

Broadband hybrid attention-based feature fusion network for printed circuit boards defect classification^{★,★★}

Feng Zhan^{a,*}, Weihan Qiu^{a,*}, Weiming Gan^a, Lingkai Hu^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 (PCBs) are integral components in a wide range of industrial applications, such as imaging sensors [1], variable voltage converters [2], wireless charging systems [3], and biomedical devices [4]. The quality of PCBs is crucial for the proper functioning of electronic devices, and as such, effective detection of PCB defects is vital. Measurement-based techniques are at the core of PCB defect detection, particularly in industrial settings where precision and accuracy are paramount. Current methods like automatic optical inspection (AOI) and visual repair systems (VRS) are widely employed, but both face significant challenges in real-world applications. Traditional AOI systems often produce a high volume of pseudo-positive defects, which are misclassified as true-positive defects due to measurement limitations in image recognition [5].

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) [6] | ✗ | ✓ | ✓ | ✗ | ✗ |
| LLD-Net (2024) [7] | ✓ | ✗ | ✗ | ✓ | ✗ |
| YOLO-HMC (2024) [5] | ✓ | ✗ | ✗ | ✓ | ✗ |
| AFRNet (2023) [8] | ✗ | ✗ | ✗ | ✗ | ✗ |
| Proposed | ✓ | ✓ | ✓ | ✓ | ✓ |

The classification of true- and pseudo-positive defects is intrinsically tied to measurement techniques that capture and process the surface texture and contextual features of the scanned images. These measurements are critical for accurately interpreting the condition of a PCB. For instance, a local break in the PCB, which results in circuit disconnection, is a true defect, while surface artifacts like skin flakes or dust on the solder mask may be falsely

*Co-first author.

**Corresponding author.

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

identified as defects but do not impact the PCB's functionality [9]. To improve the accuracy of defect classification, robust measurement techniques must be employed to extract reliable features from the scanned images while filtering out noise and irrelevant information [10]. Recent developments in PCB detection modules have shown promising results [11–15], but they are still constrained by measurement limitations such as restricted perceptual field coverage, ineffective multiscale feature extraction, and challenges in mitigating noise interference. These limitations hinder the ability to accurately classify true-positive and pseudo-positive defects, as highlighted in Table 1.

In industrial production datasets containing true- and pseudo-positive PCB defects, significant challenges arise due to class imbalance [16] and pronounced distribution discrepancies. These issues complicate the accurate classification of defects. Class imbalance, a frequent occurrence, stems from disparities in the representation of pseudo-positive and true-positive samples [17]. Meanwhile, distribution discrepancies result from texture differences between scanned and designed defect maps, leading to model underperformance and non-convergence. To address data imbalance [18–22], strategies such as data sampling [23], data generation [24], and loss function optimization [25] have been explored. However, data sampling often exacerbates model overfitting [5], while traditional data generation methods struggle to produce accurate labels and realistic distributions for synthetic data [26]. In contrast, loss function-based approaches offer enhanced robustness and flexibility for managing imbalanced datasets [27–32].

Recent advancements in computer vision have sought to address the challenges posed by class-distribution imbalances. Loss functions such as those proposed in [33–35] aim to mitigate task complexity by dynamically adjusting the gradients of positive and negative samples in imbalanced datasets. A critical task in industrial applications involves distinguishing between true-positive and pseudo-positive PCB defects. While existing loss functions effectively address sample category imbalances, they fail to account for the coexistence of class-distribution discrepancies and data distribution differences. Specifically, these loss functions are not designed to handle the discrepancies between scanned images and designed maps, which are common in PCB datasets. Consequently, directly applying conventional loss functions to true- and pseudo-positive defect classification tasks results in suboptimal model performance, highlighting the need for more comprehensive solutions.

To address these challenges, this paper proposes a broadband hybrid attention-based feature fusion network (BHAFFN). Central to this framework is the Focus-Texture-Net (FTN), designed to enhance the model's perceptual field, improve contextual multiscale feature extraction, and efficiently suppress irrelevant information. The FTN incorporates a Gather Block (G-Block) as its primary feature extraction module. The parameters of the G-Block are tailored to the perceptual field required by the training samples. It employs a broadband multiscale hybrid attention mechanism (BMHAM) to fuse features across multiple scales, enabling the effective extraction of defect characteristics from a wide range of contextual information. The G-Block maintains a large receptive field with minimal parameter complexity, reducing the risk of information loss associated with high expansion rates. It also effectively suppresses irrelevant background noise, ensuring precise feature extraction. By capturing the correlation between local defect distributions and overall data distributions, the FTN achieves superior classification accuracy compared to existing models.

To further address the problems of unbalanced data and distribution discrepancy, a loss function called clustering loss is proposed and applied in this paper. 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.

This paper contributes to the field of measurement by providing a novel solution to the challenges posed by class-distribution imbalance and distribution discrepancies in PCB defect classification. Our experimental results show that the proposed method outperforms existing state-of-the-art techniques in accurately distinguishing true-positive defects from pseudo-positive ones. This work underscores the importance of measurement-oriented methods in improving the reliability and accuracy of PCB defect classification systems. 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 [36]) 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.

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 [37, 38], 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 clustering loss (CL) loss function. This loss value will be used for backpropagation.

