

A PCB Dataset for Defects Detection and Classification

Weibo Huang, Peng Wei

Abstract—To cope with the difficulties in the process of inspection and classification of defects in Printed Circuit Board (PCB), other researchers have proposed many methods. However, few of them published their dataset before, which hindered the introduction and comparison of new methods. In this paper, we published a synthesized PCB dataset containing 1386 images with 6 kinds of defects for the use of detection, classification and registration tasks. Besides, we proposed a reference based method to inspect and trained an end-to-end convolutional neural network to classify the defects. Unlike conventional approaches that require pixel-by-pixel processing, our method firstly locate the defects and then classify them by neural networks, which shows superior performance on our dataset.

Index Terms—Printed Circuit Board, Automated Optical Inspection, PCB Dataset, Reference Based Method, Convolutional Neural Network.

I. INTRODUCTION

Printed circuit board (PCB) is the fundamental carrier in electronic devices on which a great number of elements are placed. The quality of the PCB will directly impact the performance of electronic devices. To avoid the shortcoming of manual detection, easily being fatigued, low efficiency, for instance, automated optical inspection (AOI) based on machine vision has been widely used in industry. As PCB becomes more and more complicated, the tasks of detection and classification defects are also more difficult than before. Currently, there are few public datasets on the Internet on PCB, many methods proposed in published papers used their own images, which is not convenient for other researchers to compare their new methods. For the purpose of solving above problems, we produced a public colorized synthesized PCB dataset with defects that is available to other people who want to design and evaluate their approaches.

Conventional AOI methods for inspecting printed circuit board can be divided into 3 main streams [20] : reference comparison approach, non-reference verification approach, and hybrid approach. In reference comparison approach, a standard image which is called template will be prepared firstly, and then a PCB need to be inspected will be compared with the template to find the unknown defects. Though it is straightforward and easy to use, there are also many factors that we have to take in consideration, unbalanced illumination, inaccurate registration, vast storage requirements, etc. In the non-reference verification approach, the aim of the method is

Weibo Huang, Peng Wei are with the Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Shenzhen, CN, 518052
weapon@pku.edu.cn

J. Doe and J. Doe are with Anonymous University.

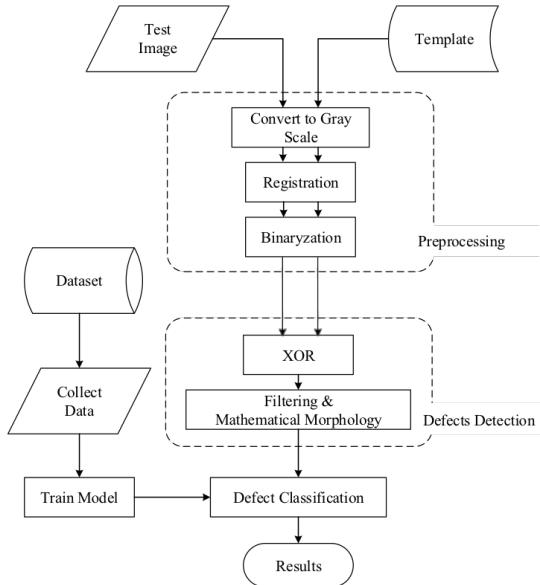


Fig. 1. The flow chart for PCB inspection. Test image and template will be separately preprocessed and compared to locate defects, then these located defects will be sent into trained neural network model to get results.

to find out if wiring track, pad and hole are in the compliance with design without a template board. This approach does not have the limits of reference method, nevertheless, it may have difficulties in detecting large defects. In the hybrid approach, reference method and non-reference method are combined, this approach will have the merits of the two basic methods, meanwhile, it requires high computation capacity.

Some various methods have been proposed on this task. Wen-Yen Wu et al. [20] introduced the development of an automated visual inspection system for PCB. It utilized 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 the inspected image and the template. LI Zheng-ming et al. [21] also used digital image processing technology based reference method to classify the defects by getting the number of connected regions, euler numbers, area of defects of the template and inspected image respectively. The result of experiment showed that the method can achieve automatic real-time detection. Vikas Chaudhary et al. [2] listed 14 kinds of defects that belong two types: positive, negative, and segmented the image into 3 parts:

wiring tracks, soldering pads and holes. Each defect can be classified by comparing pixels, number of connected components in the corresponding part. Shashi Kumar et al. [12] proposed a non-referential based approach in consideration of the difficulties in registration. In his work, inspected image was segmented into copper and non-copper parts to analyze separately, and a 3D color histogram was utilized to capture the global color 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. Rudi Heriansyah et al. [7] introduced a new technique that is to classify the defects using neural network paradigm. Various defective patterns representing corresponding defect types were designed and thousands of defective patterns had been used for training and testing. The result showed the effectiveness of defect classification technology based on neural network.

