HRIPCB: a challenging dataset for PCB defects detection and classification

Article *in* The Journal of Engineering · January 2020

DOI: 10.1049/joe.2019.1183

Weibo Huang
Peking University

The Journal of Engineering

The 3rd Asian Conference on Artificial Intelligence Technology (ACAIT 2019)

HRIPCB: a challenging dataset for PCB defects detection and classification

eISSN 2051-3305
Received on 31st October 2019
Accepted on 19th November 2019
E-First on 22nd May 2020
doi: 10.1049/joe.2019.1183
www.ietdl.org

Weibo Huang[1], Peng Wei[1] ✉, Manhua Zhang[2], Hong Liu[1]
[1]The Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Shenzhen, People's Republic of China
[2]Shenzhen Skyworth-RGB Electronic Co., Ltd., Shenzhen, People's Republic of China
✉ *E-mail: weapon@pku.edu.cn*

Abstract: To cope with the difficulties in inspection and classification of defects in printed circuit board (PCB), many methods have been proposed in previous work. However, few of them publish their datasets before, which hinders the introduction and comparison of new methods. In this study, HRIPCB, a synthesised PCB dataset that contains 1386 images with 6 kinds of defects is proposed for the use of detection, classification and registration tasks. Besides, a reference-based method is adopted to inspect and an end-to-end convolutional neural network is trained to classify the defects, which are collectively referred to as the RBCNN approach. Unlike conventional approaches that require pixel-by-pixel processing, the RBCNN method proposed in this study firstly locates the defects and then classifies them by deep neural networks, which shows superior performance on the dataset.

1 Introduction

Printed circuit board (PCB) is the fundamental carrier in electronic devices on which a great number of elements are placed. The quality of PCBs will directly impact the performance of electronic devices. To avoid the shortcomings of manual detection, for instance, easily being fatigued, low efficiency, automated optical inspection (AOI) has been widely used in industry. As PCB becomes more and more complicated because of trends towards higher precision and density, the tasks of detection and classification defects are also more difficult than before. Currently, although there are some papers on PCB defects detection, these papers use their own datasets. The lack of open datasets has led to the inability to evaluate various methods. For the purpose of solving above problems, a public colourised synthesised PCB dataset with defects that is available to other people who want to design and evaluate their approaches is presented in this paper.

Conventional AOI methods for inspecting PCB can be divided into three main streams: reference comparison approach, non-reference verification approach and hybrid approach [1]. Various methods on defects inspection and classification task have been proposed based on the three different approaches. An automated visual inspection system for PCB is introduced by Wen-Yen Wu *et al.* [1]. The system utilises an elimination-subtraction method which directly subtracts the template image from the inspected image, and then conducts an elimination procedure to locate defects in the PCB. Each detected defect is classified by three indices: the type of object detected, the difference in object numbers, and the difference in background numbers between inspected image and template. LI Zheng-ming *et al.* also use digital image processing technology based method to classify the defects by getting the number of connected regions, Euler numbers, area of defects of the template and inspected image, respectively [2]. The result of experiment shows that the method can achieve automatic real-time detection. Vikas Chaudhary *et al.* list 14 kinds of defects that belong to two types: positive, negative [3]. They segment the image into three parts: wiring tracks, soldering pads and holes, each defect can be classified by comparing pixels, number of connected components in the corresponding part. A non-referential based approach is proposed by Shashi Kumar *et al.* in consideration of the difficulties in registration [4]. In their work, inspected image is segmented into copper and non-copper parts to analyse separately, and a 3D colour histogram is utilised to capture the global colour distribution. The effectiveness of this model is evaluated on real data from PCB manufacturing industry and accuracy is compared with previously proposed non-referential approaches. A new technique that classifies the defects using neural network paradigm is introduced by Rudi Heriansyah *et al.* [5]. Various defective patterns representing corresponding defect types are designed and thousands of defective patterns have been used for training and testing. The result shows the effectiveness of defect classification technology based on neural network.

Due to the intuitiveness, simpleness and the development of computer hardware and algorithms, reference comparison methods are used to inspect defects in our approach. In addition, convolutional neural network (CNN) highlights outstanding performance in computer vision tasks, like classification, object detection, segmentation and so on. Therefore, in defect classification task, there is no need to search the features of the image, instead, an end-to-end neural network is introduced to classify the inspected defect regions. The flow chart of the whole experiment process is shown in Fig. 1. Test image and template will be separately preprocessed and compared to locate defects, then these located defects are sent into trained neural network model to get classification results.