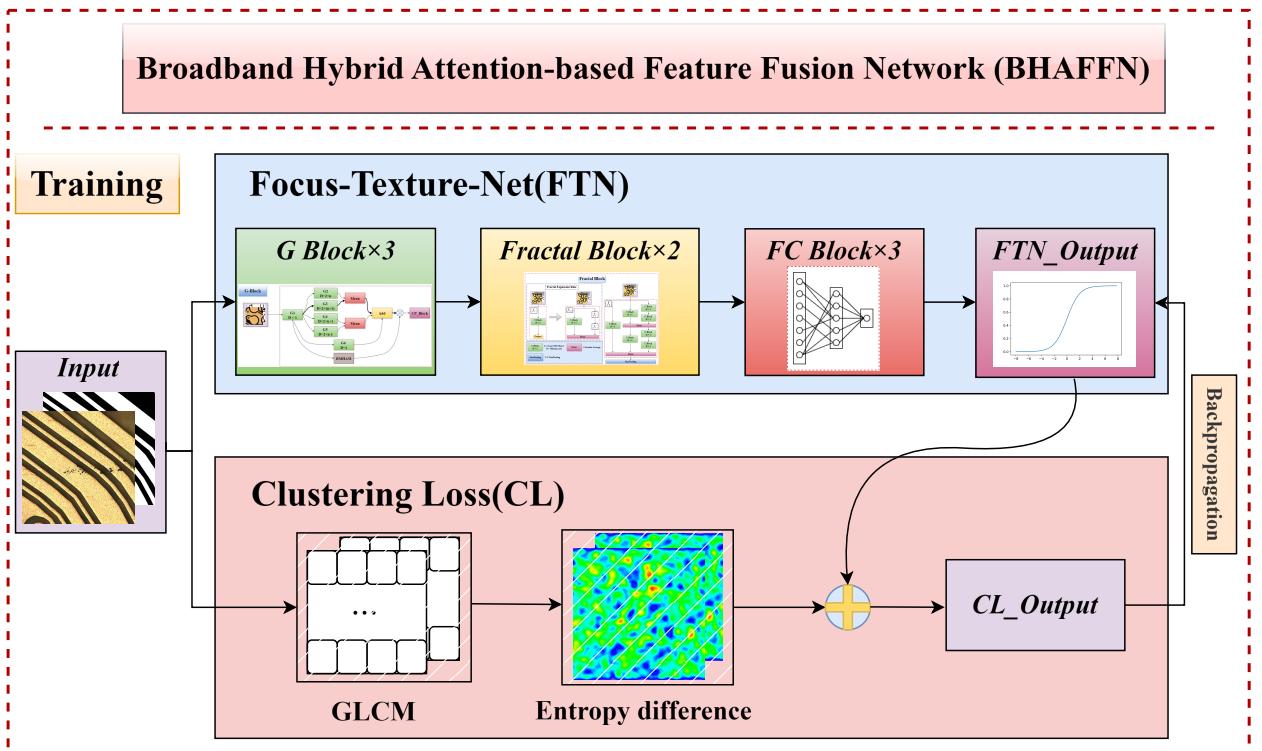


Fig. 1. General structure of the proposed BHAFFN.

2.2. G-Block

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:

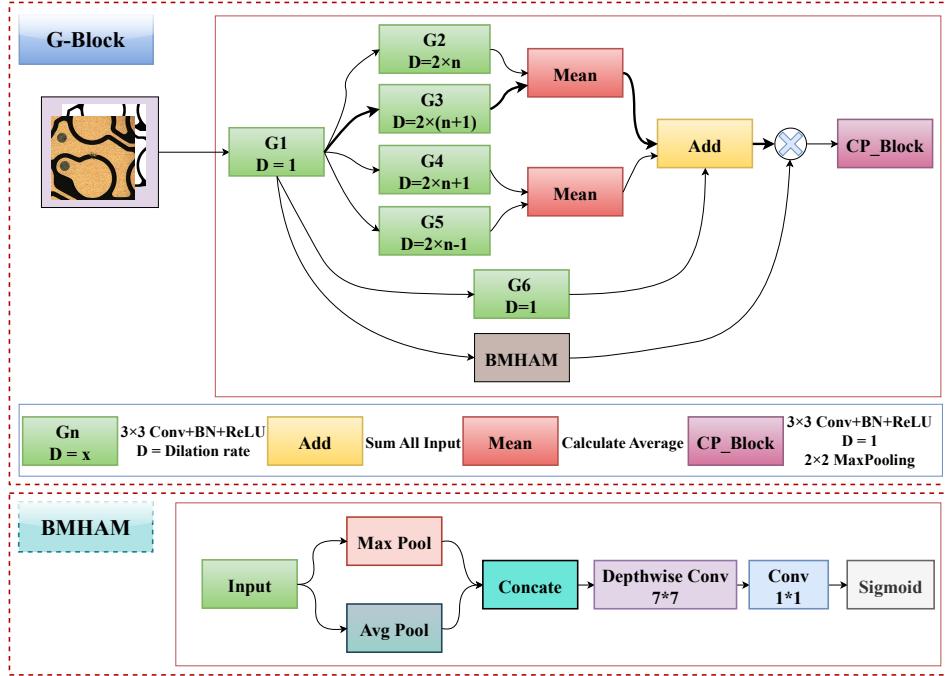


Fig. 2. G-Block: Detailed schematic diagram of the structure.