Because of the intuitiveness, simpleness, and the development of computer hardware and algorithms, reference comparison method is used to inspect defects in our approach. In addition, convolutional neural network highlights outstanding performance in computer vision tasks, like classification, object detection, segmentation, etc. Therefore, in the defect classification task, we do not search the features of the image, instead, we introduce an end-to-end neural network to classify the inspected defect regions. The experimental results prove its effectiveness. The flow chart of the whole experiment process is shown in Figure 1.

Before our work, there are some public datasets on Printed circuit board assembly (PCBA) [17] which is a kind of board after all the components and parts have been soldered and installed on the PCB and can accomplish the electronic function it was designed for. Inspection of PCBA is for the purpose of recycling when the PCBA is eliminated, however, it is not appropriate for us to use these PCBA datasets because our target is naked PCB that has no components. In this paper, we present a synthesized dataset that consists of 1386 naked PCB images. Half of them are in right orientation as templates with different defects and 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. Each PCB has 3 to 5 defects and we provide corresponding bounding box for every defect. Rotation information are also provided for PCBs rotated. 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 online¹.

The paper is organized as follows, section I introduces the backgrounds of PCB dataset and main stream methods on defects detection and classification. section II details the procedure of image acquisition, labeling and defect statistics. Our reference comparison based method and the detection part is given in section III. section IV presents convolutional neural network based model that we use to classify defects and the experiment results. Conclusions is shown in section V.

¹www.baidu.com



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

II. IMAGE ACQUISITION AND STATISTICS

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

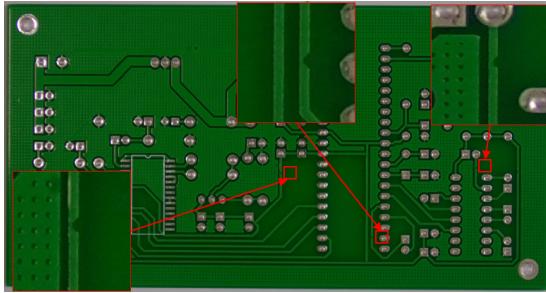
A. Image acquisition

To ensure the representativeness of the dataset, we build a PCB image acquisition system that resembles the practical AOI system used in inspection process, as is shown in Figure 2.

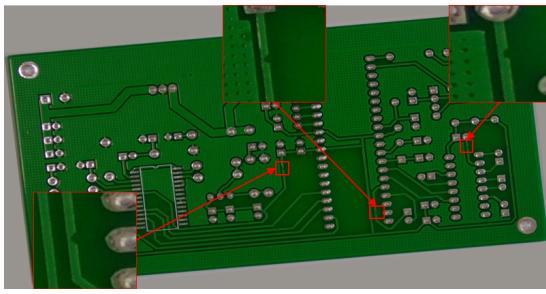
The image of template board is captured by a 16-megapixel HD industrial camera equipped with CMOS sensor, and it can be controlled by computer software or a remote control. In order to adapt to different PCB sizes and avoid edge distortion, an undistorted zoomable industrial lens is also mounted, the focal length can be adjusted between 6-12mm and the maximum aperture is f1.6. Light source is also a key part of AOI, to avoid specular reflection of the board, possible shadows and minimize the effects of uneven illumination on subsequent steps, two frosted ring LED source 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, we make 6 types of defects by photoshop, which is a graphics editor published by Adobe Systems. The defects we defined are: missing hole, mouse bite, open circuit, short, spur, spurious copper. Each image in the dataset has 3 to 5 defects of the same category in different places. Besides, we provide bounding box and coordinate information for every defect in every image, which is convenient for other researchers to know where the defect is. 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 defects images with the same position as the templates, we also provide images with random orientations to represent the situation where the 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 samples of the dataset can be seen in Figure 3.



(a) PCB with mouse bite.



(b) PCB with mouse bite and rotation.

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

B. Statistics

The dataset has four main parts, which are placed in four different folders. 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 we used in the dataset. Moreover, rotation folder has PCB images with orientations, and rotation angles are also placed with image names in .txt files in this folder. The structure of the dataset is shown in the tree diagram in Figure 4.

