PeerJ Computer Science

Defect identification of bare printed circuit boards based on Bayesian fusion of multi-scale features

Xixi Han[1], Renpeng Li[2], Boqin Wang[1] and Zhibo Lin[1]

[1] School of Electronic Information, Zhongyuan University of Technology, Zhengzhou, Henan, China
[2] Anyang Iron and Steel Automation Software Co., Ltd, Zhengzhou, Henan, China

ABSTRACT

The aim of this article is to propose a defect identification method for bare printed circuit boards (PCB) based on multi-feature fusion. This article establishes a description method for various features of grayscale, texture, and deep semantics of bare PCB images. First, the multi-scale directional projection feature, the multi-scale grey scale co-occurrence matrix feature, and the multi-scale gradient directional information entropy feature of PCB were extracted to build the shallow features of defect images. Then, based on migration learning, the feature extraction network of the pre-trained Visual Geometry Group16 (VGG-16) convolutional neural network model was used to extract the deep semantic feature of the bare PCB images. A multi-feature fusion method based on principal component analysis and Bayesian theory was established. The shallow image feature was then fused with the deep semantic feature, which improved the ability of feature vectors to characterize defects. Finally, the feature vectors were input as feature sequences to support vector machines for training, which completed the classification and recognition of bare PCB defects. Experimental results show that the algorithm integrating deep features and multi-scale shallow features had a high recognition rate for bare PCB defects, with an accuracy rate of over 99%.

Subjects Artificial Intelligence, Computer Vision
Keywords Printed circuit board, Defect identification, Feature extraction, Migration learning

Submitted 19 July 2023
Accepted 31 January 2024
Published 28 February 2024

Corresponding author
Xixi Han, 6580@zut.edu.cn

Academic editor
Jiachen Yang

Additional Information and Declarations can be found on page 22

DOI **10.7717/peerj-cs.1900**

INTRODUCTION

The printed circuit board (PCB) is an important carrier of electronic components and plays an important role in modern electronic equipment. The quality of PCBs is not only related to the reliability of electronic products but can also reflect the developmental level of a regional electronic industry. With the rapid upgrading and iteration of electronic technology, the demand for various types of electronic products continues to grow, leading to rapid growth of the PCB industry.

The quality of bare PCB determines whether electronic products can operate normally. During the production process of PCBs, it is difficult to avoid factors such as environmental interference, equipment aging, and manual operation errors. Each PCB production process may cause defects. Therefore, it is particularly important to conduct quality inspections and defect identification on bare PCBs during the PCB production process. An efficient bare PCB defect detection method can prevent defective bare PCBs from entering the next phase

How to cite this article Han X, Li R, Wang B, Lin Z. 2024. Defect identification of bare printed circuit boards based on Bayesian fusion of multi-scale features. *PeerJ Comput. Sci.* **10**:e1900 http://doi.org/10.7717/peerj-cs.1900

of the production process, which improves process efficiency, reduces the scrap rate of bare PCBs, and allows designers to analyze the defects for future improvements (*Liu & Wen, 2021*; *Zhang, Jiang & Li, 2021*). With the increasing accuracy and complexity of PCB design, the task of detecting and classifying defects has become even more challenging (*Huang et al., 2020*).

Machine vision technology is currently a research hotspot, and has great potential for application in the field of PCB defect detection (*Kim et al., 2021*). Machine vision technology uses image acquisition equipment to obtain visual information about objects and perform analysis and processing (*Zhao et al., 2022*; *Sivaranjani & Senthilrani, 2023*). It can achieve automated testing, improve production efficiency, and reduce production costs (*Dong et al., 2022*). Research on bare PCB defect detection algorithms based on machine vision theory can provide new ideas for machine vision technology, promote technological progress in related fields, and help expand the application field of machine vision and defect detection technology to areas such as medical (*Coraci, Tognolo & Masiero, 2023*; *Gao et al., 2023*), security (*Mallaiyan Sathiaseelan et al., 2021*; *Ge, Dan & Li, 2020*), and smart home technology (*Yu & Pei, 2021*; *Talaat, Arafa & Metwally, 2020*). The continuous improvement of this technology can also promote the cross-integration of machine vision and other disciplines, further promoting the improvement of industrial automation (*Kshirsagar et al., 2022*).

Image processing algorithms are key technologies in machine vision detection (*Devi & Shitharth, 2021*). *Song, Kim & Park (2019)* proposed a method based on a genetic algorithm to optimize the extraction of regions. This method extracts color features from solder joint images and then uses support vector machine (SVM) to classify solder joint defects. However, this proposed method is only applicable to defect detection of capacitors and resistors, and only uses component images obtained from RGB illumination for defect classification. *Tsai & Huang (2018)* proposed a global Fourier image reconstruction method based on traditional template matching techniques for detecting and locating small defects in non-periodic PCB images. This algorithm is insensitive to translation and lighting changes and can detect subtle defects one pixel wide in various non-periodic PCB images. It can also detect defects in the manufacturing environment. However, for very large images and images with high duplicate components, the detection efficiency and computational efficiency are not high. *Li & Li (2017)* proposed a bare PCB defect detection algorithm based on image edge features. The algorithm extracts gradient direction information entropy and the edge pixel density features of edge pixels, then combines SVM classifiers to achieve defect location. However, the calculation of neighborhood gradient direction information entropy is very sensitive to changes in gradient direction, and the defect feature extraction algorithm proposed in the article does not consider the impact of actual image distortion on this calculation. *Lu et al. (2018)* proposed a non-reference comparison framework for PCB defect detection that extracts the directional gradient histogram (HOG) and local binary patterns (LBP) features of PCB images, inputs them into SVM for supervised learning, and obtains two independent classification models. According to Bayesian fusion theory, the two models are then fused for defect classification. *Lu et al. (2018)* proved the effectiveness of fused features in defect classification problems, but their proposed algorithm cannot detect specific defect positions on the PCB surface. *Jia & Liu (2022)* improved the LeNet-5 network model for PCB character defect detection by changing the layer depth to examine the impact of different network architectures on model efficiency and adding a combination classifier in the fully connected layer to enhance the feature expression performance. However, this proposed method does not reduce the need for human intervention during the training process, and it is still a significant challenge to study the segmentation of overlapping character clusters. *Shen, Liu & Sun (2020)* established the "lightweight component category detection model" for lightweight PCB detection. This method enhanced the detection accuracy and achieved online monitoring of PCBs. Combined with a character recognition model, called component character, these two models were used for PCB component defect detection and recognition.

There are various types of defects in bare PCB boards, and most current algorithms only focus on a single feature in feature extraction, which makes it difficult to accurately describe the defects. Moreover, the dimension of feature extraction is usually too high. Defect detection methods based on deep learning rely on a large dataset of defects, and cannot guarantee the accuracy of detection for certain defects with smaller datasets. The key research focuses of the present study are feature extraction, feature reduction, and multi-feature fusion methods. This article proposes a bare PCB defect detection algorithm that combines multi-scale shallow features with deep features extracted by neural networks. Compared to single-scale features, multi-scale features have richer texture features and can better identify the detail information of an image. Convolutional neural network (CNN) can automatically learn features from datasets, allowing it to obtain new effective feature representations from new training data (*Li, Kuo & Guo, 2020*).

Building and training a deep learning network model requires a significant amount of time, expensive hardware, and a large number of labeled images. In the absence of a large number of datasets, transfer learning is a better solution (*Hu, 2023*). This article uses the VGGl6-Net network model trained on the ImageNet (*Fang et al., 2021*) dataset as the pre-trained model (*Althubiti et al., 2022*). The model was then fine-tuned, including adjusting network structure and parameters, to better fit the target dataset.

The first stage of this model is the extraction and fusion of defect features. To reduce the influence of background interference pixels on feature extraction, defect areas were segmented through image differentiation and morphological operations. Next, the multi-scale directional projection feature, the multi-scale gray-scale co-occurrence matrix feature, and the multi-scale gradient directional information entropy feature of the bare PCB defect image were extracted. Deep features were then extracted from defect images using a pre-trained VGG-16 convolutional neural network. Finally, using naive Bayesian theory, the feature information extracted by multiple algorithms was weighted and fused to construct a new feature vector.

