The Journal of Engineering

The 2nd 2018 Asian Conference on Artificial Intelligence Technology (ACAIT 2018)

IET Journals
The Institution of Engineering and Technology

Defect detection of PCB based on Bayes feature fusion

eISSN 2051-3305
Received on 17th July 2018
Accepted on 26th July 2018
E-First on 10th October 2018
doi: 10.1049/joe.2018.8270
www.ietdl.org

Zhisheng Lu[1], Qinqin He[2] ✉, Xinguang Xiang[1], Hong Liu[2]

[1]Nanjing University of Science and Technology, Nanjing, People's Republic of China

[2]Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School, People's Republic of China

✉ *E-mail: qinqinhe@pku.edu.cn*

Abstract: With the continuous development of the electronics industry, the number of printed circuit board (PCB) has grown at a rapid rate, and the requirements for the detection systems of PCB have also continuously increased. In the traditional PCB detection, the main reference is the comparison method. However, in a real scene, there are a series of problems such as non-uniform illumination, tilting of the camera angle, and the like, resulting in a less satisfactory effect of the reference comparison method. So, the authors proposed a non-reference comparison framework of PCB defects detection. This framework has achieved good results in speed and accuracy. The authors extract the histogram of oriented gradients and local binary pattern features for each PCB image, respectively, put into the support vector machine to get two independent models. Then, according to Bayes fusion theory, the authors fuse two models for defects classification. The authors have established a PCB data set that includes both defective and defect-free. It has been verified that the accuracy of the verification set is improved compared to the individual features using the fused features. The authors also illustrate the effectiveness of Bayes feature fusion in terms of speed.

1 Introduction

In industrial production, a printed circuit board (PCB) is usually composed of copper wires, pads etc. A qualified PCB requires that both the pad and the copper wire are intact and there are no defects on the panel. Therefore, the inspection of the PCB defects is extremely important in the production line.

In the traditional production inspection process, workers manually detect each board on the assembly line, which is extremely time-consuming, inefficient and has high labour costs. Also, this manual detection method may cause a second damage to the board.

With the development of image processing technology and the advancement of high-quality imaging equipment, vision-based detection technology has been widely used in today's detection systems. These methods use high-definition cameras to collect the photographs of the boards and then embed detection algorithms into industrial control computer platforms to automatically detect these defects. Different detection tasks correspond to different types of defects. The OTSU algorithm [1] is a commonly used threshold detection algorithm that performs thresholding on the image through a bimodal distribution of grey histograms. Ng [2] extended the OTSU algorithm by selecting an optimal threshold including a unimodal distribution and a bimodal distribution. Both of the above methods use a greyscale histogram. There are also many other tasks that are based on the image feature. For example, Lijun [3] used the local gradient information of the image to study the flaw of the CBB capacitor. Kumar and Pang [4] proposed a defect detection method using a Gabor filter.

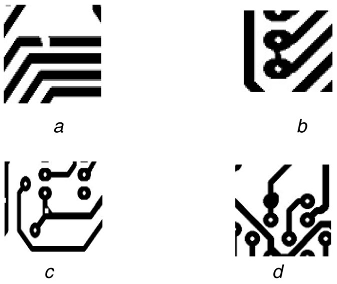


Fig. 1 *Example of different defects of PCB*
(a) Open circuit, *(b)* Short circuit, *(c)* Hollow, *(d)* Pad jam

However, the above methods are all based on image processing and perform well when the problem they deal with is simple. However, once the problem becomes complex, the result is unsatisfactory. In order to deal with complicated situations, we need to adopt more powerful technologies. Machine learning method is widely used and effective in computer vision [5], natural language processing [6], and so on. Some scholars have made a lot of contributions to the detection of industrial production. For example, Öztürk and Akdemir [7] used fuzzy c-means algorithm in unsupervised learning to detect pad defects in the PCB and get a better performance than some existing methods. (Fig. 1)

This paper proposes a new framework for PCB inspection. Firstly, after image preprocessing including image greying, image background separation, and image segmentation, we extract histogram of oriented gradients (HOG) features and local binary pattern (LBP) features separately to encode the texture information of PCB. Secondly, we put two features into support vector machine (SVM) separately for training, generating two independent models. Finally, Bayes feature fusion method is used to fuse the outcomes of two models. This method is compared with a method that uses a certain feature alone and we find that it performs better than the other two model. We have created a data set called PCBSET, and viewed the effectiveness of the verification set. The result shows that our framework has a good effect on accuracy and speed.

