# A PCB Defect Detection Algorithm with Improved Faster R-CNN

Junhao Niu [1,2], Jin Huang [1,2], Lili Cui [3*], Benxin Zhang[1,2], Aijun Zhu[1,2]

[1] *School of Electronic Engineering and Automation Guilin University of Electronic Technology, Guangxi, Guilin, 541004, China*
[2] *Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guangxi, Guilin, 541004, China*
[3] *School of Art and Design Guilin University of Electronic Technology, Guangxi, Guilin, 541004, China*

**Abstract**

The paper proposes an improved printed circuit board (PCB) defect detection algorithm based on the original faster region convolutional neural networks (Faster R-CNN) for the problems of low average accuracy mean value, poor detection of tiny defect targets and high leakage rate in PCB tiny defect detection. Firstly, a genetic algorithm is added to the K-means++ clustering algorithm to generate the initial anchor that match the data set in this paper. The standard convolution in the Resnet50 network is then replaced by a depth-separable convolution as the backbone network to reduce the number of computational parameters, and the multilayer depth features are extracted and fed into the improved feature pyramid network to train the model, effectively combining the geometric detail information in the bottom layer and the semantic contour information in the top layer to provide material for subsequent classification and localization. The experimental results show that the average accuracy of this algorithm is 95.6% and the detection speed is 0.125s, which is 9.2% higher than the current mainstream tiny object detection algorithm and has better detection accuracy for tiny defect object.

**Keywords**

printed circuit board defect detection, K-means++ clustering, genetic algorithm, anchor box, Faster R-CNN, Resnet50, depth separable convolution, feature pyramid network

## 1. Introduction

As an indispensable part of electronic products, the quality of PCB directly determines whether the electronic products can work normally, and the quality is closely related to each link of production. With the continuous development of hardware level, PCB design is developing towards the direction of multi-layer, table-pasting and densification. In addition, PCB production is composed of multiple links. For example, the production process of a single panel includes cutting, drilling, copper deposition, etching, resistance welding, hot air leveling, character and electrical measurement [1]. Problems in any of the above links may cause the final product to fail to work normally and thus increase production costs. An effective way to ensure the quality of PCB is to add PCB defect detection in the production process. Compared with electrical testing, it has the advantage of non-contact nondestructive testing, which can better protect PCB and avoid damage in the production process. So the research of PCB defect detection algorithm is very necessary.

According to whether the PCB is mounted or not, PCB defect detection is divided into PCB bare board detection and PCB component detection. Reference [2] applies the multi-scale and pyramid structure of deep convolutional network itself to the construction of feature pyramid, so as to detect the tiny defects in PCB bare board. In reference [3], it is proposed that the improved YOLOv4 can locate and identify components on the circuit board, which can realize the identification of device leakage, device wrong installation, device offset, device skew and device polarity reverse installation. At present, the research on PCB bare board defect detection is divided into two directions: automatic optical

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China
EMAIL: 30189252@qq.com (Junhao Niu); 364817579@qq.com (Jin Huang); 45105703@qq.com (Lili Cui)
ORCID: 0000-0001-6876-4566 (Junhao Niu); 0000-0002-7436-577X (Jin Huang); 0000-0002-9454-4679 (Lili Cui)

detection and machine vision defect detection based on deep learning. Automatic optical testing is affected by the changes of testing environment, circuit board types, personnel experience and other factors, so it is not robust to various PCB defects, and the cost of labor is too large. All kinds of PCB on the market have different line spacing rules and small line width. Automatic optical detection is difficult to detect such complex and diverse PCB, and the false detection rate and missed detection rate are still too high. The machine vision defect detection based on deep learning can not only save the cost of labor input, but also greatly reduce the false detection rate and missed detection rate.

With the rapid development of convolutional neural networks in recent years, some excellent PCB defect detection algorithms based on deep learning have emerged. Reference [4] proposed a redesigned clustering method based on YOLOv3 and added attention mechanism to improve the detection speed of the algorithm. Reference [5] proposed a new anchor based on YOLOv4 redesigned clustering method to improve the detection speed to 37.09FPS. However, since most PCB defects are small defects, in order to improve the detection accuracy of small defects, this paper proposes a combination of K-means++ clustering algorithm and genetic algorithm to generate an initial anchor suitable for small targets, and combines the improved Faster R-CNN detection algorithm to detect PCB defects. This paper chooses to study the defects of PCB bare board, and the detected defects include 6 common types, which are: missing hole, mouse bite, open, short, spur, and spurious copper.