There are some public datasets on printed circuit board assembly (PCBA), which is a kind of board after all the components and parts been soldered and installed on [6]. PCBA can accomplish the electronic function it is designed for. Inspection of PCBA is for the purpose of recycling when the PCBA is eliminated, however, it is not appropriate for our task because our target is naked PCB that has no components. In this paper, HRIPCB dataset that consists of naked PCB images is presented. Half of them are in right orientation as templates with different defects and the other half are manually rotated to simulate the situation when PCBs are not correctly placed. All the images originate from 10 standard template boards which are checked by human. Many PCB-related methods on detection, classification and registration problems can be conducted on this dataset, and various methods can be compared as well. The dataset is free available online (http://robotics.pkusz.edu.cn/resources/dataset/).

The paper is organised as follows. Section 1 introduces the backgrounds of PCB dataset and main stream methods on defects detection and classification. Section 2 details the procedure of image acquisition, labelling and defect statistics, evaluation metrics are also mentioned. Our reference comparison based method and the deep neural network model are given in Section 3. Section 4

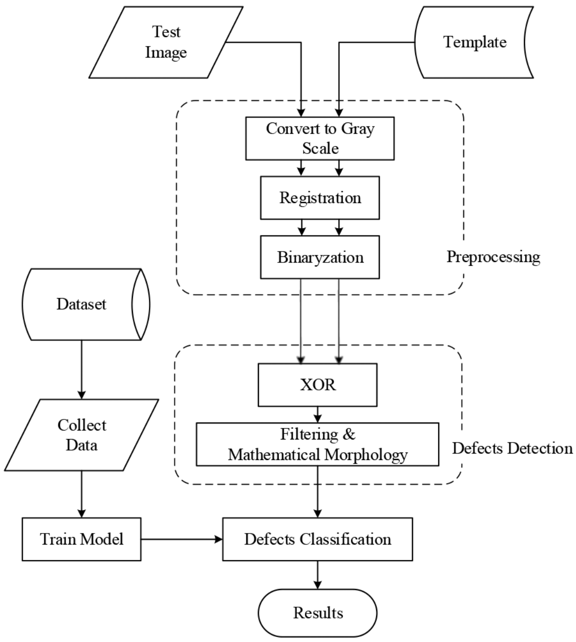


Fig. 1 *Flow chart for PCB inspection. Test image and template will be separately preprocessed and compared to locate defects, then these located defects will be sent into trained neural network model to get classification results*

Fig. 2 *PCB image acquisition system consists of light source, workbench, support, camera and image process unit*

presents experiment setup in this work and the experiment results are discussed. Conclusions are shown in Section 5.

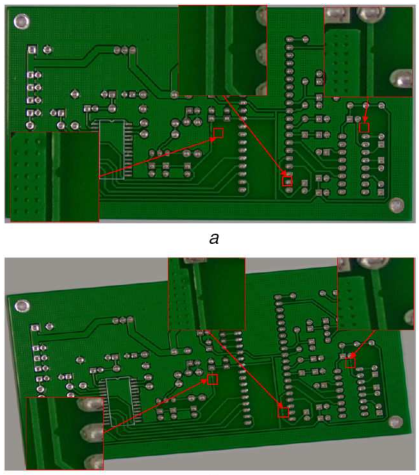
2 Dataset and evaluation

In addition to procedure and equipment related to image acquisition and dataset production, some statistics and evaluation metrics on the dataset are arranged in this section.

2.1 Image acquisition

To ensure the representativeness of the dataset, a PCB image acquisition system that resembles the practical AOI system used in inspection process is built, as is shown in Fig. 2. The images of template boards are captured by a 16-megapixel HD industrial camera equipped with CMOS sensor that can be controlled by computer software or a remote control. In order to adapt to different PCB sizes and avoid edge distortion, an undistorted zoomable industrial lens is also mounted, the focal length can be adjusted between 6 and 12 mm and the maximum aperture is f1.6. Light source is also a key part of AOI, to avoid specular reflection, possible shadow of the board and minimise the effect of uneven illumination on subsequent steps, two frosted ring LED sources equipped with special diffuse matting board are introduced to effectively overcome the adverse effects of illumination. The resolution of original photo is 4608×3456 pixels, which will be adjusted according to the size of each board when make defects.