The organisation of the rest of this paper is as follow: Section 2 contains details of the proposed algorithm. In Section 3 experiments and experimental results are discussed. In Section 4, conclusion is summarised.

2 Methodology

The framework of the proposed PCB defects detection (PCBDD) is shown in Fig. 2. PCBDD has two important modules: features extraction and defects classification. In this framework, an input image was taken with a camera and resized to 800 × 600. After PCBDD processing, the detection system will output the classify result including the defects blocks of the image.

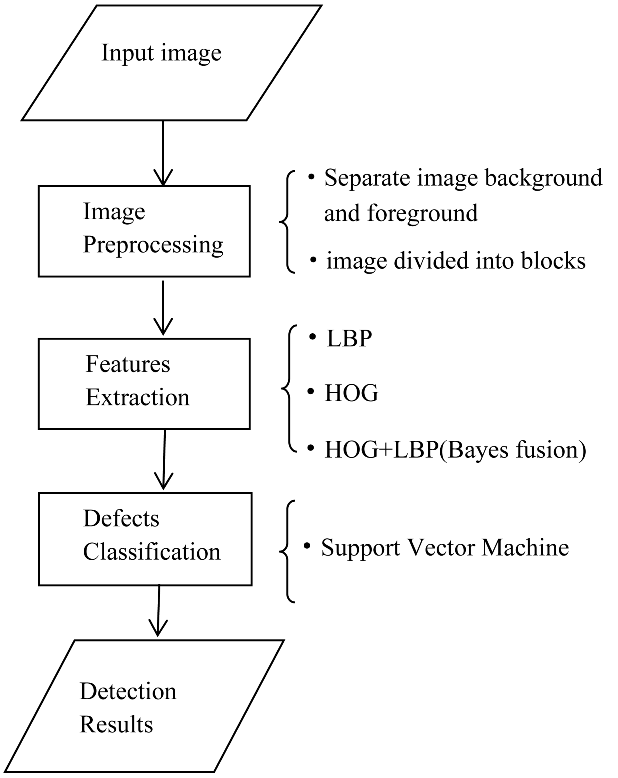


Fig. 2 *Framework of the proposed method*

2.1 Image preprocessing

As we all know, the original PCB image background is green and the illumination is not uniform, so we need to process the PCB image. First of all, we use the image segmentation technology to get the PCB above the wire, pad, and other areas, its edge is set to black, the rest of the area is set to white, including the background, so that we get a black–white PCB image. In the process, we also use the median filter [8] and mean filter [9] to denoise the image.

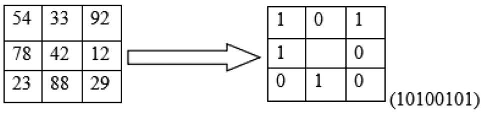
2.2 Features extraction

The proposed method is based on feature extraction and fusion and machine learning. Both LBP and HOG are image features that are widely used for image classification. LBP is usually used to describe the texture information, at the same time can very well capture the details of the image. Its most important property is its robustness to changes in greyscale caused by changes in lighting etc. Simultaneously, computational simplicity is also an advantage. HOG describes the local shape information and can describe the edge gradient information. It can reduce the dimensionality of the representation data, and it adopts the block processing method, which can well represent the connection between the local areas. PCB image has more gradient information and texture features, so it is suitable for using HOG features and LBP feature descriptions. We extract LBP features and HOG features from PCB block images and perform Bayes fusion on the two features.

We know that computers can only save one image by its pixel value. In order for a computer to 'know' or 'read' an image, it needs to extract useful information or data from the image and convert it into a vector, array, or symbol. This process is the extraction of features.

2.2.1 LBP features extraction: The LBP feature is an operator used to describe the local features of an image, with significant advantages such as grey invariance and rotation invariance, proposed by Ojala *et al.* in 1994 [10]. The LBP feature has a simple calculation as Fig. 3 shows, so it has been widely used in many areas of CV.