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

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

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

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 [39] and [40], this paper proposes BMHAM in the G-Block. The BMHAM module enhances feature fusion within the G-Block by integrating max and average pooling in parallel, capturing both discriminative and complementary information for PCB defect classification.

BMHAM leverages depthwise separable convolution to reduce cross-channel and spatial correlations while preserving critical features with minimal computational cost. The 7×7 depthwise convolution improves local-global feature extraction, crucial for detecting defects of varying scales. A subsequent 1×1 convolution refines channel attention and reduces dimensionality, optimizing efficiency without compromising performance. This lightweight design ensures robust and accurate real-time PCB defect detection, making it suitable for industrial deployment.

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

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

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

where $\text{Concat}(\cdot)$ denotes concatenation, and $\text{Depthwise_Conv}(\cdot)$ is a depthwise separable convolution. The final output is:

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

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

The G-Block captures the essential correlation between local and global information within the sample, improving the model's ability to process multi-scale global features. This structure effectively establishes the dependency between localized defect information and the broader characteristics of the PCB.

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 [41]. The concept of the receptive field is important for understanding and diagnosing how deep CNNs work [42, 43]. 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 [41]:

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

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_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

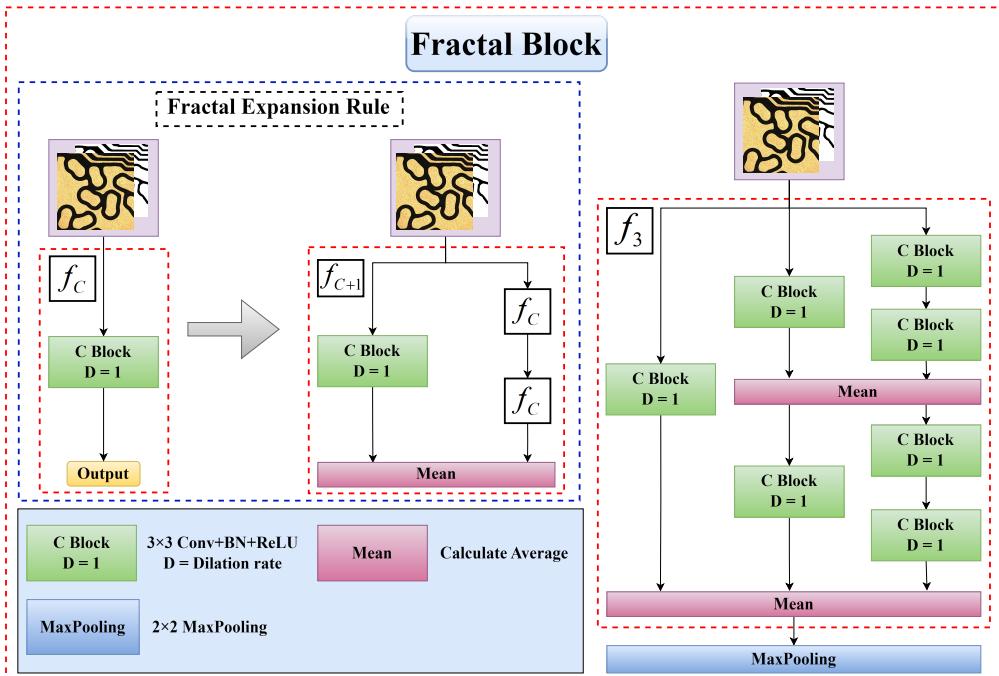


Fig. 3. Fractal Block: Detailed schematic diagram of the structure.

Fractal comes from geometry [44, 45], which analyzes data from multiple levels, perspectives, and constituents. The structure of the Fractal Block is derived from FRACTALNET [46] 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 [45]. 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 $(x + a, y + b)$, with distance d and grayscale value j . The specific mathematical expressions are as follows:

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

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

In this study, L denotes the level of grayscale, with a value of $L = 16$ selected to balance precision and computational efficiency. Previous research has primarily concentrated on scan maps that display visible defects, and many publicly available defect datasets consist solely of these types 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 $\rho_f(i, j)_k$ denotes the k -th grayscale co-occurrence matrix of the scanned image, $\rho_s(i, j)_k$ represents the k -th grayscale co-occurrence matrix of the design image, and C_k is the k -th expected value of the information entropy metric distance between the grayscale co-occurrence matrices $\rho_f(i, j)_k$ and $\rho_s(i, j)_k$ of the real and design images, respectively. 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.

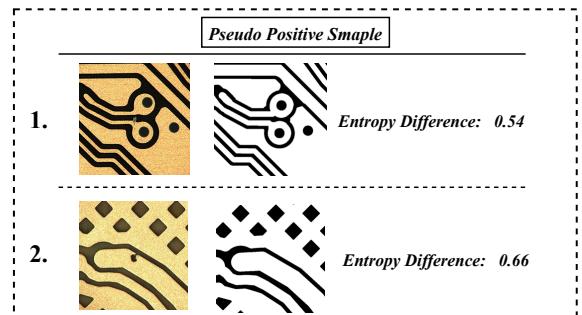
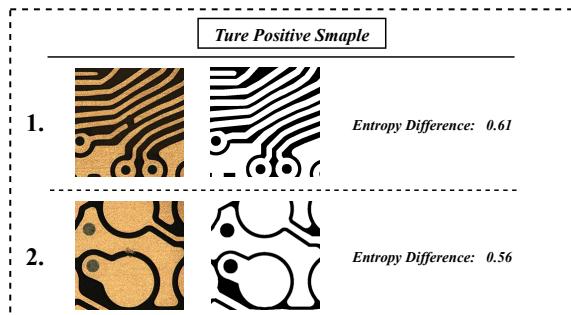


Fig. 4. The entropy difference of true-positive defects samples. **Fig. 5.** The entropy difference of pseudo-positive defects samples.