After getting cropped image, six types of defects are made by photoshop, a graphics editor published by Adobe Systems. The



a

b

Fig. 3 *Samples of the PCB with defects in the dataset*
(a) Defect image with the same position as template, ***(b)*** Defect image with random orientation

```
HRIPCB
├─Annotations
│ ├─Missing_hole
│ │ └─01_missing_hole_01.xml
│ └─...
├─Images
│ ├─Missing_hole
│ │ └─01_missing_hole_01.jpg
│ └─...
├─PCB_USED
│ ├─01.JPG
│ ├─02.JPG
│ └─...
└─rotation
  ├─Missing_hole
  ├─Missing_hole_angles.txt
  └─...
```

Fig. 4 *Four different folders of the dataset*

defects defined in the dataset are: missing hole, mouse bite, open circuit, short, spur, spurious copper. Each image in the dataset has three to five defects of the same category in different places. Besides, bounding box and coordinate information for every defect in every image are provided, which is convenient for other researchers to know where the defects are. On some inspection platforms, PCB can be fixed by mechanical devices to maintain good position. However, on the assembly line, without fixing equipments, the position and the angle of the test PCB in the taken photo may distinguish from each other. Given this circumstance, in addition to the images with the same position as the templates, images with random orientations are also provided to represent the situation where the test image is not appropriately placed in practical detection process. The angular difference between each image and the corresponding template image is also given so that the designing and evaluating of registration algorithm could be implemented on these images, the dataset samples can be seen in Fig. 3.

2.2 Statistics

The dataset has four main parts, which are placed in four different folders, for example, see Fig. 4.

The images folder stores the PCB photos with the same position as the templates, and all the photos of a defect type are put in a folder of the same name. Information of bounding boxes of each image is kept in a.xml file that saved in Annotations folder. PCB_USED folder contains the 10 template images used in the dataset. Moreover, rotation folder has PCB images with orientations, and rotation angles are placed with image names in.txt files in this folder.

Table 1 Figures for PCB and defect samples (listed in the brackets)

Category	Number (defects)
missing hole	115 (497)
mouse bite	115 (492)
open circuit	116 (482)
short	116 (491)
spur	115 (488)
spurious copper	116 (503)

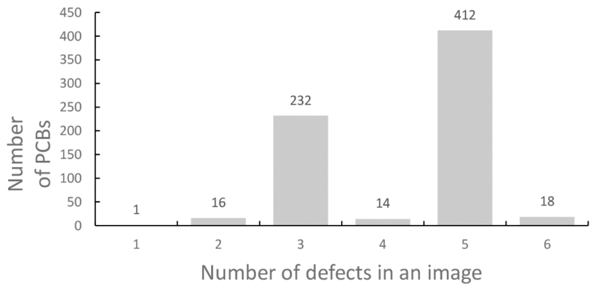


Fig. 5 *Distribution of defects per PCB*

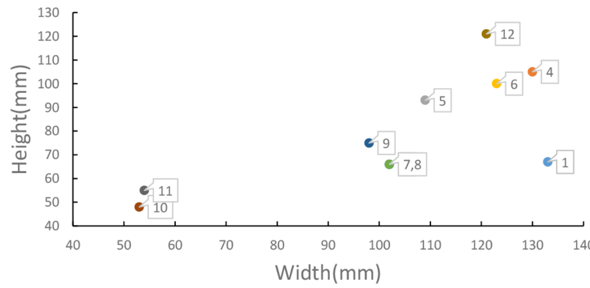


Fig. 6 *Statistics of height and width of the templates*

Details of the figures for PCB images and defect samples are listed in Table 1, where only half of the dataset is listed because the number of PCBs with and without rotation are identical.

Fig. 5 shows the distribution of defects per PCB. It is visible that the majority of PCBs have three or five defects. In Fig. 6, the height and width of every template is given. It can be seen that the largest PCB size in the dataset is 120 mm × 120 mm, while the minimum is 53 mm × 48 mm. In order to facilitate the use of our dataset, API for easy access in python is provided, the .py file in the dataset will intuitively show the bounding box of each defect in the dataset.

2.3 Evaluation metrics

The goals of PCB defects inspection are defects detection and classification, while minimising the time expenditure of the method. The metrics of defect detection is error rate P_d that is defined as follows:

$$P_d = \frac{|d - a|}{a} \times 100\%, \tag{1}$$

where d is the number of detected defect areas and a is the actual number of defect areas.

The metrics of defect classification are the classification precision rate (P_c) of each defect type and the average precision rate (AP_c). P_c is defined in the following equation:

$$P_c = \frac{c}{a} \times 100\%, \tag{2}$$

in which c is the correctly predicted number of a defect type, a is the actual number of defects of this type. The average precision rate (AP_c) is defined as

$$AP_c = \frac{1}{N}\sum_{i=1}^{N} P_c^i, \tag{3}$$

where P_c^i is the precision rate of ith defect, N denotes the number of types of defects, which is 6 in this paper.