Details of the figures for PCB images and defect samples are listed in table I. Figure 5 shows the distribution of defects per PCB. It is visible that the majority of PCBs have less than 6 defects, and most PCBs have more than 2 defects. In Figure 6, the height and width of every template is given. We can see that the largest PCB size in the dataset is $20\text{mm} \times 120\text{mm}$, while the minimum is $53\text{mm} \times 48\text{mm}$. In order to facilitate the use of our dataset, we provide API for easy access in python, the .py file in the dataset will intuitively show the bounding box of each defect in the dataset.

III. PREPROCESSING AND DETECTION

In this section, preprocessing steps like registration and binaryzation are applied, followed by XOR and mathematical morphology operation that helped to locate defects.

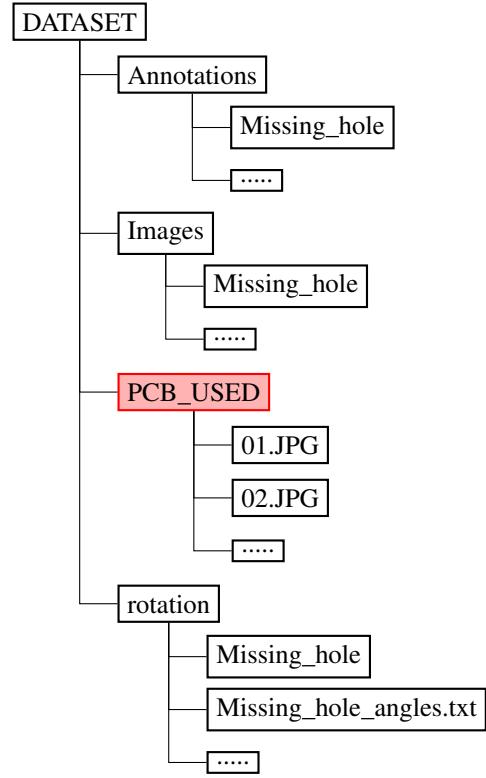


Fig. 4. Structure of the dataset, the whole dataset is divided into 4 parts.

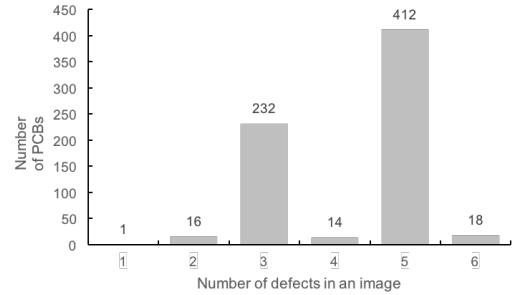


Fig. 5. The distribution of defects per PCB.

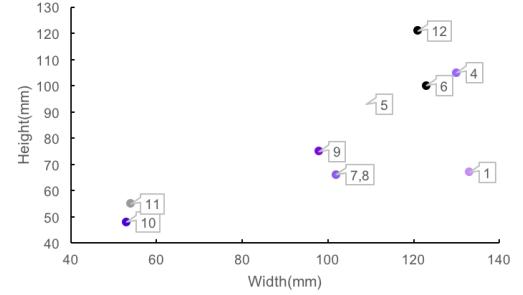


Fig. 6. The statistics of height and width of the templates.

A. Registration

The printed circuit board are placed on workbench or assembly line while photographing, which would result in the differences in direction and geometric center between

TABLE I
THE 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)

PCB to be inspected and template board. So registration is indispensable in reference comparison based method. A test image and template image will be converted into gray image first, then feature points of the two images are extracted and matched, finally the transformation matrix is calculated to transform the test image into the same orientation and position as the template image. In this paper, Speeded Up Robust Features (SURF) [1] algorithm is used to extract feature points in PCB. It is an improvement of Scale Invariant Feature Transform (SIFT) [16], with less computational complexity, and can run faster compared with SIFT. The feature points selected by SURF and SIFT are both stable and they are 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 tempalte and test image, a 2-D geometric transform will be estimated from matching points and the test image will be recovered by the geometric transform. Figure 7 shows the selected and matched feature points and transformed test image in our experiment.