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:

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

where $L_{CE}(\cdot)$ represents the cross entropy loss. 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 used in neural network training is defined by the final construction equation $CL(\hat{y}, y)$, where λ_1 and λ_2 are hyperparameters. This loss function leverages the texture complexity entropy difference between the sample design map and the scan map to define the loss metric space. By emphasizing the model's ability to fit samples with significant texture differences, it mitigates the training complexity caused by the imbalance in the PCB dataset.

3. Experiments and results

This section presents the datasets used, the experimental design, the configuration of the experimental platform, and the analysis of the results. The experiments were conducted on two datasets: the PCB-TP dataset [36] and the HRIPCB dataset [47]. The PCB-TP dataset contains both true-positive and pseudo-positive defects, while the HRIPCB dataset includes six distinct types of PCB defects.

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 [23]. 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).

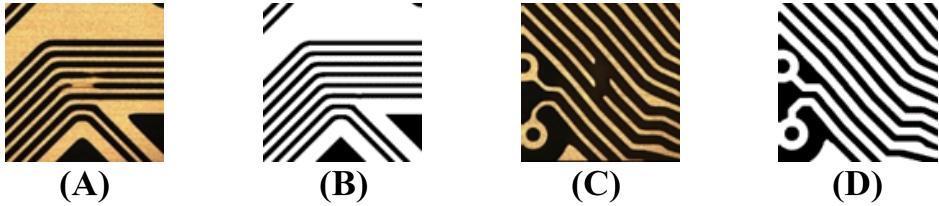


Fig. 6. Presentation of the PCB-TP Dataset.



Fig. 7. Representative examples of true defects from the PCB-TP dataset.

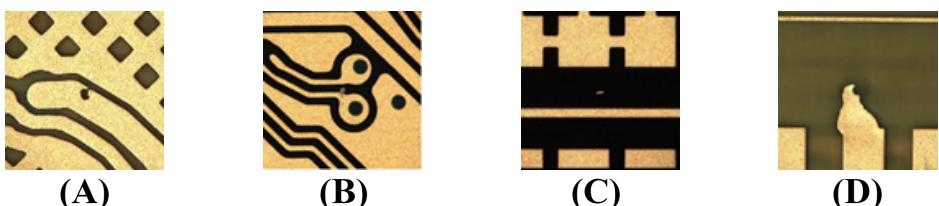


Fig. 8. Representative examples of pseudo defects from the PCB-TP dataset.

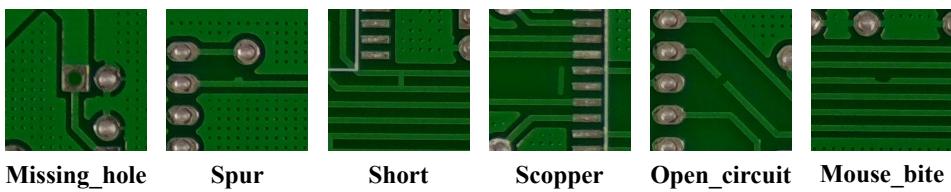


Fig. 9. Overview of the six defect types in the HRIPCB Dataset.

PCB-TP 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.

The HRIPCB dataset is a publicly available large-scale PCB defect classification dataset, which contains 10688 images with a resolution of 4068 \times 3456 pixels each, and includes six types of defects (Fig. 9). These types of defects

include “missing hole (hole),” “open,” “mouse bite (bite),” “spur,” “short,” and “spurious copper (scopper).” Due to the multi-classification task conducted on the HRIPCB dataset, the loss function has been substituted with softmax cross-entropy.

In this paper, the datasets are partitioned into training and test sets through random sampling. Specifically, 80% of the data is allocated to the training set, while the remaining 20% is used for testing.

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 [12–15, 48].

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:

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

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

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%, \quad (23)$$

$$G - \text{mean} = \sqrt{\text{Sensitivity} * \text{Specificity}}. \quad (24)$$

Among them, TP 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:

$$EH = \frac{\sum C_k}{\text{len}(C_k)}, k \in (y = 1, \hat{y} = 0), \quad (25)$$

