The Journal of Engineering

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

Improved bare PCB defect detection approach based on deep feature learning

eISSN 2051-3305
Received on 17th July 2018
Accepted on 26th July 2018
E-First on 29th October 2018
doi: 10.1049/joe.2018.8275
www.ietdl.org

Can Zhang[1], *Wei Shi*[2] ✉, *Xiaofei Li*[3], *Haijian Zhang*[1], *Hong Liu*[2]

[1]*Electronic Information School, Wuhan University, Wuhan, People's Republic of China*
[2]*Key Laboratory of Machine Perception, Peking University, Shenzhen, People's Republic of China*
[3]*INRIA Grenoble Rhône-Alpes, Saint Ismier, France*
✉ *E-mail: lusinda@whu.edu.cn*

Abstract: Robust and precise defect detection is of great significance in the production of the high-quality printed circuit board (PCB). However, due to the complexity of PCB production environments, most previous works still utilise traditional image processing and matching algorithms to detect PCB defects. In this work, an improved bare PCB defect detection approach is proposed by learning deep discriminative features, which also greatly reduced the high requirement of a large dataset for the deep learning method. First, the authors extend an existing PCB defect dataset with some artificial defect data and affine transformations to increase the quantity and diversity of defect data. Then, a deep pre-trained convolutional neural network is employed to learn high-level discriminative features of defects. They fine-tune the base model on the extended dataset by freezing all the convolutional layers and training the top layers. Finally, the sliding window approach is adopted to further localise the defects. Extensive comparisons with three traditional shallow feature-based methods demonstrate that the proposed approach is more feasible and effective in PCB defect detection area.

1 Introduction

The printed circuit board (PCB) is an essential component of various electronic devices, such as a computer, cellphone, fridge and so on. In the background of the wide application, it is of great significance to fabricate defect-free PCB. Due to the rapid development in the electronic information field, the PCB manufacturers pursue to get a final product, which is not only more qualified and integrated but also much smaller. Since the traditional way to detect PCB defect is always done by a human, it is always judged subjectively and inefficiently. Drawbacks like this have made it no longer satisfy the current manufacturing situation. Thus, it is necessary and meaningful to introduce an efficient automatic detection system of low cost into the environment. Especially for those small businesses that are not able to afford the expensive automated optical inspection (AOI) system.

This study focuses on studying computer vision based defect detection algorithm for bare PCB. It can be an alternative to the AOI equipment and also emancipate human labours from the tedious production line. The key benefits of applying the computer vision to the defect detection lie in the high efficiency and good stability. Thus, many researchers have put great efforts to incorporate the computer vision technology into the different defect detection fields. Celik *et al.* [1] developed a real-time machine vision system using neural networks for fabric defect detection. Feng *et al.* [2] proposed an automatic railway inspection system for fastener defects detection, and a probabilistic model was used. Koch *et al.* [3] presented a visual inspection system to detect defects in civil infrastructures, such as concrete bridges and tunnels. They also explored the potential of computer vision in detecting defects of other civil engineering applications. Shanmugamani *et al.* [4] utilised machine learning techniques to evaluate gun barrel defects such as wear, erosion, and rust. In PCB defect detection field, machine vision has also been applied to explore an algorithm, in which the accuracy, efficiency, and real-time performance are pulled together. As there are two types of PCB, bare and assembled PCB, the defect detection algorithm will put emphasis on different aspects for different study objectives.

For a bare PCB, which is exactly the one this paper studies, Ibrahim *et al.* [5] applied the HAAR wavelet transform to the standard and defected PCB images, and then the difference operation was done in the wavelet domain. The output was generated by performing a XOP logic operation. Chang *et al.* [6] proposed a case-based identification model which contains two steps. The first step is to store the defected images to form a concept space, and the second step is followed by calculating the relative position of similar cases with the given pattern in the concept space. Khalid *et al.* [7] made use of several image processing operations such as image subtraction and logical XOR, which successfully classified 14 common defects. Ismail *et al.* [8] adopted a new algorithm, which consists of operations such as image difference, image subtraction, and image comparison, to detect five types of defects. Putera *et al.* [9] employed a morphological image segmentation algorithm to improve the image processing algorithm. Finally, 13 defects were classified into seven categories. They further improved the algorithm by increasing categories from 7 to 11, and the production cost was reduced in the meantime [10]. Ray *et al.* [11] adopted the Hybrid approach which could not only detect PCB defects but also classify and locate the defects. To decrease the limitation of only using the single feature to represent defects, Li *et al.* [12] proposed to fuse the gradient direction information entropy and the local binary pattern (LBP) feature to construct a feature vector, which can be used to describe PCB defects. Also, finally, a support vector machine (SVM) classifier was trained for recognising defects on the bare PCB.