The second stage of this model is defect classification and identification. The fused new feature vectors were used as training samples to train the support vector machine model and then complete the classification and recognition of bare PCB defects.

The rest of this the article is organized as follows: 'Related work' analyzes related work, including image preprocessing and feature extraction; 'Method' provides specific

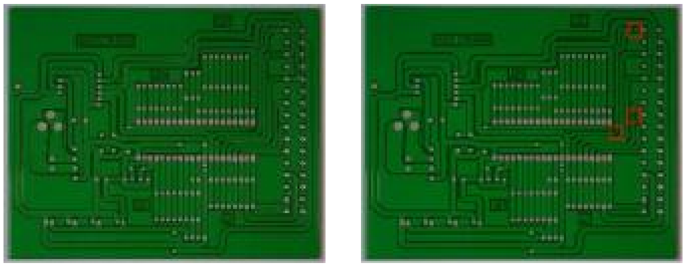

Figure 1 **Image registration results.**

Full-size DOI: 10.7717/peerjcs.1900/fig-1

methods for feature extraction of defects and feature fusion; 'Experimental result and discussion' evaluates the performance of the proposed method and presents the defect detection accuracy and classification results using different algorithms; and 'Conclusion' summarizes the findings of this article and proposes future areas of research.

RELATED WORK

mage preprocessing

The proposed model uses image enhancement and Gaussian filtering algorithms to preprocess the original image of the bare PCB to reduce the impact of lighting and noise on the image. Next, the PCB template images and PCB test images are registered using the Förstner corner detection algorithm (*Naji et al., 2023*). The pose correction and image registration of the image to be measured are completed through projection transformation. As shown in Fig. 1, the rectangular box represents the defect area obtained after calculating the difference between the template image and the test image.

The defect area of the image to be tested is then cropped on the bare PCB board. To reduce the impact of interfering pixels on feature extraction, the pixel value of the image area outside the defect area is set to 0. Some examples of defect images are shown in Fig. 2.

Feature extraction

Image recognition is actually a classification process. To identify the category to which an image belongs, the image must be distinguished from images of different categories. Feature extraction requires the selected features of an image to both describe the image well and distinguish it from different types of images. In order to increase the gap between feature information classes and improve the classification performance of feature vectors for bare PCB defect images, this article proposes a defect extraction algorithm based on multi-scale feature fusion.

Figure 2 shows that the cropped bare PCB defect image makes it difficult to extract rich feature information at a single resolution. Therefore, the proposed model performs down-sampling on the defect images. Based on the Gaussian pyramid, a multi-scale defect image is constructed. The image resolution is gradually reduced with the down-sampling, and the information is gradually enriched. Four down-sampling operations are performed

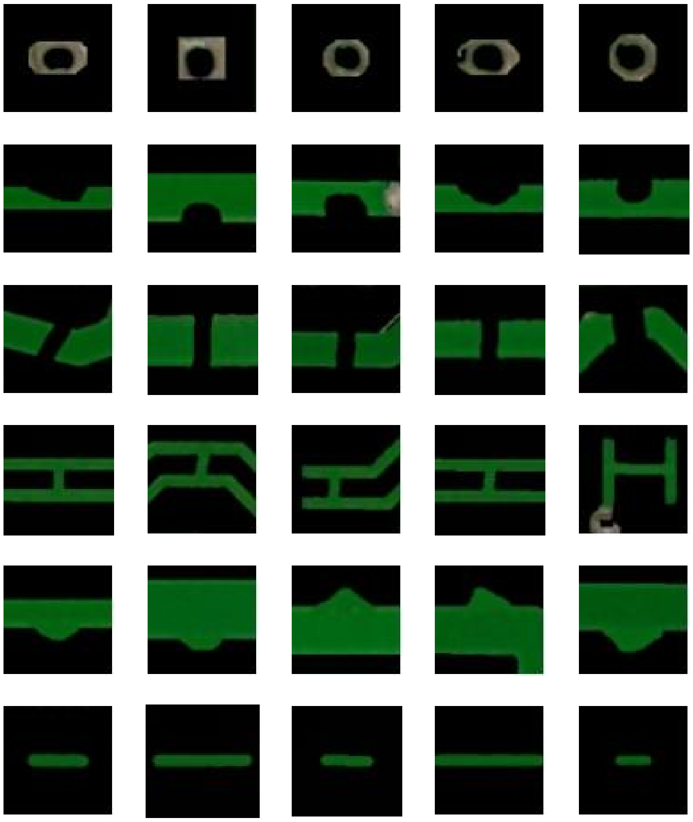

Figure 2 **Examples of defect images.**

Full-size DOI: 10.7717/peerjcs.1900/fig-2

on each image, 1–4 layers of pyramid images are constructed, and the features from each scale image are extracted, separately.

The technology roadmap of this model is shown in Fig. 3.

METHOD

Multi-scale grey scale co-occurrence matrix feature extraction

The grey scale co-occurrence matrix features are extracted in the 0° direction of each layer pyramid image. This model uses the following statistics as features of the grey scale co-occurrence matrix:

Width represents the width of the co-occurrence matrix; c_{ij} represents the term in the co-occurrence matrix.

Angular second moment (Asm) refers to the sum of squares of the values of the elements in the grayscale co-occurrence matrix, which reflects the uniformity of the image grayscale distribution and the texture thickness:

$$\mathrm{Asm} = \sum_{i,j=0}^{\mathrm{width}} c_{ij}^2 \tag{1}$$

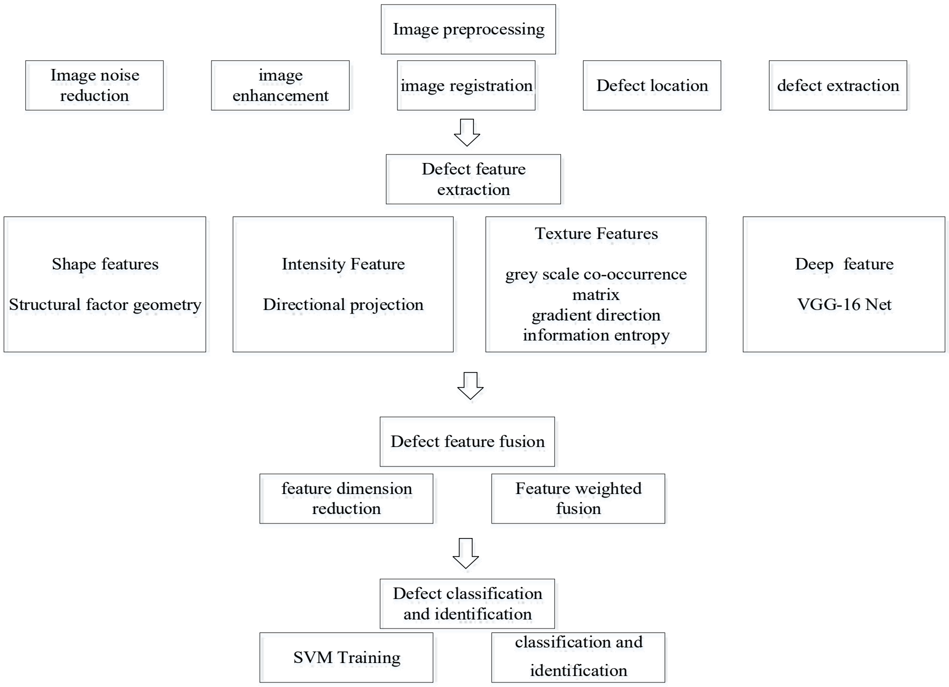

Figure 3 Technology roadmap.