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

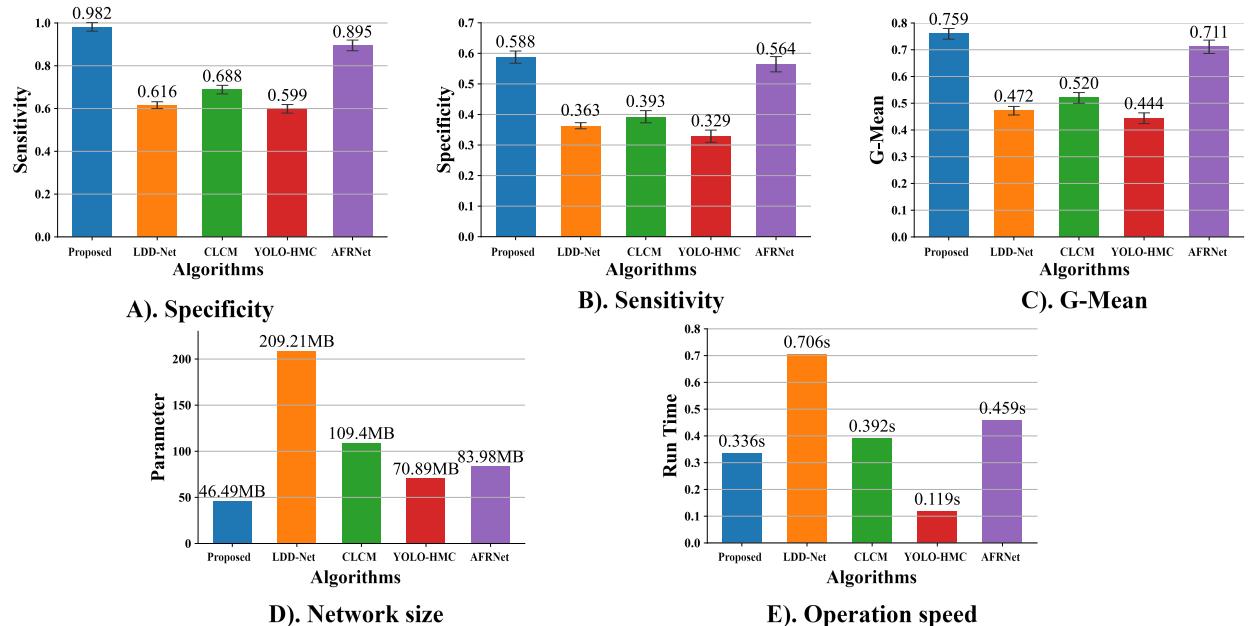


Fig. 10. Comparison of Evaluation Metrics of BHAFFN, LDD-Net, CLCM, YOLO-HMC, and AFRNet.

The experimental results comparing the most advanced models on the PCB-TP and HRIPCB datasets in recent years are shown in Fig. 10. To ensure the reproducibility and robustness of the findings, a 5-fold cross-validation approach was employed in all experiments. Each experiment used the same training and test sets. The precision metrics and their confidence intervals for the PCB-TP dataset are presented in Fig. 10 (A)-(C). The results demonstrate that the proposed BHAFFN outperforms all other models across all metrics. Regarding network size, the proposed model is the smallest, with a size of 46.49 MB, while achieving better sensitivity, specificity, and G-mean evaluation metrics than the four other models, as shown in Fig. 10 (D). Additionally, 100 random samples were selected from the test set, and the average computational speed for each sample was calculated. As illustrated in Fig. 10 (E), although YOLO-HMC [5] exhibits a faster computing speed, its evaluation metrics are inferior to those of other methods. This is attributed to the CARAFE module in YOLO-HMC, which improves computational speed but limits its ability to adapt to defects of varying sizes [49].

The proposed BHAFFN model achieves outstanding performance on the HRIPCB dataset with the highest accuracy of 99.71%, surpassing CLCM (99.53%) and significantly outperforming other models like LDD-Net (97.35%) and YOLO-HMC (96.13%), as presented in Table 4. It excels in precision, achieving 100.00% for defects such as "Hole," "Spur," and "Scopper," while maintaining high sensitivity, including 100.00% for the same categories and 99.53% for "Bite," outperforming AFRNet's 94.85%.

This balanced performance in precision and sensitivity demonstrates BHAFFN's ability to deliver reliable defect detection. Its consistent 100.00% precision and sensitivity in key defect types underscores its robustness and practicality for high-accuracy industrial applications, outperforming competitors by minimizing false positives and false negatives.

Table 4

Performance of five models on HRIPCB dataset.

| Methods | Accuracy | Precision | | | | | Sensitivity | | | | |
|-----------------|---------------|----------------|----------------|----------------|---------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | | Hole | Open | Bite | Spur | Short | Scopper | Hole | Open | Bite | Spur |
| LDD-Net | 97.35% | 96.68% | 98.49% | 96.77% | 97.34% | 96.93% | 97.87% | 97.64% | 98.83% | 97.77% | 98.31% |
| CLCM | <u>99.53%</u> | <u>99.47%</u> | 100.00% | <u>98.32%</u> | <u>99.40%</u> | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% | 98.14% |
| YOLO-HMC | 96.13% | 95.44% | 96.32% | 94.45% | 95.22% | 97.30% | 98.04% | 97.68% | 96.47% | 97.74% | 95.54% |
| AFRNet | 98.84% | 98.32% | 100.00% | 96.49% | 98.57% | 100.00% | 99.64% | 98.37% | 99.55% | 94.85% | 96.34% |
| Proposed | 99.71% | 100.00% | 98.72% | 100.00% | 99.55% | 100.00% | 100.00% | 100.00% | 99.60% | <u>99.53%</u> | <u>99.15%</u> |
| | | | | | | | | | | | 100.00% |

The preliminary comparison results of the above experiments suggest that BHAFFN surpasses other models in various aspects. To further evaluate the qualitative performance of BHAFFN, Grad-CAM [50] is employed to visualize the feature maps of different models on the test images, as shown in Fig. 11 to 13. 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. 11 to Fig. 13 (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. 13 (C), it can be observed that LDD-Net suppresses local defect information as background or noise information. Fig 11-13. (A) represents input image, (B) represents a proposed method, (C) represents LDD-Net, (D) represents CLCM, (E) represents YOLO-HMC, and (F) represents AFRNet.

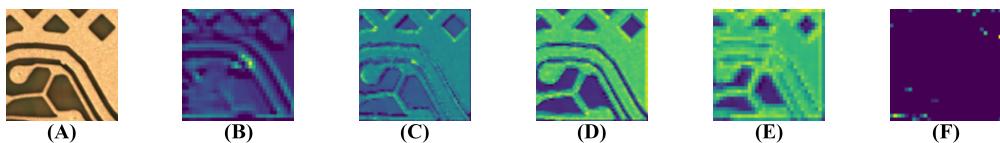


Fig. 11. Grad-CAM of sample 1.

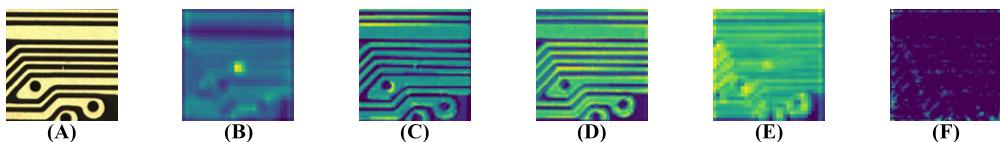


Fig. 12. Grad-CAM of sample 2.

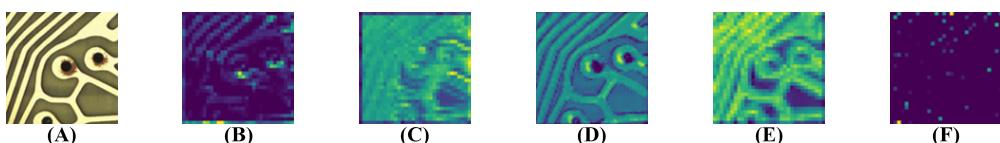


Fig. 13. Grad-CAM of sample 3.

3.6. Imbalanced data processing

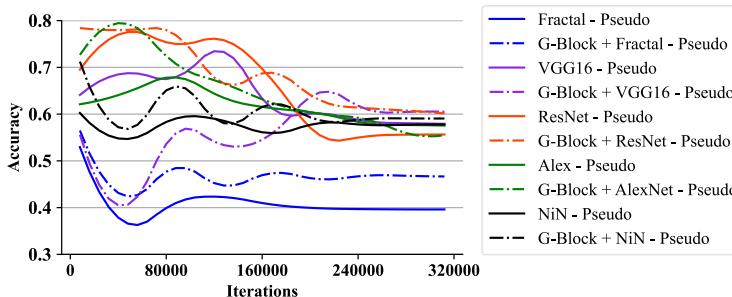
The final experimental results of Weighted Cross-Entropy (WCE) Loss are marginally lower than CL, whereas Var Focal Loss [51], Gradient Harmonizing Mechanism Loss (GHM Loss) [52], and Seesaw Loss [53] of the target detection algorithm exhibit lower performance than CL on this dataset. Although up-sampling 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 5 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 [54–56].

Table 5

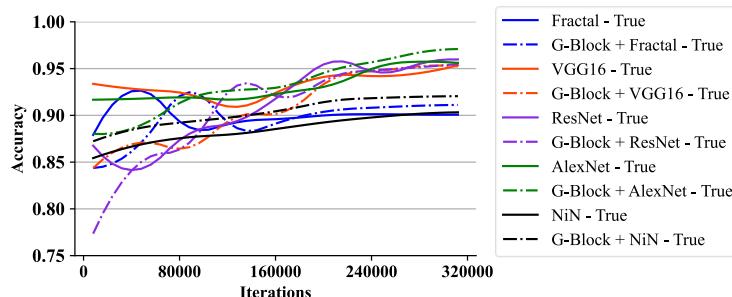
Schematic Diagram Comparison of Distribution Difference Imbalance Solutions.

| 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



(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. 14. Classification accuracy curve of the test set with the number of iterations in the 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. 14 presents the accuracy of the test set during training. Table 6 presents the final prediction results for the test set and the model

size. For VGG16 [57], 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 [58], the accuracy of pseudo defects is improved by 4.19%. For AlexNet [59], 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 [60], 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.

Table 6

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

This subsection presents the model's performance under different hyperparameter settings. Based on prior research on cost-sensitive learning [1, 48, 61], the parameters λ_1 and λ_2 in Eq.(20) regulate the balance between sensitivity and specificity. With λ_2 fixed at 1.15, the values for λ_1 are selected from the set $\lambda_1 \in \{1.5, 5.0, 8.0, 12.0, 14.0, 16.0\}$. This selection is based on preliminary validation experiments, aligning with both theoretical recommendations [48] and practical observations in industrial defect detection. Table 7 summarizes the experimental results. As λ_1 increases, sensitivity improves while specificity declines. When $\lambda_1 = 5$, specificity reaches its peak at 67.68%, though sensitivity remains lower at 93.07%. At $\lambda_1 = 16$, the highest sensitivity of 99.06% is achieved, but specificity drops to 45.68%. The configuration $\lambda_1 = 14.0$ yields a sensitivity of 98.16% and a specificity of 58.76%, providing a balanced trade-off suitable for real-world applications. Given its ability to effectively classify both true and pseudo defects, this setting is used for subsequent experimental comparisons. Overall, hyperparameters should be adjusted based on the specific requirements of practical scenarios, ensuring the model adapts to different operational needs efficiently.