B. Binaryzation

It is not easy to directly compare two colorized or gray-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. The next step is to convert gray-scale image into binary image to get the location of defects. There are many methods for image binaryzation, in our method, adaptive threshold segmentation algorithm [5] 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:

$$G_i = \alpha \times e^{-\frac{(i-(ksize-1))^2}{2 \times sigma^2}} \quad (1)$$

where $i = 1, 2, \dots, ksize - 1$ and α is the scale factor chosen so that $\sum G_i = 1$, $ksize$ indicates aperture size and it should be odd, $sigma$ is gaussian standard deviation computed from

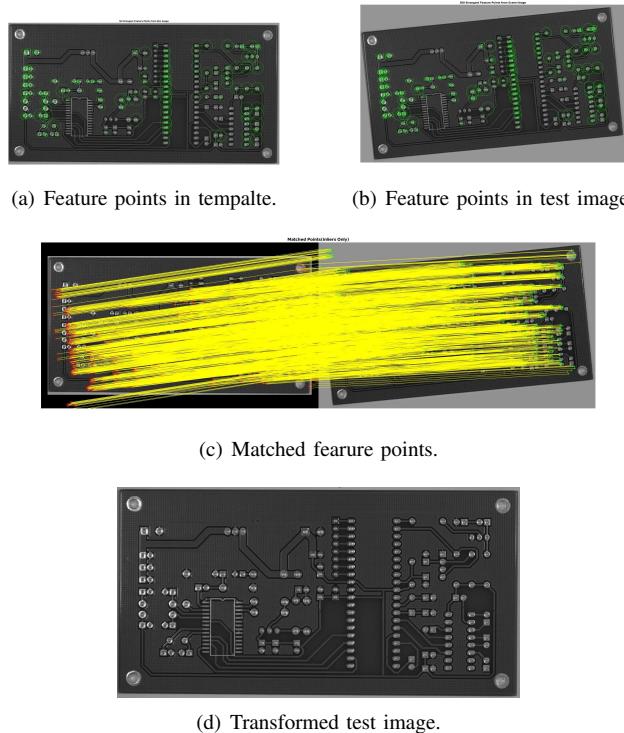


Fig. 7. The selected and matched feature points and transformed test image in our experiment.

$ksize$. Once $T(x, y)$ is calculated individually for each pixel in every region, the output value $dst(x, y)$ is defined in Equation 2:

$$dst(x, y) = \begin{cases} maxValue & \text{if } src(x, y) > T(x, y) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $maxValue$ is a non-zero value assigned to the pixels for which the condition is satisfied, usually set as 255. The bianry images coming from our dataset is shown in Figure 8.

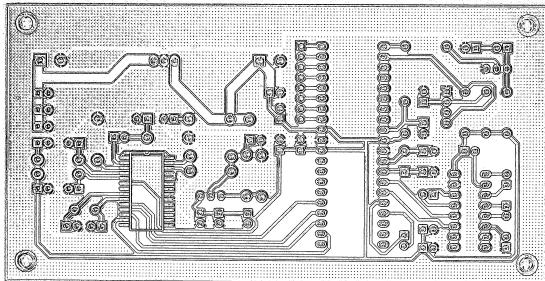
C. Localization of defects

The result binary image is obtained by XOR binary image of template and tested image, the formula of XOR operation is defined in Equation 3:

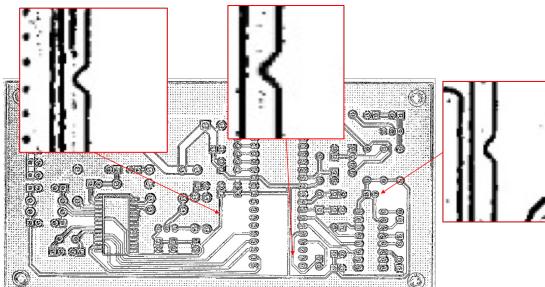
$$dst(I) = src1(I) \oplus src2(I) \quad (3)$$

where $dst(I)$ is the result binary image, $src1(I)$, $src2(I)$ are template binary map and tested binary map respectively. In the 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 [18] 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 neighborhood of a pixel point (x, y) , and take the intermediate



(a) Binary image of template.



(b) Binary image of test image.

Fig. 8. The binary images coming from our dataset. It is clear that defects are outlined in the binary map.

value to replace the value of the 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 Equation 4, 5, 6, 7 continuously.

$$A \ominus B = \{z \mid B_z \subseteq A\} \quad (4)$$

where $B_z = \{b + z \mid b \in B\}$. In erosion, if the structuring element B has a center, then the erosion of A by B can be understood as the locus of points reached by the center of B when B moves inside A. 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.

$$A \oplus B = \{z \mid (\overline{B})_z \cap A \neq \emptyset\} \quad (5)$$

where $\overline{B} = \{x \mid -x \in B\}$. In dilation, if B has a center on the origin, then the dilation of A by B can be understood as the locus of the points covered by B when the center of B moves inside A. 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.

