IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 24, 2020, accepted June 6, 2020, date of publication June 10, 2020, date of current version June 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001349

Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network

BING HU AND JIANHUI WANG
School of Information Science and Engineering, Northeastern University, Shenyang 110004, China
Corresponding author: Bing Hu (hubing607@gmail.com)

ABSTRACT Defect detection is an essential requirement for quality control in the production of printed circuit boards (PCBs) manufacturing. The traditional defect detection methods have various drawbacks, such as strongly depending on a carefully designed template, highly computational cost, and noise-susceptibility, which pose a significant challenge in a production environment. In this paper, a deep learning-based image detection method for PCB defect detection is proposed. This method builds a new network based on Faster RCNN. We use a ResNet50 with Feature Pyramid Networks as the backbone for feature extraction, to better detect small defects on the PCB. Secondly, we use GARPN to predict more accurate anchors and merge the residual units of ShuffleNetV2. The experimental results show that this method is more suitable for use in production than other PCB defect detection methods. We have also tested in other PCB defects dataset, and experiments have shown that this method is equally valid.

INDEX TERMS Defect detection, deep learning, residual network, feature pyramid, ShuffleNetV2.

I. INTRODUCTION

As the electronic device parts are shrinking down to minute sizes, printed circuit boards (PCB) as a support for electronic components is becoming more and more sophisticated and delicate. PCB defects are one of the critical factors for a high defect rate of electronic equipment. Therefore, defect detection is an important quality control technique for printed circuit boards (PCBs) industry. Different PCB defects can be generated in various production processes, such as missing values, lacking components, mistaken open circuits, and short circuits, causing the yield to drop. Therefore, it is necessary to achieve non-contact, accurate, and efficient automatic defect detection in the PCBs production process.

In recent years, an automated optical inspection (AOI) technique has been using to detect the defect during the PCB manufacturing process [1]. Compared with traditional manual detection, it has a series of advantages such as high-speed detection, cost reduction, and accuracy. In the evolution of AOI technology in the past decade, the methods are mainly divided into three categories: reference comparison methods, non-reference inspection methods, and hybrid inspection methods. The most widely used method is the reference inspection methods. In this defect detection method, the correlation between the scene images and the two window portions of the reference image is calculated. The difficulty of this method is the precise alignment of the reference image and the testing image. Performing the alignment operation requires a complicated configuration process. At the same time, the detection process is susceptible to light and noise, and even small shadows can cause false alarms.

Compared with traditional machine vision methods, deep learning-based methods can automatically extract image features, simplify the image pre-processing process, and can effectively improve the accuracy and efficiency of object detection, which has attracted the attention of many scholars.

In this paper, we propose an effective learning-based method to detect PCB defects in run-time in the Surface-mount technology (SMT) generation line, which belongs to the non-reference category. It can be used to identify six types of defects in the PCB production process. Base on the experiment on our database, it is more accurate and faster than other learning-based methods for detecting PCB defects. We focus on three challenges:

a) Deep learning methods are more capable of detecting large targets. In our scenario, the detect is always present with a small part of the image, so it is necessary to revamp the network structures to get a good performance.
b) When using high-resolution pictures, the speed of the convolution detection method is slow and cannot meet the speed requirements of real-time detection, so it needs to be improved.

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

c) For the traditional AOI detection method, corresponding fixtures and cumbersome parameter settings are required, which entails a high cost. We need a new inspection process and abandoned expensive accessories to make it more suitable for production line use.

The rest of this paper is organized as follows. Section 2 reviews previous related research. Secondly, the associated methods used in this paper are described. In Section 3, we introduce the characteristics of defects and describe the overall structure of the network. And detail the improvements to the original Faster-RCNN in the feature extraction stage and the RPN stage. In Section 4, we evaluate the proposed network and compare it with other networks. Finally, we give conclusions about our proposed network and the direction of our future work in Section 5.

II. RELATED WORKS

In this section, we review the relevant methods of PCB defect detection and introduce two methods applied in our proposed network, which is used to extract the PCB features.

A. PCB DEFECTS DETECTION

The elimination-subtraction method is the mainstream detection method as a PCB defect detection technology before using machine learning. This method calculates the pixel distance between the target image and the template image to detect obvious defects such as open circuit, short circuit, magnetic flux leakage, etc [2], [3]. Hagi *et al.* [4] proposed a new method to improve the classification accuracy of electronic circuit boards. The method first calculates the difference between the test image and the reference image and then performs high-precision detection on the defect candidate region, and finally, it extracts the feature recognition authenticity defect. Xie *et al.* [5] proposed a method using statistical appearance modeling technology (SAM), which obtains a more advantageous template than a rigid template. Wang *et al.* [6] proposed the partial information correlation coefficient (PICC) method to improve the traditional normalized cross-correlation coefficient (TNCCC). These methods are advantageous for detecting certain defects and computationally efficient. Therefore, it is widely used in production. In these methods, it is necessary to control the deviation, color change, and reflectance change of the target image from the template. And even minor changes in the PCB design require reconfiguration of the template. It increases production costs. Therefore, feature matching is proposed as an improved classic reference method. It extracts more robust features from the entire image and establishes a registration mapping relationship. Malhi and Gao [7] proposed a feature selection method based on principal component analysis (PCA). It uses supervised and unsupervised methods to classify the defects of the bearings. Local binary patterns (LBPs) are also one of the methods commonly used for feature extraction. Tajeripour *et al.* [8] proposed a fabric defect detection method based on LBPs. The method is divided into training and detection stages. The pixel-by-pixel LBP operator is applied to defect-free fabric images to calculate the reference feature vector. By comparing with the reference feature vector, a threshold suitable for defect-free windows is found. Then, a defective window can be detected using this threshold. The method has multi-resolution and grayscale invariance and can be used for defect detection of pattern fabrics and non-pattern fabrics. Ibrahim *et al.* [9] proposed an image difference algorithm based on wavelet transform. The algorithm uses the Haar wavelet and considers several different layers. One conclusion of this article is that in the application of PCB visual inspection, the second stage Haar wavelet transform should be selected. The common of these methods is uses a large amount of prior information on features for object detection. Because these features rely on hand-crafted features, there are two shortcomings. (a) It may not be possible to describe complex image scenes and objects structure. (b) cannot adapt to new views and objectives, and its generalization ability is reduced. Therefore, object detection based on traditional feature extraction methods falls into a bottleneck period.