Table 7

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 |

The hyperparameters were manually adjusted based on empirical results and domain knowledge. Automated optimization techniques, such as Bayesian optimization or genetic algorithms, were not employed in this study. Future work will explore the use of such methods to further optimize model performance and assess whether automated tuning can achieve improved results across a broader range of hyperparameter settings.

3.9. Structure-parameter ablation experiment

To verify the optimality of the combination of G-Blocks, Fractal blocks, and FC layers in the network structure, we conducted systematic ablation experiments. The aim of the experiments was to evaluate the impact of different quantity combinations on the performance of PCB defect classification, with specific metrics including Specificity, Sensitivity, and G-Mean. The experimental results are presented in Table 8. In the baseline model, the numbers of G-Blocks, Fractal blocks, and FC layers are set to 3, 2, and 3, respectively. The model achieves a specificity of 58.76%, a sensitivity of 98.16%, and a G-Mean of 0.7595.

By comparing the performance of different quantity combinations, the result shows that the number of G blocks has the most significant impact on model performance, while the quantity of Fractal blocks must be chosen carefully. The increase in the number of FC layers has a limited effect on model performance, suggesting that the quantity of FC layers is not a primary influencing factor for this task. These experiments validate the specific impact of the combination of structural parameters on PCB defect classification performance, providing important insights for subsequent model optimization.

Table 8

Sensitivity experiment detailed indicators.

| Structure-parameters | Specificity | Sensitivity | G-Mean |
|----------------------|-------------|-------------|--------|
| Baseline Model | 58.76% | 98.16% | 0.7595 |
| G-Blocks=1 | 48.29% | 90.44% | 0.6609 |
| G-Blocks=2 | 55.14% | 95.33% | 0.7250 |
| G-Blocks=4 | 58.38% | 97.23% | 0.7527 |
| Fractal blocks=1 | 56.82% | 98.03% | 0.7463 |
| Fractal blocks=3 | 57.20% | 98.22% | 0.7495 |
| Fractal blocks=4 | 53.52% | 99.06% | 0.7281 |
| FC Layer =1 | 58.45% | 97.90% | 0.7565 |
| FC Layer =2 | 58.60% | 97.78% | 0.7570 |
| FC Layer =4 | 58.61% | 98.04% | 0.7580 |

3.10. 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 9 shows that classification accuracy is relatively stable when $n = 1$ and $n = 2$.

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 9. Therefore, the model's receptive field size has a considerable impact on the final classification accuracy.

Table 9

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 |

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, BHAFFN is proposed in this paper. The FTN in BHAFFN 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 BHAFFN 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.

In future work, there is significant potential to extend this research to include the study of defects in other types of PCBs, such as flexible PCBs or IC packaging substrates. These types of boards have their own unique failure modes, which can often involve intricate internal faults that are difficult to detect using traditional methods. Future work will focus on adapting the proposed BHAFFN model to better handle the measurement complexities specific to these types of PCBs. Moreover, further research could explore the incorporation of multimodal measurement data, such as thermal imaging or X-ray scans, to improve defect detection accuracy, particularly for internal or hidden defects that are not visible through traditional surface inspection techniques [62]. Incorporating advanced measurement tools and data fusion techniques will help improve the robustness and generalizability of defect classification systems across various PCB types, enhancing both their performance and applicability in industrial environments.

Furthermore, to balance computational efficiency and accuracy, lightweight model optimization techniques, including pruning, quantization, and knowledge distillation, can effectively reduce computational overhead without compromising performance. To further enhance real-time applicability, hardware acceleration strategies such as GPU and FPGA implementations will be explored. These approaches can significantly improve inference speed, making the model more suitable for industrial deployment. By integrating efficient algorithmic refinements with hardware optimizations, future advancements aim to achieve a balance between computational cost and predictive accuracy, ensuring robust and scalable defect classification in practical applications.

CRediT authorship contribution statement

Feng Zhan: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Weihan Qiu:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Weiming Gan:** Conceptualization, Methodology, Validation, Data curation, Writing – original draft. **Lingkai Hu:** Methodology, Validation, Formal analysis, 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] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Fei Wang, Yihao Zhou, Xuan Zhang, Zhijie Li, Jiejin Weng, Guiyan Qiang, Mingjun Chen, Yang Wang, Honghao Yue, and Junyan Liu. Laser-induced thermography: An effective detection approach for multiple-type defects of printed circuit boards (pcbs) multilayer complex structure. *Measurement*, 206:112307, 2023.
- [10] Baoquan Hu, Jun Liu, and Yue Xu. A novel multi-scale convolutional neural network incorporating multiple attention mechanisms for bearing fault diagnosis. *Measurement*, 242:115927, 2025.
- [11] Feng Zhan, Lingkai Hu, Wenkai Huang, Yikai Dong, Hao He, and Guanjun Wu. Category knowledge-guided few-shot bearing fault diagnosis. *Engineering Applications of Artificial Intelligence*, 139:109489, 2025.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] Arnaud Bougaham, 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.
- [18] 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.
- [19] Weiwei Qian and Shunming Li. A novel class imbalance-robust network for bearing fault diagnosis utilizing raw vibration signals. *Measurement*, 156:107567, 2020.
- [20] Jingjing Zhang, Chuanzhi Su, Mengjie Huang, Liwen Liang, and Rui Yang. Impact of physical tool designs on user embodiment of tools in virtual reality. In *2023 9th International Conference on Virtual Reality (ICVR)*, pages 233–238. IEEE, 2023.
- [21] Jingjing Zhang, Mengjie Huang, Rui Yang, Yiqi Wang, Xiaohang Tang, Ji Han, and Hai-Ning Liang. Understanding the effects of hand design on embodiment in virtual reality. *AI EDAM*, 37:e10, 2023.
- [22] Jingjing Zhang, Mengjie Huang, Kai-Lun Liao, and Rui Yang. Exploring the effect of virtual hand realism on embodiment by subjective ratings and eye tracking. In *2022 1st IEEE International Conference on Cognitive Aspects of Virtual Reality (CVR)*, pages 000017–000020. IEEE, 2022.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] Yijin Liu, Zipeng Li, Jinglong Chen, Tianci Zhang, Tongyang Pan, and Shuilong He. A batch-adapted cost-sensitive contrastive learning network for industrial diagnosis with extremely imbalanced data. *Measurement*, page 116478, 2024.
- [29] Yihao Xue, Rui Yang, Xiaohan Chen, Zhongbei Tian, and Zidong Wang. A novel local binary temporal convolutional neural network for bearing fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 2023.

