

Broadband hybrid attention-based feature fusion network for printed circuit boards defect classification

Weiming Gan^{a,*}, Lingkai Hu^{a,*}, Feng Zhan^a, Xiaoqing Liu^{a,b}, Zhidi Huang^{c,**}

^aSchool of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou, 510006, China

^bDongguan Machong Middle School, Dongguan, 523133, China

^cShennan Circuits Co., Ltd., Shenzhen, 518117, China

ARTICLE INFO

Keywords:

Loss function

Imbalanced datasets

Printed circuit boards

Multi-scale receptive field

ABSTRACT

The classification of true- and pseudo-positive defects in printed circuit boards (PCB) has always been an essential task for industrial applications. Existing methods struggle to diagnose effectively due to the difficulties in local/global distribution extraction and imbalance data analysis. Additionally, the problems of texture distribution discrepancy imbalance between the design and scan maps cannot be effectively solved, which may affect the classification results. To address the above challenges, a broadband hybrid attention-based feature fusion Network (BHAFFN) is proposed to classify the true/pseudo-positive defects of PCB. The proposed BHAFFN can effectively extract the correlation information of the local and global defects of PCB images through a broadband multiscale hybrid attention mechanism and can solve the dataset imbalance problem by adopting a metric space loss. Experimental results demonstrate that the proposed BHAFFN can effectively improve the accuracy of PCB true- and pseudo-positive defect classification tasks and is superior over several existing algorithms.

1. Introduction

Printed circuit boards (PCB) have a wide range of applications in industrial products [1], such as imaging sensors [2], variable voltage converters [3], wireless charging systems [4], and biomedical devices [5]. The quality of PCBs can directly affect the performance of electronic devices, so it is crucial to identify and screen defective PCBs. Currently, automatic optical inspection (AOI) and visual repair system (VRS) are the two main systems applied for PCB defect detection. However, both systems have limitations in practical applications. For example, the traditional image recognition technology used in AOI can produce numerous pseudo-positive defects misjudged as true-positive defects during the working process [6]. If defective PCBs are put into the market, they may cause damage to the machine or even casualties [7]. Conversely, discarding pseudo-positive PCBs can lead to unnecessary economic losses for enterprises [8]. Additionally, the VRS system requires substantial labor to classify true-positive and pseudo-positive defects based on AOI-generated defect data, significantly prolonging PCB manufacturing time and causing heavy economic burdens. Therefore, developing intelligent classification methods for true- and pseudo-positive PCB defects is crucial.

The classification of true- and pseudo-positive defects in PCBs depends on two aspects: the surface texture of the scanned images and the contextual relationship between the design and the scanned images. For instance, a local break in the PCB that results in circuits not being connected can be classified as a true-positive defect. In contrast, skin flakes on the surface of the PCB or an unclean solder mask may appear as true defects but, in reality, do not affect the PCB's functionality, defining them as pseudo-positive defects. Therefore, to enhance the accuracy of true- and pseudo-positive defect classification, it is crucial to analyze both local defect features and contextual defect features and to attenuate irrelevant background and noise information. Some PCB detection methods have been proposed in recent years to achieve intelligent PCB defect detection and have shown extraordinary results [9, 10, 11, 12, 13]. However, these well-known methods have limitations in perceptual fields, multiscale feature extraction capabilities, and attenuation of irrelevant information, resulting in limited PCB true- and pseudo-positive classification accuracy, as shown in Table 1. To address this issue, this paper proposes a Focus-Texture-Net (FTN) to improve the model's perceptual field, contextual multiscale feature extraction capabilities, and efficiency in attenuating irrelevant information. The proposed FTN utilizes a Gather Block (G-Block) as the primary feature extraction module, with its specific variables determined

*Co-first author.

**Corresponding author.

E-mail address: huangzd@scc.com.cn (Zhidi Huang)

based on the required perceptual field of the training samples. G-Block adopts a broadband multi-scale hybrid attention mechanism (BMHAM) to fuse multi-scale feature information, allowing for the extraction of defect features from a wide range of contextual information. Moreover, this method effectively attenuates irrelevant background and noise information and avoids the information loss caused by the expansion rate. Therefore, the proposed FTN can effectively extract the correlation information between the local data distribution of defects and the overall data distribution, resulting in superior classification accuracy compared to existing models.

Table 1

Some representative approaches for PCB cosmetic defect detection.

Method	Standard image	Class imbalance	Distribution difference imbalance	Large receptive field	Attention mechanism
CLCM (2023) [14]	✗	✓	✓	✗	✗
LLD-Net (2024) [15]	✓	✗	✗	✓	✗
YOLO-HMC (2024) [6]	✓	✗	✗	✓	✗
AFRNet (2023) [16]	✗	✗	✗	✗	✗
Proposed	✓	✓	✓	✓	✓

In the dataset of true- and pseudo-positive PCB defects collected from actual industrial production lines, a serious problem of sample class imbalance [17] and large distribution discrepancy exists. It is challenging to improve the model classification accuracy due to imbalanced data and distribution discrepancy problems. Sample class imbalance, a common issue, often arises in the comparison process between pseudo-positive and true-positive defects [18]. For the inhomogeneous distribution discrepancy problem, which is caused by the texture differences between scanned and designed maps, eventually causes the non-fitting problem of the models. To address the issue of imbalanced data [19], various approaches such as data sampling [20], data generation [21], and loss functions [22] are typically proposed and applied. However, the data sampling approach may lead to model overfitting [6], and traditional data generation algorithms may face difficulties in obtaining convincing labels or data distributions for the generated data [23]. In contrast, the commonly applied loss functions demonstrate superior robustness and flexibility in handling these imbalanced datasets [24].

In recent years, efforts have been made to address the issue of class-distribution discrepancy imbalance in computer vision. Some loss functions [25, 26, 27] are developed to reduce the task complexity caused by the imbalance data problem by dynamically modifying the positive and negative sample gradients. Generally, a common task is to classify true- and pseudo-positive defects of PCB in industrial applications. In this case, the existing loss functions that can effectively solve the sample category imbalance problem lack a solution for the coexistence of data distribution difference imbalance. Additionally, existing loss functions [25, 26, 27] have not been designed to resolve the distribution differences between the original scanned and designed maps. As a result, applying the conventional loss functions directly to the true- and pseudo-positive defect classification task will limit the model performance. To further improve classification accuracy, a loss function called clustering loss is proposed and applied in this paper. This function can address the problems of unbalanced data and distribution discrepancy. The proposed clustering loss function is a dynamic factor and microscopic cross-entropy loss based on the gray-level cooccurrence matrix (GLCM) information entropy. With the changes in the sample data distribution and the confidence level of the model output, this dynamic factor can modify the sample loss value accordingly. This approach provides a better solution for addressing the class-distribution discrepancy imbalance problem in industrial settings, where it is crucial to classify true- and pseudo-positive defects accurately.

In summary, this paper presents an advanced model for true- and pseudo-positive PCB defect classification. The model integrates a multiscale information extractor with a dynamically weighted loss function. Evaluation is conducted on a PCB dataset from actual industrial production, which contains various challenging defects that are hard to classify based on local texture distribution alone.

In general, the contributions of this paper are as follows:

1. A PCB dataset with 134,712 samples of true- and pseudo-positive defects (PCB-TP) (the PCB-TP Open source address in REFERENCES [28]) that has been annotated by factory experts is published in this paper.
2. A model called G-Block is proposed for PCB true- and pseudo-positive defect classification, which extracts comprehensive contextual information by BMHAM and multiscale mixed domain.

3. A GLCM-based loss function called clustering loss (CL) is proposed to establish a loss metric space by comparing the GLCM texture complexity of the design map with that of the scan map, thus effectively relieving the task complexity by computing the loss value in the metric space.
4. The proposed method is verified on the PCB dataset, and the experimental results demonstrate that our method can effectively improve the accuracy of classifying true- and pseudo-positive defects in PCB over other recently published methods.