Full-size DOI: 10.7717/peerjcs.1900/fig-3

Correlation (Cor) measures the similarity of elements in the spatial gray level co-occurrence matrix in the row or column direction:

$$\mathrm{Cor}=\frac{\sum_{i,j=0}^{\mathrm{width}}(i-u_x)(j-u_y)c_{ij}}{s_x s_y} \tag{2}$$

where,

$$u_x=\sum_{i,j=0}^{\mathrm{width}} i\cdot c_{ij} \tag{3}$$

$$u_y=\sum_{i,j=0}^{\mathrm{width}} j\cdot c_{ij} \tag{4}$$

$$s_x^2=\sum_{i,j=0}^{\mathrm{width}}(i-u_x)^2 c_{ij} \tag{5}$$

$$s_y^2=\sum_{i,j=0}^{\mathrm{width}}\left(i-u_y\right)^2 c_{ij} \tag{6}$$

Homogeneity (Hom) measures the number of local changes in image texture:

$$\mathrm{Hom} = \sum_{i,j=0}^{\mathrm{width}} \frac{1}{1+(i-j)^2} c_{ij} \tag{7}$$

Contrast (Con) reflects the clarity of the image and the depth of the texture grooves:

$$\mathrm{Con} = \sum_{i,j=0}^{\mathrm{width}} (i-j)^2 c_{ij} \tag{8}$$

The grey scale co-occurrence matrix features of the 1–4 layers of pyramid images are calculated separately. The multi-scale grey scale co-occurrence matrix feature vectors are obtained by concatenating the feature vectors.

Multi-scale directional projection feature extraction

Directional projection refers to the horizontal and vertical projection of regional grayscale values. The calculation formula is as follows:

$$P_{\mathrm{Vert}}(c) = \frac{1}{n(c+c')} \sum_{(r+r',c+c')\in R} I\left(r+r',c+c'\right) \tag{9}$$

$$P_{\mathrm{Hor}}(r) = \frac{1}{n(r+r')} \sum_{(r+r',c+c')\in R} I\left(r+r',c+c'\right) \tag{10}$$

where R refers to region and represents the target area; I refers to image, which represents the image where the region is located; $\left(r',c'\right)$ represents the top left corner of the bounding rectangle parallel to the smallest axis in the input area; $n(x)$ represents the number of region points in the corresponding row $(r+r')$ or column $(c+c')$. Horizontal projection returns a one-dimensional function that reflects changes in vertical grayscale values. Similarly, vertical projection returns a function that reflects changes in horizontal grayscale values. Horizontal and vertical projection functions for 1–4 layers of pyramid images are then drawn separately, using open circuit defects as an example, as shown in Fig. 4.

The directional projection function cannot effectively characterize the shape features of defects. The directional projection function is also directly input into the classifier as a feature vector, which may cause data redundancy. The known one-dimensional curve can be regarded as a one-dimensional signal in time-domain space. Therefore, this model extracts the time-domain features of the multi-scale directional projection functions of each defect image as feature vectors. The time-domain feature representation method is as follows:

Root mean square:

$$X_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} g_i^2} \tag{11}$$

Root amplitude:

$$X_r = \left[\frac{1}{N}\sum_{i=1}^{N}\sqrt{|g_i|}\right]^2 \tag{12}$$

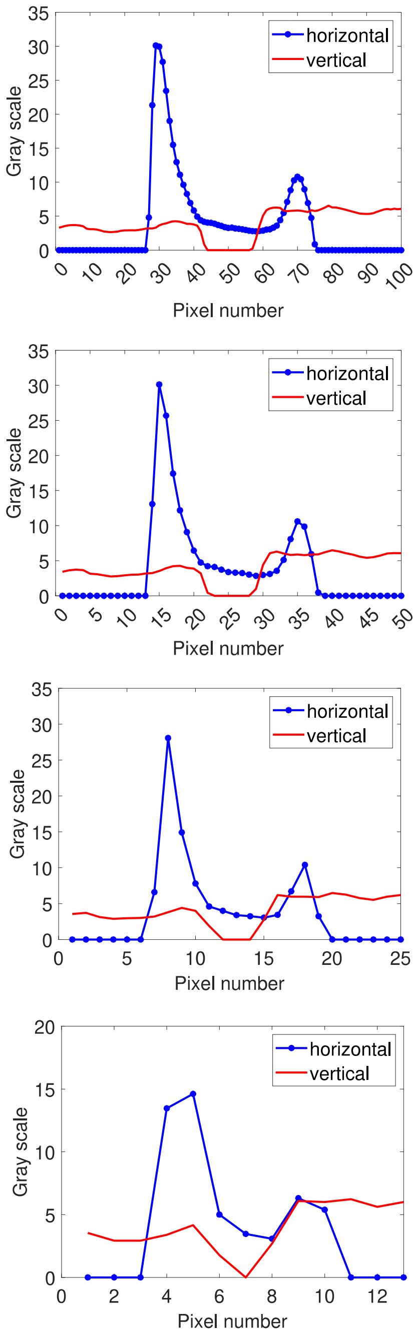

Figure 4 **Multi-scale directional projection function for open circuit images.**
Full-size DOI: 10.7717/peerjcs.1900/fig-4

Skewness:

$$X_{\mathrm{sk}} = \frac{1}{N}\sum_{i=1}^{N} g_i^3 \tag{13}$$

Average amplitude:

$$X_{\text{mean}} = \frac{1}{N}\sum_{i=1}^{N}\left|g_i\right| \tag{14}$$

Peak value:

$$X_{\text{peak}} = \max\left|g_i\right| \tag{15}$$

Kurtosis:

$$K_u = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(g_i - \overline{g}\right)^4}{X_{\text{RMS}}^4} \tag{16}$$

Waveform factor:

$$S_F = \frac{X_{\text{RMS}}}{X_{\text{mean}}} \tag{17}$$

Pulse factor:

$$I_F = \frac{X_{\text{peak}}}{X_{\text{mean}}} \tag{18}$$

Margin:

$$L = \frac{X_{\text{peak}}}{X_r} \tag{19}$$

Crest factor:

$$I_{\text{CF}} = \frac{X_{\text{peak}}}{X_{\text{RMS}}} \tag{20}$$

where N represents the number of pixel sequences, g_i represents the directional projection function value corresponding to the ith pixel sequence. The time-domain characteristics of the directional projection function of the 1–4 layers of pyramid images are then calculated and the time-domain features are concatenated to obtain the multi-scale directional projection feature vectors of the image.

Multi-scale gradient direction information entropy feature extraction

Gray information entropy

Entropy refers to the degree of chaos in a system, which is used to measure the uncertainty of information. The smaller the entropy, the more certain the information, so entropy also reflects the amount of information. Assuming there is a discrete random variable X with a probability distribution function of $p(x)$, its entropy $H(X)$ can be represented by the following equation:

$$H(X) = -\sum_{x\in\chi} p(x)\log p(x) \tag{21}$$

Image entropy reflects the average amount of information in an image, and describes the number and frequency of occurrences of grayscale values based on the analysis and statistics of the obtained image grayscale values. The one-dimensional grayscale entropy of a grayscale image can be represented by the following equation:

$$H = -\sum_{i=0}^{255} P_i \log P_i \tag{22}$$

Based on one-dimensional entropy, this model uses a feature quantity that can reflect the spatial characteristics of the gray distribution to form the two-dimensional entropy of an image:

$$P_{ij} = f(i,j)/N^2 \tag{23}$$

where $f(i,j)$ is the frequency of the occurrence of feature binary (i,j) and N isthe size of the image. The definition Eq. (24) represents the two-dimensional entropy of discrete images:

$$H = \sum_{i=0}^{255} P_{ij} \log P_{ij} \tag{24}$$

This equation can reflect the grayscale value at a certain pixel position and the comprehensive characteristics of the gray distribution of its surrounding pixels.

Image gradient

The image gradient represents the axis and partial derivative of the current pixel point to the axis, and can also be understood as the change speed of the pixel gray value. In mathematics, the Sobel mask operator is used to calculate the approximate gradient of changes in the horizontal and vertical directions of an image. The following common operators can be used, with A defined as the source image, and G_x and G_y defined as the approximate horizontal and vertical gradients of an image, respectively. The calculation method is as follows:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A, \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \tag{25}$$

where $*$ represents the convolution operation between the operator and the image.

The gradient direction feature is reflected in the image in the form of grayscale values. First, the original image is converted to a grayscale image, and then the pixels under the mask are calculated according to the formula. The obtained results replace the values of pixels at the center of the mask, and after calculating all pixels, the original image gradient direction feature map is obtained. One image was selected for each type of defect for experimentation, and the obtained gradient direction feature map is shown in Fig. 5.

This model calculates the grayscale information entropy of the gradient direction feature map of bare PCB defects based on the two-dimensional grayscale entropy calculation method of the image. Then, based on the multi-scale image construction method studied in the previous section, the gradient direction information entropy is calculated for the 1–4 layers of pyramid images. The obtained results are then serialized as multi-scale gradient direction information entropy features for bare PCB defects.