## 2. Design of improved K-means++ clustering algorithm

The anchor of the traditional Faster R-CNN method is designed manually to detect large targets such as pedestrians, vehicles and tools in PASCAL VOC2007 data set [6], which is not suitable for the detection of small defect targets in this paper, and the robustness of manual design through experience is poor. Therefore, it is necessary to design an algorithm to cluster to get the anchor suitable for micro-defect target detection, and select the size of the anchor suitable for the data set as the training parameter, so that the network can learn faster and get a better detector. This can reduce the difficulty of network fine-tuning anchor and improve the final recognition accuracy and speed.

Therefore, this paper proposes to use genetic algorithm to optimize the anchor obtained by k-means ++ algorithm. The idea is to firstly use K-means++ algorithm to select a more appropriate initial clustering center and then use k-means algorithm to get the clustering results. Finally, the final clustering results are optimized by genetic algorithm. The core of the algorithm and the three problems to be solved are how to select the initial cluster center, which distance formula between samples to select, and how to optimize the final result of the genetic algorithm. The following three parts are elaborated respectively.

### 2.1. Select the appropriate initial cluster center

Since the selection of the first clustering center of the standard k-means algorithm is randomly selected from all training samples, it is easy to finally select the size of the anchor that does not match the data set in this paper [7]. Moreover, the number of iterations and the final clustering effect of the algorithm are closely related to the selection of the initial clustering center. In order to get a better anchor, this paper uses K-means++ algorithm to select the initial cluster centers one by one, and the number of categories is K. According to the final generated five feature layers and three different proportions of anchor, we assign each feature layer in turn, so K is chosen as 15 here. Although K-means++ algorithm increases the time to select the initial cluster center, it greatly speeds up the overall convergence speed and can avoid the selection of inappropriate anchor. On the whole, it improves the performance of the clustering algorithm. The selected initial cluster center is shown in Figure 1.

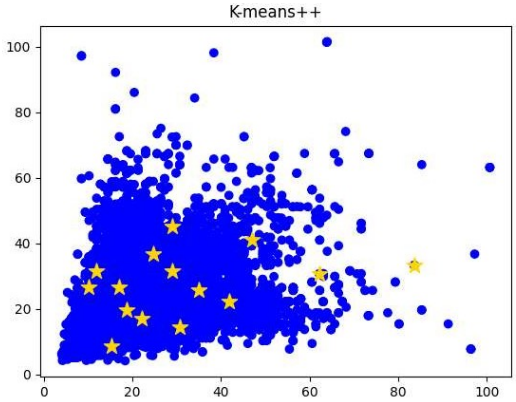

**Figure 1**: Initial cluster center

## 2.2. Distance between samples was calculated

The standard K-means algorithm uses Euclidean distance to calculate the distance between the training sample and the cluster center, in which the larger anchor will produce more errors than the smaller anchor. The defect detection problem studied in this paper requires the anchor to be as close as possible to the ground truth box size. Therefore, this paper uses the distance calculation as formula (1) and (2), where B represents the training sample and C represents the cluster center. The larger the intersection ratio, the smaller the distance, and the more likely they are to belong to the same class. It can describe the relationship between training samples and clustering centers more accurately. So as to improve the quality of the final anchor.

$$d(B,C) = 1 - IOU(B,C) \tag{1}$$

$$IOU(B,C) = \frac{area(B) \cap area(C)}{area(B) \cup area(C)} \tag{2}$$

## 2.3. Genetic algorithm

Genetic algorithm draws on the idea of biological evolution and introduces the concept of survival of the fittest into the clustering algorithm to optimize the clustering result. The characteristics of data set optimization, flexible probabilistic search and parallel computing in genetic algorithm just fill in the deficiencies of K-means algorithm [8]. The clustering center obtained by k-means algorithm is sent to the genetic algorithm, and the algorithm process is as follows:

1. The width and height data of 15 anchors were coded to get chromosomes.
2. The coincidence degree of ground truth box and anchor is defined as fitness function. The closer the size of anchor is to the size of ground truth box, the better its fitness will be. The mass of the anchor from the reaction.
3. The mutation probability was set as 90%, the number of iterations was set as 1000, and the fitness threshold was 0.25.
4. Application propagation and mutation generate the next generation population, and if the fitness of the next generation is greater than that of the previous generation, the next generation will be updated.
5. If the suboptimal solution is equal to the last optimal solution or reaches the number of iterations, the algorithm ends.