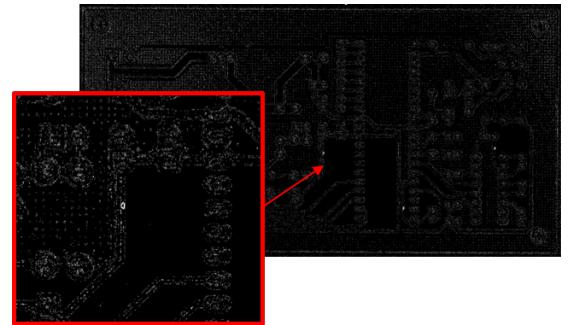
$$A \circ B = (A \ominus B) \oplus B \quad (6)$$

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.

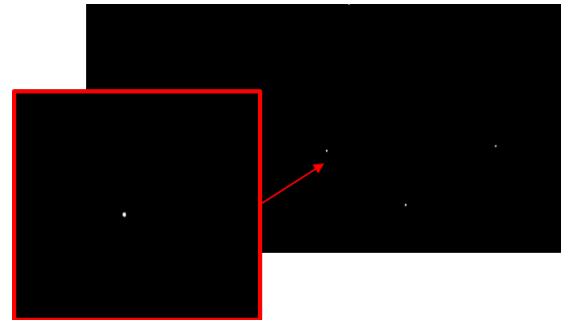
$$A \bullet B = (A \oplus B) \ominus B \quad (7)$$

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 as the opening.

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 a 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, we continue to set the area threshold to remove too small points, followed by setting non-maximum value suppression (NMS) that will remove adjacent redundant candidate regions. All the steps on filtering and mathematical morphology processing in this paper is shown in Table VI. Final result image is pure without other points except for the area that real defects locate. In this case, the location 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 Figure 9.



(a) Result image after XOR operation.



(b) Defect image after filtering and morphology operation.

Fig. 9. Result image after XOR operation (a) and defect image after filtering and mathematical morphology operation (b).

IV. EXPERIMENT AND CLASSIFICATION

In this section, experiment based on convolutional neural network for defect classification is introduced, including data preparation, model selection and analysis of experiment results.

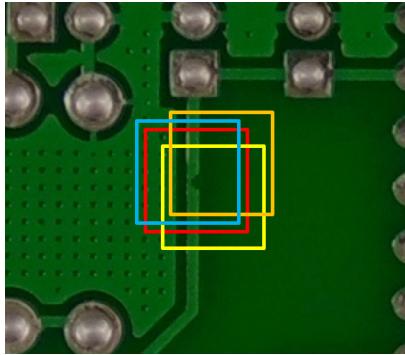


Fig. 10. The depicts of data augmentation, the red frame is original bounding box labeled in dataset, and the other 3 frames (blue, yellow, orange) are created for producing more defect images.

TABLE II
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

A. Preparing for data

After getting the location 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 [20], [21], [2], [7], which would have non-ideal result if the binaryzation is in poor condition. Nevertheless, by using an end-to-end deep learning model, the defect image can be sent to the model as input directly to obtain a classification result, thereby avoiding extracting pixel-based features from the binary image. In this paper, a convolutional neural network based method is utilized to classify defects. The prior 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, we cut the image in the bounding box as the data for neural network. In order to have data augmentation to produce more training images, we change the position of defects in the image by randomly make 5 pixel to 10 pixel offset on the existing coordinates , as the Figure 10 shows.

In this way, the size of data will be expanded and the generalization ability of the model will be enhanced. Resolution of each original defect image cropped from PCB dataset varies from one image to another. To facilitate the use of defect data, all images are resized to a resolution of 64×64 , which are divided into 3 folders: train, val and test. Further, there are 6 sub-folders under each folder including all the images of the 6 defect types. An example diagram of the training data is shown in Figure 11, and the size and distribution of the data is displayed in Table II.

