzerland, Sep. 2014, pp. 740–755.
- [31] M. D. Marco, M. Forti, M. Grazzini, and L. Pancioni, "Limit set dichotomy and convergence of semiflows defined by cooperative standard CNNs," *Int. J. Bifurcation Chaos*, vol. 20, no. 11, pp. 3549–3563, Nov. 2010.
- [32] Q. Mercier, F. Poirion, and J.-A. Désidéri, "A stochastic multiple gradient descent algorithm," *Eur. J. Oper. Res.*, vol. 273, no. 3, pp. 808–817, Dec. 2018.
- [33] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3296–3307.
- [34] S. R. Buló, G. Neuhold, and P. Kotschieder, "Loss max-pooling for semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7082–7091.



Zhong Qu received the M.S. degree in computer architecture and the Ph.D. degree in computer application visualization from Chongqing University, Chongqing, China, in 2003 and 2009, respectively.

He is currently a Professor with the Chongqing University of Posts and Telecommunications, Chongqing. His research interests are in the areas of digital image processing, digital media technology, and cloud computing.



Shi-yan Wang is currently an Associate Professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests are in the areas of machine learning and digital image processing.



Kang-hua Bao received the B.S. degree in communication engineering from Chongqing University, Chongqing, China, in 2006, and the M.S. degree in computer technology from the Chongqing University of Posts and Telecommunications, Chongqing, in 2013.

He is currently a Senior Engineer with Southwest Computer Company, Ltd., Chongqing. His research interests are in the areas of digital image processing and cloud computing.



Cui-jin Li is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.

Her main research interests are neural network and digital image processing.



Sheng-ye Wang is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.

Her main research interests are neural network and digital image processing.