As the literature reviewed above show, although researches of PCB defect detection have made great progress in recent years, most of the researchers remain to employ the image processing techniques to detect PCB defects [8–10]. Also, many researchers adopt a template matching algorithm to the standard image and defected one [7–9]. Although the feature extraction is integrated into the classifier such as SVM [12], the inspection system still cannot work very well when the detection environment changes. Since the real environment of PCB defects is much more complicated, those classical feature descriptors such as LBP are relatively too simple to represent the image. With the generation of convolutional neural networks (CNNs) [13] in the computer vision area, some more sophisticated networks are introduced to be the feature extractors. The deep features they learnt have substantially outperformed the shallow features extracted by previously handcrafted descriptors. For some well-known large dataset such as ImageNet [14] that contains 1000 classes, they can perform

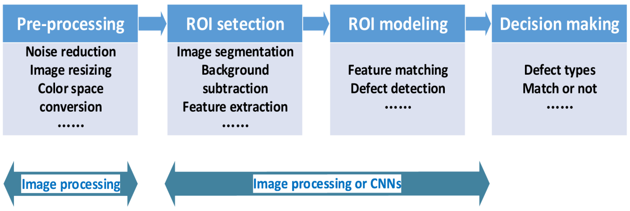


Fig. 1 *Typical PCB defect detection flowchart*

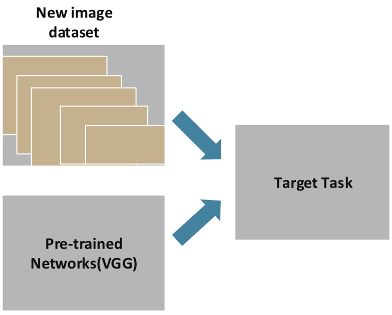


Fig. 2 *Transfer learning rout map*

excellently [15, 16], and obtain better results than traditional methods significantly. However, most research studies have not applied the deep architectures to the PCB defect detection fields, and it remains unknown whether the deep features about PCB defects could perform better than shallow features. If like most other cases, we would like to know the difference between deep features, which we learnt from a large quantitative image data, and shallow features. Despite the success, these high-level feature representations have won, it is known that a large set of labelled data are required to train a deep CNN model. However, for many problems, it is not easy to obtain so many images, which limits the application of the very deep CNNs to many fields.

The purpose of this research is to explore the possibility of deep feature learning algorithm in PCB defect detection field, especially with limited defected PCB samples. On the basis of the unified PCB defect dataset we created, this study also makes a systematic evaluation of detection results with different implementation methods. We adopt a pre-trained VGG [17] model to deal with the PCB defect detection problem. Then comparisons are made with shallow feature-based models which are combined with handcrafted feature descriptors and SVM, such as scale-invariant feature transform (SIFT) [18, 19] and LBP [20, 21]. Bag-of-Words (BoW) [22, 23] is used to improve the model performance as well. In addition, a deep feature-based model is also taken as a reference, which is implemented with AlexNet [24] and SVM. A set of experiments are carried out to evaluate their performance. The experimental results prove that the deep CNNs outperform all other traditional vision algorithms on the relatively small dataset we created and could deal with PCB defect detection effectively.

The main work of the proposed bare PCB defect detection system is discussed in a different section as listed. Section 2 describes the methodology in detail of this research, which would include the basic skeleton of the model and detection network based on the very deep CNN. Section 3 contains the results and analysis of experiments for defect detection. Finally, Section 4 concludes this work.

2 Methodology

In the traditional PCB defect detection system; there are several processing steps to be taken for the original images. Fig. 1 shows a typical defect detection flowchart, which comprises four steps: image pre-processing, regions of interest detection, defect detection and decision making. The first two steps are directly operated on the original images, which makes it greatly rely on the detailed information of input. The defect detection stage is relatively computational because this step has to process a large number of feature representations. In this section, we detail an end-to-end method to improve the defect detection system. Multiple intermediate steps are incorporated into the network, which greatly increases the detection accuracy and efficiency.

2.1 Base network

We adopt VGG-16 [17] as our base network to do the feature extraction. Since the input of VGG Net is required to be the RGB image with a size of 224 × 224, the defected images in our dataset are all resized from 128 × 128 to the designated size. The VGG model is pre-trained upon the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [25] to categorise around 1.2 million images to 1000 classes. There are five convolutional layer stacks, and very small filters of 3 × 3 size are used on the feature maps with a fixed pixel stride 1. The first two stacks have two convolutional layers, and the output of each layer is activated by rectified linear unit [24] non-linear function. So as to the other three convolutional stacks that has three convolutional layers. At the end of each stack, a max-pooling layer is connected. It is performed with 2 × 2 window size and 1 pixel stride. Three fully connected layers are then followed to the convolutional stacks and max-pooling layers. The outputs of the first two layers are all in size of 4096, while the last layer is the softmax layer which is depended on the specific classification problems. It has 1000 channels when dealing with the ImageNet ILSVRC challenge [25], while in our work, the number of channels is set to 6 according to the defect categories. Also, for the reason of additional memory consumption, the local response normalisation [24], a typical normalisation operation, is not applied to the network.