In recent years, some scholars have used Convolutional Neural Networks (CNNs) [10] as a feature extraction method for defects detection. CNN has obtained better results compared with traditional feature extracts methods. It can accurately capture defects regions without using any extra information. Besides, even if there are shadows or reflections, it can still work well to locate the boundary of the detected object area as it uses multi-level features as reference. Because of these advantages, the CNN-based object detection method refreshes the historical record on almost all existing data sets and becomes the mainstream method in object detection. Su *et al.* [11] proposed a neural-network approach for semiconductor wafer post-sawing inspection. They introduced and tested three types of neural networks: backpropagation, radial basis function network, and learning vector quantization. This method can effectively shorten the detection time to 1s per slice. Heriansyah *et al.* [12] manually designed various defect patterns representing corresponding defect types for training and testing. The results show the effectiveness of neural network-based defect classification technology. Ding *et al.* [13] proposed an approach is based on Faster-RCNN to detect tiny defects of PCB and achieved high precision. His method solves the shortcomings of deep convolutional networks in detecting small defect areas, obtains good experimental results on an open PCB defect database, and the method provides us with a good idea. Some scholars have used the method based on Faster-RCNN [14]–[16] in defect detection and achieved excellent results. However, in some studies, more attention has been paid to improving the accuracy of detection and ignoring the detection efficiency, so that real-time detection cannot be achieved in production.

Compared to traditional visual inspection methods using neural networks to detect PCB defects do not have many related types of research. One of the reasons is, the collection of the PCB defect database requires a long-time accumulation, which requires a lot of workforce and material resources.

In this article, we used a real PCB defect database that we collected from the production line as the training set and test set. The following section will introduce how we do the collection in detail.

B. GROUP CONVOLUTION AND CHANNEL SHUFFLE

Group Convolution was first proposed in AlexNet [10]. Due to limited hardware resources at the time. The author distributed feature maps to multiple GPUs for processing and finally merged the results. As shown in Fig.1 (a), the traditional convolution performs convolution operation on the input data's whole. For example, the input data size is: $H_1 \times W_1 \times C$ (H is the height, W is the width, and C is the number of the channel) and there are N convolution kernels, each of which has a size of $K \times K$, the number of channels is the same as the input, then the output data obtained by convolution is $H_2 \times W_2 \times N$, so the total parameter amount is $K \times K \times N \times C$. As shown in Fig.1 (b), the input data is divided into 3 groups (g = 3), and each group size is: $H_1 \times W_1 \times C_1/g$, the convolution kernels size is $K \times K \times C_1/g$, and the number of convolution kernels per group is N/g, and the total parameter amount is $K \times K \times N \times C_1/g$. The total parameter amount is reduced to 1/g of the original. However, there is a significant shortage of this method, which is that the connection between groups is ignored, which means the operation of merging only happens inside an individual group. In a bid to solve this, Xception [17], MobileNet [18], and other similar networks add an extra convolution layer with the size of 1×1 to merge the output from different groups. Similarly, some other networks, such as ShuffleNetV2, proposed a more advanced method called 'channel shuffling.' As shown in Fig.2 (b), the 'reorganization' of the feature map after group convolution ensures that the next group convolution input is from a different group so that information can be exchanged between different groups. Fig.2 (c) shows this ''uniformly disrupted'' process. The results show that the method effectively solves the problem of insufficient fusion of the features map without reducing the performance.

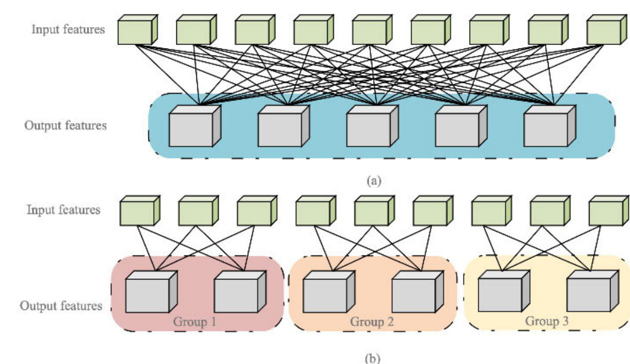

FIGURE 1. Comparison between the traditional convolution and the group convolution.

C. CHANNLE SPLIT AND DEPTHWISE CONVOLUTION

ResNet [19] is a residual learning framework, which is mainly used to solve a series of problems such as the deep neural network gradually saturates and then rapidly degenerates with the increase of network depth, including gradient dissipation and gradient explosion. It fundamentally breaks the symmetry of the network, thereby improving the ability of the representation network. The ShuffleNetV2 uses a residual block structure, such as ResNet. Instead of using group convolution, it splits the feature into two branches. The left branch is mapped equally, and the right branch contains three consecutive convolutions.

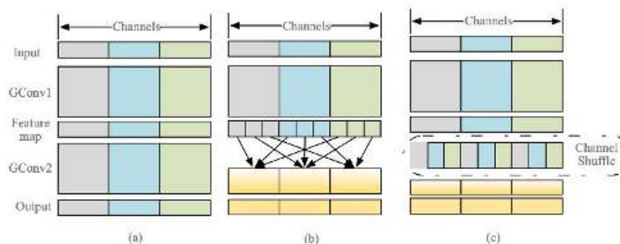

FIGURE 2. Channel Shuffle with two stacked group convolutions. (The figure is modified from [29]).

Depthwise Separable convolution (DWconv) contains two steps, Depthwise Convolution, and Pointwise Convolution. The first step is to convolve the input data with the convolution of the number of the same filters as the depth of the input data, while the second step is to convolve the input data with convolution kernel size 1×1. For example, in contrast to a traditional convolution operation, the input data size is $64 \times 64 \times 3$, and the convolution kernel size is $3 \times 3 \times 4$. The output feature map size is $64 \times 64 \times 4$ (assume that the input and output sizes are the same), the number of parameters of the convolutional layer can be calculated by the following formula $N_1 = 3 \times 3 \times 3 \times 4 = 108$. After using the 'DWconv' operation, the number of convolution layer parameters is $N_2 = 3 \times 3 \times 3 + 3 \times 1 \times 1 \times 4 = 39$. The same input also output 4 feature maps, the number of parameters of DWconv is about 1/3 of the conventional convolution. Therefore, under the premise of the same number of parameters, the number of neural network layers using 'DWconv' can be made deeper.

III. PCB DEFECTS AND DETECTION NETWORK

In this section, we observed and analyzed the characteristic of PCB defects and proposed a novel defect detection network for these characteristics. We will introduce the overall architecture of the network and detailed the core components: residual units and multi-scale regional proposal. Finally, we will introduce the industrial deployment of the network.

A. PCB DEFECTS

We focus on six common types of PCB defects in our research: (a) open circuit, (b) short course, (c) mouse bite, (d) spur, (e) pinhole, (f) solder ball, as shown in Fig.3. First, defects often only account for a small part of the PCB. Second, different mechanisms cause these defects, thus showing different characteristics, mainly including (color characteristics, shape characteristics, regional characteristics).

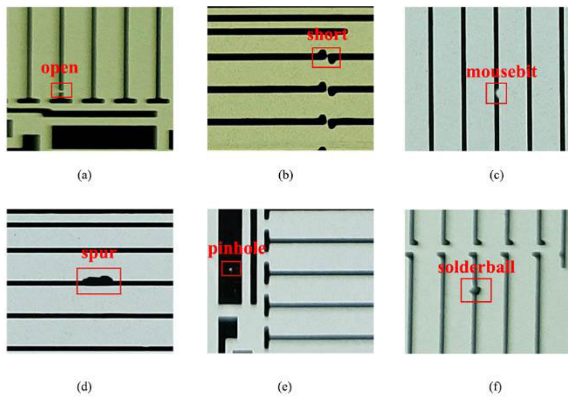

FIGURE 3. Example images of six types of PCB defects.

For example, compared with the normal mode, the short-circuit mode reduces the number of areas contained in the image. Similarly, for the open-circuit mode, the number of zones will increase. Mouse bite refers to the irregular notch on the edge of the line after etching, just like the bite by a mouse. Pinhole is due to the adsorption of hydrogen on the surface of the plated article, and it will not be released slowly. The plating solution cannot wet the surface of the plated parts so that the electroplating layer cannot be electrolyzed. As the thickness of the coating around the hydrogen evolution point increases, a pinhole is formed in the hydrogen evolution point and characterized by a shiny round hole, sometimes with a small upward tail. Solder balls will narrow the distance between the wires due to the protrusion. We need to design a reasonable network for these characteristics.

B. ARCHITECTURE OVERVIEW

Compared with the traditional convolutional neural network, which only uses strong semantic information, our proposed method also takes advantage of detailed information that contains weak features due to the PCB defects often appear with different properties. Therefore, we adapt the multi-scale features to fuse information from a multi-scale context.

The overall network framework is shown in Fig.4. The input of the model is an image in RGB format. We adopt Faster RCNN as the detector and ResNet50 [19] as the backbones. We applied the feature pyramid network (FPN) to the feature extraction part to merge deep features and shallow features, which can also improve the accuracy of small defect detection. Since a deeper network ResNet50 is used, we use Shuffle V2 residual units to replace the basic residual units to decrease the computation of the whole network. Further, we use the GARPN to gain a more accurate region proposal and reduce unnecessary anchor points, and then use ROI pooling to get object proposals. After that, use the fully connected layer to classify and bounding box regression to achieve the final defect detection results.

C. FEATURE EXTRACTION

FPN generates feature pyramids with robust semantic information at various scales to get more useful details on small objects without increasing the amount of calculation and the occupied memory significantly so that the accuracy of detecting minor defects can be dramatically improved. It uses convolutional neural networks to generate a set of hierarchical features that encode semantic information at different scales in the pyramid. The different levels of features in this hierarchical pyramid represent the objects in the image and their contextual information from different views. In our network, we use conv2, conv3, conv4, and conv5 blocks to build the pyramid model's feature map from the backbone. The reason why conv1 is not included is that it is too close to the input data, and a vast memory footprint. We choose the output of the last residual unit of each block as the bottom-up feature map, and constructed the top-down feature map by up-sampling the spatial resolution by $2\times$, which is expressed as $M2$, $M3$, $M4$, $M5$, corresponding to conv2, conv3, conv4, conv5. After the corresponding bottom-up feature map is convoluted by 1×1 to reduce the channel size and add the up-sampled map to the corresponding bottom-up map element by element. The shallow feature map contains more accurately localized information because it has not been down-sampled many times, [2]. Finally, a 3×3 convolutional layers are appended on each merged map to generate the final feature map. With purpose of eliminating the aliasing effect of up-sampling. The last sets of feature maps are denoted as $P2$, $P3$, $P4$, and $P5$ corresponding to conv2, conv3, conv4, and conv5, which are of the same spatial sizes but with more semantic information, as shown in Fig.5. Unlike the original RPN [20], where classification and bounding box regression is performed only on a single-scale signal scale, in our network, RPN takes multi-scale features as input. We will discuss it in the following subsections.

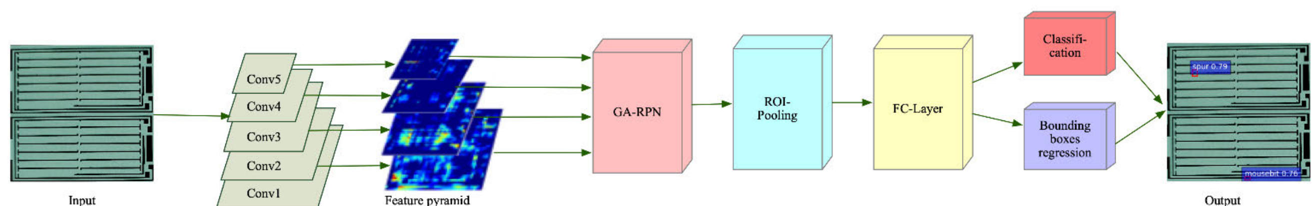

FIGURE 4. The overview of the network architecture.

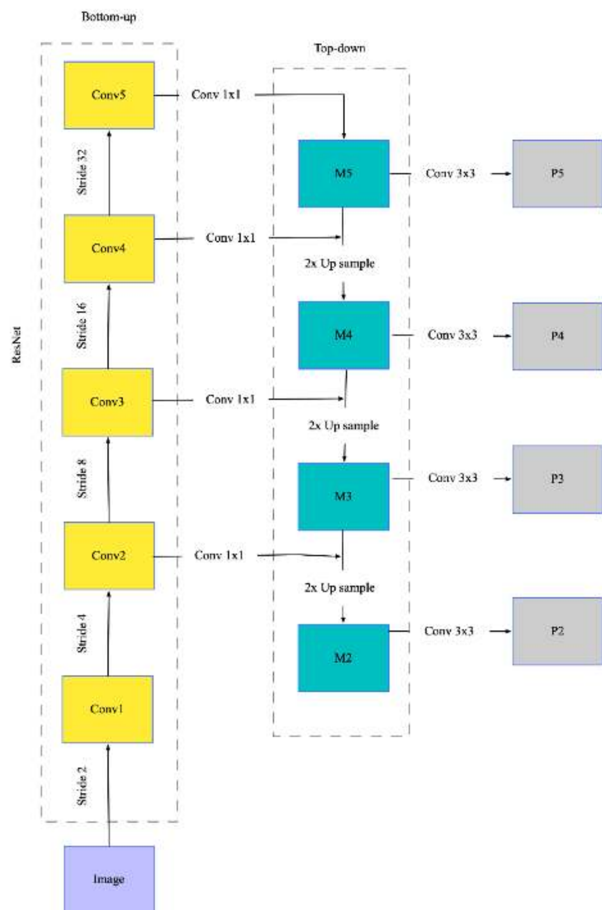

FIGURE 5. The feature pyramid structure used in this paper.

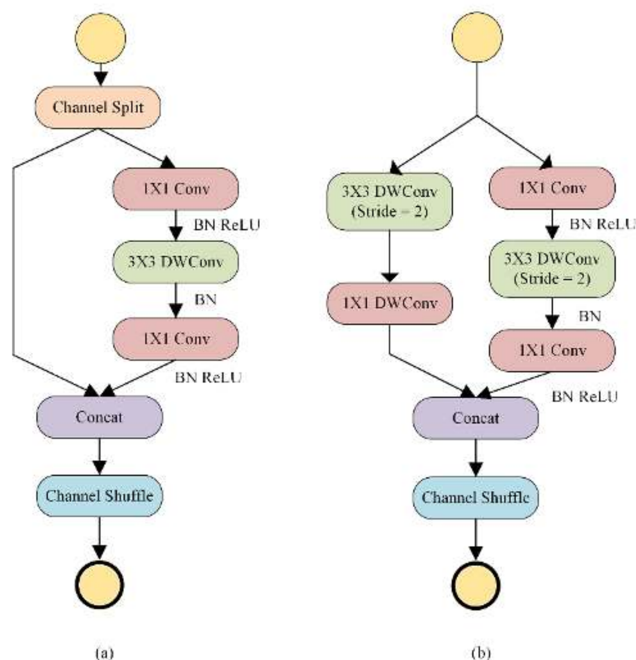

FIGURE 6. Residual units. (a) is basic residual units (b) is spatial down-sampling residual.

D. RESIDUAL BLOCK

The ResNet results on the ImageNet dataset [21] show that the classification performance of the residual structure is significantly better than the traditional convolution framework. This is because ResNet has a deeper convolutional layer. The small objects are unable to obtain salient features due to the downsampling effect in the traditional convolutional neural network. Furthermore, ResNet uses residual learning to connect the deep feature map and the previous shallow one. The high-level and low-level features are effectively utilized to combine their advantages, which can better adapt to the detection of small targets. In this paper, we use the residual unit structure based on ShuffleNetV2 to accelerate the network. In the network, each block consists of several basic units and a spatial down-sampling unit. As shown in Fig.6(a). At the beginning, "Channel Split" divides the channel dimension of the input data into two branches. One branch remains unchanged, while the other is add three extra convolutional operations. After merging the output of the two branches by the "Concat" operation, they can exchange information by "Channel Shuffle". In Fig.6(b), "channel splitting" is removed, and spatial down-sampling is performed with stride = 2.

E. MULTI-SCALE RPN

Rather than setting different scale anchors on the final map in Faster-CNN, out proposed network assigns multi-ROI of similar size on different levels of the feature map. These anchors will interfere with choosing the right ROI to get the necessary information. So, it does not have to set multi-scale anchors at each feature level. It is common practice to use a single scale anchor on each feature map. For example, the corresponding scale of anchors on $P2, P3, P4, P5$ is $8^2, 16^2, 32^2, 64^2$ and the multiple aspect ratio remains $1:2$, $1:1$, $2:1$. Therefore, each level feature map pixel will generate 3 anchors in the original picture. However, most of these anchors are still wrong and do not contribute to ROI.

In this paper, we used "Guided anchor," which was proposed by Wang *et al.* [22]. It consists of two branches and a "feature adaption as shown in Fig.7. This method can produce a small number of useful anchors. Firstly, In the position prediction branch, on the feature map at each level, the area corresponding to the center of the ground truth box is divided into the object center area, the ignored area, and the negative area according to the distance from the center. To make the feature map of each level only valid for target objects with a fixed scale range, the same area of adjacent levels is defined as the ignored area. Then at the process of position prediction, a small part of the region can be selected as the candidate center point position of the anchor, which significantly reduces the number of anchors. After predicting the position, masked Conv is used instead of Conv. The calculation is only performed at the anchors, which can accelerate the network.

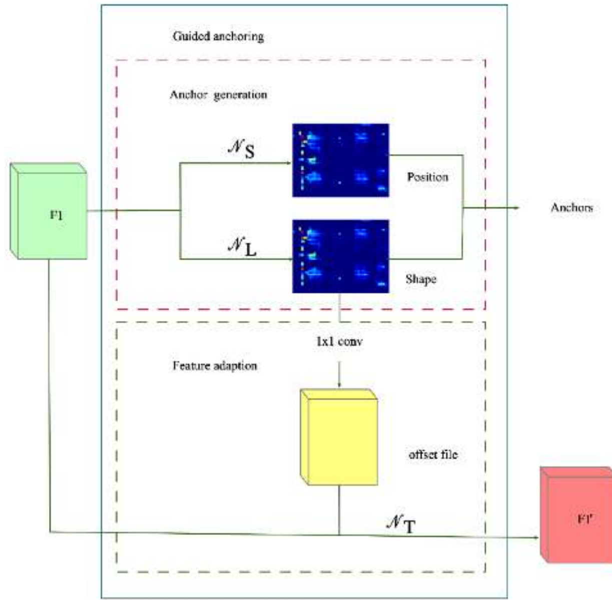

FIGURE 7. Agriculture of GARPN.

The purpose of the shape prediction branch is to predict the w and h of each anchor at a given center point using IoU as supervision. Different from original RPN, w and h here are variable and not predefined. However, the search range of w, h, for the anchor is too extensive, and it is challenging to predict directly. Therefore, use the following formula "(1)," for conversion.

$$w = \sigma \cdot s \cdot e^{dw}, \quad h = \sigma \cdot s \cdot e^{dh} \tag{1}$$

The dw and dh output from the shape prediction branch can be mapped to (w, h), where s is the stride, and σ is a hyperparameter, which is set to 8 in this paper. With $s = 32$ down-sampling feature map, dw and dh in the range of $[-1, 1]$ can search for objects in the range of [94, 696], and other levels of feature layers are similar. Finally, the anchor shape of each position of each levels of feature layers is obtained (one position has only one anchor).

After predicting the anchors' shape of each position, we use deformable convolution to transform it and integrate the shape information into the feature map, so that the new feature map can be adapted to the shape of the anchors at each position. The offset field is predicted in the shape prediction branch. Then the original feature map with offset is subjected to deformable convolution to obtain adapted feature maps, and then classification and bounding box regression are implemented on this basis.

F. INDUSTRIAL DEPLOYMENT

In this work, we use a deep learning-based method for PCB defect detect system. One of the most significant advantages is that it cuts off the effort in designing pre-templates and pre-configurations, and it also eliminates the reliance on jacks and fixtures that make them tightly aligned. In Fig.8, we compare the traditional system flow with the flow of the system proposed in the paper. We omitted the most time-consuming template configuration process and did not use costly fixtures to align them. After the industrial camera capture the PCB image to be tested, it is sent to the defect detect program. The type of defect is classified by the frozen weight file and finally output to the display terminal for manual inspection.

We apply the detection algorithm proposed in this paper to the industrial production environment. The programmable logic controller (PLC) drives the industrial camera to move quickly on the PCB and sends a part of the PCB image captured each time to the inspection unit. After the test, the system outputs the results containing the original image and the coordinates and types of defects to the center console, which is manually reviewed to distinguish the qualified and unqualified products and then flow to the next process.

To ensure that the details of the acquired images are rich enough to avoid unrecognized defects due to low resolution. The resolution of our industrial camera is 2592×1944, and the camera's field of view is 50mm $\times$ 40mm. Therefore, the size of each pixel is 0.02mm $\times$ 0.02mm. We control the camera to move horizontally and vertically, capturing six images per PCB. Furthermore, there is a 10% overlap between each adjacent picture, to avoid distortion caused by the edges of the image and not to capture the complete defect information. So, our system is suitable for detecting PCBs with a maximum size of 120mm $\times$ 100mm. Then we sent each picture to the distributed detection system to perform parallel processing of multiple model detection using multiple processes. In each detection task, we use our proposed network that has been trained and frozen.

IV. EXPERIMENTS AND DISCUSSION

We performed the above method to the PCB production environment to validate whether the expected goals can be achieved. All experiments were implemented in Python3.6 using a model developed based on Tensorflow2.0 [23], which provides a library for building an architecture for deep learning models. The experiment was performed on an Intel Core i7-8700K CPU @ 3.70 GHz, NVIDIA GeForce GTX 1080 GPU, and 16G RAM on Windows10.

A. DATA COLLECTION AND ANNOTATION

The data set was retrieved from a LED electronics factory in Wuxi, China, 2019. We use a CCD camera to take pictures for each PCB and manually screen out the pictures containing defects. We collected 1750 images with a resolution of 2592×1944, which includes multiple defects in them. Since this high resolution may cause a large amount of computation, we picked up all the defects in each picture and resized them into 600×600.

To balance the data samples of various types of defects, we removed some of the defects at the edges and unclear pictures to avoid adverse effects on training and finally cropped

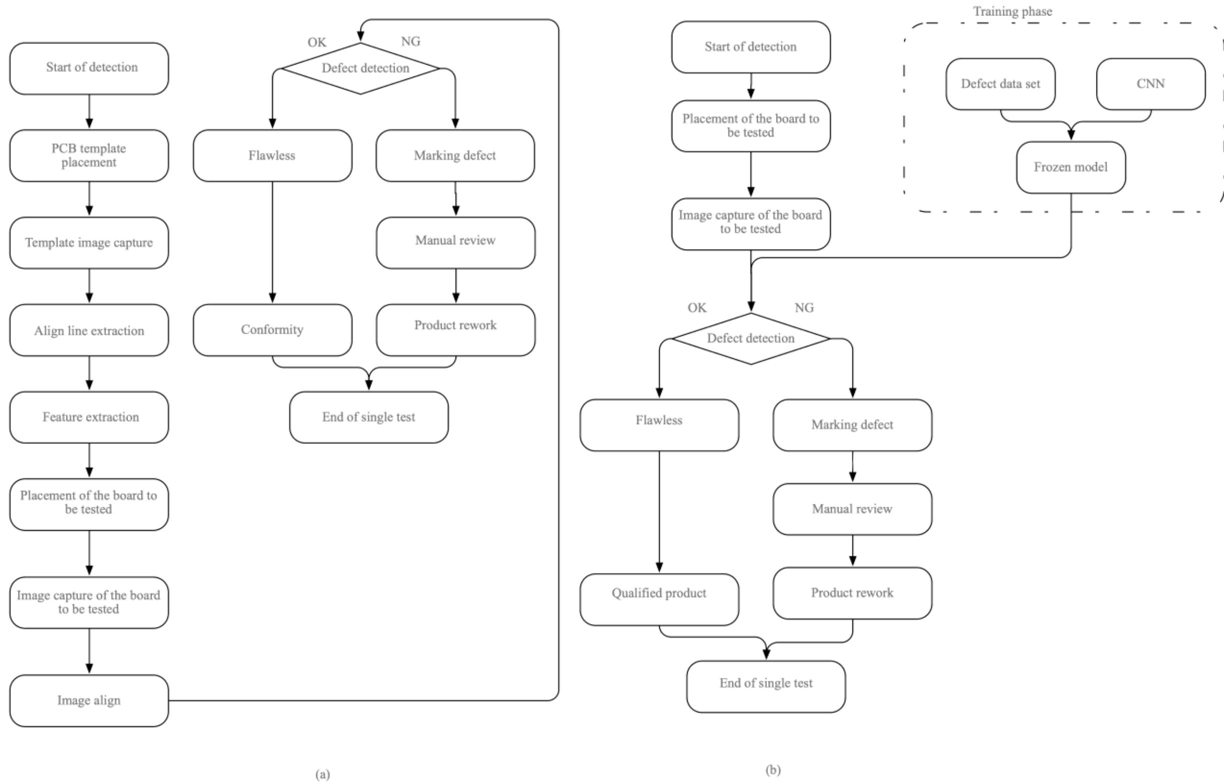

FIGURE 8. Traditional defect detection process and our propose system flow. (a) traditional system, (b) our system.

out 1500 defect pictures. However, in the production process, defects are generated as low-probability events, the labor and time costs of collecting defect data are high, so we use data enhancement methods to increase the number of PCB defect images. Rotation and brightness adjustment were introduced to our dataset to wrap the original data. Finally, 12000 defect pictures in our dataset.

In the process of labeling data, we use the image annotation tool called LabelImage [24] to mark every defect in the image. After labeling each defect has a ground-truth bounding box and a class label. The tool will generate an XML file containing information about the annotation object and the bounding box of each image, and the XML file is used as the ground-truth label for the detection model. Each defect on the picture is labeled (x_1, y_1), (x_2, y_2) and type, where are the upper left and lower right corners of the defect bounding box, and the defect type. The type is an integer ID that follows the match: 0-open, 1-short, 2-mouse bite, 3-spur, 4-pinhole, 5-solder balls.

B. EVALUATION METHOD

The detection of the defect area is performed by the greedy overlap criterion of the ground truth bound (area(G)) and the candidate bound (area(C)), that is, cross-over-union (IoU), as shown in Eq.(2). It ranges from 0 to 1, where 0 means no intersection between the area(G) and the area(C), and 1 means they are identical. In this paper, the defect area detection acceptable threshold is 0.5.

$$IoU = \frac{\text{area}(C) \cap \text{area}(G)}{\text{area}(C) \cup \text{area}(G)} \tag{2}$$

The Mean Average Precision (mAP) is our primary indicator for evaluating model performance our main indicator for evaluate model performance. It is the average of the Average Precision (AP) values of C different defects, and it reflects the accuracy of defect detection

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{3}$$

The AP value is calculated by the precision rate and the recall rate. The general definition for the Average Precision (AP) is finding the area under the precision-recall curve above.

$$AP = \int_0^1 p(r)dr = \sum_{k=1}^{N} P(k)\Delta r(k) \tag{4}$$

Accuracy measures the number of samples correctly classified cent of all proportion the number of samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{5}$$

The Recall measures the ability of the model detection for positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

The Precision measures the accurate of the model prediction.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

C. RESULTS AND EVALUATION

In the experiment, we use the cross-validation method, which aims to extract as much information as possible from the limited data and avoid local extreme values [25]. In this process, both the training samples and the test samples are learned as much as possible. We randomly selected 10% samples of each subset as test data. The remaining 90% of the pictures were equally randomly divided into six parts. In each round of training, five parts were used as training sets, and 1 part was used as a validation set. The number of each defect in the database are shown in Table 1. A total of 40,000 iterations are trained, to avoid training time is too long, we set the learning rate to 0.1 when the training step is less than 5000, and if the training step is less than 20000, the learning rate is set to be 0.01. The larger learning rate can speed up network convergence in early training. In the remaining 20,000 training sessions, we used 0.0001 as the learning rate.

TABLE 1. Number of defects in the training, validation, and testing sets, respectively.

	OPEN	SHORT	MB	SPUR	PINHOLE	SB
TRAIN	1650	1542	1290	1404	1674	1440
VALID	330	308	258	281	335	288
TEST	220	206	172	187	223	192

Like the ShuffleNetV2, our proposed network can also set the complexity scale factor by changing the number of channels of each unit, so we evaluate the impact of using different values on the performance of the model by calculating the accuracy and training speed of the algorithm. For example, when s = 1, the standard network structure, when s = 0.5, the output and input channels of each stage are half of the number of channels in the standard network, and others are similar. The experimental results are shown in Table 2.

TABLE 2. The network with different complexity.

Scale factor (s)	0.25	0.5	1.0
Accuracy	0.871	0.878	0.896
Parameters	36M	98M	317M
Training speed (step/second)	12.6	10.2	7.3

By comparing the results of the network under different complexity, we conclude that as the complexity of the model increases, the parameters involved in the calculation will increase, and with a lower complexity of the model. At the same time, the accuracy of the model will increase. In different object detection tasks, it is often necessary to make various choices in terms of faster speed and higher accuracy. In the industrial environment, reliability is paramount, so in the subsequent experiments, we chose our network with a complexity factor of 1.0 as the detection model.

Here we compared our method with state-of-the-art defect detection algorithms, including the FasterRCNN, RetinaNet [26], and YOLOv3 [27] with different backbones, table 3 lists details of varying performance indicators (mAP, recall, Efficiency).

As we can see from table 3, deeper networks with combined FPN contribute most to the performance of the network. For Faster RCNN with Resnet101-FPN as the backbone network, our proposed network improved the mAP and recall increased by 3.4% and 3.3%, respectively, which verified that GARPN had a small backbone could achieve much better performance than RPN with larger backbones. Although, when using Resnet101 as the backbone of our proposed network can increase mAP by 1.4%, we did not use it, because, in an industrial production environment, we must make a balance between detection accuracy and detection efficiency. Yolo3 + MobileNet [18] has faster detection speed, which is also benefited from its use of the depthwise separable convolution, and the backbone network has only 28 layers. Still, it cannot meet the industrial requirements in terms of detection accuracy. Therefore, our network is more suitable in general.

We also obtained the precision and recall values under different confidence thresholds by calculating the precision-recall curve, as shown in Fig.9, our proposed network is better than the Faster RCNN and RetinaNet with the same ResNet50-FPN backbone. However, all of the networks we proposed cannot get a good result of detecting the defects with type "open", there are some false positives. We found that due to the different causes of the "open" defect, the image defects in the training set show different characteristics. The reasons for the formation of this defect is as the follows, (a) An Open circuit is produced by damage to the copper sheet and scratches, (b) Uncoppered in the production process due to poor plating or other reasons, (c) Open circuit at a specific location due to film damage. For (a), (c), the edge of the "open" position is rough, and the position forming the "open" is occasionally accompanied by trachoma or bubbles. These characteristics are available for learning. For (b), the resulting "open" tends to be edge smooth, so that it cannot be distinguished from the normally open path by feature learning. Therefore, we remove all the images belonging to the (b) type and re-add new (a), (c) type pictures to the defect data set, and then re-train. For (b), we detection by assisted non-deep learning method. After testing from the new training, the detection accuracy and recall rate was improved, and the false positive rate was also significantly reduced. The overall result was acceptable.

To verify that using GARPN can effectively reduce low-quality anchors while still being able to high-precision detection, we compare it with the traditional RPN method.

TABLE 3. Detection results on PCB data set.

Network	Backbone	mAP (%)	Recall (%)	Runtime (s/img)
Faster RCNN	ResNet50	85.2	72.6	0.12
	ResNet50-FPN	86.4	73.8	0.09
	ResNet101	87.1	78.5	0.31
	ResNet101-FPN	90.8	79.2	0.27
RetinaNet	ResNet50-FPN	86.5	76.3	0.10
	ResNet101-FPN	89.7	79.2	0.42
YOLOv3	MobileNet	70.6	72.9	0.052
Proposed method	ResNet50-FPN	94.2	82.5	0.078
	ResNet101-FPN	95.6	83.7	0.213

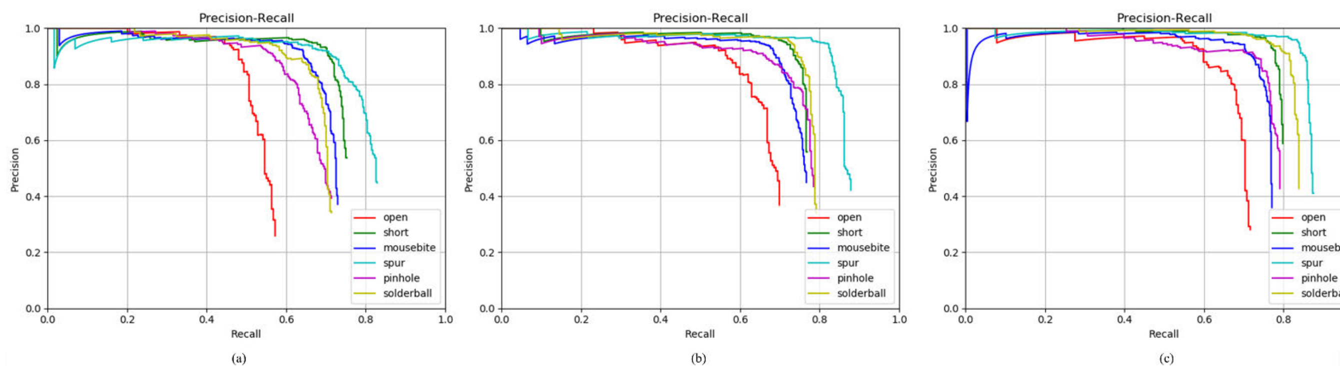

FIGURE 9. Precision-recall curve for each class drawn in different color. (a) Faster RCNN. (b) RetinaNet. (c) Our proposed.

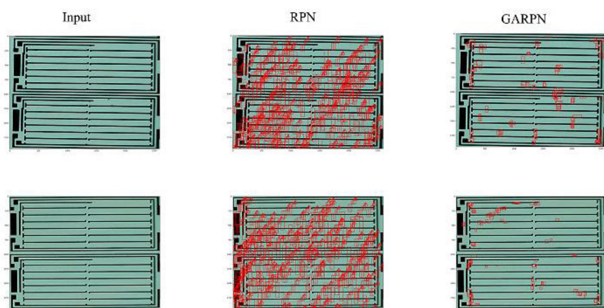

FIGURE 10. Examples of RPN proposals (middle) and GARPN proposals (right).

As shown in Fig.10, the anchors generated by GARPN can generate anchors around the defect location more accurately than the anchors generated by RPN+9 anchors (three scales and three ratios). Anchors generated by GARPN are densely distributed around the defect and sparsely distributed in other non-target areas.

We use ablation experiments to verify the advantages of the network proposed in this article. We designed 4 experiments.

a) The original Faster-Rcnn based on ResNet as the backbone.
b) The addition of FPN on faster-Rcnn+ ResNet50.
c) Added APRPN based on previous design.
d) The addition of ShuffleNetV2

Table 4 shows the ablation experiment results. The addition of the FPN has better accuracy in detecting small defects

TABLE 4. Ablation experiment on our network.

method(s)	Runtime (s/img)	mAP(%)	UP(%)
Original+ResNet50	0.12	85.2	
+FPN	0.09	86.4	1.2
+GARPN	0.087	91.2	4.8
+ShuffleNetV2	0.078	94.2	3.0

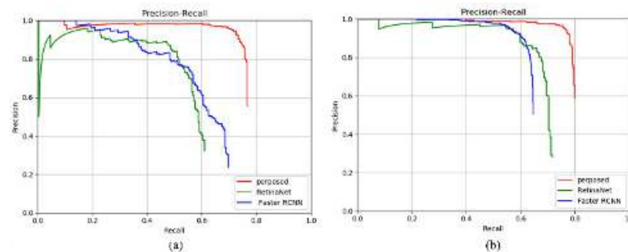

FIGURE 11. Precision-recall curve for different detector.

than the first network. After adding GARPN, the detection accuracy has been improved significantly, and the detection speed has also been improved. It can reduce 90% unnecessary anchors then RPN and generate more accurate proposals [22]. After using ShuffleNetV2, the network's detection efficiency is improved, which can meet the needs of real-time detection.

To verify the robustness of the method, we apply this method to an open PCB defect database (http://robotics.pkusz.edu.cn/resources/dataset/) [28]. The database contains 639 PCB images and 2953 defects that have been correctly labeled. It contains six types of defects. (missing hole, mouse bite, open circuit, short, spur, and spurious copper). We only

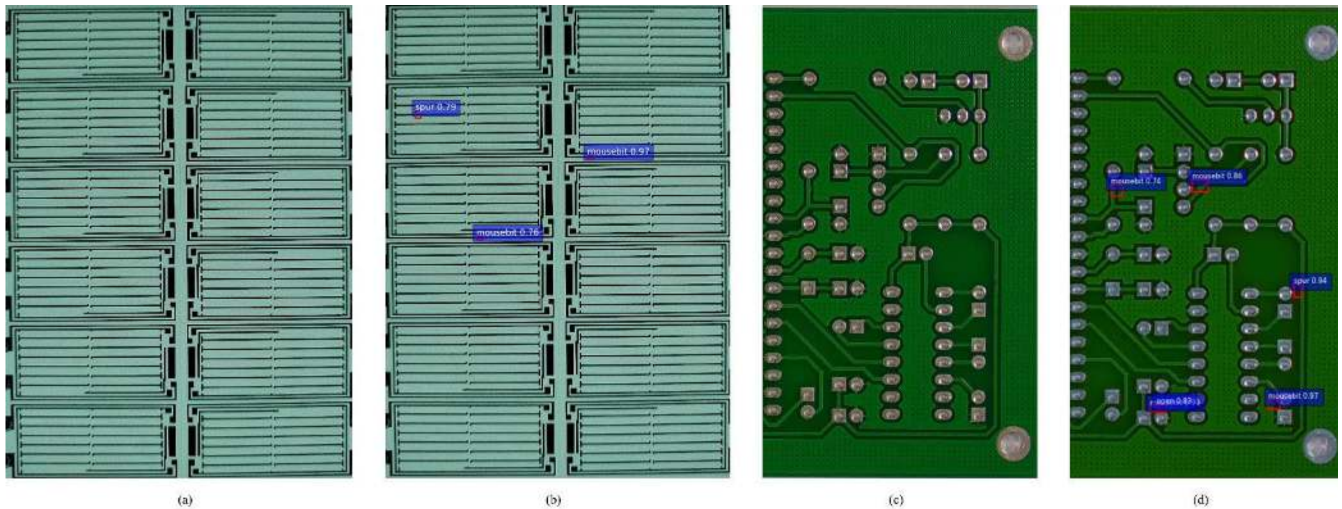

FIGURE 12. Detection results. (a) and (c) are the original pictures, (b) and (d) are test results.

used the same four types of defects as in our defect database. We first selected 400 images for each defect dataset for testing. We used three networks for comparison, the results are shown in Fig.11 (a). The results show that our proposed network has higher accuracy and recall than other networks, but the results are not as good as expected from the original test set. Although the defect characteristics context information of the defect caused by the different PCB, meanwhile the model is trained from a single data set, so the detection effect is not good.

In another experiment, we cropped 400 sub-images for each defect type in the dataset because one image may contain multiple defect types. A total of 650 images were collected and resize to 600 $\times$ 600. The collected pictures are added to the training and test data sets. As shown in Fig.11(b), the result is better than the above experiment. This may be because the model trained by adding a new defect picture to the training set incorporates more defect information, so it has better robustness in defect detection. The number of new effect pictures added in the new training set is too small compared to the original data set, the results are not as good as those in Experiment 2. However, they are still available. Some detection results of our method in Fig.12.

V. CONCLUSION

In this paper, a PCB automated inspection method based on a convolutional neural network is proposed. In response to the characteristics of PCB defects, we modified the original Faster RCNN. Firstly, we use a deeper backbone ResNet50 for feature extraction, and we used a Feature Pyramid Networks method to detect smaller defects better. We replace RPN with GARPN to generate anchors adaptively. To speed up the network, we also use the residual unit in ShuffleNetV2. In our method, no external mechanical fixtures needed and strict template alignment operations, which reduced testing cost. The experimental results show that the mAP of the improved model is 94.2, and the detection speed is 0.08s/img, which is improved by 9% and 0.042 s/img in accuracy and speed compared to the original Faster R-CNN with ResNet50, respectively. And it performs better than other detection networks for our database.

We also use other defect data sets for testing our network. When we use our pre-trained model, the detection performance has declined, which because even the same type of defects will show different characteristics due to different materials and processes in PCB production. When we join these pics to our training set, the result was as good as we expect. In the future, we will collect more defect samples to join the training set and fine-tune the network to adapt to defects detection of more types.

LIST OF ABBREVIATIONS

PCB Printed Circuit Boards
AOI Automatic Optical Inspection.
SMT Surface-Mount Technology
FPN Feature Pyramid Network
RPN Region Proposal Network
SAM Statistical Appearance Modeling
PICC Partial Information Correlation Coefficient
TNCCC Traditional Normalized Cross-Correlation Coefficient
PCA Principal Component Analysis
LBPs Local Binary Patterns
CNNs Convolutional Neural Networks
GPU Graphics Processing Unit.
DWconv Depthwise Separable convolution
RGB Red, Greeb, Blue.
ROI Regions of Interest.
PLC Programmable Logic Controller
CPU Central Processing Unit.
LED Light-emitting Diode.
CCD Charge-Coupled Device.
XML Extensible Markup Language

IoU Intersection over Union.
mAP Mean Average Precision
AP Average Precision
TN True Negative.
TP True Positive.
FN False Negative.
FP False Positive.
MB Mouse Bite
SB Solder Balls

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] W. Y. Wu, M. J. J. Wang, and C. M. Liu, ''Automated inspection of printed circuit boards through machine vision,'' (in English), *Comput. Ind.*, vol. 28, no. 2, pp. 103–111, May 1996.

[3] K. Sundaraj, ''PCB inspection for missing or misaligned components using background subtraction,'' *WSEAS Trans. Inf. Sci. Appl.*, vol. 6, no. 5, pp. 778–787, 2009.

[4] H. Hagi, Y. Iwahori, S. Fukui, Y. Adachi, and M. K. Bhuyan, ''Defect classification of electronic circuit board using SVM based on random sampling,'' *Procedia Comput. Sci.*, vol. 35, pp. 1210–1218, Aug. 2014.

[5] H. W. Xie, Y. C. Kuang, and X. M. Zhang, ''A high speed AOI algorithm for chip component based on image difference,'' in *Proc. Int. Conf. Inf. Autom.* New York, NY, USA: IEEE, vols. 1–3, Jun. 2009, pp. 948–953.

[6] C.-C. Wang, B. C. Jiang, J.-Y. Lin, and C.-C. Chu, ''Machine vision-based defect detection in IC images using the partial information correlation coefficient,'' (in English), *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 3, pp. 378–384, Aug. 2013.

[7] A. Malhi and R. X. Gao, ''PCA-based feature selection scheme for machine defect classification,'' *IEEE Trans. Instrum. Meas.*, vol. 53, no. 6, pp. 1517–1525, Dec. 2004.

[8] F. Tajeripour, E. Kabir, and A. Sheikhi, ''Fabric defect detection using modified local binary patterns,'' *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, Dec. 2007, Art. no. 783898.

[9] Z. Ibrahim, S. A. R. Al-Attas, and Z. Aspar, ''Analysis of the wavelet-based image difference algorithm for PCB inspection,'' in *Proc. 41st SICE Annu. Conf. (SICE)*, vol. 4, Aug. 2002, pp. 2108–2113.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''Imagenet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[11] C.-T. Su, T. Yang, and C.-M. Ke, ''A neural-network approach for semiconductor wafer post-sawing inspection,'' *IEEE Trans. Semicond. Manuf.*, vol. 15, no. 2, pp. 260–266, May 2002.

[12] R. Heriansyah, S. A. R. Al-attas, and M. M. A. Zabidi, ''Neural network paradigm for classification of defects on PCB,'' *Jurnal Teknologi*, vol. 39, no. 1, pp. 87–104, Dec. 2003.

[13] 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.

[14] Y. T. Li and J. I. Guo, ''A VGG-16 based faster RCNN model for PCB error inspection in industrial AOI applications,'' in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2018, pp. 1–2.

[15] J. C. P. Cheng and M. Wang, ''Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,'' *Autom. Construct.*, vol. 95, pp. 155–171, Nov. 2018.

[16] X. Xu, Y. Lei, and F. Yang, ''Railway subgrade defect automatic recognition method based on improved faster R-CNN,'' *Sci. Program.*, vol. 2018, Jun. 2018, Art. no. 4832972.

[17] F. Chollet, ''Xception: Deep learning with depthwise separable convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[18] A. G. Howard, M. Zhu, B. Chen, W. Wang, T. Weyand, M. Andreetto, H. Adam, and D. Kalenichenko, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' *Corrosion Sci.*, vol. abs/1704.04861, Mar. 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[20] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015, pp. 91–99.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[22] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, ''Region proposal by guided anchoring,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2965–2974.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and X. Zheng, ''TensorFlow: Large-scale machine learning on heterogeneous distributed systems,'' Apr. 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467

[24] *Tzutalin. LabelImg*. Accessed: Dec. 31, 2017. [Online]. Available: https://github.com/tzutalin/labelImg

[25] M. W. Browne, ''Cross-validation methods,'' *J. Math. Psychol.*, vol. 44, no. 1, pp. 108–132, 2000.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, ''Focal loss for dense object detection,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[27] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' *CoRR*, vol. abs/1804.02767, Feb. 2018.

[28] W. Huang and P. Wei, ''A PCB dataset for defects detection and classification,'' *CoRR*, vol. abs/1901.08204, Jan. 2019.

[29] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNet V2: Practical guidelines for efficient CNN architecture design,'' in *Proc. European Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 116–131.

BING HU received the M.S. degree in computer science from Northeastern University, Shenyang, China, in 2012, where he is currently pursuing the Ph.D. degree in control theory and control engineering with the School of Information Science and Engineering. From 2013 to 2018, he was a Research Assistant with the Institute of Microelectronics (IME), Chinese Academy of Sciences (CAS).

JIANHUI WANG was born in Liaoning, China, in 1957. She received the B.S., M.S., and Ph.D. degrees in electrical engineering from Northeastern University, Shenyang, China, in 1982, 1986, and 1999, respectively, where she is currently a Full Professor and a Doctoral Supervisor with the School of Information Science and Engineering. Her research interests include complex industrial process modeling, control and optimization, rehabilitation robots, and intelligent control. Her current research interest includes intelligent control theory and its applications.