The definition of the original LBP feature is to traverse each pixel point and compare the pixel point with the surrounding eight-pixel values. If this pixel point is larger than the surrounding pixel points, the surrounding pixel point is marked as 0, otherwise, it is marked as 1. Therefore, each pixel around it is given a value of 0 or 1. Arranged according to certain rules, you can get an 8-bit binary number, which is the LBP value of this pixel [11].



54	33	92
78	42	12
23	88	29

1	0	1
1		0
0	1	0

(10100101)

Fig. 3 *Example of original LBP feature calculation*

00000000 0 jumps

00001111 1 jumps

00001100 2 jumps

Fig. 4 *Uniform LBP mode*

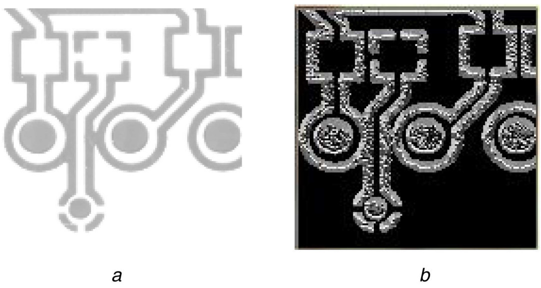


Fig. 5 *Example of a uniform LBP feature of a PCB image*
(a) Original image, ***(b)*** Result image

An improvement on the original LBP is Uniform LBP, i.e. equivalent mode or uniform mode LBP, which can greatly reduce the types of LBP features. This LBP feature is divided into two modes, one is an uniform mode, the LBP value of this mode has at most two jumps, and the other is a mixed mode, and its LBP value has at least three jumps [12]. Our so-called jump is a change from 0 to 1 or 1 to 0, such as 01 or 10 (Fig. 4).

In this paper, one improvement we have made is block Uniform LBP. If only the Uniform LBP feature is used to represent the image, the dimension is relatively small and it is difficult to represent all the information of the image. So we divide the image into many blocks of the same size, then calculate the Uniform LBP feature of each block, and then connect each one to form the LBP feature of the entire image. An example of a block-by-block Uniform LBP feature histogram of a PCB image is shown in Fig. 5.

2.2.2 HOG features extraction: The HOG feature, which is a directional gradient histogram, is commonly used in the image processing field and was originally used as a feature descriptor for pedestrian detection [13].

The HOG feature has many advantages that other features do not have. For example, the HOG feature is calculated in some areas of the image, and it has good robustness against uneven lighting. At the same time, the image's stretch scaling has little effect on the extraction of HOG feature. What's more, the histogram of each cell is calculated and then combined to reduce the dimension of the HOG feature of the entire image, depicting the contour in the image.

The HOG normalisation process can achieve the same effect as image normalisation colour values and gamma correction, no image preprocessing is performed [14]. A one-dimensional discrete gradient template is usually applied to the horizontal and vertical directions, such as the convolution kernel $[-1, 0, 1]$ and $[-1, 0, 1]^T$, and the gradient of the pixel (x, y) in the image can be calculated as

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (1)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \quad (2)$$

where, $G_x(x, y)$, $G_y(x, y)$, and $H(x, y)$ are, respectively, the horizontal gradient, the vertical gradient, and the pixel value at (x, y). Its gradient magnitude and gradient direction are [15]

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{3}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \tag{4}$$

As Fig. 6 shows, the process of extracting HOG feature is complex. To perform HOG feature calculations, we need to divide the image into several cells. The cell is the smallest unit of calculation. For each cell, for example, we use nine bin histograms to store the cell's gradient direction information, then we can divide the 360° direction into 9 parts. The gradient direction of the pixel is in any of these nine parts. The corresponding bin is incremented by one. At this point, the gradient direction of each pixel in the cell is totally mapped to the histogram, and a gradient direction histogram of the cell can be obtained. There are nine bins, it is a nine-dimensional feature vector [16]. Of course, the number of bins can be arbitrarily set.

The cell is relatively small, the gradient information may be less than enough, so according to a certain rule, several cells are combined into one block, and the histogram of each block is the connection of the gradient histogram of each cell under it. The area adjacent to the block may be duplicated, but the benefit is that the gradient information is continuous. Also, the direction gradient histogram in the block needs to be normalised. The purpose of this is to compress the brightness and shadow [17].