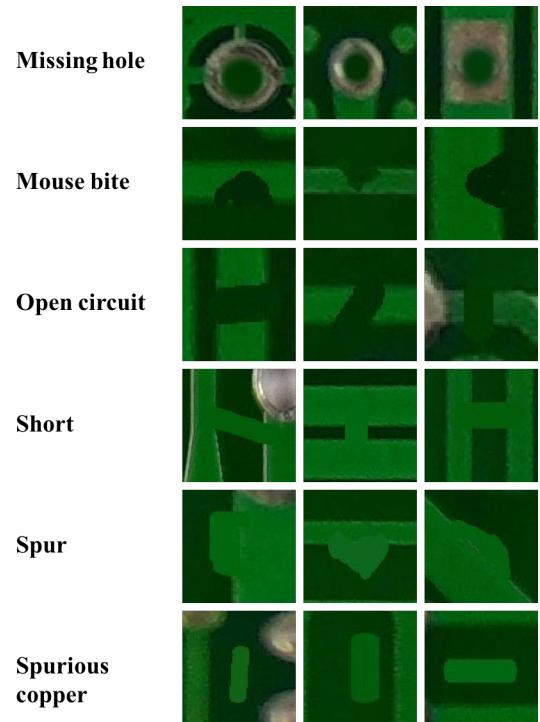


Fig. 11. An example of the training data for neural network. All the resolutions of the images are resized to 64×64 .

B. Models

Convolutional neural network (CNN) has powerful ability to extract features in pictures, and it has been widely used in many computer vision tasks such as classification [6], [9] , segmentation [15], object detection [14], etc. Recently, in the field of defect inspection, a lot of methods based on convolutional neural network have been adopted [3], [8], [22]. 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 flow 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 our method, inspired by Densenet [9], to utilize 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 Figure 12(a). Each block has 6 convolutional layers, in which every layer takes all outputs of previous layers as input. Hence, the output of l^{th} layer that has l inputs (including outputs from previous block) can be defined as:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (8)$$

where x_l is the output of l^{th} layer, $H_l(\cdot)$ denotes a compose of functions in l^{th} layer including Batch Normalization (BN) [11], Rectified Linear Units (ReLU) [4], Pooling [13], and Convolution (Conv). In our experiment, each H_l contains 2 convolutions of size 1×1 and size 3×3 with stride 1 and padding 1 respectively, and there are BN and ReLU function