Figure 2 shows the final clustering results.

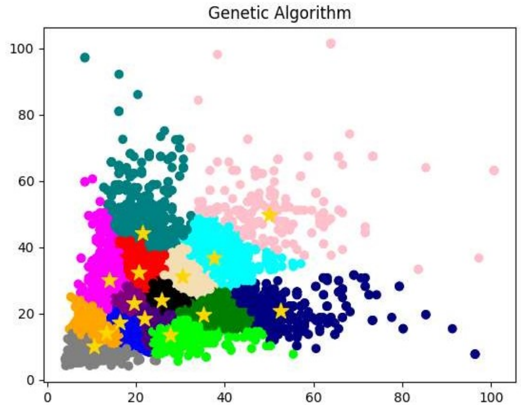

**Figure 2**: Final clustering results

## 3. Design of improved Faster R-CNN algorithm

In a nutshell, the object detection task is to input an image into the network and output an image containing the object bounding box and the category label score. With the iteration of the algorithm and people's pursuit of speed and accuracy, the target detection algorithm has gradually developed into one-stage target detection algorithm and two-stage target detection algorithm. Relatively successful one-stage target detection algorithms include SSD[9] and YOLO[10] series, which have the advantage of fast detection speed but poor detection effect on dense and small targets. However, the defects on the circuit board in this paper belong to dense and small targets, so we do not consider this algorithm. Two-stage target detection algorithms with good results include R-CNN[11], Fast R-CNN[12], Faster R-CNN[6], Mask R-CNN[13], etc., which have the advantage of high detection accuracy. Because PCB defect detection in industry needs to accurately detect all kinds of small PCB defects, this paper adopts the Faster R-CNN algorithm in the two-stage detection algorithm as the basis to improve the detection of PCB small defect targets. The process of Faster R-CNN algorithm is described below before the improvement.

### 3.1. Faster R-CNN algorithm

The overall process of Faster R-CNN algorithm is as follows, and its framework is shown in Figure 3:

1. Firstly, the shortest side of the input image is scaled to 600 according to the original aspect ratio. At this time, the other side is automatically scaled according to the original aspect ratio. In this way, the image is not distorted and the original information in the image is retained.
2. The scaled image is input into the main feature extraction network, and the feature map is obtained after a series of convolution operations.
3. The feature map is input to generate the suggestion box network, in which the feature map is processed by sliding window, and the suggestion box and the probability of having or not having a target are obtained by category prediction and position prediction.
4. The feature map and the suggestion box are input into the ROI Pooling layer. Due to the different dimensions of the suggestion box, the size of the local feature map generated after mapping to the feature map is also inconsistent, which affects the unified management of data by the code. Therefore, the Pooling operation is carried out in the ROI Pooling layer to solve the above problems. The size of this series of local feature maps is adjusted to the same size and then spliced onto the same channel.
5. This series of local feature maps are flattened and then input into the classifier for prediction and the regressor for prediction. The prediction bounding box and the confidence score are obtained.