3 RBCNN approach

In this section, the RBCNN approach is introduced, first explaining the preprocessing steps like registration and binarisation, followed by XOR and mathematical morphology operation that help to locate defects, as well as deep neural network based model.

3.1 Registration

The PCB is placed on workbench or assembly line while photographing, which would result in the differences in direction and geometric centre between PCB to be inspected and template board. So, registration is indispensable in reference comparison based method. A test image and template will be converted into grey image first, then feature points of the two images are extracted and matched, the transformation matrix is calculated to transform the test image into the same orientation and position as the template image at last. In this paper, speeded up robust features (SURF) [7] algorithm is used to extract feature points in PCB. It is an improvement of scale invariant feature transform (SIFT) [8], with less computational complexity, and can run faster compared with SIFT. The feature points selected by SURF and SIFT are both stable and rotation, scale, luminance invariant. Although SIFT has better matching effect than SURF in the case of scale and rotation transformation, SURF has better matching effect under brightness change, considering the practical application scenarios, SURF is chosen for PCB registration. Once get the SURF feature points of the template and test image, a 2D geometric transform will be estimated from matching points and the test image will be recovered by the geometric transform.

3.2 Binarisation

It is not easy to directly compare $x \in \chi$ two colourised or grey-scale images due to the fact that they are easily influenced by illumination. Nevertheless, by using a binary map, the outline and shape of the PCB are only expressed in black and white, which is more convenient for comparison. There are many methods for image binarisation, in this paper, adaptive threshold segmentation algorithm [9] is chosen. Instead of using a global value as threshold value, adaptive threshold algorithm calculates thresholds for small regions of the image, because PCB image may have different lighting conditions in various areas. For every pixel (x, y), the threshold value $T(x, y)$ is the weighted sum of $blocksize \times blocksize$ neighbourhood values where weights are a Gaussian window. The Gaussian kernel is defined as follows:

$$G_i = \alpha \times e^{(-(i-(ksize-1)^2)/2)/(2\times sigma^2)}, \tag{4}$$

where $i = 1, 2, \ldots, ksize - 1$ and α is the scale factor chosen so that $\sum_i^G = 1$, $ksize$ indicates aperture size and it should be odd, $sigma$ is Gaussian standard deviation computed from $ksize$. Once $T(x, y)$ is calculated individually for each pixel in every region, the output value $dst(x, y)$ is defined as

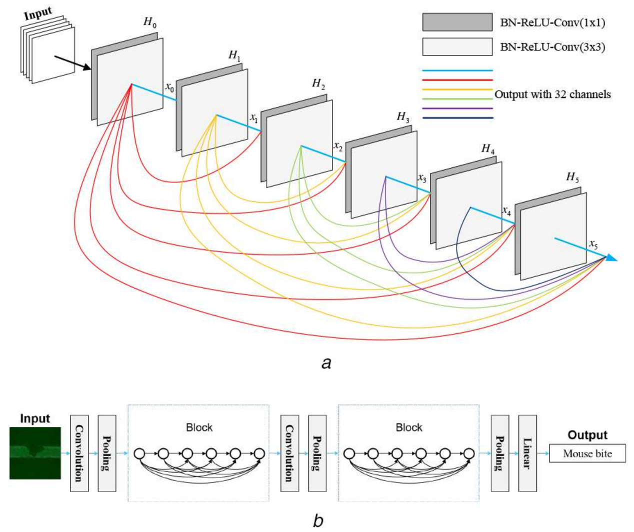


Fig. 7 *Basic block and schematic network structure in our paper*
(a) Block with six layers, ***(b)*** Schematic structure of the network in our method

$$dst(x, y) = \begin{cases} maxValue, & src(x, y) > T(x, y) \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $maxValue$ is a non-zero value assigned to the pixels for which the condition is satisfied, usually set as 255.

3.3 Localisation of defects

The result binary image is obtained by XOR binary image of template and test image, the formula of XOR operation is defined as

$$dst(I) = src1(I) \oplus src2(I), \tag{6}$$

where $dst(I)$ is the result binary image, $src1(I)$, $src2(I)$ are template binary map and test binary map, respectively. In XOR, if the pixel values in the corresponding positions of the template and test image are the same, the pixel value of the position in result image will be 1 after XOR, if not, the result value will be 0.