Finally, combine the descriptors of each block to form HOG features. The HOG feature dimension of an image is calculated as follows. The number of bins is multiplied by the number of cells in the block to obtain the dimension of the feature vector within the block. Then this dimension is multiplied by the quotient of the length, width, and step size of the image to obtain the final HOG feature's dimension.

2.2.3 Bayes fusion:

Feature fusion is mainly used to describe various different feature fusion methods. The common methods are pre-fusion, which is to splice various features together as described previously. There are also late-phase fusions that involve the process of multi-core learning, usually after feature extraction and training alone.

Bayes fusion is a kind of late-phase fusion [18]. The advantage of Bayes fusion is that, when learning with SVM, the most appropriate kernel functions for different features may not be the same. If you simply string together all the feature vectors, how to select the kernel function becomes the biggest problem. This is because if only one kernel function is selected, it may not be applicable to all features, resulting in poor results. The Bayes fusion can select the corresponding kernel function according to the feature vector. On the other hand, a single feature may not be able to represent the information of the entire image, and the advantage of Bayes fusion is that it can refer to the effects of multiple features, thus making the model more accurate.

Bayes estimation uses Bayes' theory to combine new evidence and prior probabilities to get new probabilities. It provides a method for calculating hypothetical probabilities based on hypothetical prior probabilities, the probability of observing different data under a given hypothesis, and the observed data itself [19].

Known mode space Ω contains c modes, denoted as $\Omega = \{\omega_1, \ldots, \omega_c\}$, unknown sample x consists of N-dimensional real-valued features, defined as $x = [x_1, x_2, \ldots, x_N]^N$. According to Bayes decision theory with minimum error rate, if the sample is divided into the j-th class, this class is the pattern class with the greatest posterior probability under the condition of known sample x [20]. This decision process can be expressed as

$$F(\omega_j) \overset{x \to \omega_j}{=} \max_{k = 1, \ldots, c} P(\omega_k | x) \tag{5}$$

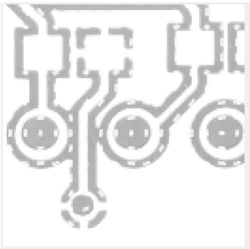


Fig. 6 *HOG feature of Fig. 5a*

Where $P(\omega_k | x)$ represents the posterior probability of the k-th class, $k \in \{1, 2, \ldots, c\}$.

Assuming x as the output of the classifier, the Bayes classifier fusion algorithm can be obtained. Assuming there are M classifiers, then each classifier will output a result y_i, so the characteristics obtained at this time are: $y = [y_1, \ldots, y_M]$. Then for an unknown sample y, the decision process can be expressed as

$$F(\omega_j) \overset{y \to \omega_j}{=} \max_{k = 1, \ldots, c} P(\omega_k | y_1, \ldots, y_M) \tag{6}$$

where $P(\omega_k | y_1, \ldots, y_M)$ represents the posterior probability of k-th class under the condition that M classifier output is known, $k \in \{1, 2, \ldots, c\}$. Based on this, we introduce the classifier independence assumption and combine the following formula:

$$P(\omega_k | y_i, \ldots, y_M) = \frac{p(y_1, \ldots, y_M | \omega_k) p(\omega_k)}{p(y_1, \ldots, y_M)} \tag{7}$$

then we can get the classifier fusion multiplication rules

$$F(\omega_j) \overset{y \to \omega_j}{=} \max_{k = 1, \ldots, c} P^{-(M-1)}(\omega_k) \prod_{i=1}^{M} P(\omega_k | y_i) \tag{8}$$

there will be a problem with this above, which is that when $p(w_k | x_i)$ is 0, it will cause problems. Based on the multiplication rules, the re-introduction of prior and posterior probabilities is approximately equal

$$P(\omega_k | x_i) = P(\omega_k)(1 + \delta_{ki}) \tag{9}$$

where δ_{ki} is very small.