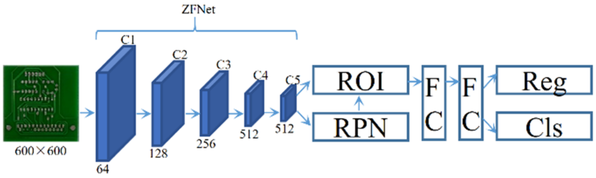

**Figure 3**: Faster R-CNN framework diagram

## 3.2. Improved feature extraction network

Feature extraction in object detection task is to extract the details of the image, such as color, contour, texture and edge information. Using these features as the input of the detection algorithm can reduce the overhead of the algorithm in time and space. Many excellent feature extraction networks have been derived by improving and stacking modules such as convolutional layer, fully connected layer, activation function and pooling layer. In 1998, LeNet excellent performance in the visual task of handwritten digit recognition drew people's attention to the power of convolution [14]. In 2013, ZFNet explained how the convolutional neural network works and demonstrated the functions of the intermediate feature layer and the operation process of the classifier by using the visualization technology [15]. This network is also the feature extraction network used by the traditional Faster R-CNN. In 2014, VGGNet uniformly adopted a convolution kernel size of 3×3 in order to simplify the selection of hyper parameters in the convolution layer, which simplified the design of the model and the number of network parameters and deepened the depth of the convolutional neural network, but at this time, the network depth was up to 19 layers [16]. In 2015, ResNet was proposed to solve the problem of gradient disappearance caused by network layer stacking, so that the network layer number can be deep [17]. In this paper, ResNet50 is selected as the basis of the backbone feature extraction network and the idea of deep separable convolution is introduced [18]. We call the improved feature extraction network DS-Resnet50.

As shown in Figure 4, Depthwise Convolution and Pointwise Convolution divide Depthwise Convolution into two stages. The former uses three single-channel convolution pairs to carry out two-dimensional convolution with step size of 1 and padding of 1 to extract spatial information in the direction of length and width of a single feature layer. Based on the output of Depthwise Convolution, the latter uses N three-channel 1×1 Convolution to perform 3D Convolution with step size of 1 and padding of 0 to make up for the lost cross-channel channel information of the former. Therefore, the number of parameters and the calculation amount are the sum of the two, as shown in Equations (3) and (4).

$$Params_{dp} = 9N_{in} + 3N_{out} \tag{3}$$

$$FLOP_{dp} = 9H * W * N_{in} + 3H * W * N_{out} \tag{4}$$

The formula of parameter number and calculation amount of standard convolution is (5), as shown in (6).

$$Params_{st} = 9N_{in} * N_{out} \tag{5}$$

$$FLOP_{st} = H * W * Params_{st} \tag{6}$$

By comparing the number of parameters and calculation amount of the above two operations, it can be seen that when the number of channels N in the feature extraction network is large, the depth separable convolution can effectively concentrate the training model and reduce the redundancy of the algorithm compared with the standard convolution.

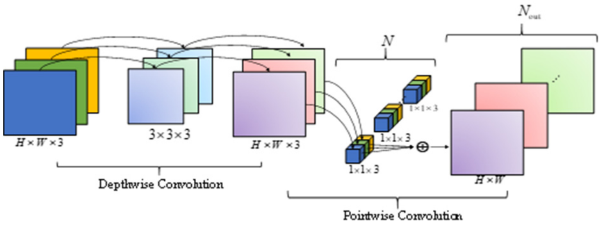

**Figure 4**: Schematic diagram of depth-separable convolution

We input images with holes missing defects that need to be detected into the improved feature extraction network. Five feature layers of different sizes are the inherent multi-scale pyramid structure of deep convolutional network, and the extracted features can be seen by using visualization technology on different feature layers. In Figure 5, we can clearly see that the features of defects are clearly distinguished, which provides help for subsequent defect detection. However, in the operation process of the trunk feature extraction network, the small target details of the bottom feature layer are gradually replaced by the whole information through the pooling layer, which makes our PCB defect detection task more difficult. Therefore, we improve it in Section 3.3.

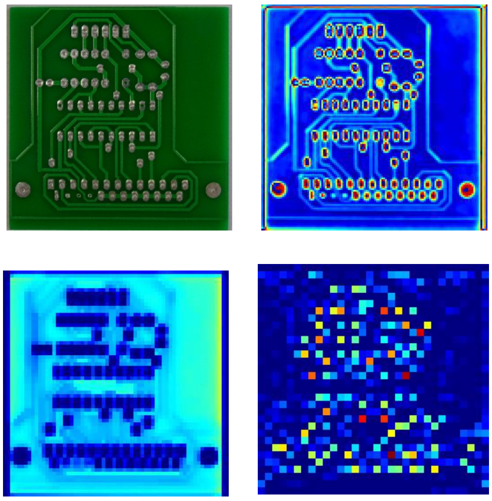

**Figure 5**: Original and depth feature maps

## 3.3. Improved multi-scale feature fusion

The results obtained by observing the same object at different distances belong to multi-scale, and the feature layers in each stage of deep learning field are called multi-scale. Because PCB defects vary in size, we use characteristic pyramid network to accurately detect PCB defects of various sizes.