The disadvantage of CNNs is a huge amount of labelled sample images for training, especially for the deeper networks which have even more parameters to learn, such as 22 layers [26]. Therefore, if we directly use only thousands of images in our small dataset to train the deep networks, facing so many parameters, it will easily fall into the serious over-fitting situation. On the challenge of making the deep network VGG fit for our small dataset, we combine the transfer learning method with data augmentation and then use the pre-trained CNN to extract the deep discriminative features. This approach not only enlarges the training dataset but also reduces the high requirements on the image quantity.

2.2 Deep feature learning

As mentioned before, to obtain millions of parameters in the deep CNNs, we need training samples in large quantities. The lack of defected PCB images motivates us to explore the scheme to address the problem. Therefore, this study applies two methods: parameters transfer learning and data augmentation.

2.2.1 Parameters transfer learning: For a CNN model, although its configuration is known, the way how it works remains a mystery to us. The knowledge that we can use is that image features in low-level such as edges are extracted in lower convolutional layers, and the higher convolutional layers mainly extract more detailed and complicated features. The fully connected layers act to capture features related to the specific categories for the classification task. The handcrafted feature descriptors only extract the shallow representations, which is difficult to be transferred to identify other new objects. Fortunately, the deep discriminative features can be obtained in CNNs about the compositions of images and the specific combination such as shapes and edges. All the critical information to recognise the objects are contained in the last several convolution layers and fully connected layers. Thus, in this study, we need not learn new parameters in the network by training such a small dataset, we can simply use what has been pre-trained in a state-of-art VGG model on ImageNet. These pre-trained parameters are then fine-tuned with an optimal learning rate to fit for our classification task. Fig. 2 shows the routing map of transferring parameters from a large existed dataset to a small target dataset.

2.2.2 Pre-trained model obtaining: Since the learning process of CNNs is usually computational slow, a powerful graphic processing unit (GPU) is always needed to accelerate it. Training a

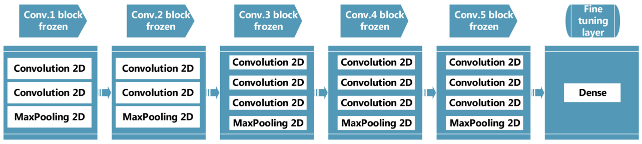


Fig. 3 *Configuration of our fine-tuned model*

VGG model usually takes several days, with millions of iterations, to learn the weights of the networks. It not only requires high-computational power, as well as long time and large dataset. As many people have released their training model parameters, in this study, we use the shared weights to fine tune the network.

2.2.3 Fine-tuning: This strategy is adopted by freezing all the convolutional layers and replacing the fully connected layers with a new dense layer. Parameters are then updated by our own data. As Fig. 3 shows, the entire fully connected layer is removed from the VGG model which is pre-trained on ImageNet. Then the remained convolutional layers are regarded as a feature extractor. In the proposed network, the features extracted will be transferred to a dense layer and fine-tuning technique is followed on our own small dataset. The output layer can be replaced by a trained linear classifier, such as SVM or softmax.

For the front convolutional layers, most of them contain the generic representations which could be transferred to other recognition tasks, such as edges and shapes. The details of the images are mainly in the last several layers. Therefore, in our study, we just make the first five convolutional layer blocks frozen, for the purpose of making full use of the information learnt in ImageNet. Then we use back propagation to fine-tune the weights of the proposed network on our PCB defect dataset. Due to the small size of our dataset, it would be better to only train the dense layer, which can effectively avoid the over-fitting problem.

2.2.4 Data augmentation: On the basis of the limited defected PCB images, in order to improve the detection accuracy, we use data augmentation techniques to expand the created dataset. There are many means to implement it, such as linear and nonlinear transformation. In our work, we apply two methods. One is some simple affine transformations, such as rescaling, shift, mirror and so on. These basic techniques help a lot to prevent over-fitting on such a small dataset.

2.3 Localisation

The sliding window approach is adopted in our defect localisation stage. This method in object localisation field has been used for a long time [27–29]. It mainly depends on the evaluation score of the classifier upon the window area in the image. Select the maximum as the final recognised label and make annotation by drawing rectangular in this region. Formally, it can be written as

$$\boldsymbol{R}_{\text{obj}} = \arg\max\nolimits_{\boldsymbol{R} \subseteq \boldsymbol{I}} f(\boldsymbol{R}), \tag{1}$$

where $\boldsymbol{R}$ represents all the window-region matrixes over the image $\boldsymbol{I}$. Since it is too computational to slide a window pixel by pixel over an $M \times M$ image, especially when M is very large, we set a fixed size between each window. The experiment shows that this coarse search approach could reduce the number of windows effectively and has little effect on the localisation results by comparing three different strides.