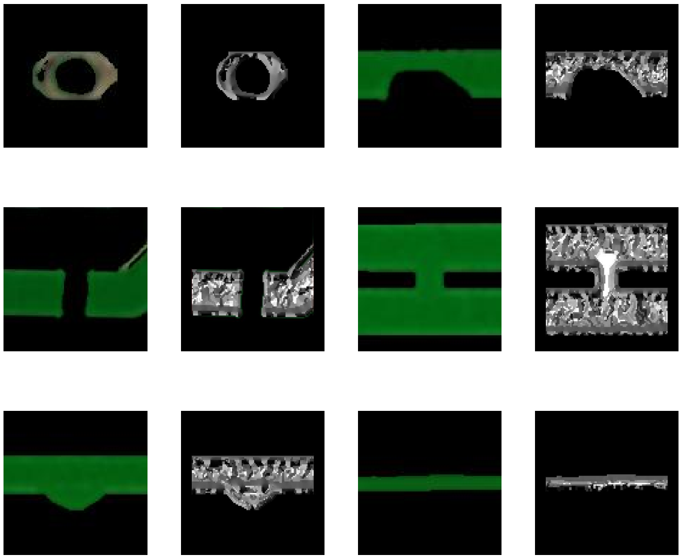

Figure 5 **Gradient direction feature map.**

Full-size DOI: 10.7717/peerjcs.1900/fig-5

Deep semantic feature extraction based on transfer learning

To further explore the deep semantic features of bare PCB defect images, the model proposed in this article uses a convolutional neural network to extract features from the datasets. The VGG-16 network, proposed by *Simonyan & Zisserman (2014)* was selected as the pre-training model. The VGG-16 model consists of a convolutional layer, a pooling layer, and a fully connected layer. The network model structure is shown in Table 1.

Each convolutional layer applies a series of convolutional checks to the input data through convolutional operations for feature extraction, generating a feature mapping set. The maximum pooling layer aggregates input data through sliding windows to reduce feature mapping. Pooling operations can also improve the ability of features to describe images. The fully connected layer is used to generate feature vectors. After each layer, there is a nonlinear layer, ReLU, represented as $f(x) = \max(0,x)$, used to achieve rapid convergence of network training.

In the process of transfer learning, there are two models for fine-tuning. The first method is to freeze all convolutional feature extraction layers of the pre-trained model and only perform fine-tuning training operations on the classification layer; the second method is to fine tune all convolutional feature extraction and classification layers of the pre-trained model (*Lamsiyah et al., 2023*).

This model uses the first fine-tuning method. The known convolutional layer feature maps are output in the form of matrix sets. If directly converted into a one-dimensional vector, the dimensionality of the vector is very large, leading to overfitting problems and affecting classification efficiency (*Bai & Tang, 2018*). In order to reduce the risk of overfitting, the number of parameters and computational complexity are reduced. This

Table 1 **VGG-l6 network structure.**

Number of layers	Model	Kernel configuration		
		Size	Quantity	Step size
1	Convolution	3 × 3	64	1
2	Convolution	3 × 3	64	1
3	Max-pooling	2 × 2	–	2
4	Convolution	3 × 3	128	1
5	Convolution	3 × 3	128	1
6	Max-pooling	2 × 2	–	2
7	Convolution	3 × 3	256	1
8	Convolution	3 × 3	256	1
9	Convolution	3 × 3	256	1
10	Max-pooling	2 × 2	–	2
11	Convolution	3 × 3	512	1
12	Convolution	3 × 3	512	1
13	Convolution	3 × 3	512	1
14	Max-pooling	2 × 2	–	2
15	Convolution	3 × 3	512	1
16	Convolution	3 × 3	512	1
17	Convolution	3 × 3	512	1
18	Max-pooling	2 × 2	–	2
19	Full-connection	4096	–	–
20	Full-connection	4096	–	–
21	Full-connection	1000	–	–
22	SoftMax	–	–	–

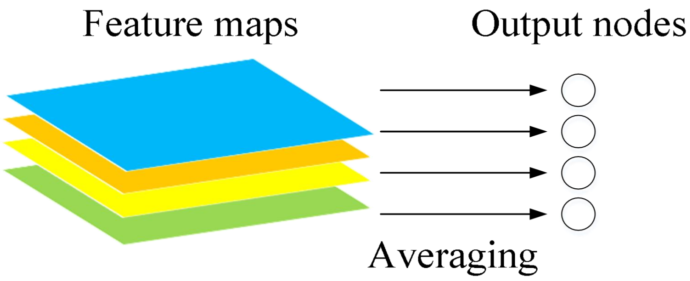

Figure 6 **Schematic diagram of global pooling layer.**
Full-size DOI: 10.7717/peerjcs.1900/fig-6

model removes the last five layers of the VGGl6-Net network model and replaces them with a global average pooling layer and a fully connected layer with 512 nodes.

The function of global average pooling is to add all the pixel values of the feature map and find a balance to obtain a numerical value. By using this numerical value to represent

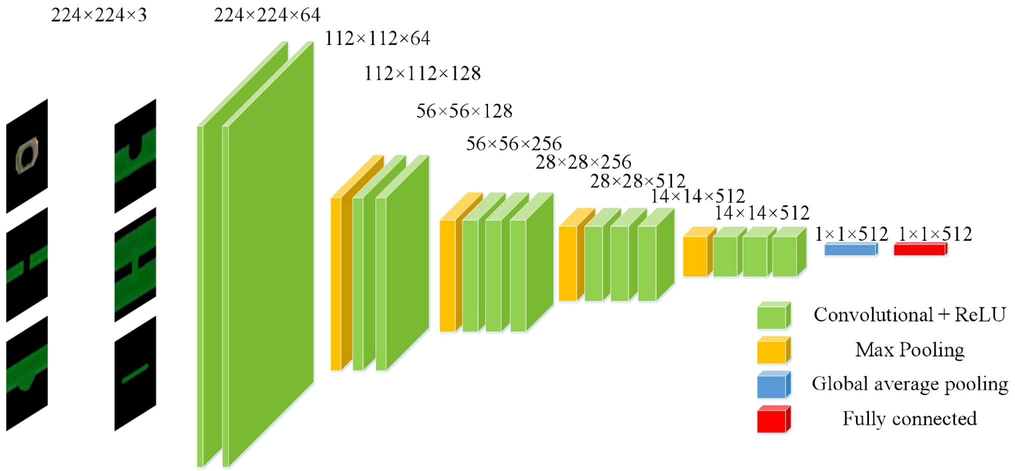

Figure 7 The VGG-l6 model after fine tuning.

Full-size DOI: 10.7717/peerjcs.1900/fig-7

the corresponding feature map, the spatial parameters are reduced, and the robustness of the network is improved (*Hsiao et al., 2019*). The schematic diagram is shown in Fig. 6.

The feature maps output from the global average pooling layer are used as inputs to the fully connected layer, and the 512-dimensional feature vectors outputted from the fully connected layer are used as the deep features of the bare PCB defect images. The structure of the adjusted VGGl6 net model is shown in Fig. 7.

Feature fusion

The different semantic features extracted from the same bare PCB defect image reflect different image information. This model integrates multi-scale grayscale co-occurrence matrix features, multi-scale directional projection features, multi-scale gradient directional information entropy features, and deep semantic features extracted by the VGG-l6 neural network to increase the diversity of feature information. If each group of feature vectors were simply added or subtracted, it would cause problems, such as high dimensionality of feature vectors and high computational complexity, so it is necessary to reduce the dimensionality of the feature vectors. It is also necessary to assign different weights to different feature vectors because different feature vectors have different scale ranges and classification performance, and direct addition and subtraction would ignore the imbalance of the data.

This model uses principal components analysis (PCA) to reduce the dimensionality of the extracted feature vectors (*Jiang & Sun, 2023*), and the high contribution features are extracted as effective features for bare PCB defect images.

Assuming there are M samples $\{X_1, X_{2,}, \ldots, X_M\}$ andeach sample has an N-dimensional feature vector $X_i = \left(x_1^i, x_2^i, \ldots, x_N^i\right)^T$, then each feature x_j has its own characteristic value.