However, the result binary image may contain a great number of noises and unwanted pseudo-defects. To get real defects, median filtering [10] and mathematical morphological [11] processing are used. Median filtering is a non-linear filtering technique used to eliminate tiny noise points in the image. The basic idea is to sort the pixel values of the neighbourhood of a pixel point (x, y), and take the intermediate value to replace the value of original pixel. Morphological processing is a theory and technique for the analysis and processing of geometrical structures, the basic morphological operators are erosion, dilation, opening and closing which are defined in (7)–(10) continuously.

In erosion, if the structuring element B has a centre, then the erosion of A by B can be understood as the locus of points reached by the centre of B when B moves inside A, which can be defined as follows:

$$A \ominus B = \{z | B_z \subseteq A\}, \tag{7}$$

where $B_z = \{b + z | b \in B\}$. Generally speaking, erosion can make the range of the target area smaller, which can be used to eliminate small and meaningless objects in an image.

In dilation, if B has a centre on the origin, then the dilation of A by B can be understood as the locus of the points covered by B when the centre of B moves inside A

$$A \oplus B = \left\{z \middle| (\bar{B})_z \cap A \neq \varnothing\right\}, \tag{8}$$

where $\bar{B} = \{x | -x \in B\}$. The dilation can be used to make the target boundary to expand outward to fill in some holes and eliminate small particle noises existing in the target area.

The opening of A by B is obtained by the erosion of A by B, followed by dilation of the resulting image by B, which will remove isolated points, burrs and bridges, while the overall position and shape of the target area remain unchanged, opening can be formulated as

$$A \circ B = (A \ominus B) \oplus B. \tag{9}$$

The closing of A by B is obtained by the dilation of A by B, followed by erosion of the resulting structure by B. It can fill the small holes and close the small cracks, keeping the overall position and shape unchanged. Closing operator can be formulated as

$$A \bullet B = (A \oplus B) \ominus B. \tag{10}$$

3.4 Models

CNN has powerful ability to extract features in pictures, and it has been widely used in many computer vision tasks such as classification [12, 13], segmentation [14], object detection [15] and so on. Recently, in the field of defect inspection, a lot of methods based on CNN have been adopted [16–18]. The results showed their superiorities compared with conventional approaches. As tasks become more and more complicated, the CNNs also become increasingly deep to make sure that more features would be extracted to contribute to the final result. However, another problem called gradient diffusion occurs when gradient flows back to the beginning if the network is so deep. In this case, one common solution for the problem above is creating shortcut from early layers to later layers. In this paper, inspired by Densenet [13], to utilise the densely connection structure, a small and efficient network is designed to handle PCB defect classification problem.

The network mainly consists of two basic blocks, as is illustrated in Fig. 7*a*. Each block has six convolutional layers, in which every layer takes all outputs of previous layers as input. Hence, the output of lth layer that has l inputs (including outputs from previous block) can be defined as

$$x_l = H_l([x_0, x_1, \ldots, x_{l-1}]), \tag{11}$$

where x_l is the output of lth layer, $H_l(\cdot)$ denotes a compose of functions in lth layer including batch normalisation (BN) [19], rectified linear units (ReLU) [20], pooling [21] and convolution (Conv). In our experiment, each H_l contains two convolutions of size 1×1 and size 3×3 with stride 1 and padding 1, respectively, and there are BN and ReLU functions before every convolution. The structure of a layer can be simply summarised as BN-ReLU-Conv (1×1)-BN-ReLU-Conv (3×3). Each H_l is set to produce fixed 32 feature-maps, which will result in lth layer having $k_0 + 32 \times (l-1)$ input feature-maps, here k_0 is the number of channels in the input layer. This 1×1 convolution can be introduced as bottleneck layer [22] to change number of input feature-maps, our method let each 1×1 convolution produces 4×32 feature-maps in block. More precisely, before sending into a layer, the feature-maps from previous layers are concatenated instead of combining them, so the 6-layer block will have 21 connections at last.

In addition, before entering the first basic block, the input image will pass through a convolution of size 7×7 with stride 2 and padding 3, followed by BN, ReLU and Maxpooling functions of size 3×3 with stride 2 and padding 1. Then the output will be passed to the first block which is followed by a transition layer where the number and size of feature-maps will be halved for compacting the model. The structure of transition layer is like BN-ReLU-Conv (1×1)-AvgPool (2×2). After the second block, an adaptive AvgPool is used and then a linear layer is employed to produce 6×1 vector. Detailed architecture of the network is demonstrated in Fig. 7*b*.