2.4 Loss function

CNNs mostly have the same configuration, and it is essentially a supervised learning process of minimising loss function to optimise a CNN. The mathematic formulation can be described as

$$L = f_{\text{loss}}(\boldsymbol{y}, F(\boldsymbol{W}, \boldsymbol{x})), \tag{2}$$

where L denotes the cost of the total training images. We use F as a collection of operations in the network, such as convolution operations and pooling operations. Let W represent the parameters that can be adjusted by the iterative learning process. The predicted output label $F(W, \boldsymbol{x})$ and the desired category label matrix $\boldsymbol{y}$ are put into the loss function f_{loss} to calculate the cost.

In a CNN, the commonly used loss function is softmax matching with cross entropy loss function. Since the outputs of the CNN are the evidence of input instances, the softmax is used to transfer this evidence into the form of probability

$$p_i = \frac{\mathrm{e}^{s_i}}{\sum_{k=1}^{T} \mathrm{e}^{s_k}}, \tag{3}$$

where s_i denotes the ith CNN output. The probability p_i represents the confidence level of input belonging to the ith category in the total T categories. Then we use it to compute the cross-entropy loss:

$$f_{\text{loss}} = -\sum_{i=1}^{T} y_i \log(p_i), \tag{4}$$

where y_i is the ith category label. The parameter update formula can be obtained by using the stochastic gradient descent (SGD) optimiser

$$W_{\text{new}} = W_{\text{old}} - \lambda \nabla L(\boldsymbol{W}_{\text{old}}), \tag{5}$$

where λ is the learning rate and the gradient of L is computed with the current parameters W_{old}. The new parameters W_{new} are then updated towards the direction of maximum change.

3 Experiments and analysis

3.1 Dataset description

Our work is based on the PCB images provided by the author of ‘defect detection and recognition of bare PCB based on computer vision’ [30]. However, there are only 99 original PCB images, and the number of defects in each PCB is rather small. Just cutting out the small defect area of the PCB as our dataset is surely far from enough to deal with our classification task. Therefore, we enlarge the dataset with some artificial defects on the bare PCB and then mix them with true defects together. The final dataset is divided into three sets, one for training, one for validation, and the other for testing. Each category contains six types of defected images. Since there are more images of the normal type in the PCB, we make samples of this type as much as possible to avoid misclassification in our dataset. The training set has a total of 3800 images, with 1000 normal type and each with 560 defects for the other five types, including copper, short, open, spurs and mouse-bite. The validation set accounts for 20% of the entire set, namely 950 images, which has the same size as the testing set.

3.2 Evaluation metrics

We adopt the evaluation criteria for multi-class image dataset classification task from [31]. It is designed to access the accuracy of correctly classified images, the cost of misclassification, and the trade-off of detection errors between different defect types. To have a quantitative evaluation of the algorithms, the average precision (AP) is used in the defect classification. Precision is the percentage of positive instances in all selected samples, while recall is the ratio of correctly selected instances in the entire positive samples. The computation formula can be represented by the true positive (TP), false positive (FP) and false negative (FN)

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{6}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{7}$$

Table 1 Statistics of the PCB defect dataset

Defect type	Training set	Validation set	Testing set
normal	1000	250	250
copper	560	140	140
short	560	140	140
open	560	140	140
spur	560	140	140
mouse bite	560	140	140
total	3800	950	950

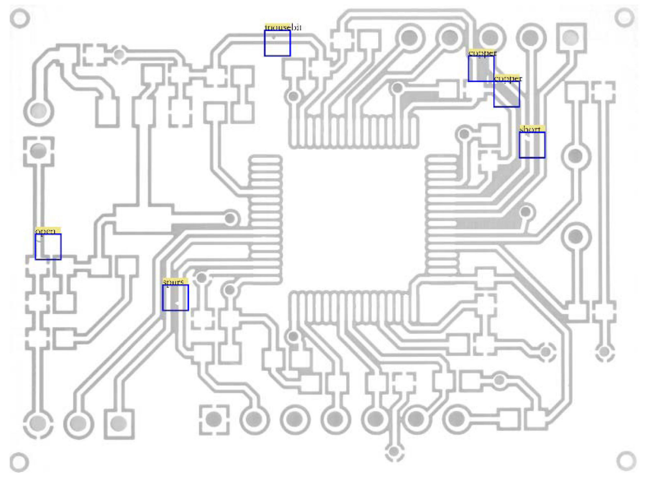


Fig. 4 *Detection results of the proposed method. Four bounding boxes upper (left to right) are labelled as mouse bite, copper, copper, and short. The boxes downside (from left to right) are open and spur*