The rest of this paper is organized as follows: Section 2 explains the construction method for each part of the proposed BHAFFN and provides an in-depth analysis of the adopted strategies. Section 3 presents the research material and provides a detailed analysis of the experimental results. Finally, Section 4 concludes the paper and provides directions for future research.

2. Methodology

This section provides an overview of the general architecture of BHAFFN. It then provides detailed descriptions of the two components included in BHAFFN: the specific structure of the FTN and the construction strategy of the CL loss function.

2.1. General structure

To improve the robustness of the model, particularly for industrial applications, data enhancement is typically performed at the input layer of the model [29], which involves spatial and color space transformations.

The Focus-Texture-Net (FTN) in BHAFFN comprises four components: the G-Block, Fractal Block, and fully connected Block (FC Block), as illustrated in Fig. 1. One of the components of FTN is G-Block. It captures the effective correlation between local and global information in the sample by using a multiscale mixed domain and broadband multiscale hybrid attention mechanism. As shown in Fig. 1, FTN has three G-Blocks in series. After extracting a wide range of multi-scale contextual information with G-Block, this article employs Fractal Block to perform deep data mining and information reorganization on these multi-scale contexts. As shown in Fig. 1, FTN comprises two concatenated Fractal Blocks.

BHAFFN then calculates the loss value using the cluster-ing loss (CL) loss function. This loss value will be used for backpropagation.

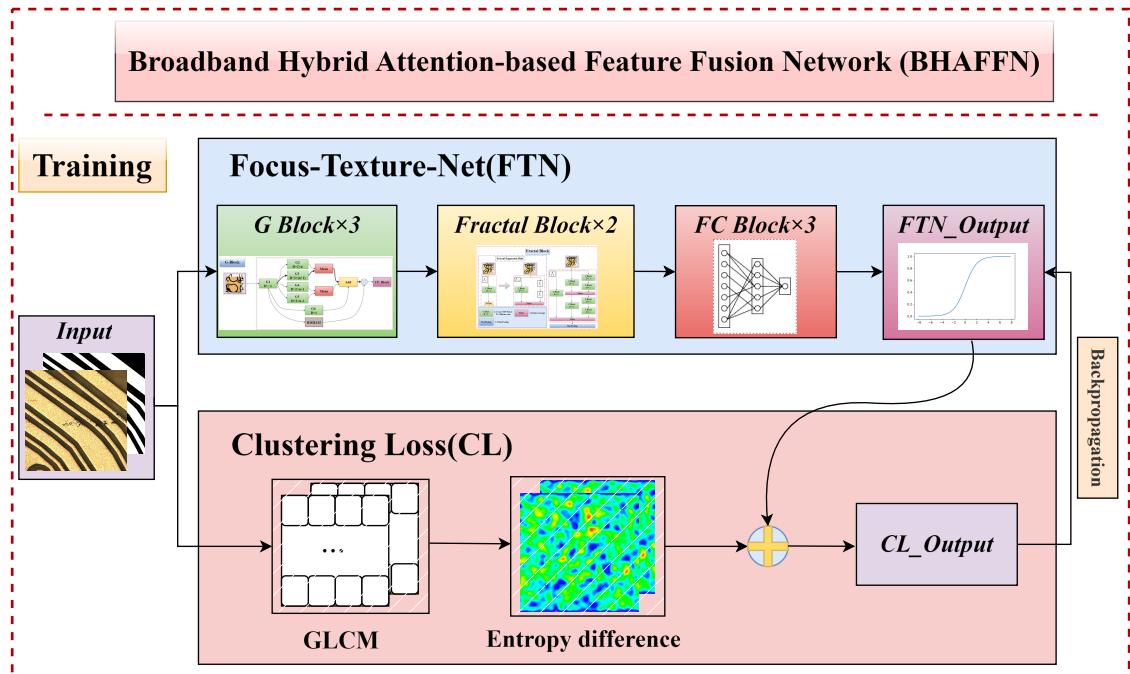


Fig. 1. General structure of the proposed BHAFFN.

2.2. G-Block

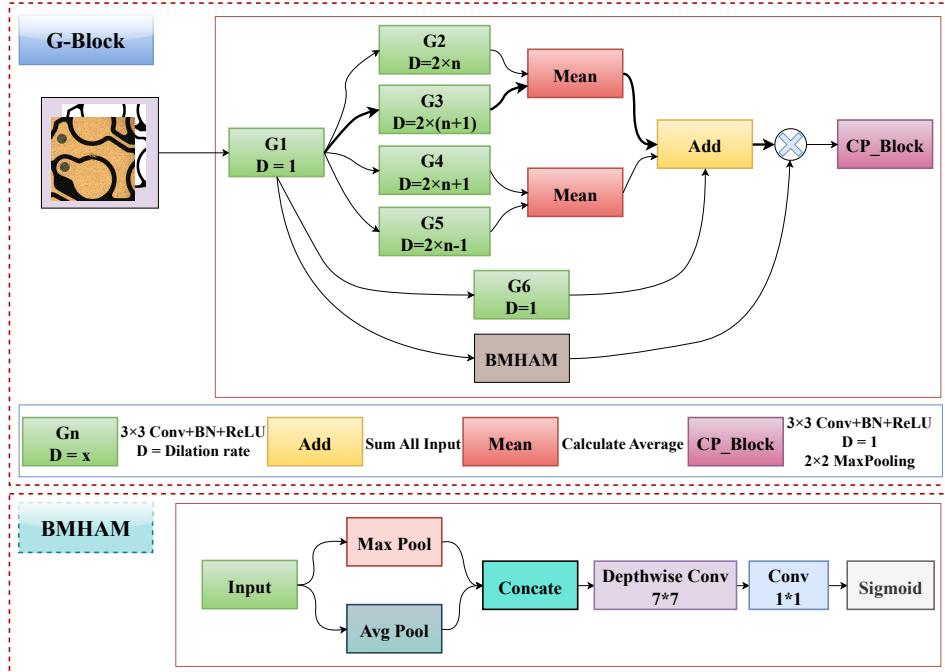


Fig. 2. G-Block: Detailed schematic diagram of the structure.

As illustrated in Fig. 2, the G-Block contains BMHAM and a multiscale mixed domain. In the G-Block, D represents the dilation rate, with a larger dilation rate indicating a larger perceptual field of the convolution kernel. These modules include G_2-G_5 , information balancing layers G_1 and G_6 , Mean(\cdot) and Add(\cdot) and BMHAM. The G-Block module works as follows:

First, encodes the input data once and is computed as follows:

$$R(x) = \text{ReLU}(\text{BN}(\text{Conv}(x)_{3 \times 3})), \quad (1)$$

$$G_1 = R(\text{input}), \quad (2)$$

where $\text{Conv}(\cdot)_{k \times k}$ denotes the convolution operation using a convolution kernel with expansion factor kernel size = k , $D = 1$, $\text{BN}(\cdot)$ denotes batch normalization, and $\text{ReLU}(\cdot)$ is the activation function. Then, the results of G_1 are input into $G_2 - G_5$, where $G_i D$ defines the calculation of the convolution kernel expansion rate in $G_2 - G_5$ as follows:

$$G_i D = \begin{cases} 2n_i, & i = 2 \\ 2(n_i + 1), & i = 3 \\ 2n_i + 1, & i = 4 \\ 2n_i - 1, & i = 5 \end{cases}. \quad (3)$$

In this paper, $n_2 = n_3 = n_4 = n_5$ in $G_i D$. The first output of information fusion is calculated using the following equation:

$$\text{output}_1 = \text{Add}(\text{Mean}(G_2, G_3), \text{Mean}(G_4, G_5), G_6(G_1)). \quad (4)$$

Among the parts of the G-Block shown in Fig. 2, the information balance layer G_6 is implemented as a skip connection. Mean(\cdot) indicates the addition and averaging of two matrices, whereas Add(\cdot) refers to the summation of all output matrices.