The steps for the PCA are as follows:

Step 1: All features are centralized using demeaning.

Each feature is averaged and then its own mean is subtracted.

$$\overline{x_n} = \frac{1}{M}\sum_{i=1}^{M} x_n^i \tag{26}$$

Step 2: The Covariance matrix C, is calculated taking features A and S as examples.

$$C = \begin{bmatrix} \mathrm{cov}(x_1, x_1) & \mathrm{cov}(x_1, x_2) \\ \mathrm{cov}(x_2, x_1) & \mathrm{cov}(x_2, x_2) \end{bmatrix}. \tag{27}$$

In the above matrix, the variances of features x_1 and x_2 are on the diagonal, and the covariance is found on the non-diagonal. A covariance greater than 0 indicates that if one of x_1 and x_2 x_2 increases, the other also increases; a covariance less than 0 means when one increases, the other decreases; when the covariance is 0, the two are independent. The larger the absolute value of covariance, the greater the impact the two have on each other. The solution formula for $\mathrm{cov}(x_1, x_1)$ is shown in Eq. (28), and the same applies to the others.

$$\mathrm{cov}(x_1, x_1) = \frac{\sum_{i=1}^{M}\left(x_1^i - \overline{x_1}\right)\left(x_1^i - \overline{x_1}\right)}{M-1} \tag{28}$$

The covariance matrix C of M samples under N-dimensional features are obtained in this covariance calculation formula.

Step 3: The eigenvalues of the covariance matrix are and the corresponding eigenvectors are identified.

The eigenvalue λ of the Covariance matrix C and the corresponding eigenvector μ(each eigenvalue corresponds to a eigenvector) is calculated with the following formula:

$$C\mu = \lambda\mu. \tag{29}$$

There are N eigenvalues λ, and each λ_i corresponds to a feature vector μ_i. The feature values are sorted in descending order, with the largest k selected, producing a set of $\{(\lambda_1, \mu_1), (\lambda_2, \mu_2), \ldots, (\lambda_k, \mu_k)\}$.

Step 4: The original features are projected onto the selected feature vector to obtain the new K-dimensional features after dimensionality reduction.

Dimensionality reduction is performed by selecting and projecting the largest eigenvalues and corresponding eigenvectors. For each sample X^i, the original feature is $\left(x_1^i, x_2^i, \ldots, x_n^i\right)^T$, and the projected new feature is $\left(y_1^i, y_2^i, \ldots, y_k^i\right)^T$. The new feature is calculated using Eq. (30):

$$\begin{bmatrix} y_1^i \\ y_2^i \\ \cdot \\ \cdot \\ \cdot \\ y_k^i \end{bmatrix} = \begin{bmatrix} \mu_1^T\left(x_1^i, x_2^i, \ldots, x_n^i\right)^T \\ \mu_2^T\left(x_1^i, x_2^i, \ldots, x_n^i\right)^T \\ \cdot \\ \cdot \\ \cdot \\ \mu_k^T\left(x_1^i, x_2^i, \ldots, x_n^i\right)^T \end{bmatrix} \tag{30}$$

so that each sample X^i is changed from the original $X^i = \left(x_1^i, x_2^i\right)^T$ to the current $X^i = y_1^i$.

This model selects appropriate features by determining the amount of information contained in the feature values. According to Eq. (30), the new feature vectors calculated by principal component analysis are arranged in descending order. The size of the eigenvalues corresponds to the amount of information contained in the eigenvalues, which also indicates the contribution rate of the principal components. This model selects the feature values with a cumulative contribution rate of over 99.8% and ranking first as the fusion feature values. After the principal component calculation, the first six principal components of the multi-scale gray level co-occurrence matrix feature vector, the first seven principal components of the multi-scale directional projection feature vector, the first two principal components of the multi-scale gradient directional information entropy feature vector, and the first 68 principal components of the deep feature vector were used as the dimensionality reduced feature vectors for feature fusion.

The naive Bayesian theory is then used to calculate the weights of each group of feature vectors (*Shitharth et al., 2022*). The posterior probability of each group of eigenvectors is calculated through the dataset, and then their respective weight values are calculated. The dataset is first divided and then the conditional probability and prior probability of each category are calculated according to the training set. The posterior probability of the eigenvector is then calculated using the Bayesian formula. The maximum posterior probability is used as the final posterior probability of the eigenvector (*Krishna & Divya, 2022*).

The specific steps to calculate the posterior probability of eigenvectors and assign weights to each group of eigenvectors are as follows:

Step 1: Assuming there is a feature vector $x = \{x_1, x_2, \ldots, x_n\}$ and a category set $C = c_1, c_2, \ldots, c_k$, then the posterior probability calculation formula of the Naive Bayes classifier is as follows:

$$P(c_i|x) = \frac{P(c_i)P(x|c_i)}{P(x)} \tag{31}$$

where $P(c_i)$ represents the prior probability of the category in the training set, $P(x|c_i)$ represents the conditional probability of eigenvector x under category c_i, and $P(x)$ represents the marginal probability of feature vector x. According to the naive Bayesian theory, denominator a is a constant in classification and can be omitted, so the posterior probability can be simplified as $P(c_i|x) \propto P(c_i)P(x|c_i)$.

Step 2: The prior probability $P(c_i)$ and conditional probability $P(x|c_i)$ are calculated. The prior probability $P(c_i)$ represents the frequency of occurrence of category c_i in the training set, and the calculation method is as follows:

$$P(c_i) = \frac{|D_{c_i}|}{|D|} \tag{32}$$

where $|D_{c_i}|$ represents the number of samples belonging to category c_i in the training set and $|D|$ represents the total number of samples in the training set. The calculation of conditional probability $P(x|c_i)$ is based on the naive Bayesian classifier assumption, also called the feature conditional independence assumption. According to this assumption, conditional probability can be defined as the product of the conditional probability of each feature, as shown in the following formula:

$$P(x|c_i) = \prod_{j=1}^{n} P(x_j|c_i) \tag{33}$$

where x_j represents the jth feature of feature vector x and $P(x_j|c_i)$ represents the conditional probability of the occurrence of feature x_j under category c_i. For discrete features, $P(x_j|c_i)$ can be directly defined as the frequency of feature x_j appearing under category c_i, and the calculation formula is as follows:

$$P(x_j|c_i) = \frac{|D_{c_i,x_j}|}{|D_{c_i}|} \tag{34}$$

where $|D_{c_i,x_j}|$ represents the number of samples in the training set that belong to category c_i and feature x_j appears, and $|D_{c_i}|$ represents the number of samples belonging to category c_i in the training set. For continuous features, assuming that they conform to the Gaussian distribution, the conditional probability $P(x_j|c_i)$ can be calculated using the probability density function of the Gaussian distribution, as shown in the following formula:

$$P(x_j|c_i) = \frac{1}{\sqrt{2\pi\sigma_{c_i,j}^2}} \exp\left(-\frac{(x_j - \mu_{c_i,j})^2}{2\sigma_{c_i,j}^2}\right) \tag{35}$$

where $\mu_{c_i,j}$ and $\sigma_{c_i,j}$ represent the mean and standard deviation of feature x_j under category c_i, respectively, as calculated from the training set.

Step 3: The posterior probability of the feature vector is calculated according to the naive Bayesian classifier, as follows:

$$P(c_i|x) \propto P(c_i) \prod_{j=1}^{n} P(x_j|c_i) \tag{36}$$

where $P(c_i)$ represents the prior probability of category c_i in the training set and $P(x_j|c_i)$ represents the conditional probability of the occurrence of feature x_j under category c_i. Then the maximum a posteriori estimation is the final posterior probability of the eigenvector, and is calculated as follows:

$$P_{\max}(c_i|x) = \mathrm{argmax}_{c_i \in C} P(c_i) \prod_{j=1}^{n} P(x_j|c_i) \tag{37}$$

Step 4: The weight coefficients w_i of each group of feature vectors, defined as the normalization result of the maximum a posteriori estimation, are calculated as follows:

$$w_i = \frac{P_{\max}(c_i|x)}{\sum_{i=1}^{n} P_{\max}(c_i|x)} \tag{38}$$

Due to the presence of four sets of feature vectors in this model, $n = 4$.