Feature pyramid network integrates feature maps from different layers in horizontal and vertical dimensions [19], as shown in Figure 6, which is mainly used to solve the shortcomings of PCB defect detection algorithm when dealing with multi-scale changes. In Faster R-CNN, detection is based on the last feature layer. An obvious defect of this approach is that it is not friendly to detect small targets, because the underlying feature map has high resolution, small receptive field and strong representation ability of geometric details, which is helpful to the localization function of target detection task. However, the high-level feature map has low resolution, large receptive field and strong representation ability of semantic contour information, which is helpful to the classification function of object detection task. If only the last layer is used for detection, it obviously does not make full use of the underlying geometric details, resulting in unsatisfactory detection effect of small objects.

The bottom-up path is the sequential execution process of the algorithm, and the feature map generated by DS ResNet50 network is divided into four pyramid levels C2, C3, C4, C5 according to its size. Due to the high resolution and a large number of parameters. Conv1 is not considered to be included in the pyramid level because its large memory footprint affects the real-time performance of the algorithm.

The top-down path is performed by enlarging the smaller feature map to the same width and height as the neighboring feature map by a 2 times up-sampling operation. Lateral linkage refers to the idea of residual network, and the feature map obtained by up-sampling in the previous layer is added to the feature map obtained by correcting the number of channels in the current layer. We fuse the fusion information from the top layer based on the bottom layer C2. The final multi-scale feature layers P1, P2, P3, P4 and P5 are formed by effectively combining the bottom level geometric detail information and the top-level semantic contour information. Finally, the generated multi-scale feature layers are input into the ROI Pooling layer to generate a series of local feature maps with the same size of 7×7.

After flattening, a vector is obtained, and the final detection map is obtained through classifiers and regressors.

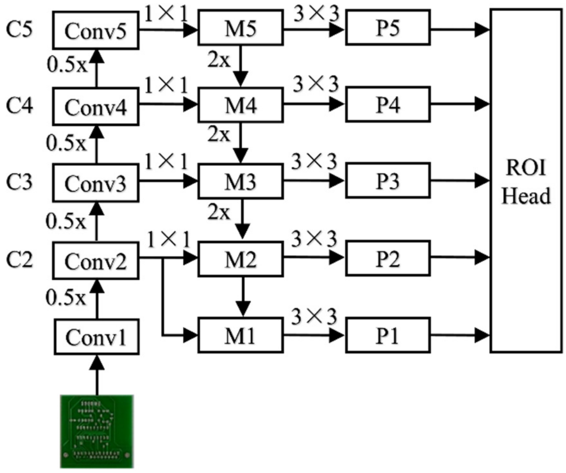

**Figure 6**: Characteristic pyramid structure

## 4. Experiment and Analysis

### 4.1. Experimental environment

Considering the maturity of the deep learning framework and the ability of the environment to schedule hardware resources, this program is programmed in Python language, the experimental platform is Pycharm, PyTorch1.7.1 is used as the deep learning framework, and the running environment is configured as Ubuntu16.04 and CUDA10.0.130, Cudnn7.6.5, Quadro P5000 GPU, 16GB of video memory. The CPU is Intel Xeon CPU E5-2699.

### 4.2. The experimental data

The PCB DATA SET used in this paper is from Peking University [2]. After data enhancement processing, a total of 10668 images were generated in the data set, which contained six kinds of PCB bare board defects, namely missing hole, mouse bite, open, short, spur, and spurious copper.

### 4.3. Model training

15 anchor boxes were obtained by using the improved K-means++ clustering algorithm in this paper on the real box data in the PCB DATA SET. Its size, respectively [9,10]、[14,13]、[14,18]、[19,19]、[27,13]、[16,25]、[27,19]、[22,24]、[16,37]、[23,31]、[40,19]、[30,28]、[23,44]、[36,36]、[47,53]. Figure 7 shows the final generated 15 anchor, each color corresponds to a feature layer, and each color has three anchors with different proportions to adapt to detection targets with different aspect ratios.