The precision-recall curve (PRC) is computed to evaluate the performance of the classification problems, which set the recall value as the *x*-axis and precision as the *y*-axis. The area below PRC is summarised and averaged upon the recall levels spaced at a fixed value between zero and one. Take interval *m* for example, the AP is obtained by doing the average operation upon the maximum measured precision $p_{\max}(r)$ when it exceeds recall level *r*

$$\mathrm{AP} = \frac{m+1}{m} \sum_{r \in [0, m, \ldots, 1]} p_{\max}(r)\,. \tag{8}$$

3.3 Implementation details

3.3.1 Dataset: Our dataset is created using a sliding window with the size 128 × 128 pixels over the 4072 × 3046 PCB images. The step is set 64 pixels, and half-overlapped windows are then cropped and saved in the set. Choosing 128 as the window size is for the reason that this area contains all the features of the defects and will not carry too much background information to the classifier as well. The cropped small ROIs are then manually selected for different defect sets. The validation set and training set are randomly divided at a ratio of 1:4. The testing set is made the same size as the validation set. Table 1 is the number of the dataset in each category.

3.3.2 Pre-processing: Due to the small number of labelled defects, we firstly make some pre-processing techniques to enlarge the dataset. The rotation operation is used with a range of 0–40°. Also, the images randomly zoom in the range of 0.8–1.2. Apart from these, the shift operation is also applied. Small images can shift either horizontally or vertically by 20% of the image height pixels. The Boolean value of horizontal flip is set to true, which means it can randomly flip. When the whole pre-processing process is finished, the dataset is greatly expanded. As a fixed size of the model input is required, the images are resized to 224 × 224 pixels. Some research studies resize the images to a larger size to augment the dataset, e.g. 256 × 256 pixels. Also, then they randomly crop the image with the required size, which proves to perform well in some classification cases [24]. However, the defects in our dataset are not always located in the centre, and most are situated in the peripheral area instead, such as the corner of the small window region. Therefore, in this study, we do not take our dataset that way, and only scale the entire small images to the required size. Also, then our model is fine-tuned on each defect set.

3.3.3 Classification methods: Besides the proposed method, three shallow feature-based methods and 1 deep feature-based method are also adopted as references. The overall methods include the algorithms which combine SVM with the LBP and the histogram-oriented gradient (HOG) feature respectively. The SIFT feature described by the BoW model is also used to train the SVM classifier. A transferred model which regards the AlexNet as the feature extractor and SVM as the classifier is set as another reference. Our proposed method uses the pre-trained VGG network to learn the high-level feature contained in the defects. All the convolutional layers are frozen, and the top layers are replaced with a new softmax layer, the parameters of which are then fine-tuned on the target small dataset. The evaluation metrics are used to compare the performance of each method objectively.

3.3.4 Training details: In the study, we use stochastic gradient descent (SGD) to optimise the proposed network, and the momentum is set to 0.9. The initial learning rate is usually set in the range of 0.01–0.0001. Considering the network configuration and the dataset we used, we take value 0.0001 as the learning rate and 100 as the training epochs. The update formula of the learning rate in every iteration is shown as

$$\mathrm{lrn} = \mathrm{lrn}_{\mathrm{ini}} \times \left(1 - \frac{\mathrm{iter}}{\max\ \_\mathrm{iter}}\right)^{0.5}. \tag{9}$$

Mini-batches learning is adopted. To make the training process executable and GPU parallel performance using rational, a suitable batch size is searched from a relatively small value, which is fixed to the power of 2. In the end, we take eight as the batch size, which helps to avoid the local minimum value but converge to a flat minimum. Also, compared with the large batch, the small value improves the generalisation of the network.

3.4 Comparisons of PCB defect detection approaches

Fig. 4 shows the detected defects in the PCB image based on the proposed method. The defects detected contain 1 short, 1 mouse bite, 2 coppers, 1 open and 1 spur. Each predicted defect is marked out by a rectangular box in blue, and the predicted type is annotated in the yellow box. This method successfully detected all the defects in the image without omission. Also, it outputs a non-misclassification result, in which the model does not recognise normal type as a defect and vice versa. The detection result for coppers shows that our model is relatively robust even when the defect resides in the peripheral regions of the detecting windows. This could help us set a larger stride of the sliding window to reduce the detection time, and in the meantime, the accuracy is maintained. Furthermore, Fig. 5 gives the detection result of the model based on the combination of AlexNet and SVM, which achieves the highest AP among the other four methods. The output shows that the normal area in the PCB images is easily recognised as the defect. Also, some defects are too small to be detected. However, our model successfully avoids the mistakes and obtains a relative ideal detection output. The result qualitatively proves that the proposed method outperforms the other classification methods.