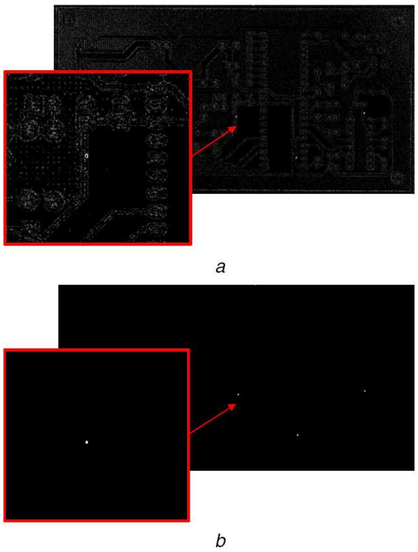


Fig. 8 *Result image and defect image*
(a) Result image after XOR operation, ***(b)*** Defect image after filtering and morphology operation

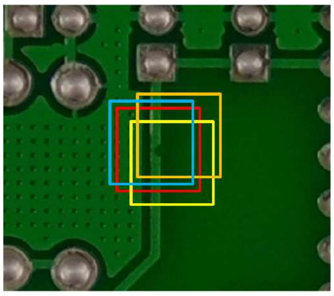


Fig. 9 *Depicts of data augmentation, the red frame is original bounding box labelled in dataset, and the other three frames (blue, yellow, orange) are created for producing more defect images*

4 Experiments and discussions

In this section, experiment setups on defects location, data preparation and model training are detailed and experiment results are discussed.

4.1 Locate defects

In this paper, the result image of XOR operation is filtered by a 5×5 kernel first to get rid of some small isolate points, then closing operation with 15×15 rectangle element is taken so that local parts of defects would be connected and enhanced, followed by an opening operation with 3×3 rectangular element. The main object in a binary image will be highlighted by using closing and opening operation continuously. In addition, continue to set the threshold of area to remove too small points, followed by setting non-maximum value suppression that will remove adjacent redundant candidate regions. The final result image is pure without other points except for the areas that real defects locate. In this case, the locations of defects can be obtained from the connected areas, the result image after XOR operation and defect image after filtering and mathematical morphology operation are drawn in Fig. 8.

4.2 Preparing for data

After getting the locations of defects, the next step is to identify the defect category. Conventional methods are based on pixel-by-pixel comparison between template and test image to select enough features to represent defects [1–3, 5], which would have non-ideal result if the binarisation is in poor condition. Nevertheless, by using an end-to-end deep learning model, the image of defect area can be sent to the model as input directly to obtain a classification

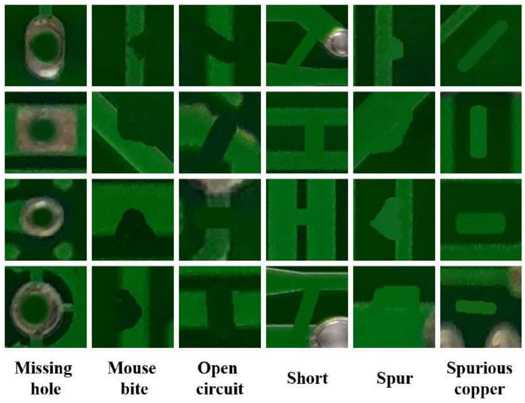


Fig. 10 *Example of the training data for neural network. All the resolutions of the images are resized to* 64×64

result, thereby avoiding extracting pixel-based features from the binary image. The priori task for training and testing neural network is to prepare enough data. Considering that bounding box in our dataset has already given the coordinates of each defect, the data for neural network is clipped by the image in the bounding box. In order to have data augmentation to produce more training images, position of defect in the image is changed by randomly making 5 pixel to 10 pixel offsets on existing coordinates, as Fig. 9 shows.

In this way, the size of data will be expanded so that the generalisation ability of the model will be enhanced. Resolution of each original local defect image cropped from PCB dataset varies from one to another. To facilitate the use of defect data, all images are resized to a resolution of 64×64, which are divided into three folders: train, val and test. Further, there are six sub-folders under each folder including all the images of six defect types. An example diagram of the training data is shown in Fig. 10, and the distribution of data is displayed in Table 2.

4.3 Training

The training process is executed on a computer with Intel Xeon E5-2640 CPU, 128 GB RAM, and a NVIDIA GTX 1080Ti GPU is used during training. Stochastic gradient descent with momentum 0.9 is used to update parameters. The initial learning rate is set to 0.01 and decay 0.1 every 7 epochs. The model is trained using batch size 8 for 50 epochs and the whole training procedure takes about 25 min. L2 penalty is 1×10^{-5} in the experiment to prevent over-fitting.