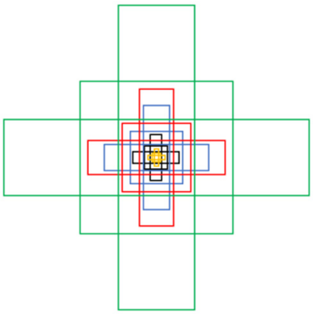

**Figure 7**: The resulting anchor

The hyper parameter initialization Settings in this article's code are shown in Table 1.

**Table 1**
Hyper parameter setting during training

| Hyper parameter Name | Value |
|---|---|
| Epoch | 15 |
| Learning rate | 0.01 |
| Momentum | 0.9 |
| Weight decay | 0.0001 |
| Batch size | 8 |
| Gamma | 0.33 |
| Optimization Type | SGD |

## 4.4. The evaluation index

The evaluation indexes used in this paper are Accuracy (A), Precision (P), Recall (R), Frames Per Second (FPS), weighted harmonic average (F1), and detection Accuracy (mAP). Such as formula (7) - (11).

$$A = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \tag{7}$$

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{8}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{9}$$

$$F_1 = \frac{2PR}{P + R} \tag{10}$$

$$mAP = \frac{\sum_{i=0}^{n} AP(i)}{n} \tag{11}$$

Where: $N_{TP}$ is the number of samples with positive prediction results and positive actual samples. $N_{FP}$ represents the number of samples with positive prediction results and negative actual results. $N_{TN}$ is the number of samples with negative prediction results and negative actual results. $N_{FN}$ is the number of samples that were predicted to be negative but actually were positive. FPS is the number of images transmitted per second, and F1 is the average of P and R.

## 4.5. Results analysis

Experiment 1 is to verify the detection effect of anchor obtained by the improved K-means++ clustering algorithm in PCB defect detection algorithm. We compared the detection speed and accuracy of anchor obtained by different clustering algorithms in the same detection algorithm and the same data set, as shown in Table 2.

**Table 2**
Different clustering methods are used to check the speed and accuracy

| Model | Clustering method | mAP/% | Time/s |
|---|---|---|---|
| Faster-RCNN | Standard clustering | 86.4% | 0.189 |
| Faster-RCNN | My clustering | 90.7% | 0.131 |
| My Model | Standard clustering | 91.8% | 0.129 |
| My Model | My clustering | 95.6% | 0.125 |

From the data in bold in Table 2, it can be seen that the clustering method in this paper and the improved detection algorithm have been used to improve the accuracy and speed. The detection accuracy was improved by 9.2%, and the detection speed was improved by 0.064s. This is due to the design of suitable anchor and the improvement of detection algorithm.

In Experiment 2, the traditional Faster R-CNN algorithm was compared with the proposed algorithm in the same PCB DATA SET and the same hardware conditions by using the single variable comparison method. Figure 8 shows the detection result diagram of the traditional Faster R-CNN algorithm. It can be seen that 2 - defects were missed in the notch diagram, 1 - in the open circuit diagram, 1 - in the short circuit diagram and 3 - in the residual copper diagram. It can be seen that the original Faster R-CNN algorithm did not have a good detection effect on minor defect targets and could not accurately detect the defect positions in the diagram.

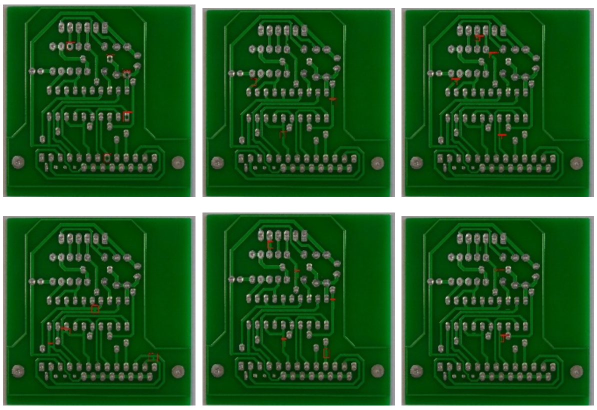

**Figure 8**: Traditional Faster R-CNN detection results

Figure 9 is the detection result chart of the algorithm in this paper. It can be seen that the improved algorithm can mark all the 6 types of defects contained in the figure. Compared with the standard Faster R-CNN algorithm, it is more suitable for small target detection and has significantly improved the missed detection rate and confidence.