- [30] Yihao Xue, Rui Yang, Xiaohan Chen, Baoye Song, and Zidong Wang. Separable convolutional network-based fault diagnosis for high-speed train: A gossip strategy-based optimization approach. *IEEE Transactions on Industrial Informatics*, 2024.
- [31] Jingjing Zhang, Mengjie Huang, Yonglin Chen, Kai-Lun Liao, Jiajia Shi, Hai-Ning Liang, and Rui Yang. Touchmark: Partial tactile feedback design for upper limb rehabilitation in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 2024.
- [32] Jingjing Zhang, Mengjie Huang, Yonglin Chen, Xiaoyi Xue, Kai-Lun Liao, Rui Yang, and Jiajia Shi. Tacpoint: Influence of partial visuo-tactile feedback on sense of embodiment in virtual reality. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 943–944. IEEE, 2024.
- [33] 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.
- [34] 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.
- [35] Carlos Solorzano and Du-Ming Tsai. Accurate vision-based pcb positioning using cosine-convolutional neural networks. *IEEE Transactions on Automation Science and Engineering*, 2023.
- [36] PCB-TP Open Source Address, 2024. Available at: https://pan.baidu.com/s/1qUyi0_wH22jSmVwlFwQ0mA (Access code: dikc).
- [37] 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.
- [38] Lingkai Hu, Feng Zhan, Wenkai Huang, Weiming Gan, Haoxiang Hu, Hao He, and Kunbo Han. Weird-net: Weighted relative distance attention for efficient and robust sequence processing. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2024.
- [39] 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.
- [40] 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.
- [41] Roland Gao. Rethink dilated convolution for real-time semantic segmentation. *arXiv preprint arXiv:2111.09957*, 2(3):6, 2021.
- [42] 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.
- [43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.
- [44] Bogdan-Cristian Anghelina and Radu Miculescu. On the localization of hutchinson–barnsley fractals. *Chaos, Solitons & Fractals*, 173:113674, 2023.
- [45] Jaqueline Junko Tenguam, Leonardo H da Costa Longo, Guilherme Freire Roberto, Tháína 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.
- [46] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *International Conference on Learning Representations*, 2016.
- [47] Weibo Huang, Peng Wei, Manhua Zhang, and Hong Liu. Hripcb: a challenging dataset for pcb defects detection and classification. *The Journal of Engineering*, 2020(13):303–309, 2020.
- [48] 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.
- [49] Zuo Zuo, Jiahao Dong, Yue Gao, and Zongze Wu. Hyperdefect-yolo: Enhance yolo with hypergraph computation for industrial defect detection. *arXiv preprint arXiv:2412.03969*, 2024.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] Tian-Yi Jiang, Zhi-Yang Liu, and Guang-Zhu Zhang. Yolov5s-road: Road surface defect detection under engineering environments based on cnn-transformer and adaptively spatial feature fusion. *Measurement*, 242:115990, 2025.
- [55] Xujie He, Jing Jin, Fujiang Yu, She Zhao, Duo Chen, and Xiang Gao. Msidetector: Detecting multi-scenario industrial defects using an adapted visual foundation model and dual thresholding discriminator. *Measurement*, 242:115753, 2025.
- [56] Fangfang Liang, Zhaoongyang Wang, Wei Ma, Bo Liu, Qing En, Dong Wang, and Lijuan Duan. Hdfa-net: A high-dimensional decoupled frequency attention network for steel surface defect detection. *Measurement*, 242:116255, 2025.
- [57] 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.
- [58] 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.
- [59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [60] Shuicheng Yan Min Lin, Qiang Chen. Network in network. *International Conference on Learning Representations*, 2013.

- [61] Jingjing Tang, Yan Li, Zhaojie Hou, Saiji Fu, and Yingjie Tian. Robust two-stage instance-level cost-sensitive learning method for class imbalance problem. *Knowledge-Based Systems*, 300:112143, 2024.
- [62] Yuer GAO, Tongqing XU, Zhongyuan Liao, and Yi CAI. Autonomous navigation algorithm for underwater robots based on global-local fusion under multi-objective optimization. August 2024.