4.4 Results

Experiment results on defects location, classification and time consumption are presented and discussed.

4.4.1 Defects inspection: To verify the effectiveness of reference comparison based method in this paper, the preprocessing and detection algorithms are implemented on our dataset. The statistics of the result is listed in Table 3. It is clear that only a mouse bite and open circuit defect are needlessly detected, the former is a wrong detection (i.e. false detection of non-defect area) and the latter is an overlapped one (i.e. a defect produces two adjacent overlapped results).

4.4.2 Defects classification: The classification model is tested on the test data produced in Section 4.1 by bounding box and all defect samples produced in Section 3 by reference comparison based method. It should be noted that before classifying the defects, the repeatedly and incorrectly detected samples in the defect detection results are removed to avoid the impact on the classification procedure. The result shown in Table 4 indicates that the proposed method acquire superior performances on both groups, with average precision of 97.74 and 99.40%, respectively. The reason for this case is that original defect image obtained by

Table 2 Distribution of defects in training, val and test folders

	Train	Val	Test
missing hole	599	203	192
mouse bite	600	190	194
open circuit	598	189	177
short	600	189	177
spur	600	187	189
spurious copper	600	203	203
total	3597	1161	1148

Table 3 Defects detection results, first row lists number of defects provided by dataset, second row lists number of defects got by our method and third row are the error rate P_ds

	Missing hole	Mouse bite	Open circuit	Short	Spur	Spurious copper
actual number	497	492	482	491	488	503
detected number	497	493 (+1 error)	483 (+1 over lapped)	491	488	503
error rate (P_d)	0%	0.2%	0.2%	0%	0%	0%

Table 4 Defects classification results, the first row represents the P_cs and AP_c obtained in the test data produced in Section 4.1, and the second row represents the P_cs and AP_c obtained from all samples in Section 3

	Missing hole	Mouse bite	Open circuit	Short	Spur	Spurious copper	Average (AP_c)
test data (P_c)	98.96%	97.94%	97.74%	99.48%	93.65%	98.52%	97.74%
all samples (P_c)	100%	99.6%	99.18%	99.39%	99.39%	98.80%	99.40%

Table 5 Time consumption of each step, registration takes up the most time, followed by localisation and binarisation

Procedure	Time, s
registration	0.6219
binarisation	0.1650
localisation	0.1808
classification	0.0212
total	0.9889

the reference comparison method is smaller than the image cropped by the bounding box given by manual annotation, resulting in the defect body accounts for a larger proportion when the image is resized to a fixed resolution, which is more beneficial for classification.

4.4.3 Time consumption: Taking the detection efficiency into account, the time required to spend in each step of inspecting a PCB is recorded, as described in Table 5. It takes a total of 0.9899 s to execute the entire process on a computer with Intel Core i7-7700 CPU @ 3.60 GHz, 8 GB RAM. In these steps, registration accounts for the most of total time because searching feature points and calculating descriptors are all time consuming tasks.

5 Conclusion

In this paper, in consideration of lack of public shared PCB dataset, a synthesised PCB dataset called HRIPCB dataset that has 1386 images with 6 types of common defects is presented and published, including missing hole, mouse bite, open circuit, short, spur and spurious copper. Half of the images are for the situation where a test PCB is placed correctly, while the other half is set for simulating the situation when the test board is randomly orientated in the workbench. Bounding box of every defect is provided in our dataset so that the location of each defect can be affirmed. Besides, the existing of bounding box makes it possible for the images to be utilised as labelled data in object detection tasks. The transformation information is also provided to facilitate other researchers to study registration problems.

Based on reference comparison method, an end-to-end CNN model is introduced to classify the defects. The method proposed in this paper is referred to as RBCNN approach which combines conventional method with CNN model and reaches impressive performance on our dataset. In order to learn more effectively, instead of choosing simply stacking convolutional layers, dense shortcuts inspired from Densenet are used to achieve high accuracy with relatively few layers.

Future work may focus on continuously increasing the size of the dataset, improving the robustness of the algorithm, reducing the time consumption of the entire detection process while achieve higher efficiency, what is more, designing effective non-reference comparison method to avoid using template.

6 Acknowledgment

This work is supported by National key R&D program of China (2018YFB1308600, 2018YFB1308602), Specialized Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (no.ZLZBCXLJZI20160729020003).