Inspired by [30] and [31], this paper proposes BMHAM in the G-Block. The BMHAM module aggregates channel information using maximum pooling and mean pooling operations. Then, cross-channel correlation and spatial correlation are reduced by using a depth-separable convolution of 7×7 . Finally, BMHAM uses 1×1 convolution to learn channel attention correlation and perform dimensionality reduction. The BMHAM is calculated by the following formula:

$$\text{BMHAM}_1(x) = \text{Concat}(\text{MaxPool}(x), \text{avgPool}(x)), \quad (5)$$

$$\text{BMHAM}_2(x) = \text{Depthwise_Conv}(\text{BMHAM}_1(x))_{7 \times 7}, \quad (6)$$

$$\text{BMHAM}(x) = \sigma(\text{Conv}(\text{BMHAM}_2(x))_{1 \times 1}), \quad (7)$$

where $\text{Concat}(\cdot)$ is the concatenate operation, which splices the last dimension of the matrix. $\text{Depthwise_Conv}(\cdot)$ is a depth-wise separable convolution. $\sigma(\cdot)$ represents the sigmoid activation function. The final output of the G-Block is calculated using the following equation:

$$\text{Output} = \text{MaxPooling}(R(\text{output}_1 \times \text{BMHAM}(\text{input})))_{2 \times 2}, \quad (8)$$

$\text{MaxPooling}(\cdot)_{2 \times 2}$ denotes a 2×2 maximum pooling operation.

2.3. G-Block parameter determination

To determine the specific parameters for the G-Block, the required receptive field of the model is calculated to establish the value of parameter n in Eq. (3). In general, the field of view (FOV) of the pixel of interest obtained by convolution is also referred to as the receptive field [32]. The concept of the receptive field is important for understanding and diagnosing how deep CNNs work [33, 34]. In general, the receptive field of the convolution is computed iteratively. The specific iterative formula for a convolution kernel with an expansion rate is as follows:

$$R_l = R_{l-1} + (k_l - 1) D \prod_{i=0}^{l-1} S_i, \quad (9)$$

where R_l is the receptive field of a pixel point in the current convolutional layer corresponding to the input layer, k_l is the convolutional kernel of the current convolutional layer, D is the expansion factor of the convolutional kernel of the current convolutional layer, and S_i is the moving step. Although a convolution kernel with an expansion rate can effectively enlarge the receptive field, to prevent gaps between the weights, it is necessary to ensure that the expansion rate D is upper bounded, as described in [32]:

$$\frac{k}{s} \geq D. \quad (10)$$

The k in Eq. (10) represents the size of the convolution kernel, whereas s denotes the stride of the convolutional operation. Moreover, the size of the convolution kernel, denoted as k_l , with an expansion rate can be calculated using the following formula:

$$k_l = 2D + 1. \quad (11)$$

By substituting k_l from Eq. (11) into k from Eq. (10), the following equation is obtained:

$$2 + \frac{1}{D} \geq s. \quad (12)$$

Given the expansion rate $D \in \mathbb{N}^*$ and the convolution step $s = 1$ in the G-Block, Eq. (12) holds constantly. Therefore, the specified expansion rate remains well below this upper bound.

The calculation path for the receptive field is shown by the bold arrow in Fig. 2. The other parts are not involved in the calculation of the receptive field, and we calculate the receptive field of the color pathway part in the G-Block. First, the receptive field of this path is the largest. Secondly, the information with a smaller receptive field can be fused through the ADD(\cdot) operation shown in Fig. 2. The receptive field of this pathway under each parameter n was iteratively calculated using Eq. (9), and the results are shown in Table 2.

Table 2

G-Block field of view on each layer.

$G_i D$ Para.	FOV			Final Coverage Rate
	G-Block 1	G-Block 2	G-Block 3	
n=1	21	101	421	82.22%
n=2	25	121	505	82.22%
n=3	29	141	589	115.03%

2.4. Fractal Block

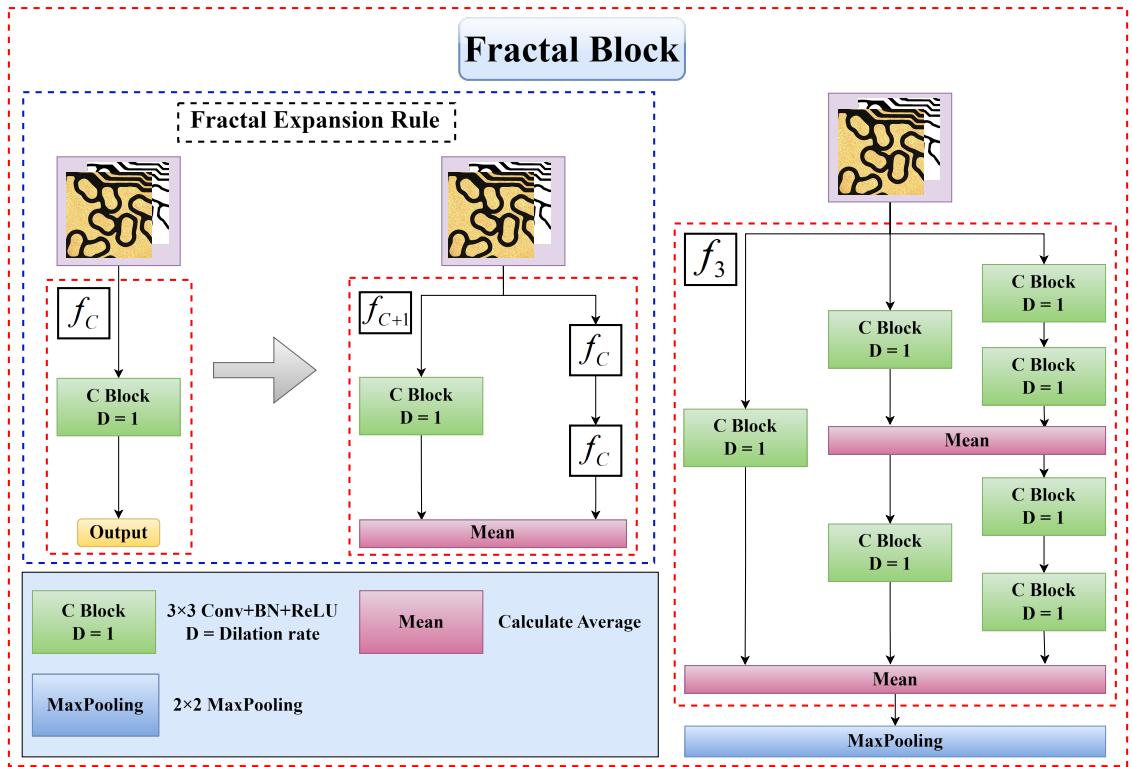


Fig. 3. Fractal Block: Detailed schematic diagram of the structure.

Fractal comes from geometry [35, 36], which analyzes data from multiple levels, perspectives, and constituents. The structure of the Fractal Block is derived from FRACTALNET [37] and is shown in Fig. 3. This network structure's shallow subnetworks provide a quick answer, whereas deeper subnetworks with higher latency provide a more accurate answer [36]. Assuming that $f_C(z) = R(\text{input})$, the specific expression for the multi-branch $f_{C+1}(z)$ is shown as follows:

$$f_{C+1}(z) = [(f_C \circ f_C)] \oplus [R(z)]. \quad (13)$$

The specific model construction is shown in Fig. 3 by the fractal expansion rule. $f_{C+1}(z)$ is stacked by $f_C(z)$ with the same rule. According to Eq. (13), it is known that when $f_3(z)$, the model construction result is shown in the right end block in Fig. 3, where the specific formula for the C Block is the same as Eq. (1).

2.5. Construction strategy of clustering loss

The gray-level co-occurrence matrix (GLCM) is a powerful tool for analyzing image texture features due to its robustness and stability. However, it cannot be directly incorporated into the loss function of neural networks for two reasons. First, the GLCM is calculated for a single image and thus cannot be used to directly compare texture differences between two images. Second, the eigenvalue entropy of the GLCM typically falls between 1 and 10. If this value is directly added to the loss function, it can often result in difficulty in decreasing the gradient of the model. To address these limitations, the properties of industrial production datasets are utilized, and CL is proposed to enhance the GLCM method.

The grayscale co-generation matrix starts from the image element (x, y) with grayscale value i and counts the frequency $\rho = (i, j \mid a, b, \theta)$, which appears simultaneously with the image element $(x + \alpha, y + \beta)$, with distance d and grayscale value j . The specific mathematical expressions are as follows:

$$\rho(i, j \mid \alpha, \beta) = \{(x, y) \mid f(x, y) = i, f(x + \alpha, y + \beta) = j\}, \quad (14)$$

$$(i, j) \in [0, L - 1].$$

Most previous studies have focused solely on scan maps with visible defects and even publicly available defective datasets consisting only of scan maps. However, the value of design maps without surface defects is often underestimated. In fact, design map data can provide a wealth of a priori knowledge to identify defects, and acquiring such data is often straightforward in many manufacturing scenarios. Leveraging this property, a formula is constructed to calculate the entropy difference of the texture distribution of the samples, as follows:

$$C_k = \frac{1}{L^2} \sum_{k=0}^{L^2} \left| \log(\rho_f(i, j)_k) - \log(\rho_s(i, j)_k) \right|, \quad (15)$$

where $\rho_f(i, j)$ represents the grayscale co-occurrence matrix of the scanned image, $\rho_s(i, j)$ represents the grayscale co-occurrence matrix of the design image, and C_k represents the expected value of the information entropy metric distance between the grayscale co-occurrence matrix $\rho(i, j)$ of the real image $\rho_f(i, j)$ and the design image $\rho_s(i, j)$. The expectation value C_k and the true labels of the corresponding samples can be used to cluster each sample in the space, thus establishing the metric space of the samples. In this metric space, different data clusters represent the different metric coefficients that should be incorporated into the loss function.

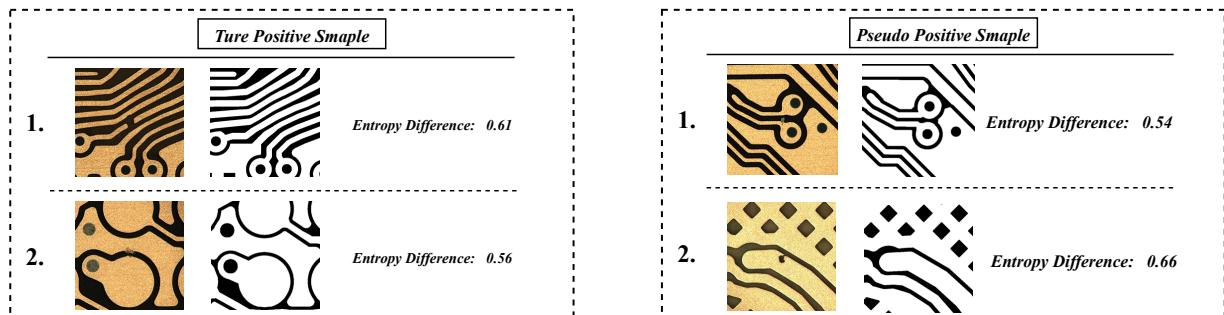


Fig. 4. The entropy difference of true-positive defects samples. **Fig. 5.** The entropy difference of pseudo-positive defects samples.

For true-positive samples, the smaller the difference between the entropy of the scanned image and the design map (i.e., the texture distribution is similar), the higher the network's ability to fit it. As shown in Fig. 4, sample 2 has a data distribution similar to the standard graph and some pseudo-positive defect samples (e.g., sample 1 in Fig. 4). However, most of the true-positive defect samples (e.g., sample 1 in Fig. 5) have more prominent local data distributions. The model often struggles to fit sample 2 because it is labeled as a true-positive defect but resembles pseudo-positive samples. To address this issue, the entropy difference based on GLCM is utilized as prior knowledge, and the complexity of model training is reduced by increasing the optimization weight of the loss function for these samples. For pseudo-positive samples, the larger the difference between the entropy of the scanned image and the

design map (i.e., the texture distribution is not similar), the higher the fitting ability of the network to this sample. Samples with different labels have different metric coefficients under different metric values.

$$S_k = \frac{C_k - \text{Min}(C_k)}{\text{Max}(C_k) - \text{Min}(C_k)} + a, (a > 0) \quad (16)$$

$\text{Max}(\cdot)$ is used to find the maximum value in the sequence. $\text{Min}(\cdot)$ is used to find the minimum value in the sequence. a is a hyperparameter used to prevent S_k from going to zero and causing the gradient to vanish. It is necessary to normalize C_k such that $C_k \in (0, 1)$ and map it into a probability form to match the parameter distribution of the neural network.

Let $f^w(x)$ be the output of the neural network. The following probabilistic model is defined for the binary classification task, where the probabilistic output of the model is expressed as follows:

$$\hat{y} = \sigma(f^w(x)), \quad (17)$$

$$\hat{y}_{\text{class}} = \begin{cases} 1, \hat{y} \geq 0.5 \\ 0, \hat{y} < 0.5 \end{cases}, \quad (18)$$

$\sigma(\cdot)$ is the sigmoid function. From Eq. (18), if the model output $\hat{y} \in [0, 0.5]$, it is judged to be a pseudo defect. If $\hat{y} \in [0.5, 1]$, it is judged to be a true defect. The log-likelihood equation for the model's probabilistic output \hat{y} is derived as follows:

$$\delta = \left| \frac{\partial L_{CE}}{\partial x} \right| = |\hat{y} - y|, \quad (19)$$

the final loss function $CL(\hat{y}, y)$ is constructed as follows:

$$CL(\hat{y}, y) = \begin{cases} L_{CE}(\hat{y}, y)(1 - S_k)\delta\lambda_1, & \text{if } y = 1 \text{ and } \lambda_1 \neq 0 \\ L_{CE}(\hat{y}, y)S_k\delta\lambda_2, & \text{if } y = 0 \text{ and } \lambda_2 \neq 0 \end{cases} \quad (20)$$

The loss function applied to neural network training is given by the final constructive equation $CL(\hat{y}, y)$, where λ_1 and λ_2 are hyperparameters. The loss function uses the texture complexity entropy difference between the sample design map and the scan map to establish the loss metric space and reduces the training task complexity caused by the imbalance of this kind of dataset by strengthening BHANN's ability to fit the samples with high texture differences.

3. Experiments and results

This section outlines the dataset utilized, the experimental design employed, the configuration of the experimental platform, and the analysis of the experimental results. This paper reports on a multitude of experiments conducted on the PCB-TP dataset [28], which contains true- and pseudo-positive defects on PCB.

3.1. Experimental data

This section describes the true and pseudo-positive effects of PCB in detail and introduces the data set used in this experiment briefly. (In the download link of the dataset, the "datasets introduction" document will provide a more detailed introduction to the dataset)

The judgment of true- and pseudo-PCB defects is equivalent to the judgment of the impact of defect functions [20]. The classification of defects can also be referred to as functional defect classification based on whether the defect affects the original function of the PCB. For instance, when a "line defect" appears on the circuit in the PCB (as shown in Fig 7. (C)), although the defect does not completely cut off the circuit, the cross-sectional area of the copper wire connecting the circuit is small. In the process of product use, it may cause phenomena such as excessive heat generation, thereby affecting its original function. This is considered a functional (true) defect, and the exact defect is a "line defect." If a defect appears on the circuit in the PCB, but only a small part of the original circuit is cut off (as shown in Fig. 8(A)), the defect will not affect the continuity of the circuit or the use of the final product, so it will not affect its original function. This type of defect is considered non-functional (pseudo). In addition to defects on copper wires, weak rust

or foreign matter adhering to PCB surfaces can also cause false alarms (as shown in Fig. 8(B)). However, after cleaning this type of PCB, it can be used normally. In this article, this type of defect is also considered non-functional (pseudo).

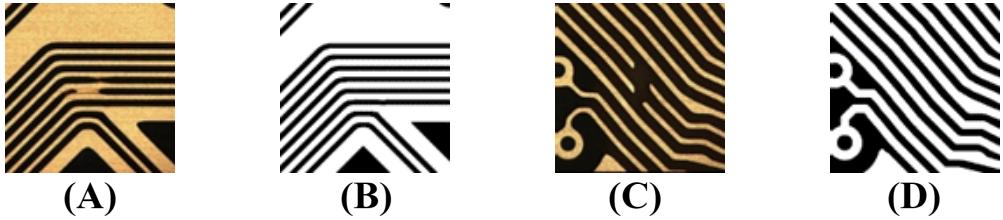


Fig. 6. PCB-TP datasets presentation.

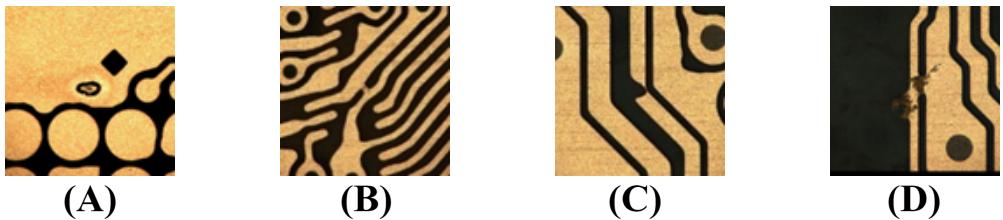


Fig. 7. Representative images of true defects.

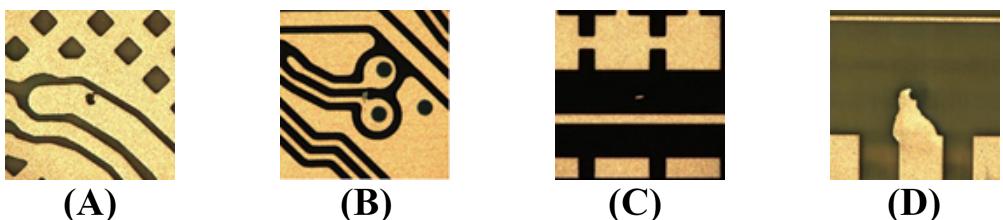


Fig. 8. Representative images of pseudo defects.

This dataset contains 134,712 sets of image data, with each set comprising two images, as illustrated in Fig. 6. Within this dataset, channels 1 to 3 correspond to images collected directly from the production line, whereas channel 4 represents the design standard map. Each image in the dataset has a dimension of 512 pixels \times 512 pixels, and the associated labels classify defects into two categories: true defects and pseudo defects. Figs. 7 and 8 display representative images of external PCB defects.

In this paper, the PCB-TP [28] dataset is partitioned into a training set and a test set through random sampling. In particular, 80% of the dataset is allocated to the training set, and the remaining 20% is assigned to the test set.

3.2. FTN parameter determination

When $n_2 = n_3 = n_4 = n_5 = 1$ in Eq. (3), the amount of high-dimensional feature information output by three G-Blocks in a series can not only cover 82.22% of the whole image (as shown in Table 2) but also reduce the number of model operations at the same time. The specific parameters of each G-Block are shown in Table 3, taking G-Block 1 as an example. $n = 1$ refers to $n_2 = n_3 = n_4 = n_5 = 1$ in Eq. (3), and *Output Channels* refers to the number of output hidden layers of G-Block 1.

All the Fractal Blocks used in this study are of f_4 type and their specific construction methods are illustrated in Eq. (13). In Table 3, the *Output Channels* parameter of Fractal Block 1 indicates the number of output hidden layers of the module, which are set to 128 and 256.

Table 3

Schematic diagram of the input parameters of FTN modules.

Block Type	Block Name	Parameter	
		Output Channels	n
Input	Data Enhancement	4	\
G-Block	G-Block 1	16	1
	G-Block 2	32	1
	G-Block 3	64	1
Fractal Block	Fractal Block 1	128	\
	Fractal Block 2	256	\
FC Block	FC Layer 1	32	\
	FC Layer 2	16	\
	FC Layer 3	1	\
Output	Sigmoid	1	\

3.3. Experimental platform and experimental design

The algorithms are implemented in the deep learning framework TensorFlow 2.0 and trained on computers equipped with an NVIDIA GeForce RTX 3090 GPU, Python 3.7.6, CUDA 11.0, and CUDNN 8.0.5. To evaluate the effectiveness of BHAFFN, its performance is compared with several deep classification methods used in recent years for identifying true- and pseudo-positive PCB defects, including [9, 10, 11, 12, 13].

To assess the performance of CL, it is replaced with several classical methods commonly used to address the problem of sample class imbalance and distribution difference imbalance. Ablation experiments are conducted on the G-Block using several classical models to evaluate the effectiveness of the proposed G-Block in FTN. Additionally, model parameter sensitivity experiments are performed with different hyperparameters. Finally, experiments are carried out under varying expansion rates. In particular, the maximum number of epochs is set to 30, and the batch size is 16. The optimization algorithm used is Adam Optimizer, with an adaptive learning rate, a basic learning rate of 0.01, and a decay factor of 0.99 for the learning rate.

3.4. Evaluation metrics

The definitions of the evaluation metrics Specificity, Sensitivity, and G-Mean used for the category imbalance problem are presented as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100\%, \quad (21)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100\%, \quad (22)$$

$$G - \text{mean} = \sqrt{\text{Sensitivity} * \text{Specificity}}. \quad (23)$$

Among them, TP in Eq.(21) and Eq.(22) stands for true positive defects, TN stands for true negative defects, FP stands for false positive defects, and FN stands for false negative defects. The evaluation metric EH for the learning ability of distribution imbalance data is specifically defined as follows:

$$EH = \frac{\sum C_k}{\text{len}(C_k)}, k \in (y = 1, \hat{y} = 0), \quad (24)$$

where $\text{len}(\cdot)$ is expressed as the number of elements in finding C_k . The specific expression of C_k is Eq. (15), which is the entropy difference calculated from the real image and the standard image. Therefore, the greater the chaos in the data distribution of the two images, the smaller the difference in entropy, which indicates a lower similarity between the two images. In the case of true-positive defects, the larger the entropy difference between the true image and the standard image, the more challenging it becomes for the model to fit the data distribution.

Thus, the smaller the EH of the samples in the test set, the more effective the model is in solving the distribution difference imbalance problem, and vice versa.

3.5. PCB classification model

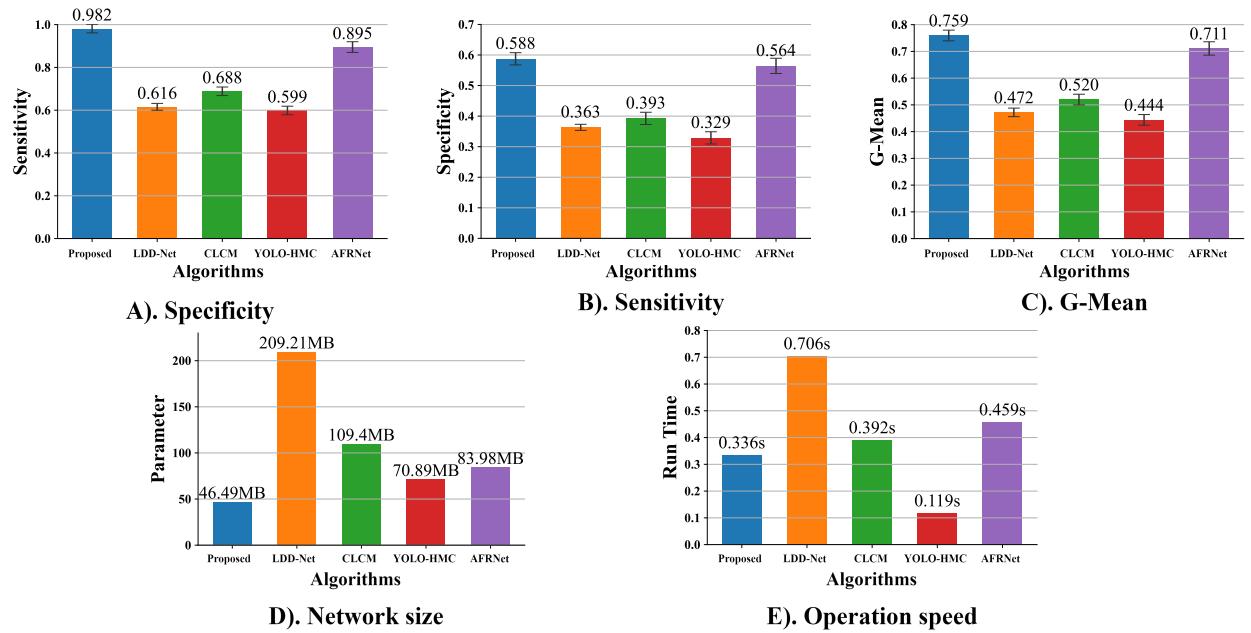


Fig. 9. Comparison of Evaluation Metrics of BHAFFN, LDD-Net, CLCM, YOLO-HMC, and AFRNet.

The experimental results of PCB true- and pseudo-positive defect classification methods on this dataset in recent years are depicted in Fig. 9. In this paper, every experiment is trained and evaluated using the same training and test sets. Evidently, our proposed BHAFFN outperforms other competitors in all metrics in the experiment. Moreover, in terms of network size, the proposed network in this paper has the smallest size of 46.49 MB, while exhibiting better sensitivity, specificity, and G-mean evaluation metrics than the four other networks, as depicted in Fig. 9 (D). In addition, 100 samples are randomly selected from the test set in this subsection, and the average computing speed for each sample is calculated. As illustrated in Fig. 9 (E), although YOLO-HMC [6] has a faster computing speed, all its evaluation metrics in this dataset are lower compared to other methods.

The preliminary comparison results in the bar chart suggest that BHAFFN surpasses other models in various aspects. To further evaluate the qualitative performance of BHAFFN, Grad-CAM [38] is employed to visualize the feature maps of different models on the test images, as shown in Fig. 10 to 12. The feature maps reveal that BHAFFN can effectively extract and emphasize the defect features by filtering out more invalid information and attenuating the irrelevant background and noise information, thus improving its ability to recognize defects. Based on Fig. 10 to Fig. 12 (B), (C), (D), and (E), LDD-Net, CLCM, YOLO-HMC, and AFRNet exhibit a limited ability to suppress irrelevant background and noise information. Moreover, the effective information used for defect classification is not very accurate. In Fig. 12 (C), it can be observed that LDD-Net suppresses local defect information as background or noise information. Fig 10-12. (A) represents input image, (B) represents a proposed method, (C) represents LDD-Net, (D) represents CLCM, (E) represents YOLO-HMC, and (F) represents AFRNet.

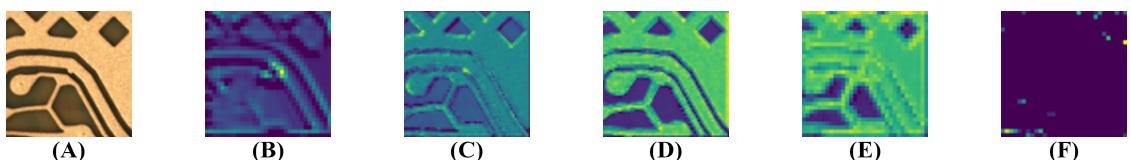


Fig. 10. Grad-CAM of sample 1.

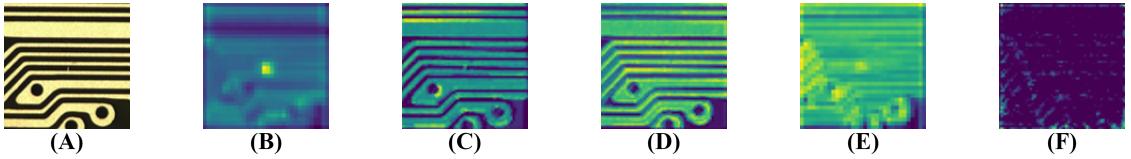


Fig. 11. Grad-CAM of sample 2.

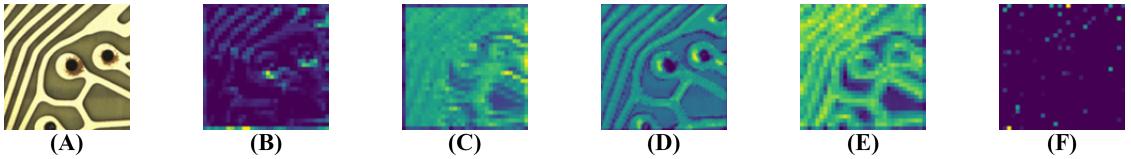


Fig. 12. Grad-CAM of sample 3.

3.6. Imbalanced data processing

The final experimental results of Weighted Cross-Entropy (WCE) Loss are marginally lower than CL, whereas Var Focal Loss [39], Gradient Harmonizing Mechanism Loss (GHM Loss) [40], and Seesaw Loss [41] of the target detection algorithm exhibit lower performance than CL on this dataset. Although up-sampling and down-sampling techniques yield better results on the training set, their accuracy on the test set is poor, with a true defect recognition rate of only 49.23% and 50.75%, respectively. Table 4 shows that the EH (24) values obtained on the test set are the lowest for the method employed, indicating that the network has a superior ability to fit data samples with imbalanced classes and distribution differences.

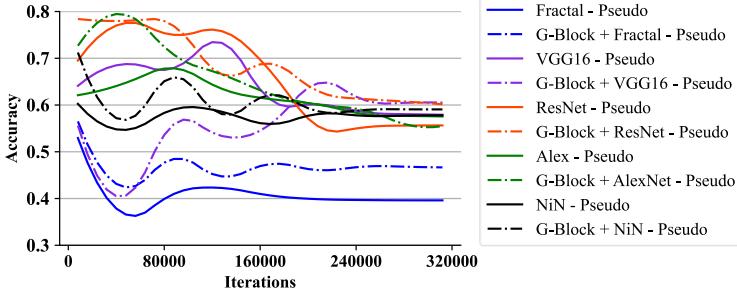
Table 4

Schematic Diagram Comparison of Distribution Difference Imbalance Solutions.

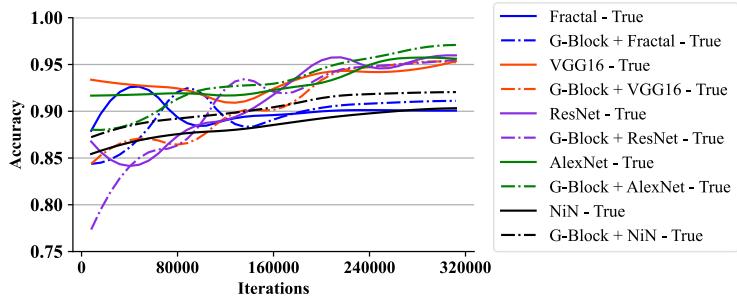
Method	Specificity	Sensitivity	G-Mean	EH (24)
WCE Loss	56.71%	97.58%	0.7438	2.66
Var Focal Loss	47.53%	92.52%	0.6750	2.63
GHM Loss	50.13%	90.98%	0.2160	2.68
Seesaw Loss	48.21%	88.64%	0.6537	2.67
Up Sampling	76.17%	49.23%	0.6123	2.96
Down Sampling	64.37%	50.75%	0.5715	3.21
Move Threshold	56.87%	88.65%	0.7100	2.68
Proposed	58.76%	98.16%	0.7595	2.61

3.7. G-Block ablation experiment

This subsection summarizes the G-Block ablation experiments, where a layer of G-Block with an output hidden layer of 8 was added in front of each classical network, and comparison experiments were conducted. Fig. 13 presents the accuracy of the test set during training. Table 5 presents the final prediction results for the test set and the model size. For VGG16 [42], the size of the model is reduced from 612MB to 228.3MB because the added G-Block comes with a maximum pooling layer. Although the model size is decreased, the true defect recognition rate is improved by 2.48%, with a 0.97% reduction in the pseudo defect recognition rate. For ResNet50 [43], the accuracy of pseudo defects is improved by 4.19%. For AlexNet [44], the pseudo defect recognition rate is decreased by 0.86%, but the model size is increased by 0.04MB after adding the G-Block. Although the recognition accuracy of true defects is decreased by 0.07%, the true defect recognition accuracy is improved by 1.61% after the reduction. For NiN [45], although the model size is increased by 2.96MB after adding the G-Block, the recognition accuracy of true and pseudo defects is improved by 1.73% and 1.34%, respectively. Through this experiment, it is demonstrated that the G-Block proposed in this paper can enhance the classification accuracy of the model by increasing the model's perceptual field.



(A) Classification accuracy curves of test set pseudo-defects in G-Block ablation experiments with the number of iterations.



(B) Classification accuracy curves of test set true-defects in G-Block ablation experiments with the number of iterations.

Fig. 13. Classification accuracy curve of the test set with the number of iterations in the G-Block ablation experiment.

Table 5

Ablation experiments with G-Block.

Network	Specificity	Sensitivity	G-Mean	Param.
Fractal Block*6	39.62%	90.09%	0.5974	38.62MB
G-Block + Fractal Block*6	46.65%	91.12%	0.6519	45.54MB
VGG16	57.86%	95.95%	0.7450	612.21MB
G-Block + VGG16	60.34%	94.98%	0.7569	228.30MB
ResNet50	55.62%	95.74%	0.7297	69.27MB
G-Block + ResNet50	59.81%	95.67%	0.7564	69.31MB
AlexNet	57.45%	95.41%	0.7403	1.12GB
G-Block + AlexNet	56.59%	97.02%	0.7409	290.06MB
NiN	57.74%	90.33%	0.7221	7.83MB
G-Block + NiN	59.08%	92.06%	0.7374	10.79MB

3.8. Sensitivity Analysis of Parameter

In this subsection, the final run of the model under different hyperparameters is primarily presented. In the case of $\lambda_2 = 1.15$ in Eq.(20), λ_1 in Eq.(20) is selected from the fixed set $\lambda_1 \in \{1.5, 5.0, 8.0, 12.0, 14.0, 16.0\}$ for the parameters to perform sensitivity analysis on the model. The final experimental results are listed in Table 6. Table 6 shows that when λ_1 is larger, the sensitivity is larger, whereas the specificity gradually decreases. When $\lambda_1 = 5$, the model obtains the maximum specificity of 67.68%, but the model only has a lower sensitivity of 93.07% at this time. When $\lambda_1 = 16$, although the highest sensitivity of 99.06% is obtained, the model only has a lower specificity of 45.68% at this time. When $\lambda_1 = 14.0$, the model obtains a sensitivity of 98.16% and a specificity of 58.76%. Evidently, the model is more suitable for application in a real production environment. Given that the classification accuracy of both true defects and pseudo defects of this parameter is relatively high, the experimental comparisons in this paper are made at $\lambda_1 = 14.0$. In general, the hyperparameters of the model are then adjusted according to the usage requirements of the actual scenario so that the model can be adaptive to meet realistic usage requirements.

Table 6

Sensitivity experiment detailed indicators.

Hyper-parameters	Specificity	Sensitivity	G-Mean
$\lambda_1 = 1.5$	81.21%	85.10%	0.8716
$\lambda_1 = 5.0$	67.68%	93.07%	0.7936
$\lambda_1 = 8.0$	66.36%	94.80%	0.7931
$\lambda_1 = 12.0$	62.19%	97.26%	0.7778
$\lambda_1 = 14.0$	58.76%	98.16%	0.7595
$\lambda_1 = 16.0$	45.68%	99.06%	0.6726

3.9. Experimental study on receptive field

In this section, the impact of different receptive fields on classification accuracy is observed by displaying the different cases of variables $G_2D - G_5D$ in Eq. (3) under different receptive fields. Table 7 shows that classification accuracy is relatively stable when $n = 1$ and $n = 2$.

Table 7

Accuracy of models under different receptive domains

Hyper-parameters	G_2D	G_3D	G_4D	G_5D	Specificity	Sensitivity	G-Mean
n=1	2	4	3	1	58.76%	98.16%	0.7595
n=2	4	6	5	3	58.24%	97.81%	0.7547
n=3	6	8	7	5	57.16%	95.32%	0.7381
\	1	1	1	1	54.21%	93.44%	0.5065

However, when $n = 3$, the output feature layer of G-Block 3 covers 115.03% of the original image, as shown in Table 2. This suggests that an excessively large receptive field may introduce irrelevant information and impair the final classification accuracy. Conversely, setting all parameters in $G_2D - G_5D$ to 1 results in a significantly smaller receptive field, which also reduces classification accuracy, as seen in the last row of Table 7. Therefore, the model's receptive field size has a considerable impact on the final classification accuracy.

4. Conclusions and future works

Current methods for classifying true- and pseudo-positive PCB defects often overlook the correlation between the local and global data distribution of sample defects and the imbalance of class and distribution differences, despite these being common issues in the PCB manufacturing industry. To address these challenges, BHAFNN is proposed in this paper. The FTN in BHAFNN enhances the model's receptive field through the use of a broadband multi-scale mixed attention mechanism. This mechanism integrates the context-related information from different scales in the sample. The CL loss function calculates the difference in the texture entropy of GLCM between the sample design map and scan map to establish a metric space for sample loss. This allows for the dynamic adjustment of sample loss during model training, leading to improved classification performance. The proposed BHAFNN is compared with recent classification methods for PCB true- and pseudo-positive defects and multiple ablation experiments are conducted. The results show that our model outperforms the existing methods. In the future, it is worth extending the two-category problem to a four-category problem (repairable true defects, non-repairable true defects, pseudo defects, and those requiring manual confirmation). This approach can reduce the loss of PCBs in the production process and significantly decrease the economic expenditure associated with labor, which is crucial for real-world industrial applications.

CRediT authorship contribution statement

Weiming Gan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Lingkai Hu:** Conceptualization, Methodology, Software, Validation, Formal

analysis, Investigation, Data curation, Writing – original draft. **Feng Zhan:** Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Xiaoqing Liu:** Formal analysis, Resources, Writing – review & editing, Supervision, Project administration. **Zhidi Huang:** Formal analysis, Resources, Writing – review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data are available based on request from the first author.

References

- [1] Yongbing Zhou, Minghao Yuan, Jian Zhang, Guofu Ding, and Shengfeng Qin. Review of vision-based defect detection research and its perspectives for printed circuit board. *Journal of Manufacturing Systems*, 70:557–578, 2023.
- [2] Yayun Cheng, Li Zhang, Dalu Guo, Nannan Wang, Jiaran Qi, and Jinghui Qiu. Subregional polarization fusion via stokes parameters in passive millimeter-wave imaging. *IEEE Transactions on Industrial Informatics*, 2024.
- [3] Peisong Ma, Deshang Sha, and Debin Zhang. High efficiency and full range zvs single-stage semi-dab ac-dc converter with improved modulation and variable frequency control. *IEEE Transactions on Power Electronics*, 2024.
- [4] Ronghuan Xie, Yuanchao Wu, Hongmin Tang, Yizhan Zhuang, and Yiming Zhang. A strongly coupled vehicle-to-vehicle wireless charging system for emergency charging purposes with constant-current and constant-voltage charging capabilities. *IEEE Transactions on Power Electronics*, 2024.
- [5] Grace Maxted, Pedro Estrela, and Despina Moschou. Employing electrochemically derived ph gradients for lab-on-pcb protein preconcentration devices. *Microsystems & Nanoengineering*, 10(1):10, 2024.
- [6] Minghao Yuan, Yongbing Zhou, Xiaoyu Ren, Hui Zhi, Jian Zhang, and Haojie Chen. Yolo-hmc: An improved method for pcb surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 2024.
- [7] Lei Lei, Han-Xiong Li, and Hai-Dong Yang. Reliable and lightweight adaptive convolution network for pcb surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 2024.
- [8] Minghao Yang, Yongjia Zhao, Jiaxin Li, Wei Tang, Hang Pan, Jinlong Chen, and Fuchun Sun. Accurate flexible printed circuit (fpc) position identification using haptic information in industrial robotic mobile phone assembly. *IEEE Transactions on Industrial Electronics*, 2024.
- [9] Huan Zhang, Liangxiao Jiang, and Chaoqun Li. Cs-resnet: Cost-sensitive residual convolutional neural network for pcb cosmetic defect detection. *Expert Systems with Applications*, 185:115673, 2021.
- [10] Yilin Miao, Zhewei Liu, Xiangning Wu, and Jie Gao. Cost-sensitive siamese network for pcb defect classification. *Computational Intelligence and Neuroscience*, 2021(1):7550670, 2021.
- [11] Yu-Shan Deng, An-Chun Luo, and Min-Ji Dai. Building an automatic defect verification system using deep neural network for pcb defect classification. In *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pages 145–149. IEEE, 2018.
- [12] Ziheng Xiao, Yu Jiang, Tengfei Sun, Yue Wu, and Yi Tang. Refining power converter loss evaluation: A transfer learning approach. *IEEE Transactions on Power Electronics*, 2024.
- [13] Yihao Xue, Rui Yang, Xiaohan Chen, Weibo Liu, Zidong Wang, and Xiaohui Liu. A review on transferability estimation in deep transfer learning. *IEEE Transactions on Artificial Intelligence*, 2024.
- [14] Xinyi Yu, Li Han-Xiong, and Haidong Yang. Collaborative learning classification model for pcbs defect detection against image and label uncertainty. *IEEE Transactions on Instrumentation and Measurement*, 72:1–8, 2023.
- [15] Longxin Zhang, Jingsheng Chen, Jianguo Chen, Zhicheng Wen, and Xusheng Zhou. Ldd-net: Lightweight printed circuit board defect detection network fusing multi-scale features. *Engineering Applications of Artificial Intelligence*, 129:107628, 2024.
- [16] Danqing Kang, Jianhuang Lai, Junyong Zhu, and Yu Han. An adaptive feature reconstruction network for the precise segmentation of surface defects on printed circuit boards. *Journal of Intelligent Manufacturing*, 34(7):3197–3214, 2023.
- [17] Changlin Chen, Qiman Wu, Jin Zhang, Haojie Xia, Pengrong Lin, Yong Wang, Mengke Tian, and Rencheng Song. U 2 d 2 pcb: Uncertainty-aware unsupervised defect detection on pcb images using reconstructive and discriminative models. *IEEE Transactions on Instrumentation and Measurement*, 2024.
- [18] Arnaud Bougamah, Mohammed El Adoui, Isabelle Linden, and Benoît Frénay. Composite score for anomaly detection in imbalanced real-world industrial dataset. *Machine Learning*, 113(7):4381–4406, 2024.
- [19] Dongxu Bai, Gongfa Li, Du Jiang, Juntong Yun, Bo Tao, Guozhang Jiang, Ying Sun, and Zhaojie Ju. Surface defect detection methods for industrial products with imbalanced samples: A review of progress in the 2020s. *Engineering Applications of Artificial Intelligence*, 130:107697, 2024.
- [20] Dan Shi, Na Sun, Yanchi Liu, Cheng Lian, Xingyu Chen, Xiaoya Zhou, Xiaoyong Liu, Qiwei Liu, and Juejia Zhou. Fast calculation method for crosstalk of high-density printed circuit board by using machine learning. *IEEE Transactions on Electromagnetic Compatibility*, 2024.
- [21] Naifu Yao, Yongqiang Zhao, Seong G Kong, and Yang Guo. Pcb defect detection with self-supervised learning of local image patches. *Measurement*, 222:113611, 2023.

- [22] Ka Chun Fung, Kai-Wen Xue, Cheung-Ming Lai, Kwan-Ho Lin, and Kin-Man Lam. Improving pcb defect detection using selective feature attention and pixel shuffle pyramid. *Results in Engineering*, 21:101992, 2024.
- [23] Niegong Cao, Daehan Won, and Sang Won Yoon. Pads: Predictive anomaly detection for smt solder joints using novel features from spi and pre-aoi data. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2024.
- [24] Mehrnaz Mirzaei, Marzieh Hashemzadeh Sadat, and Farnoosh Naderkhani. Application of machine learning for anomaly detection in printed circuit boards imbalance date set. In *2023 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 128–133. IEEE, 2023.
- [25] Dhruv Makwana, Sparsh Mittal, et al. Pcbsegclassnet—a light-weight network for segmentation and classification of pcb component. *Expert Systems with Applications*, 225:120029, 2023.
- [26] Panke Zhu, Yiwen Cai, Fen Li, Jing Kan, Kewei Chen, and Fangyan Dong. A deep learning method for pcb defect detection based on improved yolov8. In *Proceedings of the International Conference on Computer Vision and Deep Learning*, pages 1–7, 2024.
- [27] Carlos Solorzano and Du-Ming Tsai. Accurate vision-based pcb positioning using cosine-convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 2023.
- [28] PCB-TP Open Source Address, 2024. Available at: https://pan.baidu.com/s/1qUyi0_wH22jSmVwlFwQ0mA (Access code: dikc).
- [29] Nianyin Zeng, Peishu Wu, Zidong Wang, Han Li, Weibo Liu, and Xiaohui Liu. A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Transactions on Instrumentation and Measurement*, 71:1–14, 2022.
- [30] Jun-Gi Jang, Chun Quan, Hyun Dong Lee, and U Kang. Falcon: lightweight and accurate convolution based on depthwise separable convolution. *Knowledge and Information Systems*, 65(5):2225–2249, 2023.
- [31] Jun Wang, Peng Yin, Yuanyun Wang, and Wenhui Yang. Cmat: integrating convolution mixer and self-attention for visual tracking. *IEEE Transactions on Multimedia*, 26:326–338, 2023.
- [32] Roland Gao. Rethink dilated convolution for real-time semantic segmentation. *arXiv preprint arXiv:2111.09957*, 2(3):6, 2021.
- [33] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.
- [34] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.
- [35] Bogdan-Cristian Anghelina and Radu Miculescu. On the localization of hutchinson–barnsley fractals. *Chaos, Solitons & Fractals*, 173:113674, 2023.
- [36] Jaqueline Junko Tenguam, Leonardo H da Costa Longo, Guilherme Freire Roberto, Thaína AA Tosta, Adriano B Silva, Marcelo Zanchetta do Nascimento, and Leandro Alves Neves. Higuchi fractal dimension with a multidimensional approach for color images. *Software Impacts*, 21:100690, 2024.
- [37] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *International Conference on Learning Representations*, 2016.
- [38] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [39] Xiang Li, Chengqi Lv, Wenhui Wang, Gang Li, Lingfeng Yang, and Jian Yang. Generalized focal loss: Towards efficient representation learning for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 45(3):3139–3153, 2022.
- [40] Huanlong Zhang, Jiapeng Zhang, Guohao Nie, Jilin Hu, and WJ Chris Zhang. Residual memory inference network for regression tracking with weighted gradient harmonized loss. *Information Sciences*, 597:105–124, 2022.
- [41] Oscar Moll, Manuel Favela, Samuel Madden, Vijay Gadepally, and Michael Cafarella. Seesaw: interactive ad-hoc search over image databases. *Proceedings of the ACM on Management of Data*, 1(4):1–26, 2023.
- [42] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition.” arxiv, apr. 10, 2015. doi: 10.48550. *arXiv preprint arXiv:1409.1556*, 2023.
- [43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [45] Shuicheng Yan Min Lin, Qiang Chen. Network in network. *International Conference on Learning Representations*, 2013.