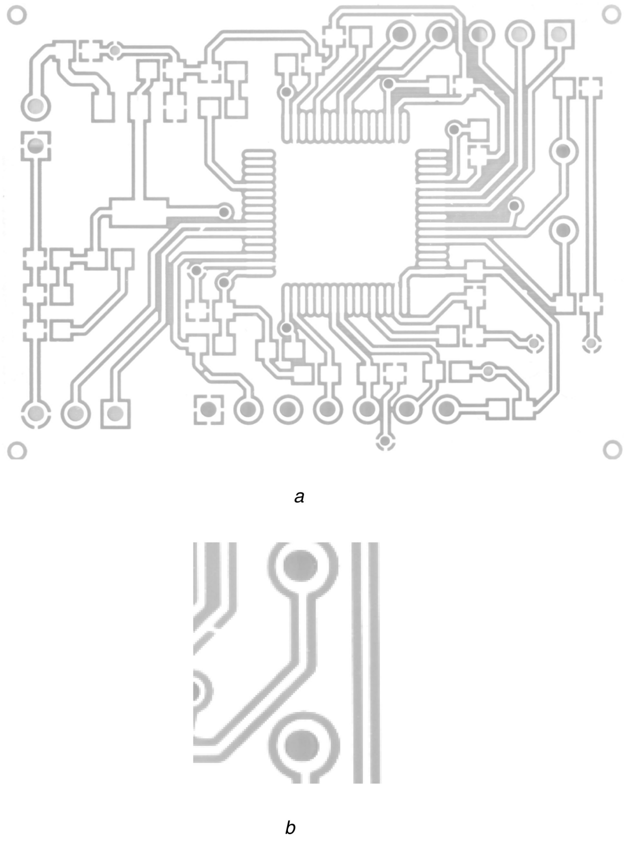
Ultimately, the classifier fusion addition rule can be derived

$$F(w_j) \overset{y \to \omega_j}{=} \max_{k = 1, \ldots, c} \left[(1 - M)P(\omega_k) + \sum_{i=1}^{M} P(w_k | y_i)\right] \tag{10}$$

2.3 Defects classification

In the actual industrial production process, a large number of qualified products and unqualified products have been accumulated. In order to make full use of these valuable information and improve the accuracy of PCB inspection, we adopt the traditional method of supervised machine learning: SVM to build the detector. The proposed algorithm marks eligible products as positive samples and unqualified products as negative samples. After extracting the features of these samples, the classifier is trained using SVM. When a new PCB arrives, the PCBDD uses a classifier to examine the image and output of the block when the sample is defective.

Given a set of training sets, each sample in the training set is marked as either a positive sample or a negative sample. The SVM creates a model by training. The training is performed by a certain algorithm. After the model is successfully trained, it is the classifier that can determine a new sample as a positive or negative sample. More intuitively, considering the data as a p-dimensional vector, the SVM classifier needs to find a p-1 hyper-plane that separates the points in space as much as possible. There may be


a

b

Fig. 7 *Sample image of PCBSET*
(a) Black-white PCB image, ***(b)*** One of PCB image block

many hyper-planes that can classify data, so we need to choose the optimal hyper-plane. The optimal hyper-plane is the longest Euclidean distance between the two classes. However, this SVM classifier is only suitable for linearly separable cases. When it is inseparable linearly, it is necessary to increase the dimension of the original dimension and divide the positive and negative samples in the higher dimensional space.

3 Experimental results and analysis

In this section, we first introduce the database PCBSET and the experiment setup. Then we evaluate the proposed method on the basis of PCB data sets from the aspects of accuracy, speed, and so on, and further validated the effectiveness of PCBDD.

3.1 Database introduction

In this experiment,we established a new dataset named as PCBSET as the database. The original PCB image size is ∼4000 × 3000, which is relatively large, so we normalised them to a size of 800 × 600 as Fig. 7*a* shows. Although the image size has become smaller, the image information is still much, and if it is used directly for training or detection, the effect will not be very good. Therefore, after normalising to the size of 800 × 600, we divide the image into blocks and divide the image into 256 × 256 image blocks in 64 steps as Fig. 7*b* shows. After blocking, it is obvious that there is a lot less information in the blocks. These blocks are usually innocent or contain only one kind of defect. PCBSET is consist of 148 qualified blocks and 148 unqualified blocks. Also we choose 56 images of each kind of blocks to be the test set.