After feature dimensionality reduction and weight allocation, the feature vectors need to be connected together to complete feature fusion. This model uses parallel fusion as the feature vector stitching method. Assuming there is a bare PCB defect image sample I, after

Table 2 The pseudocode of the proposed algorithm.

Algorithm Multi-feature fusion based on principal component analysis and Bayesian theory

```
1:Intput: Image $I$ of PCB bare board defects to be detected
2:Construct 4-layer Gaussian pyramid image $\left\{I_1, I_{2,}, I_3, I_4\right\}$ of defect image I of PCB bare board to be detected
3:Calculate the multi-scale gray level co-occurrence matrix feature vector $\mathrm{GLCM} = [\mathrm{GLCM}_1, \mathrm{GLCM}_2, \mathrm{GLCM}_3, \mathrm{GLCM}_4]$ of $I$ via Eqs. (1)–(8)
4:Calculate the multi-scale directional projection feature vectors $\mathrm{pro} = \left[X_{\mathrm{RMS}}, X_r, X_{\mathrm{sk}}, X_{\mathrm{mean}}, X_{\mathrm{peak}}, K_u, S_F, I_F, L, I_{\mathrm{CF}}\right]$ for $I$ via Eqs. (9)–(20)
5:Calculate the multi-scale gradient direction information entropy feature vector $\mathrm{Ent} = [H_1, H_2, H_3, H_4]$ of $I$ for $I$ via Eqs. (21)–(25)
6:Calculating the deep semantic feature vector $D = [d_1, d_2, d_3, \ldots, d_n], n = 512$ of $I$ based on the improved VGG-16 neural network
7:Extract features from image $I$ to obtain M feature vectors $\left\{X_1, X_{2,}, \ldots, X_M\right\}$
8:Reduce the dimension of $\left\{X_1, X_{2,}, \ldots, X_M\right\}$ via Eqs. (26)–(30), obtain N feature vectors $\left\{X_1, X_{2,}, \ldots, X_n\right\}$
9:Calculate the weight $\left\{W_1, W_{2,}, \ldots, W_n\right\}$ of each feature vector of $\left\{X_1, X_{2,}, \ldots, X_N\right\}$ via Eqs. (31)–(38)
10:Calculate the feature vector $V$ after parallel weighted fusion via Eq. (39)
11:Output: the feature vector $V = \{C_1, C_2, \ldots, C_n\}$ after parallel weighted fusion
```

dimensionality reduction and weighting, $A = \{A_1, A_2, \ldots, A_n\}$ and $B = \{B_1, B_2, \ldots, B_n\}$ are obtained, and the feature vector after parallel fusion is $V = \{C_1, C_2, \ldots, C_n\}$, where C_i is defined as the following equation:

$$C_i = \sqrt{(A_i)^2 + (B_i)^2} \tag{39}$$

If the number in digits of feature vectors A and B is not equal, then 0 is added to the low dimensional feature vectors.

Before dimensionality reduction, the parallel fusion dimension of feature vectors had a total of 512 dimensions. After dimensionality reduction, the parallel fusion dimension of feature vectors was reduced to 68 dimensions. The feature dimensionality was reduced significantly, and the noise was eliminated. Different weight coefficients were assigned to each group of feature vectors based on different classification performance; the important features were strengthened and the non important features were weakened.

The pseudocode of the proposed algorithm is shown in Table 2.

EXPERIMENT RESULTS AND DISCUSSION

The Open Laboratory of Intelligent Robotics at Peking University has a bare PCB dataset with various defects (leaks, mouse bites, open circuits, short circuits, burrs, and residual copper) (*Huang et al., 2020*), which can be used for image detection, classification, and registration tasks. Each type of defect is distributed on 10 different bare PCB images, with a total of 690 defect images in the database. Each image has 2–6 defective areas. After defect extraction, the sample size statistics are shown in Table 3.

Table 3 **Statistics of defect sample quantity.**

Defect type	Number of images/piece	Number of defect samples/piece
Leaks	115	442
Mouse bites	115	442
Open circuits	116	419
Short circuits	116	499
Burrs	114	424
Residual copper	114	434
Total	690	2660

This experiment used the following hardware: an Intel(R) Core(TM) i5-8300H CPU @ 2.30 GHz 2.30 GHz with a GTX 1050Ti graphics card. Bare PCB defect detection requires machine vision algorithms that can be developed quickly and efficiently, as well as a stable operating environment. This article proposes a newly-designed and developed bare PCB defect detection algorithm based on the Halcon platform. Halcon is a commercial software with powerful machine vision capabilities and high scalability, which supports multiple programming languages and development environments.

The experiments were conducted on the Halcon and Matlab platforms. First, based on the feature extraction method outlined previously, the multi-scale gray level co-occurrence matrix feature vector $\mathrm{GLCM} = [\mathrm{GLCM}_1, \mathrm{GLCM}_2, \mathrm{GLCM}_3, \mathrm{GLCM}_4]$ of 1~4 layers of Gaussian pyramid image were extracted for each bare PCB defect image. The feature vector of gray level co-occurrence matrix of the first layer Gaussian pyramid image was $\mathrm{GLCM}_1 = [\mathrm{Asm}_1, \mathrm{Cor}_1, \mathrm{Hom}_1, \mathrm{Con}_1]$, the feature vector of gray level co-occurrence matrix of the second layer Gaussian pyramid image was $\mathrm{GLCM}_2 = [\mathrm{Asm}_2, \mathrm{Cor}_2, \mathrm{Hom}_2, \mathrm{Con}_2]$, the feature vector of gray level co-occurrence matrix of the third layer Gaussian pyramid image was $\mathrm{GLCM}_3 = [\mathrm{Asm}_3, \mathrm{Cor}_3, \mathrm{Hom}_3, \mathrm{Con}_3]$, and the feature vector of gray level co-occurrence matrix of the fourth layer Gaussian pyramid image was $\mathrm{GLCM}_4 = [\mathrm{Asm}_4, \mathrm{Cor}_4, \mathrm{Hom}_4, \mathrm{Con}_4]$. The multi-scale gradient direction information entropy eigenvector of 1~4 layers of Gaussian pyramid image was $\mathrm{Ent} = [H_1, H_2, H_3, H_4]$, and the multi-scale directional projection feature vector was $\mathrm{pro} = \left[X_{\mathrm{RMS}}, X_r, X_{\mathrm{sk}}, X_{\mathrm{mean}}, X_{\mathrm{peak}}, K_u, S_F, I_F, L, I_{\mathrm{CF}}\right]$. Fused in parallel with the deep semantic feature vector $D = [d_1, d_2, d_3, \ldots, d_n], n = 512$ extracted from the convolutional neural network to form a new feature vector. Then, the new feature vector was used as input to SVM, and the SVM classifier model was trained to classify bare PCB defect images.

The dataset was divided as follows: 70% of the dataset was used as the training set, 10% as the validation set, and 20% as the testing set. To verify the superiority of the multi-feature fusion algorithm in image description compared to a single feature, the extracted multi-scale gray level co-occurrence matrix feature vector, multi-scale gradient direction information entropy feature vector, multi-scale direction projection feature vector, and deep feature vector were respectively input into the SVM classifier for classification and recognition.

The confusion matrix obtained based on deep feature recognition is shown in Table 4.

Table 4 **Deep feature confusion matrix.**

	Leaks	Mouse bites	Open circuits	Short circuits	Burrs	Residual copper	FP
Leaks	88	0	1	2	0	0	3
Mouse bites	0	78	15	0	0	0	15
Open circuits	0	0	68	0	0	0	0
Short circuits	0	2	0	95	0	0	2
Burrs	0	8	0	1	80	0	9
Residual copper	0	0	0	0	4	87	4
FN	0	10	16	3	4	0	33

Table 5 **Multi-scale gradient direction information entropy feature confusion matrix.**

	Leaks	Mouse bites	Open circuits	Short circuits	Burrs	Residual copper	FP
Leaks	88	0	1	8	3	1	13
Mouse bites	0	75	9	12	1	0	22
Open circuits	0	11	72	7	3	1	22
Short circuits	0	1	2	68	4	0	7
Burrs	0	1	0	3	73	0	4
Residual copper	0	0	0	0	0	85	0
FN	0	13	12	30	11	2	68

Table 6 **Multi-scale gray co-occurrence matrix feature confusion matrix.**

	Leaks	Mouse bites	Open circuits	Short circuits	Burrs	Residual copper	FP
Leaks	66	0	0	0	0	0	0
Mouse bites	0	82	1	0	0	0	1
Open circuits	0	0	63	0	0	0	0
Short circuits	22	5	20	98	48	6	101
Burrs	0	1	0	0	36	0	1
Residual copper	0	0	0	0	0	81	0
FN	22	6	21	0	48	6	103

The confusion matrix obtained based on multi-scale gradient direction information entropy feature is shown in Table 5.