To quantitatively compare our method with other algorithms, we list the AP and mean AP (mAP) in Table 2. Also, the PRCs of six defect types are plotted in Fig. 6. These evaluation metrics help to give a specific understanding of the overall performance. From Table 2, we can see that the proposed method obtains the highest mAP at 99.59%, and it exceeds the second highest method by 8%which is based on a combination of Alexnet and SVM. Comparing with the shallow feature-based methods, mAps of deep networks in the last two columns show an increase of 20% at least. Take the row of the copper defect, e.g. the AP is around 65% for

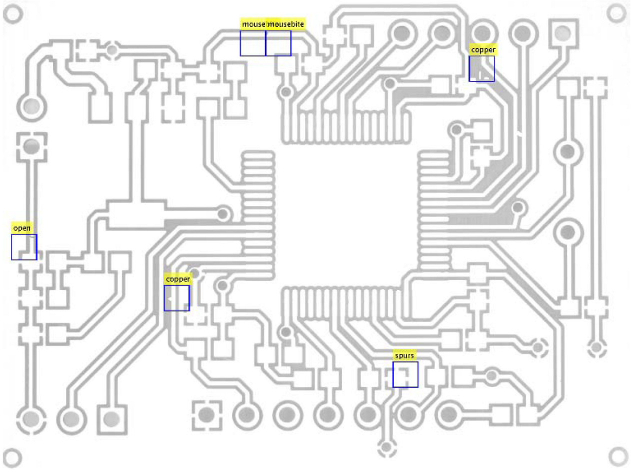
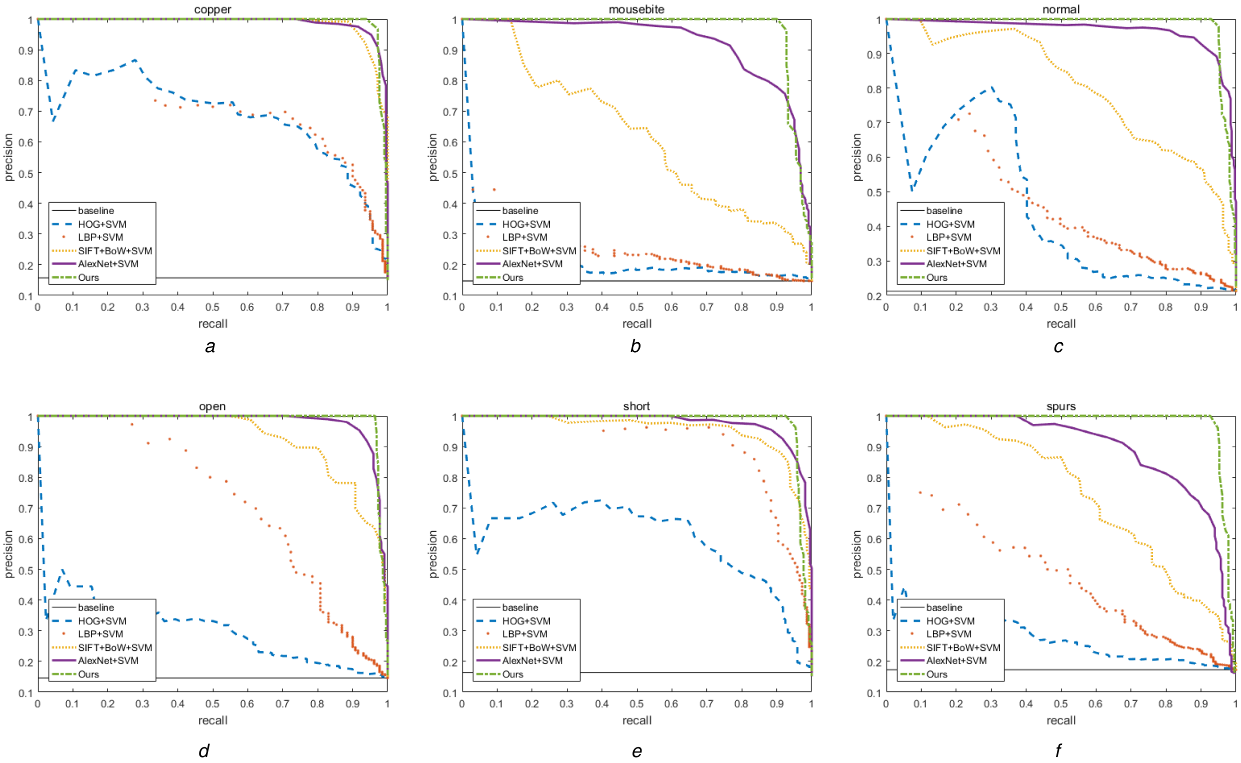


Fig. 5 *Detection results based on the combination of AlexNet and SVM. Short defect at the upper right corner is missing. The bottom bounding box is the misclassification of normal type*