3.2 Experiments setup

In our experiment, each image of PCBSET is divided and scaled to a uniform 256 × 256 pixel size. In feature extraction, we used block-by-block Uniform LBP for LBP extraction. The block size is 16 × 16 pixel size. For HOG extraction, the cell size is 8 × 8 pixels and the block size is 16 × 16 pixels, the number of the bins of the histogram of oriented gradient is 9.

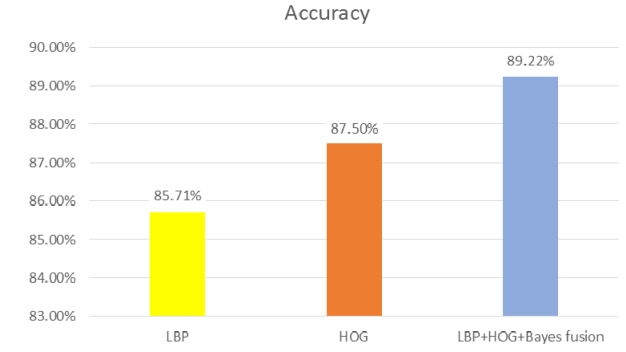


Fig. 8 *Result of defect classification accuracy*

Table 1 Speed of PCBDD

	Total time (train and test), s
LBP	11
HOG	18
LBP + HOG	20

The platform used in our experiment is Samsung 450rj with Windows 7 operating system, Intel Core I5-4210U processor. Feature extraction and SVM classifiers were built on MATLAB R2016a.

3.3 Detection accuracy

We selected 28 non-defective images and 28 defective ones from the PCB data set as test sets, and the remaining images were used as training sets.

For defect classification accuracy experiment, we use LBP feature, HOG feature, Bayes fusion of LBP feature, and HOG feature separately to train the detectors.

As Fig. 8 shows, after the Bayes fusion of LBP features and HOG features, the accuracy rate is higher than that of a single feature. This also shows that the proposed Bayes feature fusion method is effective. At the same time, it can be seen that the HOG feature is more suitable for the detection of PCB defects than the LBP feature. It also shows that most features in the PCB images are gradient information rather than texture information, so there is a big difference in local gradient information between different PCB images.

3.4 Detection speed

In this section, we provide an analysis of the detection speed for the proposed PCBDD framework. The time is calculated from reading the training set image to finish detecting the PCB image.

According to Table 1, the LBP method is the fastest defects detection algorithm in the PCBDD framework. Although HOG method achieved better classification accuracy in the detection accuracy experiment, it took a longer time to train the detector. From this experiment, we can see that the time spent on feature fusion of LBP and HOG is not very long, just two seconds more than the HOG method, which shows that our method is feasible in terms of time.

3.5 Defects selecting

After training the model, the PCB board image can be predicted. Since the image we train is a 256 × 256 PCB sub-image, after reading a complete PCB image, we need to block the PCB image into 256 × 256 sub-image in 64 steps and then put into the SVM for prediction whether the image is flawed.

Then we will put the detected sub-images into the original image, highlighting defect areas by deepening the blue channel as Fig. 9 shows. Basically, defects have been detected and highlighted.

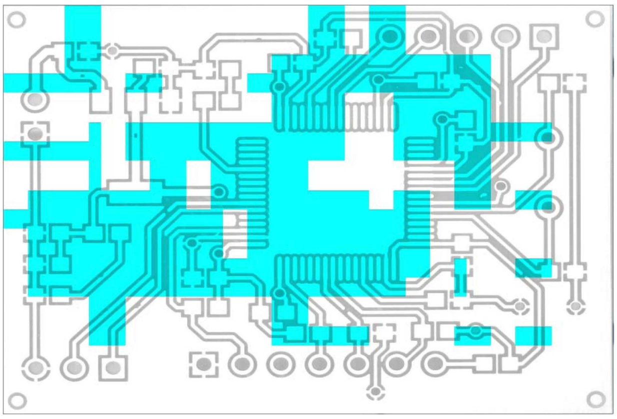


Fig. 9 *Example of defects selecting*

4 Conclusion

In this paper, we present a complete PCB defect inspection (PCBDD) framework based on machine learning. The framework solves the problems in the traditional reference comparison method, such as uneven illumination. Specifically, in this method, LBP features and HOG features are used to describe the features of a PCB image, training by SVM to get two models. Bayes feature fusion is used to fuse those two models and then obtain a fused model. This method is compared with a method that uses a single feature. It is found that there is a significant improvement in the accuracy rate, and at the same time, the speed is not much slower, indicating that our proposed algorithm is effective as the calculation is fast and accuracy is higher. Therefore, the proposed PCBDD frame can be used for actual PCB surface defect detection.