7 References

[1] Wu, W.Y., Wang, M.J.J., Liu, C.M.: ‘Automated inspection of printed circuit boards through machine vision’, *Comput. Ind.*, 1996, **28**, (2), pp. 103–111

[2] Zheng-Ming, L.I., Hong, L.I., Sun, J.: ‘Detection of PCB based on digital image processing’, *Instrum. Techn. Sens.*, 2012, **61**, (8), pp. 87–89

[3] Chaudhary, V., Dave, I.R., Upla, K.P.: ‘Automatic visual inspection of printed circuit board for defect detection and classification’. Proc. of the 2017 Int. Conf. on Wireless Communications, Signal Processing and Networking, Chennai, India, 2017, pp. 732–737

[4] Kumar, S., Iwahori, Y., Bhuyan, M.K.: ‘PCB defect classification using logical combination of segmented copper and non-copper part’. Proc. of Int. Conf. on Computer Vision and Image Processing, Roorkee, India, 2017, pp. 523–532

[5] Heriansyah, R., Al-attas, S.A.R., Zabidi, M.M.A.: ‘Neural network paradigm for classification of defects on PCB’, *Jurnal Teknologi*, 2003, **39**, (1), pp. 87–103

[6] Pramerdorfer, C., Kampel, M.: ‘A dataset for computer-vision-based PCB analysis’. Proc. of the 14th IAPR Int. Conf. on Machine Vision Applications, Tokyo, Japan, 2015, pp. 378–381

[7] Bay, H., Tuytelaars, T., Van Gool, L.: ‘Surf: speeded up robust features’, European Conf. on Computer Vision, Graz, Austria, 2006, pp. 404–417

[8] Lowe, D.G.: ‘Distinctive image features from scale-invariant keypoints’, *Int. J. Comput. Vis.*, 2004, **60**, (2), pp. 91–110

[9] Gonzalez, R.C., Woods, R.E., Eddins, S.L.: ‘*Digital image processing using Matlab*’ (Publishing House of Electronics Industry, China, 2009)

[10] Huang, T., Yang, G., Tang, G.: ‘A fast two-dimensional median filtering algorithm’, *IEEE Trans. Acoustic. Speech. Signal Process.*, 1979, **27**, (1), pp. 13–18

[11] Soille, P.: ‘*Morphological image analysis: principles and applications*’, (Springer Science & Business Media, USA., 2013), pp. 800–801

[12] He, K., Zhang, X., Ren, S., *et al.*: 'Deep residual learning for image recognition'. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 2016, pp. 770–778

[13] Huang, G., Liu, Z., Laurens, V.D.M., *et al.*: 'Densely connected convolutional networks'. IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, HI, USA., 2017, pp. 2261–2269

[14] Long, J., Shelhamer, E., Darrell, T.: 'Fully convolutional networks for semantic segmentation'. IEEE Conf. on Computer Vision and Pattern Recognition, Boston, Massachusetts, USA., 2015, pp. 3431–3440

[15] Liu, W., Anguelov, D., Erhan, D., *et al.*: 'Ssd: single shot multibox detector'. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 21–37

[16] Faghih-Roohi, S., Siamak, H., Núnez, A., *et al.*: 'Deep convolutional neural networks for detection of rail surface defects'. Int. Joint Conf. on Neural Networks, Vancouver, BC, Canada, 2016, pp. 2584–2589

[17] Hanzaei, S.H., Afshar, A.: 'Automatic detection and classification of the ceramic tiles' surface defects'. *Pattern Recognit.*, 2017, **66**, pp. 174–189

[18] Zhou, S., Chen, Y., Zhang, D., *et al.*: 'Classification of surface defects on steel sheet using convolutional neural networks', *Materiali Tehnologije*, 2017, **51**, (1), pp. 123–131

[19] Ioffe, S., Szegedy, C.: 'Batch normalization: accelerating deep network training by reducing internal covariate shift'. Proc. of Int. Conf. on Machine Learning, Lille, France, 2015, pp. 448–456

[20] Glorot, X., Bordes, A., Bengio, Y.: 'Deep sparse rectifier neural networks'. Proc. of Int. Conf. on Artificial Intelligence and Statisitics, Ft. Lauderdale, FL, USA, 2011, pp. 315–323

[21] Lecun, Y., Bottou, L., Bengio, Y., *et al.*: 'Gradient-based learning applied to document recognition', *Proc. IEEE*, 1998, **86**, pp. 2278–2324

[22] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.*: 'Rethinking the inception architecture for computer vision'. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA., 2016, pp. 2818–2826