Table 2 AP and mAP of five methods based on shallow features and deep features

Methods	HOG + SVM	LBP + SVM	SIFT + BoW + SVM	AlexNet + SVM	Ours
copper	0.6571	0.8214	0.8851	0.9718	0.9989
mousebite	0.1298	0.2748	0.4392	0.8468	0.9869
normal	0.4656	0.4444	0.6824	0.9721	0.9996
open	0.3615	0.5692	0.7297	0.9595	0.9934
short	0.4863	0.8219	0.8581	0.9259	0.9998
spurs	0.3117	0.4416	0.6757	0.8268	0.9899
mAP	0.4040	0.5622	0.7506	0.9172	0.9959



a *b* *c*

d *e* *f*

Fig. 6 *PRC of six defect types with five methods. (HOG + SVM method is blue (dashed). LBP + SVM is red (dotted). BoW method is orange (small dotted). AlexNet is purple (solid). Ours is green (dash-dotted))*
(a) Copper defect, ***(b)*** Mouse bite defect, ***(c)*** Normal, ***(d)*** Open defect, ***(e)*** Short defect, ***(f)*** Spur defect

the HOG method and 99% for the proposed method, where the increase is up to 34%. Also, the AP of the mouse bite defect nearly doubles over the BoW based method. This significant increasing value demonstrates the great performance of deep feature learning, in which the high-level features are extracted to discriminate the difference among the defects.

From the curves in Fig. 6, we can see that the deep networks obtain a higher area, which means both high precision and recall. The high precision proves that the deep feature-based methods have much better accuracy. Also, a much greater number of positive samples are returned, which is shown by the high recall value. The HOG- and LBP-based models show a result of the high recall and low precision, especially in the mouse bite defect, which represents the less FN samples and more FP samples. Many false results are wrongly predicted to be positive. In addition, our method outperforms the SVM model combined with AlexNet and still achieves high precision in the high recall regions. Though the AP in Table 2 is generally higher in the proposed method, it is only tested on the small dataset created on our own. Therefore, the data presented in the study can just be as a reference to the feasibility of the deep neural network in the PCB defect detection area. This attempt motivates us to further the research in the future work and to solve more complicated circumstances with larger datasets.

4 Conclusion

The highlights of the proposed CNN-based model are the end-to-end configuration of the entire defect detection system. From the original images to the final detection results, the parameters of the feature extractor and the classifier are all obtained by the training process, which is achieved by convolutional layers and a dense layer, respectively. This configuration can make up for the complexity of traditional vision methods which contain several image processing steps. Furthermore, the proposed approach makes a new attempt to the deep neural network in the PCB defect detection area with the small dataset self-created. Also, its outcome shows a better performance in defect classification than other traditional methods based on handcrafted feature extraction. The proposed deep model has two special strengths: (i) the deep feature learnt is discriminative enough to distinguish the difference between defects. (ii) The system is much more complex, so it is able to be used in diverse situations.

Although the model performs well using the deep features upon the current dataset, the feature selection algorithms are not studied in the network. This algorithm can greatly help us to build a more robust model for the multi-class identification task. Also, the top layers of the model can also be improved, since it is constructed only by a simple dense layer currently. Thus, our future work will place emphasis on the feature selection algorithms and the architecture of the network.

5 Acknowledgments

We express great gratitude to Jianjie Ma, the author of paper [30], for providing some PCB defect images that we used to create our own dataset. We are also grateful for the additional assistance by Zhongqiang Cai and Wei Guo. We are grateful that this work was supported by the Specialised Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (number ZLZBCXLJZI20160729020003).

6 References

[1] Celik, H.I., Dulger, L.C., Topalbekiroglu, M.: 'Development of a machine vision system: real-time fabric defect detection and classification with neural networks', *J. Text. Inst.*, 2014, **105**, (6), pp. 575–585

[2] Feng, H., Jiang, Z.G., Xie, F.Y*., et al.*: 'Automatic fastener classification and defect detection in vision-based railway inspection systems', *IEEE Trans. Instrum. Meas.*, 2014, **63**, (4), pp. 877–888

[3] Koch, C., Georgieva, K., Kasireddy, V*., et al.*: 'A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure', *Adv. Eng. Inf.*, 2015, **29**, (2), pp. 196–210

[4] Shanmugamani, R., Sadique, M., Ramamoorthy, B.: 'Detection and classification of surface defects of gun barrels using computer vision and machine learning', *Measurement*, 2015, **60**, pp. 222–230

[5] Ibrahim, Z., Al-Attas, S.A.R.: 'Wavelet-based printed circuit board inspection system', *Int. J. Signal Process.*, 2004, **1**, (1), pp. 73–79

[6] Chang, P.C., Chen, L.Y., Fan, C.Y.: 'A case-based evolutionary model for defect classification of printed circuit board images', *J. Intell. Manuf.*, 2008, **19**, (2), pp. 203–214

[7] Khalid, N.K., Ibrahim, Z., Abidin, M.S.Z*., et al.*: 'An algorithm to group defects on printed circuit board for automated visual inspection', *Int. J. Simul. Syst. Sci. Technol.*, 2008, **9**, (2), pp. 1–10