In the future, we will convert the two categories problem into a multiple categories problem and implement an algorithm to detect the location of the defect. Also, we will try to fuse different features for testing.

5 Acknowledgments

This work is supported by Specialised Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (No.ZLZBCXLJZI20160729020003).

6 References

[1] Otsu, N.: 'A threshold selection method from gray-level histograms', *IEEE Trans. Syst. Man Cybern.*, 1979, **9**, (1), pp. 62–66

[2] Ng, H.F.: 'Automatic thresholding for defect detection', *Pattern Recognit. Lett.*, 2006, **27**, (14), pp. 1644–1649

[3] Lijun, X.: '*Research on target surface helium detection system based on machine vision*' (Hangzhou Dianzi University, Hangzhou, 2015)

[4] Kumar, A., Pang, G.K.H.: 'Defect detection in textured materials using Gabor filters', *IEEE Trans. Ind. Appl.*, 2002, **38**, (2), pp. 425–440

[5] Lafferty, J., McCallum, A., Pereira, F.C.N.: 'Conditional random fields: probabilistic models for segmenting and labeling sequence data', 2001

[6] Manning, C.D.: '*Foundations of statistical natural language processing*' (The MIT Press, Cambridge, 1999)

[7] Öztürk, Ş., Akdemir, B.: 'Detection of PCB soldering defects using template based image processing method', *Int. J. Intel. Syst. Applic. Eng.*, 2017, **5**, (4), pp. 269–273

[8] Tukey, J.W.: 'Exploratory data analysis', 1977

[9] Bovik, A.C.: '*Handbook of image and video processing*' (Elsevier Academic Press, London, 2010, 2nd edn.)

[10] Ojala, T., Pietikainen, M., Maenpaa, T.: 'Multiresolution gray-scale and rotation invariant texture classification with local binary patterns', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, **24**, (7), pp. 971–987

[11] Fang, Y., Luo, J., Lou, C.: 'Fusion of multi-directional rotation invariant uniform LBP features for face recognition'. IEEE, 2009, pp. 332–335

[12] Wu, M.L., Fahn, C.S., Chen, Y.F.: 'Image-format-independent tampered image detection based on overlapping concurrent directional patterns and neural networks', *Appl. Intell.*, 2017, **47**, (2), pp. 347–361

[13] Dalal, N., Triggs, B.: 'Histograms of oriented gradients for human detection'. 2005 IEEE Computer Society Conf. on Computer Vision and Recognition (CVPR), San Diego, USA, June 2005, pp. 886–893

[14] Newell, A.J., Griffin, L.D.: 'Multiscale histogram of oriented gradient descriptors for robust character recognition', *IEEE Comput. Soc.*, 2011, pp. 1085–1089

[15] Wang, X., Han, T.X., Yan, S.: 'An HOG-LBP human detector with partial occlusion handling'. 2009 IEEE 12th International Conf. on Computer Vision, Kyoto, Japan, 2009, pp. 32–39

[16] Zhu, Q., Yeh, M.C., Cheng, K.T., *et al.*: 'Fast human detection using a cascade of histograms of oriented gradients'. 2006 IEEE Computer Society Conf. on Computer Vision and Recognition (CVPR), New York, USA, June 2006

[17] Llorca, D.F., Arroyo, R., Sotelo, M.A.: 'Vehicle logo recognition in traffic images using HOG features and SVM'. IEEE, 2013, pp. 2229–2234

[18] Juang, B.H., Hou, W., Lee, C.H.: 'Minimum classification error rate methods for speech recognition', *IEEE Trans. Speech Audio Process.*, 1997, **5**, (3), pp. 257–265

[19] Yang, J., Yang, J.Y., Zhang, D., *et al.*: 'Feature fusion: parallel strategy vs serial strategy', *Pattern Recognit.*, 2003, **36**, (6), pp. 1369–1381

[20] 'CSDN': https://blog.csdn.net/wuye999/article/details/79152049, accessed 2 January 2018.5