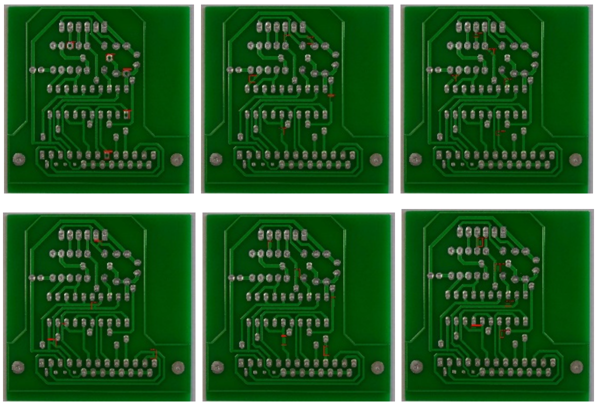

**Figure 9**: The detection results of the proposed algorithm

## 5. Acknowledgements

I would like to thank the National Natural Science Foundation of Guilin University of Electronic Technology (11901137) and Guangxi Young Teachers' Basic Scientific Research Ability Improvement (2019KY0232) and Graduate Education Innovation Program of Guilin University of Electronic Technology (2019YCXS093) for the financial support. I would like to thank the members of Guangxi Key Laboratory of Automatic Detection Technology and Instruments for their help. Finally, I would like to thank Aleksandr Ometov for providing the writing template.

## 6. References

[1] Jifeng Li, Research and implementation of PCB surface defect detection technology based on deep neural network, Master's thesis, Foshan Institute of Science and Technology, Guangdong, China, 2020.

[2] D. Runwei, D. Linhui, L. Guangpeng. "TDD-net: a tiny defect detection network for printed circuit boards." CAAI Transactions on Intelligence Technology (2019): 110-116.

[3] Li Xie, Xiaofang Yuan, Baixin Yin. "Defect detection of circuit board components based on improved YOLOv4 network." Measurement and control technology (2022): 19-27.

[4] Wen Li, Xiaochun Li, Haolei Yan. "PCB defect detection based on improved YOLOv3." Electro optic and Control (2022): 106-111.

[5] Xianyu Zhu, Jie Xiong, Ningsha Wang. "Research on Defect detection method of PCB Bare Board based on improved YOLOv4." Industrial control computer (2021): 39-40.

[6] S. Ren, K. He, R. Girshick. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence (2017): 1137-1149.

[7] Yuxia Lai, Jianping Liu, Guoxing Yang. "K-means clustering analysis based on genetic algorithm." Computer Engineering (2008): 200-202.

[8] Yongliang Feng, Hao Li. "Research on K-means clustering improvement based on genetic algorithm." Computer and Digital Engineering (2020): 1831-1834.

[9] W. Liu, D. Anguelov, D. Erhan. "SSD: Single shot multi box detector." European conference on computer vision (2016): 21-37.

[10] R. Girshick, J. Donahue, T. Darrell. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition (2016): 779-788.

[11] J. Redmon, S. Divvala, R. Girshick. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition (2014): 580-587.

[12] R. Girshick. "Fast R-CNN." Proceedings of the IEEE international conference on computer vision (2015): 1440-1448.

[13] K. He, G. Gkioxari, P. Dollár. "Mask R-CNN." Proceedings of the IEEE international conference on computer vision (2017): 2961-2969.

[14] Y. L, L. B, Y. B. "Gradient-based learning applied to document recognition." Proceedings of the IEEE (1998): 2278-2324.

[15] M.D. Zeiler, R. Fergus. "Visualizing and understanding convolutional networks." European conference on computer vision (2014): 818-833.

[16] K. Simonyan, A. Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv (2014): 1409-1556.

[17] K. He, X. Zhang, S. Ren. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition (2016): 770-778.

[18] F. Chollet. "Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition (2017): 1251-1258.

[19] T.Y. Lin, P. Dollár, R. Girshick. "Feature pyramid networks for object detection." Proceedings of the IEEE conference on computer vision and pattern recognition (2017): 2117-2125.