[8] Ibrahim, I., Ibrahim, Z., Khalil, K*., et al.*: 'An algorithm for classification of five types of defect on bare printed circuit board', *Int. J. Comput. Sci. Eng. Syst.*, 2011, **5**, (3), pp. 201–208

[9] Putera, S.H.I., Ibrahim, Z.: 'Printed circuit board defect detection using mathematical morphology and MATLAB image processing tools'. Proc. Int. Conf. on Education Technology and Computer, Shanghai, China, June 2010, pp. V5-359–V5-363

[10] Putera, S.H.I., Dzafaruddin, S.F., Mohamad, M.: 'Matlab based defect detection and classification of printed circuit board'. Proc. Int. Conf. on Digital Image Computing Techniques and Applications, Bangkok, Thailand, May 2012, pp. 115–119

[11] Ray, S., Mukherjee, J.: 'A hybrid approach for detection and classification of the defects on printed circuit board', *Int. J. Comput. Appl.*, 2015, **121**, (12), pp. 42–48

[12] Li, Y.F., Li, S.Y.: 'Defect detection of bare printed circuit boards based on gradient direction information entropy and uniform local binary patterns', *Circuit World*, 2017, **43**, (4), pp. 145–151

[13] LeCun, Y., Boser, B.E., Denker, J.S*., et al.*: 'Back propagation applied to handwritten zip code recognition', *Neural Comput.*, 1989, **1**, (4), pp. 541–551

[14] Deng, J., Dong, W., Socher, R*., et al.*: 'ImageNet: a large-scale hierarchical image database'. Proc. Int. Conf. on Computer Vision and Pattern Recognition, Miami, FL, USA, June 2009, pp. 248–255

[15] Donahue, J., Jia, Y.Q., Vinyals, O*., et al.*: 'Decaf: a deep convolutional activation feature for generic visual recognition'. Proc. Int. Conf. on Machine Learning, Beijing, China, June 2014, pp. 647–655

[16] Razavian, A.S., Azizpour, H., Sullivan, J*., et al.*: 'CNN features off-the-shelf: an astounding baseline for recognition'. Proc. Int. Conf. on Computer Vision and Pattern Recognition Workshop, Columbus, OH, USA, June 2014, pp. 512–519

[17] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556, 2014

[18] Lowe, D.G.: 'Object recognition from local scale-invariant features'. Proc. Int. Conf. on Computer Vision, Kerkyra, Greece, Sept. 1999, pp. 1150–1157

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

[20] He, D.C., Wang, L.: 'Texture unit, texture spectrum, and texture analysis', *IEEE Trans. Geosci. Remote Sens.*, 1990, **28**, (4), pp. 509–512

[21] Wang, L., He, D.C.: 'Texture classification using texture spectrum', *Pattern Recognit.*, 1990, **23**, (8), pp. 905–910

[22] Csurka, G., Dance, C., Fan, L.X*., et al.*: 'Visual categorization with bags of keypoints'. Workshop on Statistical Learning in Computer Vision, ECCV, Prague, Czech Republic, May 2004, vol. 1, (1–22), pp. 1–2

[23] Sivic, J., Zisserman, A.: 'Video Google: a text retrieval approach to object matching in videos'. Proc. Int. Conf. on Computer Vision, Nice, France, October 2003, pp. 1470–1477

[24] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'ImageNet classification with deep convolutional neural networks'. Proc. Int. Conf. on Neural Information Processing Systems, Lake, Tahoe, USA, December 2012, pp. 1097–1105

[25] Russakovsky, O., Deng, J., Su, H*., et al.*: 'ImageNet large scale visual recognition challenge', *Int. J. Comput. Vis.*, 2015, **115**, (3), pp. 211–252

[26] Szegedy, C., Liu, W., Jia, Y.Q*., et al.*: 'Going deeper with convolutions'. Proc. Int. Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, June 2015, pp. 1–9

[27] Rowley, H.A., Baluja, S., Kanade, T.: 'Human face detection in visual scenes'. Proc. Int. Conf. on Neural Information Processing Systems, Denver, USA, December 1996, pp. 875–881

[28] Dalal, N., Triggs, B.: 'Histograms of oriented gradients for human detection'. Proc. Int. Conf. on Computer Vision and Pattern Recognition, San Diego, CA, USA, June 2005, pp. 886–893

[29] Ferrari, V., Fevrier, L., Jurie, F*., et al.*: 'Groups of adjacent contour segments for object detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, **30**, (1), pp. 36–51

[30] Ma, J.J.: 'Defect detection and recognition of bare PCB based on computer vision'. Chinese Control Conf., Dalian, China, July 2017, pp. 11023–11028

[31] Everingham, M., Van Gool, L., Williams, C.K*., et al.*: 'The Pascal visual object classes (VOC) challenge', *Int. J. Comput. Vis.*, 2010, **88**, (2), pp. 303–338