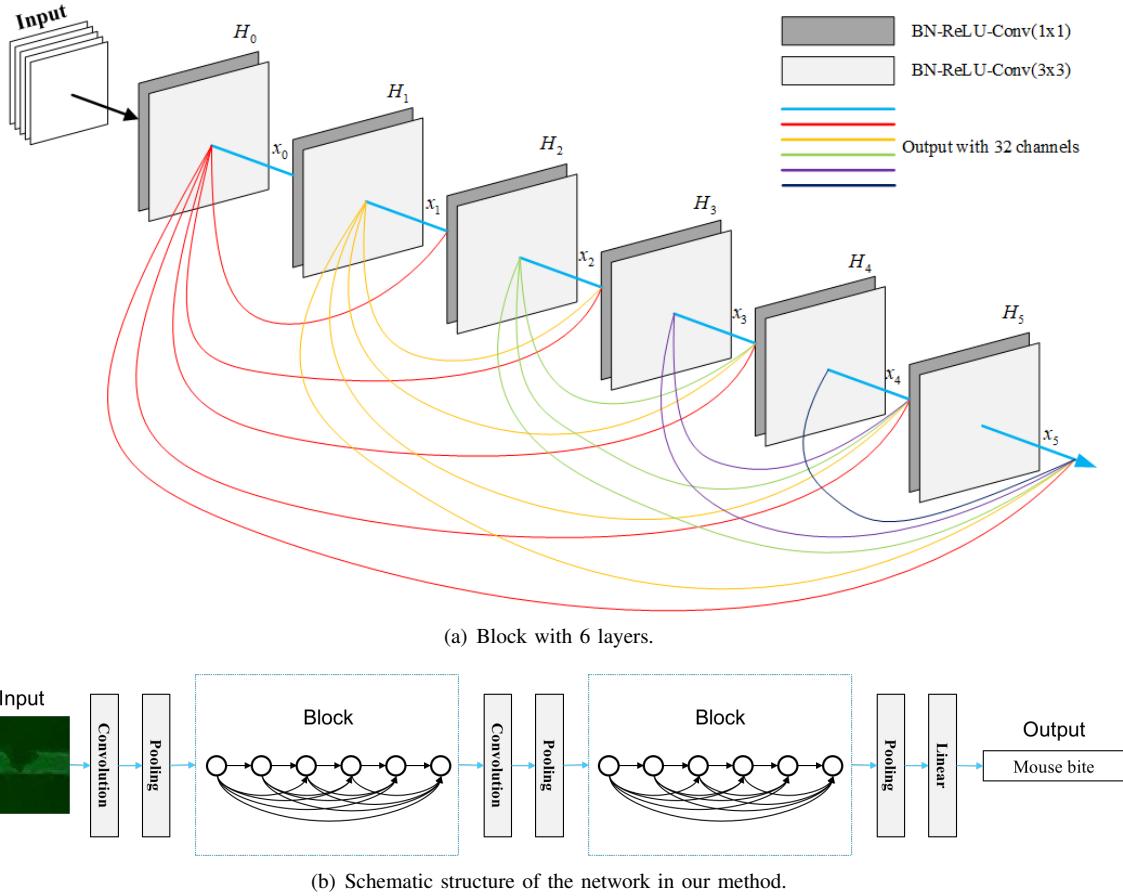


Fig. 12. Basic Block and schematic network structure in our paper, (a) draws the 6-layer block and (b) illustrates the process on how output generates from an input image.

before every convolution. The structure of a layer can be simply summarized 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 l^{th} 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 [19] to change number of input feature-maps, our method let each 1×1 convolution produce 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 function 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 Figure 12(b) and parameter setting is listed in Table III.

TABLE III
DETAILED PARAMETER SETTING OF THE NETWORK IN OUR EXPERIMENT.

Layers	Output Size	Operation
Convolution	32×32	7×7 Conv, stride 2
Pooling	16×16	3×3 Max Pool, stride 2
Block1	16×16	$\begin{bmatrix} 1 \times 1 & \text{Conv} \\ 3 \times 3 & \text{Conv} \end{bmatrix} \times 6$
Transition Layer	16×16	1×1 Conv
	8×8	2×2 Avg Pool
Block2	8×8	$\begin{bmatrix} 1 \times 1 & \text{Conv} \\ 3 \times 3 & \text{Conv} \end{bmatrix} \times 6$
Classification Layer	1×1	7×7 Adaptive Avg Pool
		6D fully-connected, softmax

C. Training

The training process is executed on a computer with Intel Xeon E5-2640 CPU, 128GB RAM, and a NVIDIA GTX 1080Ti GPU is used during training. Stochastic gradient descent (SGD) 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. We train the model using batch size 8 for 50 epochs, the whole training procedure takes about 25 minutes. L2 penalty is 1×10^{-5} in the experiment to prevent over-fitting.

D. Results

The goal of PCB defect inspection is defects detection and classification, while also minimizing the time expenditure of the method. Then, the performances of defect detection and defect classification are discussed. The metrics of defect detection is error rate P_d that defined as follows:

$$P_d = \frac{|d - a|}{a} \times 100\% \quad (9)$$

where d is the number of detected defect areas, a is the actual number of defect areas. And the metrics of defect classification are the classification precision rate (P_c) of each type of defect and the average precision rate (AP_c). P_c is defined in the following equation:

$$P_c = \frac{c}{a} \times 100\% \quad (10)$$

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

$$AP_c = \frac{1}{N} \sum_{i=1}^N P_c^i \quad (11)$$

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

1) *Defects inspection*: To verify the effectiveness of our reference comparison based method, we implemented the preprocessing and detection algorithm on our dataset. The statistics of the result is listed in Table IV. We can see that only a mouse bite and open circuit defect are needlessly detected the former is a wrong detection (ie, false detection of non-defect area) and the latter is an overlapped one (ie, a defect produces two similar overlapped results).

2) *Defects classification*: We test our classification model on the test data produced in section IV (A) by bounding box and all defect samples produced in section III by our reference comparison based method. It should be noted that before classifying the defects, we remove the repeatedly and incorrectly detected samples in the defect detection results to avoid the impact on the classification procedure. The result showed in Table V indicates that our 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 the reference comparison method is smaller than the image cropped by the bounding box given by the manual annotation. Resulting in the defect body accounts for a larger proportion when the image is resized to fixed resolution, which is more beneficial for classification.

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

V. CONCLUSION

In this paper, in consideration of lack of public shared PCB dataset, we produce and publicize a synthesized PCB dataset that has 1386 images with 6 types of common defects, including missing hole, mouse bite, open circuit, short, spur, spurious copper. Half of the images are for the situation where a test PCB is placed correctly, while other half is set for simulating the situation when the test board is randomly orientated in the workbench. Bounding box of every defect are 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 utilized as labeled data in object detection tasks. The transformation information is also provided to facilitate other researchers to study registration problems.