The confusion matrix obtained based on multi-scale gray co-occurrence matrix feature is shown in Table 6.

The confusion matrix obtained based on multi-scale directional projection feature is shown in Table 7.

The analysis results of different confusion matrices show that fused features can better describe image detail features compared to single features, with a significant improvement in accuracy.

Table 7 Multi-scale directional projection feature confusion matrix.

	Leaks	Mouse bites	Open circuits	Short circuits	Burrs	Residual copper	FP
Leaks	80	6	0	1	0	0	7
Mouse bites	0	81	11	1	0	0	12
Open circuits	5	0	68	10	0	0	15
Short circuits	0	1	0	80	8	0	9
Burrs	3	0	5	3	76	3	14
Residual copper	0	0	0	3	0	84	3
FN	8	7	16	18	8	3	60

To verify the classification effect of the SVM fusion multi-feature classification algorithm on bare PCB defect images, the algorithm proposed in this article was compared with the CNN algorithm using AlexNet (*Kavitha & Rao, 2019*), VGG-16, and ResNet-50 as the selected CNN network models. Accuracy (acc), precision (pre), recall, and F1 score were used as evaluation indicators for classification performance and were calculated, as follows:

$$\mathrm{acc}(A,B)=\frac{A}{B} \tag{40}$$

$$\mathrm{pre}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{41}$$

$$\mathrm{recall}(A_i,B_i)=\frac{A_i}{B_i} \tag{42}$$

$$F1=\frac{2\cdot\mathrm{pre}\cdot\mathrm{recall}}{\mathrm{pre}+\mathrm{recall}} \tag{43}$$

where, A represents the sum of the correct number of sample discrimination, Brepresents the total number of samples, B_i represents the number of samples of each type, TP represents the number of correctly identified samples in the category, FP represents the number of cost class samples misjudged by other class samples, and FNrepresents the number of samples in this category that were mistakenly classified as other categories.

The comparative classification results are shown in Table 8.

Figure 8 shows the detection and recognition results of defects in the tested images of different types of bare PCB boards using the algorithm proposed in this article. The rectangle in the figure represents a short circuit, the ellipse represents an open circuit, the diamond represents a burr, the square represents a leak, the circle represents a mouse bite, and the square with rounded corners represents residual copper. The comparative experimental results (Table 8) shows that, compared to traditional CNN classification algorithms, the SVM classification algorithm that integrates multi-scale shallow features and deep semantic features had a significant improvement in the accuracy of bare PCB

Table 8 Comparative classification results.

Algorithm	Evaluating indicator	Leaks	Mouse bites	Open circuits	Short circuits	Burrs	Residual copper	Average value
Algorithm in this article	Accuracy Rate/%				99.43			
	Precision Rate/%	100	100	98.82	100	97.67	100	99.42
	Recall rate/%	100	97.73	100	98.98	100	100	99.45
	F1 score/%	100	98.85	99.41	99.49	98.82	100	99.43
AlexNet	Accuracy Rate/%				97.16			
	Precision Rate/%	92.63	97.67	100	96.04	97.65	100	97.33
	Recall rate/%	100	95.45	95.24	98.98	98.81	94.25	97.12
	F1 score/%	96.17	96.55	97.56	97.49	98.22	97.04	97.17
VGG-16	Accuracy Rate/%				98.11			
	Precision Rate/%	100	98.84	98.75	96.00	95.45	100	98.17
	Recall rate/%	100	96.59	94.05	97.96	100	100	98.10
	F1 score/%	100	97.7	96.34	96.97	97.67	100	98.11
ResNet-50	Accuracy Rate/%				98.68			
	Precision Rate/%	100	96.67	100	97.00	100	98.86	98.76
	Recall rate/%	100	98.86	96.43	98.98	97.62	100	98.65
	F1 score/%	100	97.75	98.18	97.98	98.8	99.43	98.69

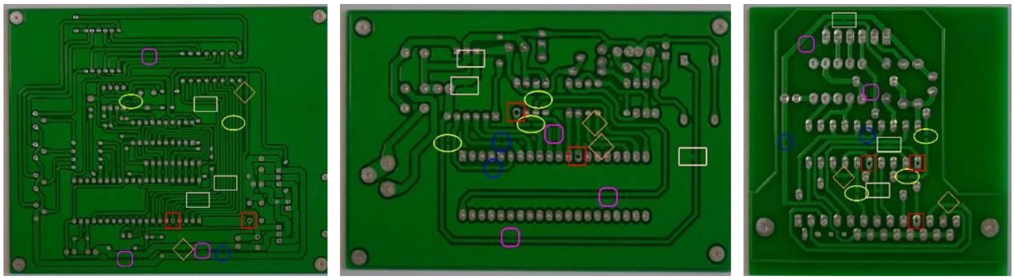

Figure 8 Defect identification results.

Full-size DOI: 10.7717/peerjcs.1900/fig-8

defect image classification, demonstrating that the feature extraction algorithm proposed in this article can obtain more effective information about defects.

CONCLUSION

Aiming at the problems existing in the field of bare PCB defect detection, this article proposes a multi-scale feature extraction method based on Gaussian pyramid that extracts multi-scale gray level co-occurrence matrix features, multi-scale directional projection features, and multi-scale gradient direction information entropy features, and uses a convolutional neural network to further extract image deep semantic features to enrich feature information. This article establishes a feature fusion method based on principal component analysis and Bayesian theory. First, based on principal component analysis, the extracted feature vectors are dimensionally reduced, greatly reducing both feature dimension and computational complexity. Then, Bayesian theory is used to calculate the weight values of different feature vectors, and multiple sets of vectors are fused in parallel. Compared to a single feature, fused features have stronger image representation capabilities. In addition, SVM has unique advantages in handling small sample problems. The proposed model fuses all the extracted features into a new feature vector to enhance the ability of the features to describe the image and converts it into a feature sequence, which is input into SVM for training to achieve bare PCB defect classification and recognition. The experimental results prove that the proposed algorithm can quickly and accurately identify leaks, mouse bites, open circuits, short circuits, burrs, and residual copper defects, with an accuracy rate of over 99% and a certain degree of stability.

The research in this article provides some ideas for bare PCB detection, but improvements are needed in future work. This article only focuses on some specific issues of bare PCB defect detection, and the proposed method is not yet suitable for all complex production environments. The interference mode, interference type, and interference degree in actual production environments cannot be predicted, such as lens interference, electromagnetic interference, or light source interference, which were not considered in the research process of this article. Therefore, improving the anti-interference ability of PCB defect detection algorithms in complex noise environments has significant practical significance. In addition, the defect detection algorithm proposed in this article can detect bare PCB boards in production and quickly detect defects that occur during the production process, but cannot evaluate the severity of defects. Future research should include measuring the size of different defects and evaluating the impact of defects on the quality of bare PCBs to better improve the production process.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The authors received no funding for this work.

Competing Interests

The authors declare there are no competing interests. Renpeng Li is employed by Anyang Iron and Steel Automation Software Co., Ltd.

Author Contributions

- Xixi Han conceived and designed the experiments, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Renpeng Li conceived and designed the experiments, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Boqin Wang performed the experiments, prepared figures and/or tables, and approved the final draft.
- Zhibo Lin analyzed the data, prepared figures and/or tables, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The code is available in the Supplementary File and Zenodo: Han, X. (2023). Code [Data set]. Zenodo. https://doi.org/10.5281/zenodo.10068752

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.1900#supplemental-information.

REFERENCES

Althubiti SA, Alenezi F, Shitharth S, Sangeetha K, Reddy CVS. 2022. Circuit manufacturing defect detection using VGG16 convolutional neural networks. *Wireless Communications and Mobile Computing* **2022**:1070405 DOI 10.1155/2022/1070405.

Bai S, Tang H. 2018. Softly combining an ensemble of classifiers learned from a single convolutional neural network for scene categorization. *Applied Soft Computing* **67**:183–196 DOI 10.1016/j.asoc.2018.03.007.

Coraci D, Tognolo L, Masiero S. 2023. Rehabilitation of fall in elderly: the paradigm of a new medical vision. *Applied Sciences* **13(3)**:1898 DOI 10.3390/app13031898.

Devi BT, Shitharth S. 2021. Multiple face detection using Haar-AdaBoosting, LBP-AdaBoosting and neural networks. *IOP Conference Series: Materials Science and Engineering* **1042(1)**:012017 DOI 10.1088/1757-899X/1042/1/012017.

Dong G, Sun S, Wang Z, Wu N, Huang P, Feng H, Pan M. 2022. Application of machine vision-based NDT technology in ceramic surface defect detection—a review. *Materials Testing* **64(2)**:202–219 DOI 10.1515/mt-2021-2012.

Fang J, Chen Y, Zhang X, Zhang Q, Huang C, Meng G, Liu W, Wang X. 2021. EAT-NAS: elastic architecture transfer for accelerating large-scale neural architecture search. *Science China* **64(09)**:103–115 DOI 10.1007/s11432-020-3112-8.

Gao H, Mokhtarzadeh AA, Li S, Fei H, Geng J, Wang D. 2023. Multi-angle face expression recognition based on integration of lightweight deep network and key point feature positioning. *Journal of Physics: Conference Series* **2467(1)**:012008 DOI 10.1088/1742-6596/2467/1/012008.

Ge L, Dan D, Li H. 2020. An accurate and robust monitoring method of full-bridge traffic load distribution based on YOLO-v3 machine vision. *Structural Control and Health Monitoring* **27(12)**:e2636 DOI 10.1002/stc.2636.

Hsiao TY, Chang YC, Chou HH, Chiu CT. 2019. Filter-based deep-compression with global average pooling for convolutional networks. *Journal of Systems Architecture* **95**:9–18 DOI 10.1016/j.sysarc.2019.02.008.

Hu P. 2023. A rice pest identification method based on a convolutional neural network and migration learning. *Journal of Circuits. Systems and Computers* **32(05)**:2350089 DOI 10.1142/S0218126623500895.

Huang WB, Wei P, Zhang MH, Liu H. 2020a. HRIPCB: a challenging dataset for PCB defects detection and classification. *The Journal of Engineering* **13**:303–309 DOI 10.1049/joe.2019.1183.

Jia XJ, Liu ZH. 2022. Character identification for integrated circuit components on printed circuit boards using deep learning. *Journal of Electrical Engineering & Technology* **17(1)**:601–616 DOI 10.1007/s42835-021-00885-4.

Jiang X, Sun HW. 2023. Learning performance of uncentered kernel-based principal component analysis. *International Journal of Wavelets, Multiresolution and Information Processing* **21(03)**:2250059 DOI 10.1142/S021969132250059X.

Kavitha K, Rao BT. 2019. Evaluation of distance measures for feature based image registration using AlexNet. *International Journal of Advanced Computer Science & Applications* **9(10)**:284–290 DOI 10.14569/IJACSA.2018.091034.

Kim J, Ko J, Choi H, Kim H. 2021. Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* **21(15)**:4968 DOI 10.3390/s21154968.

Krishna CS, Divya G. 2022. Accuracy measure for identification of fake blog articles in social website using random forest over Naive Bayesian classifier. *ECS Transactions* **107(1)**:12863 DOI 10.1149/10701.12863ecst.

Kshirsagar PR, Manoharan H, Selvarajan S, Althubiti SA, Alenezi F, Srivastava G, Lin JCW. 2022. A radical safety measure for identifying environmental changes using machine learning algorithms. *Electronics* **11(13)**:1950 DOI 10.3390/electronics11131950.

Lamsiyah S, Mahdaouy AE, Ouatik SEA, Espinasse B. 2023. Unsupervised extractive multi-document summarization method based on transfer learning from BERT multi-task fine-tuning. *Journal of Information Science* **49(1)**:164–182 DOI 10.1177/0165551521990616.

Li YT, Kuo P, Guo JI. 2020. Automatic industry PCB board DIP process defect detection system based on deep ensemble self-adaption method. *IEEE Transactions on Components, Packaging and Manufacturing Technology* **11(2)**:312–323 DOI 10.1109/TCPMT.2020.3047089.

Li Y, Li S. 2017. Defect detection of bare printed circuit boards based on gradient direction information entropy and uniform local binary patterns. *Circuit World* **43(4)**:145–151 DOI 10.1108/CW-06-2017-0028.

Liu G, Wen H. 2021. Printed circuit board defect detection based on MobileNet-Yolo-Fast. *Journal of Electronic Imaging* **30(4)**:043004–043004 DOI 10.1117/1.JEI.30.4.043004.

Lu Z, He Q, Xiang XG, Liu H. 2018. Defect detection of PCB based on Bayes feature fusion. *The Journal of Engineering* **2018(16)**:1741–1745 DOI 10.1049/joe.2018.8270.

Mallaiyan Sathiaseelan MA, Paradis OP, Taheri S, Asadizanjani N. 2021. Why is deep learning challenging for Printed Circuit Board (PCB) component recognition and how can we address it? *Cryptography* **5(1)**:9 DOI 10.3390/cryptography5010009.

Naji OAAM, Shah HNM, Anwar NSN, Johan NF. 2023. Square groove detection based on Frstner with Canny edge operator using laser vision sensor. *The International Journal of Advanced Manufacturing Technology* **125(5-6)**:2885–2894 DOI 10.1007/s00170-023-10862-y.

Shen J, Liu N, Sun H. 2020. Defect detection of printed circuit board based on lightweight deep convolution network. *IET Image Processing* **14(15)**:3932–3940 DOI 10.1049/iet-ipr.2020.0841.

Shitharth S, Kshirsagar PR, Balachandran PK, Alyoubi KH, Khadidos AAO. 2022. An innovative perceptual pigeon galvanized optimization (PPGO) based likelihood Naïve Bayes (LNB) classification approach for network intrusion detection system. *IEEE Access* **10**:46424–46441 DOI 10.1109/ACCESS.2022.3171660.

Simonyan K, Zisserman A. 2014. Very deep convolutional networks for large-scale image recognition. ArXiv arXiv:1409.1556v6.

Sivaranjani A, Senthilrani S. 2023. Computer vision-based cashew nuts grading system using machine learning methods. *Journal of Circuits, Systems and Computers* **32(03)**:2350049 DOI 10.1142/S0218126623500494.

Song JD, Kim YG, Park TH. 2019. SMT defect classification by feature extraction region optimization and machine learning. *International Journal of Advanced Manufacturing Technology* **101(5-8)**:1303–1313 DOI 10.1007/s00170-018-3022-6.

Talaat M, Arafa I, Metwally HMB. 2020. Advanced automation system for charging electric vehicles based on machine vision and finite element method. *IET Electric Power Applications* **14(13)**:2616–2623 DOI 10.1049/iet-epa.2020.0380.

Tsai DM, Huang CK. 2018. Defect detection in electronic surfaces using template-based fourier image reconstruction. *IEEE Transactions on Components, Packaging, and Manufacturing Technology* **9(1)**:163–172 DOI 10.1109/TCPMT.2018.2873744.

Yu C, Pei H. 2021. Face recognition framework based on effective computing and adversarial neural network and its implementation in machine vision for social robots. *Computers & Electrical Engineering* **92(1)**:107128 DOI 10.1016/j.compeleceng.2021.107128.

Zhang H, Jiang L, Li C. 2021. CS-ResNet: cost-sensitive residual convolutional neural network for PCB cosmetic defect detection. *Expert Systems with Applications* **185(1)**:115673 DOI 10.1016/j.eswa.2021.115673.

Zhao W, Gurudu S, Taheri S, Mallaiyan Sathiaseelan MA, Asadizanjani N. 2022. PCB component detection using computer vision for hardware assurance. *Big Data and Cognitive Computing* **6(2)**:39 DOI 10.3390/bdcc6020039.