Based on reference comparison method, we introduce an end-to-end convolutional neural network model to classify the defects, which reaches impressive performance on our dataset. In order to learn more effectively, we do not choose simply stacking convolutional layers, instead, we use dense shortcuts inspired from Densenet 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 consumed of the entire detection process while achieve higher efficiency, what's more , designing effective non-reference compaison method to avoid using template.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Eccv*, 110(3):404–417, 2006.
- [2] Vikas Chaudhary, Ishan R. Dave, and Kishor P. Upla. Automatic visual inspection of printed circuit board for defect detection and classification. *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, 2018-Janua:732–737, 2018.
- [3] Shahrad Faghrib-Roohi, Hajizadeh Siamak, Alfredo Nez, Robert Babuska, and Bart De Schutter. Deep convolutional neural networks for detection of rail surface defects. In *International Joint Conference on Neural Networks*, 2016.
- [4] Xavier Glorot, Antoine Bordes, and Y Bengio. Deep sparse rectifier neural networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS) 2011*, 15:315–323, 01 2011.
- [5] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins. *Digital Image Processing Using Matlab*. Publishing House of Electronics Industry, 2009.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, pages 770–778, 06 2016.
- [7] Rudi Heriansyah, Syed Abdul Rahman Al-attas, and Muhammad Munim Ahmad Zabidi. Neural Network Paradigm for Classification of Defects on PCB. *Jurnal Teknologi*, 39(1):87–103, 2003.
- [8] Saeed Hosseinzadeh Hanzaei and Ahmad Afshar. Automatic detection and classification of the ceramic tiles surface defects. *Pattern Recognition*, 66, 11 2016.
- [9] Gao Huang, Zhuang Liu, Van Der Maaten Laurens, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017.
- [10] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Trans.on Acoustic.speech. & Signal Processing*, 27(1):13–18, 1979.

TABLE IV

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 V

DEFECTS CLASSIFICATION RESULTS, THE FIRST ROW REPRESENTS THE P_c s AND AP_c OBTAINED IN THE TEST DATA PRODUCED IN SECTION IV (A), AND THE SECOND ROW REPRESENTS THE P_c s AND AP_c OBTAINED FROM ALL SAMPLES IN SECTION III.

	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 VI

STEPS OF FILTERING AND MATHEMATICAL MORPHOLOGY PROCESSING IN OUR EXPERIMENT.

Step	Operation	Kernel(shape)
1	median filtering	5x5
2	closing	15x15(rectangle)
3	opening	3x3(rectangle)
4	median filtering	5x5
5	closing	29x29(ellipse)
6	opening	3x3(rectangle)
7	opening	1x1(rectangle)

and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

- [20] Wen Yen Wu, Mao Jun J Wang, and Chih Ming Liu. Automated inspection of printed circuit boards through machine vision. *Computers in Industry*, 28(2):103–111, 1996.
- [21] L. I. Zheng-Ming, L. I. Hong, and Jun Sun. Detection of pcb based on digital image processing. *Instrument Technique & Sensor*, 61(8):87–89, 2012.
- [22] Shiyang Zhou, Youping Chen, Dailin Zhang, Jingming Xie, and Yunfei Zhou. Classification of surface defects on steel sheet using convolutional neural networks. *Materiali in tehnologije*, 51:123–131, 02 2017.

TABLE VII

TIME CONSUMPTION OF EACH STEP, REGISTRATION TAKES UP THE MOST TIME, FOLLOWED BY LOCALIZATION AND BINARYZATION.

Procedure	Time(s)
Registration	0.6219
Binaryzation	0.1650
Localization	0.1808
Classification	0.0212
Total	0.9889

- [11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pages 448–456, 2015.
- [12] Shashi Kumar, Yuji Iwahori, and M K Bhuyan. Proceedings of International Conference on Computer Vision and Image Processing. 459, 2017.
- [13] Yann Lecun, Leon Bottou, Y Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.
- [14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016.
- [15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Arxiv*, 79, 11 2014.
- [16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [17] Christopher Pramerdorfer and Martin Kampel. A dataset for computervision-based PCB analysis. *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, pages 378–381, 2015.
- [18] Dr. Pierre Soille. Morphological image analysis: Principles and applications. Springer. *Sensor Review*, 28(5):800–801, 1999.
- [19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens,