Expert Systems With Applications 246 (2024) 123199

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A real-time anchor-free defect detector with global and local feature enhancement for surface defect detection

Qing Liu [a], Min Liu [a,*], Q.M. Jonathan [b], Weiming Shen [c]

[a] School of Electronics and Information Engineering, Tongji University, Shanghai, 201800, China
[b] Department of Electrical and Computer Engineering, University of Windsor, Windsor, N9B 3P4, ON, Canada
[c] The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China

ARTICLE INFO

Keywords:
Industrial surface defect detection
Real-time anchor-free defect detector
Global feature enhancement module
Local feature enhancement module
Box refinement module

ABSTRACT

Industrial surface defect detection (ISDD) is vital for manufacturing enterprises to control product quality. Many general object detection (GOD) methods are utilized in this field. However, they rarely take into full account the characteristics of industrial defects. We identify three crucial characteristics in ISDD: complex background, small size defect and irregular shape. To copy with it, in this paper, we proposed a novel real-time anchor-free defect detector for ISDD. Firstly, to reduce noise interfere from complex background, we proposed global feature enhancement module (GFEM) to enhance high-level feature's ability in modeling global information so that background noises are alleviated. Secondly, to enhance small size defect's feature, we introduced local feature enhancement module (LFEM). It enhances small size defect's feature by amplifying local peaks in low-level features. Thirdly, we introduced box refinement module (BRM) to capture defect's shape information to provide more accurate prediction result. Lastly, we evaluated the proposed defect detector's effectiveness using three public ISDD datasets. The experimental results are promising: our detector achieves a mAP of 92.0% on PVEL_AD, 99.6% on the PCB defect dataset, and 81.6% on NEU-DET. These scores outperform state-of-the-art methods, proving the superiority of our proposed detector. Additionally, it reached 46.1 FPS on the PVEL_AD dataset, showing its capability for real-time detection.

1. Introduction

With the rapid development of intelligent manufacturing (Liu et al., 2022), manufacturing enterprises have been improved equipment productivity and reliability significantly, however, product quality monitoring is still challenging for manufacturing enterprises, because it is easily affected by the quality of raw material, working conditions and existing technology during production process. Among all defects, surface defects are the most intuitive form that affects product quality (Singh & Desai, 2023). Consequently, it is necessary to detect surface defect so as to ensure product quality.

Defect detection is fundamentally an object detection task. Over the past decade, with the rapid advancement of convolutional neural network (CNN) (He, Zhang, Ren, & Sun, 2016; Simonyan & Zisserman, 2014; Tan & Le, 2019), object detection methods have witnessed significant breakthroughs. Object detection methods can be divided into two-stage and one-stage based on their processing steps. Two-stage methods represented by R-CNN series first generates candidate regions and then classifies them, usually with higher accuracy but slower speed (Girshick, 2015; Girshick, Donahue, Darrell, & Malik, 2014; Ren, He, Girshick, & Sun, 2015). One-stage methods, represented by YOLO series (Bochkovskiy, Wang, & Liao, 2020; Redmon, Divvala, Girshick, & Farhadi, 2016; Wang, Bochkovskiy, & Liao, 2023), SSD (Liu et al., 2016) and RetinaNet (Lin, Goyal, Girshick, He, & Dollár, 2017), directly predict an object's class and location without generating region proposals beforehand, which is faster than two-stage methods because of eliminating the additional computation time required for proposal generation. Early object detection methods utilized anchor-based designs with specific predefined shapes and sizes for position predictions. In pursuit of simplicity and adaptability, researchers transitioned to anchor-free methods (Duan et al., 2019; Tian, Shen, Chen, & He, 2019). To optimize the allocation strategy of positive and negative samples during the training process, some researchers have introduced methods like ATSS (Zhang, Chi, Yao, Lei, & Li, 2020), AutoAssign (Zhu et al., 2020) and DW (Li, He, Li, & Zhang, 2022).

However, these general object detection (GOD) methods are designed for datasets like COCO and VOC (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010; Lin et al., 2014), which might not be ideal for industrial defects due to the difference between them and

* Corresponding author.
E-mail addresses: 201042@tongji.edu.cn (Q. Liu), lmin@tongji.edu.cn (M. Liu), jwu@uwindsor.ca (Q.M. Jonathan), wshen@ieee.org (W. Shen).

https://doi.org/10.1016/j.eswa.2024.123199
Received 23 September 2023; Received in revised form 25 December 2023; Accepted 8 January 2024
Available online 13 January 2024

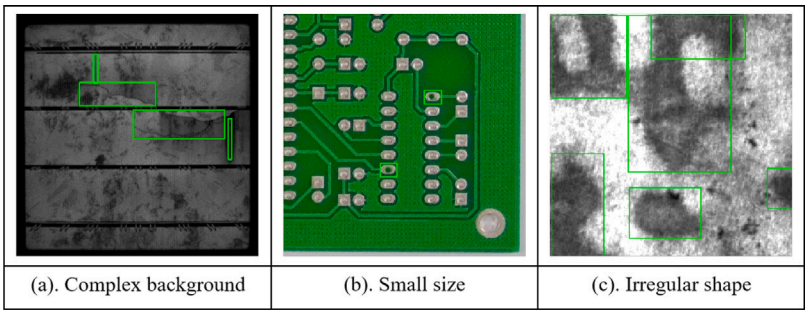

Fig. 1. Examples of surface detects in industrial scenes. (a) defects in photovoltaic cell with complex background. (b) defects in PCB panel with small size. (c) defects in hot-rolled steel strip with irregular shape.

natural scenes. As illustrated in Fig. 1, industrial defects exhibit several distinct characteristics when compared to natural scenes.

(1) **Complex background.** Industrial defects appear against a complex background filled with irrelevant noise. As illustrated in Fig. 1(a), the boundary between the defect and the background is subtle, leading to detection results easily influenced by the background.
(2) **Small size.** Industrial defects manifest in small sizes. As depicted in Fig. 1(b), a small defect covers fewer pixels, resulting in insufficient and weak representative features for detection.
(3) **Irregular shape.** Industrial defect presents irregular shape. As shown in Fig. 1(c), defect with irregular shape contains rich geometric information while it is easily ignored by general object detection.

By combining GOD framework with the characteristics of industrial defects, many methods have been proposed for surface defect detection (Chen et al., 2023; Huang, Wang, Li, & Zou, 2021; Lu & Huang, 2023; Zhi et al., 2023). Literature (Liang & Sun, 2022; Wang et al., 2022; Yu, Lyu, Zhou, Wang, & Xu, 2022) proposed detection methods tailed for small size defect while literature (Lu, Fang, Qiu, & Xu, 2022) proposed detection method for complex background. While existing methods have achieved promising results, they mostly focus on certain aspects of industrial defect characteristics. To the best of our knowledge, there has no research that fully takes into account the aforementioned characteristics of industrial defects. This implies that there is still room for improvement in detection performance by comprehensively considering the characteristics of industrial defects.

To address the identified research gap, we focus on studying detection methods based on the GOD framework, taking into account the comprehensive characteristics of industrial defects. A GOD framework typically comprises three components: the backbone extracts foundational features, the neck refines and enhance them, and the head makes final predictions based on the processed features. In the features extracted by the backbone, high-level features represent semantic information, while low-level features highlight detailed information. We believe that suppressing background noise is of primary importance. The key lies in understanding the entire image through the semantic information provided by high-level features, which in turn offers a clear context to guide the low-level features in their suppression of background noise. However, CNN has a limitation in modeling global information due to its local receptive fields and struggles to take all the contextual information into consideration (Wang, Girshick, Gupta, & He, 2018). To this end, we firstly propose a Global Feature Enhancement Module (GFEM) to strengthen the high-level features' understanding of the entire image. Next, We introduced a Local Feature Enhancement Module (LFEM) to amplify the features of small-sized defects distributed in the low-level layers. Lastly, we introduced a Box Refinement Module (BRM) to capture defect's shape information for more accurate prediction results. In this paper, we proposed a novel real-time anchor-free detector for surface defect detection. Specifically, within the backbone, we introduced the GFEM to enhance the high-level features' ability in modeling global information, thereby mitigating interference from background noise. In the neck, we introduced the LFEM to enhance the low-level features' local peaks associated with small-sized defects, thereby rendering the features of small-sized defects more prominent. In the head, we introduced BRM to make prediction result aware of defect's shape. Finally, the proposed detector is trained using the DW loss function. The proposed detector was tested on three public industrial surface defect datasets. The experimental results demonstrate its effectiveness and superiority over state-of-the-art methods. The main contributions of this paper are concluded as follows:

(1) We proposed a novel real-time anchor-free defect detector for industrial surface defects. Specifically, to address challenges of complex background, small size, and irregular shape in industrial defects, we introduced GFEM, LFEM, and BRM in the detector, each tailored for these challenges respectively.
(2) We proposed a plug-and-play GFEM, which is a channel-level non-local operation. It treats channels and their relationships as a fully connected graph and executes graph channel reasoning to enhance higher-level features' understanding of global information, thereby improve ability in distinguishing background noise and defects.
(3) We evaluate our proposed detector on three different types of public industrial surface defect datasets. Experimental results indicate that the proposed detector achieves real-time detection performance and outperforms state-of-the-art methods across various metrics.

The rest of this paper is arranged as follows: Section 2 reviews the related work of surface defect detection and global information modeling. In Section 3, the proposed detector is described in detail, which mainly includes GFEM, LFEM, BRM and DW training loss. In Section 4, the proposed detector is validated and experimental results are analyzed. Finally, Section 5 presents the conclusion and the future work.

2. Related work

This section offers a brief review of surface defect detection and global information modeling.

2.1. Surface defect detection

Many surface detect detection methods have been proposed using GOD framework. Some scholars proposed detection methods for Steel Surface (Cheng & Yu, 2020; He, Song, Meng, & Yan, 2019; Yu, Cheng, & Li, 2021). He et al. (2019) proposed an end-to-end steel surface defect detection method where multiple hierarchical features are fused to improve detection performance. Huang et al. (2021) proposed

Tri-DFPN for detecting the defects on the texture surface of plastic relays and achieved remarkable improvement. However,it is a two-stage detection method and cannot achieve real-time detection. Chen et al. (2023) proposed a lightweight YOLO-ADPAM detection method for thin film transistor-liquid crystal display (TFT-LCD) panels where a designed parallel attention module is utilized to refine feature maps. However, its anchor-based design is complex and struggles with irregular shapes. Lu and Huang (2023) proposed a texture-aware one-stage detection network for fabric defect detection where an adaptive feature fusion module and a multi-task defect detection head are designed for improving its performance. Some scholars proposed detection method for small size defect (Liang & Sun, 2022; Wang et al., 2022; Yu et al., 2022). Yu et al. (2022) proposed ES-Net for small size defect where the aggregated feature guidance module and dynamic scale-aware head are utilized to improve defection performance on small size defect. Liang (Liang & Sun, 2022) et al. proposed ELCNN for small defect detection of magnetic tile. Apart from small size detection, Lu et al. (2022) proposed an anchor-free defect detector for complex background where an enhanced feature pyramid network is designed to reduce background noise interfere. In addition to CNN-based detection methods, some scholars proposed transformer-based detection methods (Gao, Zhang, Yang, & Zhou, 2022; Liu, Lin, et al., 2021; Lu et al., 2022). However, these methods require a substantial amount of computation, leading to poor detection efficiency.

Previous works have entailed only certain characteristics of industrial defects. In this paper, different from previous work, we provided comprehensive insight into the characteristics of industrial defects and proposed a novel real-time defect detector.

2.2. Global information modeling

Global information modeling can capture rich semantic information in an image and thus effectively distinguish between the background and objects. Global information modeling methods can be divided into attention-based methods (Hou, Zhou, & Feng, 2021; Liu, Shao, & Hoffmann, 2021; Zhang & Yang, 2021) and non-local(NL) operation (Song, Li, Guo, & Huang, 2023; Wang et al., 2018; Yin et al., 2020) based on their processing mechanisms. Attention-based methods, such as SE and CBAM use global average pooling (GAP) or global max pooling (GMP) to model global information. Through excitation, they generate attention weights that guide the reconstruction of feature maps to emphasize important contents in the image (Hu, Shen, & Sun, 2018; Woo, Park, Lee, & Kweon, 2018). Unlike attention-based methods that primarily weigh local or channel-wise information, the NL operation captures long-range dependencies by assessing relationships between all pixel pairs. This allows distant features to influence each other, leading to a richer global context understanding. However, the NL operation introduces significant computational overhead due to its pix-level operation, making it less suited for real-time applications. Recently, some scholars utilized the Graph Convolutional Network (GCN) (Kipf & Welling, 2016) to realize NL operation (Li et al., 2020; Zhang, Chen, Arnab, Xue, & Torr, 2022), where local structures in images are treated as nodes and information is propagated between nodes in a non-Euclidean space through graph reasoning.

Inspired by previous work, we proposed a plug-and-play GFEM, which is a channel-level NL operation. It is efficient and can notably enrich the semantic information of high-level features to suppress background noise.

3. Methodology

In this section, we first provide an overall of the proposed defect detector. We then discuss the details of GFEM, LFEM, and BRM. Lastly, we present the training loss.

3.1. Overall

The structure of the proposed defect detector is illustrated in Fig. 2. Given the requirements of real-time defect detection and the necessity of avoiding complex anchor designs, our proposed defect detector is based on an one-stage, anchor-free GOD (Tian et al., 2019). This design ensures both speed and simplicity, making it well-suited for industrial defect detection. Specifically, the proposed defect detector consists of three parts: backbone, neck and head. We utilized ResNet18 as the backbone for our detector. For a given industrial surface image processed through ResNet18, the extracted multi-level feature maps are $\{C1, C2, C3, C4, C5\}$. Due to the higher resolution of $C1$ and $C2$, utilizing them might compromise detection efficiency. Therefore, we only employ $C3$, $C4$, and $C5$ for defect detection. The resolutions of $C3$, $C4$, and $C5$ are 1/8, 1/16, and 1/32 of the original image, respectively. $C3$ and $C4$ are low-level features, corresponding to local detail information in industrial surface images, which contains features of small-sized defects. Meanwhile, $C5$ is high-level feature, reflecting global semantic information in the image, such as objects and background. To enrich $C5$'s global semantic information and mitigate the influence of background noise on defects, we adopt GFEM to enhance $C5$ into $F5$. Next, in the neck, we employ FPN (Lin, Dollár, et al., 2017) to fuse $F5$ top-down into $C4$ and $C3$, yielding $F4$ and $F3$ with enhanced semantic information. Besides, LFEM is utilized to enhance small-size defect features in $F4$ and $F3$. Finally, the three enhanced features, along with BRM, are employed for defect detection. BRM effectively learns the shape information of defects, leading to more precise prediction results.

3.2. Global feature enhancement module

Inspired by NL operation (Wang et al., 2018) and Graph-based NL operation (Li et al., 2020; Zhang et al., 2022), we proposed an efficient, plug-and-play GFEM for industrial defect detection, which is a channel-level NL operation and its structure is illustrated in Fig. 3. Assume that the high-level feature $C5 \in \mathbb{R}^{C \times H \times W}$, where C represents the number of channels, while H and W correspond to feature maps' height and width, respectively. To reduce the computational complexity of GFEM, as show in formula (1), we utilize a 1×1 convolutional layer to transform $C5$ to X, decreasing its channel dimension from C to N, where $N = C/r$ and r is the reduction rate. The GFEM pipeline primarily comprises three steps: node representation, node graph reasoning and feature reconstruction. Each of these steps is elaborated upon below.

$$X = conv_{1\times1}(C5) \tag{1}$$

Firstly, during the node representation phase, as shown in formula (2)–(4), we use GAP and GMP to compress N channels into N vectors, subsequently aggregating their results to form the node representation, where n denotes the n^{th} channel in X, and X is represented by two $N \times 1 \times 1$ vectors. Next, we divide each vector into K groups and each group is regarded as a node. Here, each node is represented by N/K channels. The combination of GAP and GMP captures both overall context and peak activations, providing a holistic node representation for further node graph reasoning.

$$GAP(X(n)) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} X(n)_{i,j} \tag{2}$$

$$GMP(X(n)) = \max_{i,j} X(n)_{i,j}, i \in [1, W], j \in [1, H] \tag{3}$$

$$Represent_1(X(n)) = GAP(X(n)) + GMP(X(n)) \tag{4}$$

Secondly, during the node graph reasoning phase, we establish a fully connected graph where nodes represent vertices, and their inter-relationships are denoted by the edges within the graph. As shown in formulas (5) and (6), we use GCN for graph reasoning, The Laplacian

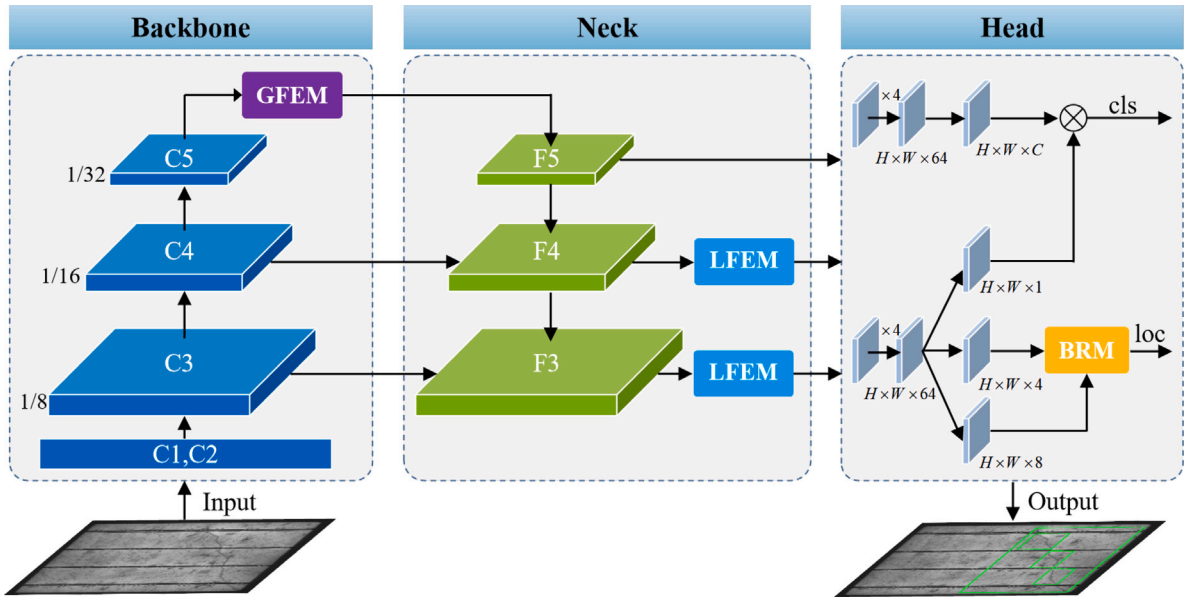

Fig. 2. The structure of the proposed defect detector. It consists of three parts: backbone, neck and head. Firstly, for an industrial surface image, ResNet18 is used as backbone to extract its features maps $\{C1, C2, C3, C4, C5\}$. $C5$ is enhanced as $F5$ using GFEM. Next, $F4$ and $F3$ are obtained by FPN in the neck, then LFEM is utilized to further enhance $F4$ and $F3$. Finally, in the head, enhanced features and BRM are utilized for defect detection.

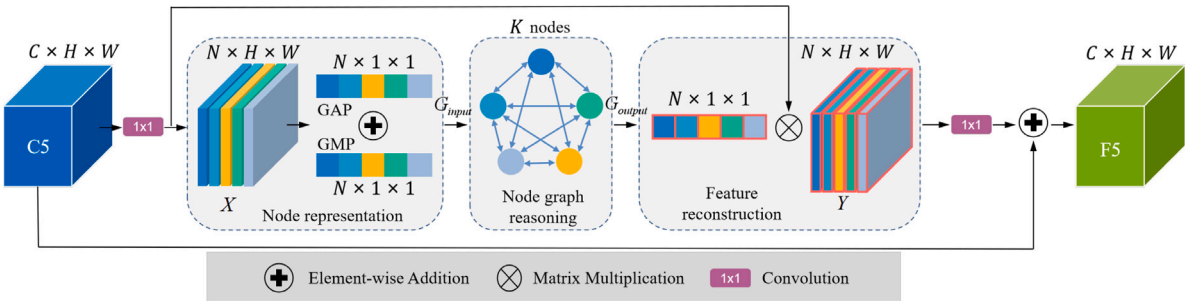

Fig. 3. The structure of the proposed GFEM. GFEM mainly consists of three steps: node representation, node graph reasoning and feature reconstruction. We combine GAP and GMP to create a well-rounded node representation. Node graph reasoning generates rich global semantic information in the defect image. The reconstructed $F5$ has enhanced ability in distinguish background noise and defect.

matrix $(I - A_g)$ is employed for information propagation between vertices. Here, $Represent_2(X)$ is the input of GCN, σ represents the activation function, A_g is a learnable matrix, and k denotes the k^{th} node. Both A_g and W_g can be implemented by a 1x1 convolution. After node graph reasoning, G_{output} reflects relationships between each node.

$$G_{input} = Represent_2(X(k)) \tag{5}$$

$$G_{output} = \sigma(WXA) = \sigma(W_g G_{input}(I - A_g)) \tag{6}$$

Lastly, in the feature reconstruction phase, as shown in formula (7), by multiplying X with G_{output}, we obtain Y, which represents the reconstructed features. As shown in formula (8), then we use a 1×1 convolutional layer to transform Y to original channel dimension C. Besides, to prevent any specific channel response from being overly amplified or suppressed, $C5$ is incorporated into the final output.

$$Y = G_{output} X \tag{7}$$

$$F5 = C5 + conv_{1\times1}(Y) \tag{8}$$

GFEM enhances the global information modeling capability of $C5$, enriching its semantic information, which is beneficial for distinguishing between background noise and defects. Furthermore, this capability is propagated from $F5$ to $F4$ and $F3$ through top-down feature fusion using FPN.

3.3. Local feature enhancement module

The challenge in detecting small size defects stems from their subtle feature representations. We introduced the LFEM to enhance the feature representation of small size defects. Our LFEM is implemented using the Local Attention Pyramid (LAP) (Shim, Hyun, Bae, & Heo, 2022), which effectively amplifies local peaks within regions with different scale in each channel. LFEM employs a spatial pyramid to divide each channel into patches with different sizes, then both normalization and sigmoid operations are performed on each patch independently to generate local attention maps, which is utilized to recursively amplify local peaks within each patch.

The structure of the LFEM is illustrated in Fig. 4. Given the pyramid level as p, the feature map is divided into patches whose size is $\frac{1}{2^p}$ of the original. Consider an example where the value of p ranges from 2 to 0. Initially, when p is set to 2, the feature map is divided into 16 patches. Subsequently, each patch undergoes instance normalization (Ulyanov, Vedaldi, & Lempitsky, 2016). This operation is individually applied to each channel within a patch. Following this, a sigmoid function is employed to derive the attention map. This attention map is then employed to multiply with the original input to generate the output of this pyramid level. Each level's output is then used as the input for the next, ascending from the base to the top of the pyramid. This process continues until it reaches the peak level when $p = 0$. The output at this peak level then serves as the final output of the LFEM.

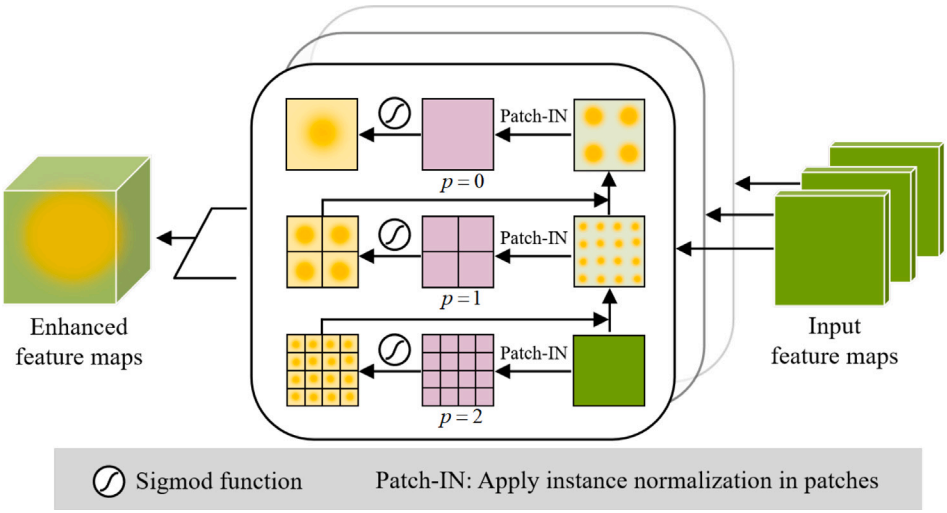

Fig. 4. The structure of the LFEM. LFEM divides the input feature maps into multi-scale patches by using spatial pyramid (purple part).

Consider the input feature maps are F with dimensions $C \times H \times W$ ($H = W$) and the initial level in LFEM is p. As shown in formula (9), F_p is the input. Then local attention maps w_p are obtained by formula (10) where F_p are divided into $2^p \times 2^p$ patches, and the dimension of w_p is $C \times 2^p \times 2^p$, In denotes instance normalization and σ represents sigmod function.

$$F_p = F \tag{9}$$

$$w_p = L(F_p, p) = \sigma(\text{Patch-In}(F_p, p)) \tag{10}$$

Next, as shown in formula (11), F_{p-1} is obtained by a weighted sum of the original value and the value processed through w_p, where α is an adjustable parameter and $\odot$ is element-wise multiplication. The aforementioned calculation is recursively performed until $p = 0$, where the patch size becomes the original input size, covering the entire channel. Then the final output is F_{-1}.

$$F_{p-1} = \alpha F_p + (1 - \alpha) w_p \odot F_p \tag{11}$$

In this paper, we employ LFEM to process both $F3$ and $F4$. LFEM emphasizes small-sized defect features amplifying local peaks across multi-scale patches in each channel. Furthermore, through the local attention mechanism, LFEM can also weaken background noises by assigning them lower attention values.

3.4. Box refinement module

GOD ignores the shape information of defects, compromising the accuracy of its predictions. Box refinement technique is an effective method to address this issue (Zhang, Wang, Dayoub, & Sunderhauf, 2021; Zhang, Wen, Bian, Lei, & Li, 2018). Inspired by Zhang et al. (2021), we designed BRM to compensate for the ignorance of the defect's shape. Unlike Zhang et al. (2021), which designs a complex feature representation for box refinement. Our method is simpler. We focus on points near the defect boundary, as they have a stronger correlation with the defect's shape, leading to more accurate predictions. It refines the initial prediction result by predicting offsets for the boundary points through an learnable module.

The designed BRM is illustrated in Fig. 5. Consider that the feature map HF in the head's location branch predicts the initial offset $B \in \mathbb{R}^{H \times W \times 4}$, where $B(i, j) = \{\Delta l, \Delta t, \Delta r, \Delta b\}$ represents the distances from the anchor point (i, j) to the leftmost, topmost, rightmost, and bottommost boundaries of the ground truth (red), respectively. Then as shown in formula (12), we utilized an learnable module l_{offset} to predict two offsets for each boundary point (green) based on HF, where

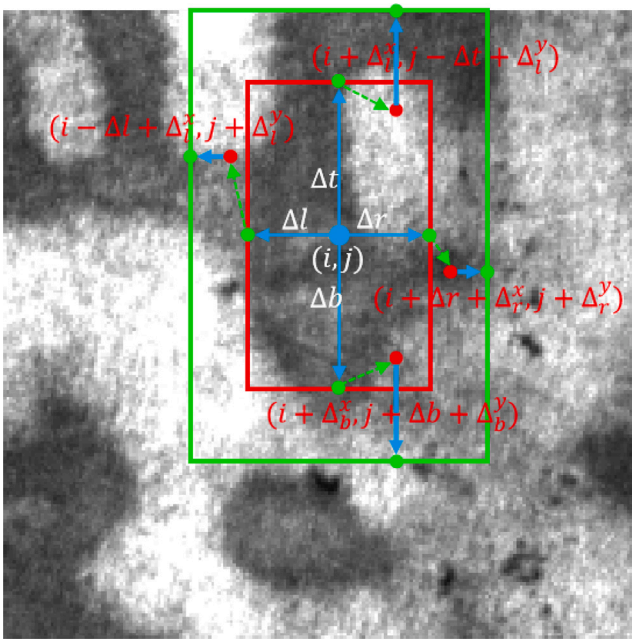

Fig. 5. Illustration of the BRM. A initial offset to the leftmost, topmost, rightmost, and bottommost boundaries of the ground truth (red) is firstly predicted as $B(i, j) = \{\Delta l, \Delta t, \Delta r, \Delta b\}$ at location (i, j). Next, an learnable module l_{offset} predicts two offsets B_{offest} for each boundary point (green point). Lastly, the final refined bounding box (green) is predicted by Deformable Convolution Network (DCN) using B_{offest}.

l_{offset} is implemented by a convolutional layer, $B_{offest} \in \mathbb{R}^{H \times W \times 8}$. $B_{offest}(i, j) = \{\Delta l_x, \Delta l_y, \Delta t_x, \Delta t_y, \Delta r_x, \Delta r_y, \Delta b_x, \Delta b_y\}$. Subsequently, the new boundary points are denoted as formula (13). The final refined bounding box, illustrated in green, is predicted by Deformable Convolution Network (DCN) (Dai et al., 2017) using B_{offest}. Lastly, the refined box is utilized to update the initial offset.

$$B_{offest} = l_{offset}(HF) \tag{12}$$

$$\begin{aligned} N_l(i, j) &= (i - \Delta l + \Delta l_x, j + \Delta l_y) \\ N_t(i, j) &= (i + \Delta t_x, j - \Delta t + \Delta t_y) \\ N_r(i, j) &= (i + \Delta r + \Delta r_x, j + \Delta r_y) \\ N_b(i, j) &= (i + \Delta b_x, j + \Delta b + \Delta b_y) \end{aligned} \tag{13}$$

BRM refines the positions of boundary points using learnable offsets. These offsets capture the defect's shape information to a certain extent, leading to more accurate prediction results.

3.5. Training loss

Label assignment, which allocates positive or negative loss to training samples, is vital for object detection loss computation. In this paper, We employ the state-of-the-art DW method from Li, He, et al. (2022) to assign positive and negative weights to anchors. Different from previous coupled weighting methods (Tian et al., 2019; Zhang et al., 2020; Zhu et al., 2020), DW assigns dual weights to each anchor using consistency and inconsistency metrics. The detailed calculation process of loss is as follows.

Firstly, W_{pos} is defined by formula (14) and (15), where t is denoted as consistency metric, s is the predicted class score, β and μ are hyperparameters.

$$t = s \times Iou^{\beta} \tag{14}$$

$$W_{pos} = e^{\mu t} \times t \tag{15}$$

Next, W_{neg} is defined by formula (16), where $\gamma 1$ and $\gamma 2$ are hyperparameters and k and b are determined by $\gamma 1$ using undetermined coefficients.

$$W_{neg} = \begin{cases} s^{\gamma 2} & \text{if IoU} < 0.5 \\ (-k \times IoU^{\gamma 1} + b) \times s^{\gamma 2} & \text{if IoU} \in [0.5, 0.95] \\ 0 & \text{if IoU} > 0.95 \end{cases} \tag{16}$$

Lastly, $\mathcal{L}_{cls}$, $\mathcal{L}_{reg}$ and $\mathcal{L}_{total}$ are calculated as formula (17)–(19), where N represents the total number of anchors within the candidate bag, while M indicates those outside of it. FL stands for Focal-loss as described in Lin, Goyal, et al. (2017), and $GIoU$ refers to the regression loss detailed in Rezatofighi et al. (2019). b represents the predicted box's location, while b' denotes the location of the ground truth.

$$\mathcal{L}_{cls} = \sum_{n=1}^{N} -w_{pos}^{n} \times \ln(s^{n}) - w_{neg}^{n} \times \ln(1 - s^{n}) + \sum_{m=1}^{M} FL(s^{m}, 0) \tag{17}$$

$$\mathcal{L}_{reg} = \sum_{n=1}^{N} w_{pos}^{n} \times GIoU(b, b') \tag{18}$$

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \beta \mathcal{L}_{reg} \tag{19}$$

4. Empirical study

In this section, we use three public industrial surface defect datasets to validate the effectiveness of our proposed defect detector. We initiate with a brief description of the datasets. This is followed by an introduction of the evaluation metrics and the implementation details. Finally, we present our experimental results and provide a thorough analysis.

4.1. Dataset description

4.1.1. PVEL_AD

PVEL_AD, provided by Su, Zhou, and Chen (2022), is a public dataset for anomaly detection in Photovoltaic electroluminescence solar cells. It comprises 36,543 near-infrared images, spanning 12 defect types and one anomaly-free. Following the setting in Su et al. (2022), we selected crack, finger, black_core, and thick_line to construct our dataset. Each image, originally at a resolution of 1024 × 1024, is resized to 640 × 640. The dataset is split with an 8:2 train-test ratio (see Fig. 6).

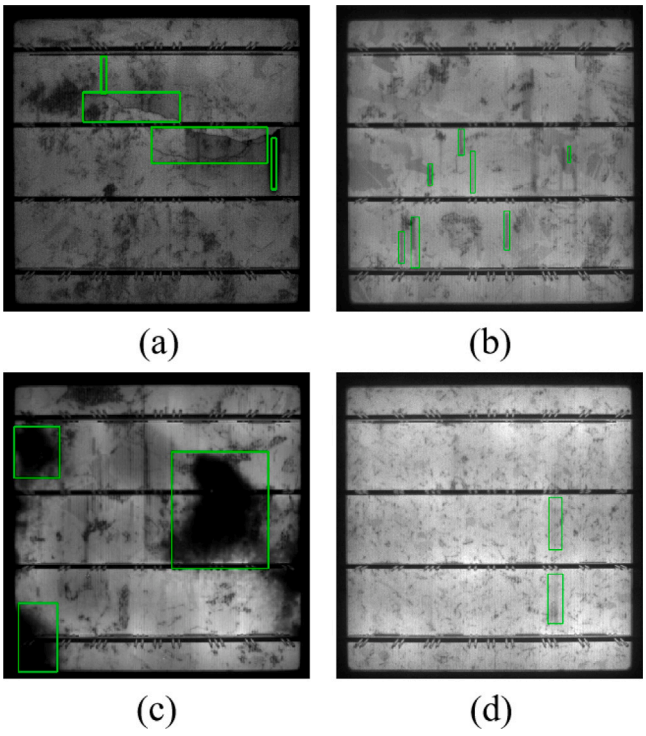

Fig. 6. Defect samples from PVEL_AD: (a) crack, (b) finger, (c) black_core, and (d) thick_line.

4.1.2. PCB defect dataset

PCB defect dataset, provided by Ding, Dai, Li, and Liu (2019) is a public dataset for defect detection in printed circuit boards. It consists of 12,428 images covering 6 defect types: missing hole, mouse bite, open circuit, short, spur, and spurious copper. Each image, originally at a resolution of 600 × 600, is resized to 640 × 640. The training dataset contains 9,920 images, while the test dataset contains 2,508 images (see Fig. 7).

4.1.3. NEU-DET

NEU-DET, provided by He et al. (2019) is a public dataset for defect detecion in hot-rolled steel strip. It consits of 1,800 images covering 6 defect types: Crazing (Cr), Rolled-in Scale (RS), Patches (Pa), Pitted Surface (PS), Inclusion (In), Scratches (Sc). Each image, originally at a resolution of 200 × 200, is resized to 320 × 320. The dataset is split with an 7:3 train-test ratio (see Fig. 8).

4.2. Experimental setting

4.2.1. Evaluation metrics

We use AP (Average Precision) to assess the detection performance of each class, while mAP measures the overall detection performance. A higher AP value indicates better detection performance. Formulas (20) and (21) illustrate the computation methods for AP and mAP respectively, where P denotes precision, R denotes recall, C denotes the total number of classes. In this paper, we use AP50 as the evaluation metric.

$$\text{AP} = \int_{0}^{1} P(R)\, dR \tag{20}$$

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_{c} \tag{21}$$

The computation methods of P and R are shown in formula (22) and (23), where TP, FP, and FN denote the counts of True Positives, False

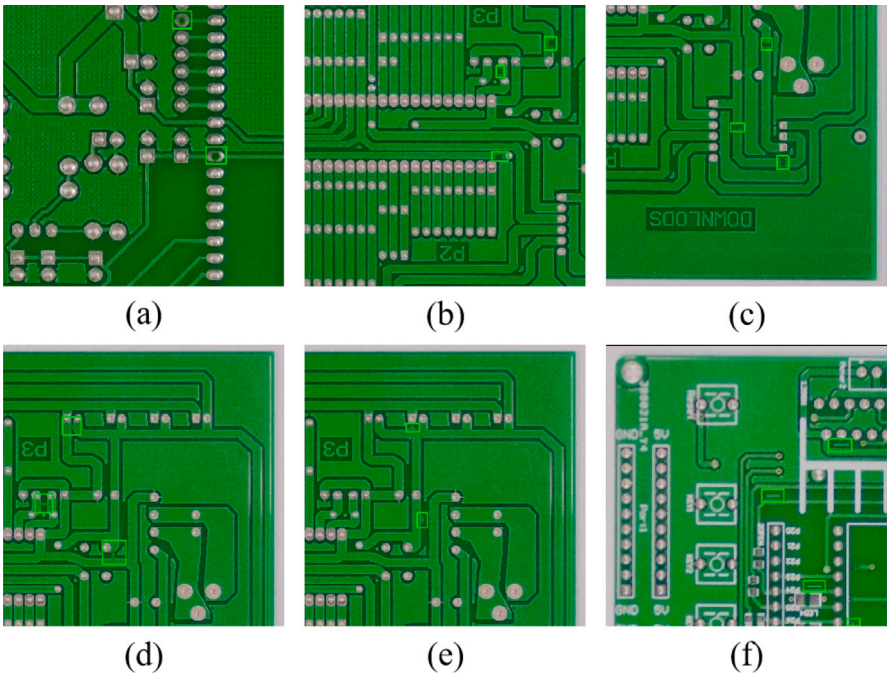

Fig. 7. Defect samples from PCB defect dataset: (a) missing_hole, (b) mouse_bite, (c) open_circuit, (d) short, (e) spur and (f) spurious_ copper.

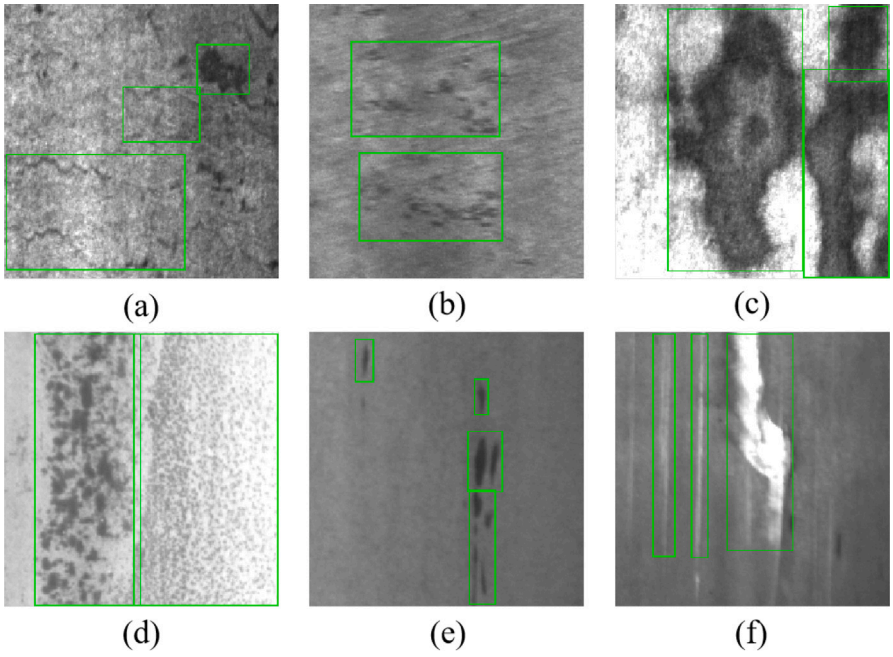

Fig. 8. Defect samples from PCB defect dataset: (a) Cr, (b) RS, (c) Pa, (d) PS, (e) In and (f) Sc.

Positives, and False Negatives, respectively.

$$P = \frac{TP}{TP + FP} \tag{22}$$

$$R = \frac{TP}{TP + FN} \tag{23}$$

4.2.2. Implementation details

In the GFEM, the node group K is set to 16 and the channel reduction rate r is set to 2. In the LFEM, we set $\alpha = 0.5$ and the highest level p of the spatial pyramid to 2. For the PVEL_AD and PCB defect datasets, patch sizes in $F3$ and $F4$ are $\{20 \times 20, 40 \times 40, 80 \times 80\}$ and $\{10 \times 10, 20 \times 20, 40 \times 40\}$, respectively. For NEU-DET, they are set to $\{10 \times 10, 20 \times 20, 40 \times 40\}$ for $F3$ and $\{5 \times 5, 10 \times 10, 20 \times 20\}$ for $F4$. For the parameters used in calculating the training loss, we follow Li, He, et al. (2022). The backbone is pre-trained on ImageNet.

We train for 80 epochs using SGD with a learning rate of 0.01, momentum of 0.9, weight decay of 0.0001, and a batch size of 32. The learning rate undergoes a linear warm-up over the initial 500 iterations and is scaled to 0.1 of its value after the 50th epoch. All networks are built on mmdetection 2.26.0. All programs are installed in python 3.7 and the hardware platform is Inter (R) CoreTM i9-10920X processor with basic frequency 3.50 GHZ and RAM 96 GB. The GPU is NVIDIA GeForce RTX 3090.

Table 1
Comparison with state-of-the-art methods on PVEL_AD.

Method	Backbone	Params	FPS	mAP@0.5 (%)	Crack (%)	Black_core (%)	Thick_line (%)	Finger (%)
RetinaNet	ResNet50	36.17 M	43.1	87.2	67.1	98.8	90.5	92.3
FCOS	ResNet50	31.89 M	42.9	86.4	73.3	98.1	85.5	88.7
YOLOv3	DarkNet53	61.54 M	52.5	85.9	65.4	98.1	88.5	91.6
YOLOv5	CSPDarkNet53	21.20 M	56.3	87.5	70.3	99.1	88.6	92.0
YOLOX	CSPDarkNet53	25.30 M	51.2	88.2	73.7	98.8	87.5	92.6
YOLOv6	EfficientRep	34.30 M	53.8	88.6	73.0	99.4	89.1	93.0
YOLOv7	CSPDarkNet53	36.90 M	50.5	89.8	75.4	**99.5**	91.0	93.3
OURS	ResNet18	11.65 M	46.1	**92.0**	**81.3**	99.3	**92.9**	**94.3**

Table 2
Comparison with state-of-the-art methods on PCB defect dataset.

Method	Backbone	Params	FPS	mAP@0.5 (%)	Missing_hole (%)	Mouse_bite (%)	Open_circuit (%)	Short (%)	Spur (%)	Spurious_copper (%)
FCOS	ResNet50	31.89 M	42.9	94.0	94.5	95.2	93.6	93.8	93.7	92.9
YOLOv3	Darknet53	61.54 M	52.5	87.2	87.1	87.3	87.2	88.3	86.9	86.4
YOLOv4	CSPDarknet53	27.60 M	55.2	93.3	93.6	92.9	94.1	93.6	92.8	92.7
YOLOv5	CSPDarknet53	21.20 M	56.3	94.5	95.5	94.3	94.4	94.7	94.3	93.8
YOLOX	CSPDarknet53	25.30 M	51.2	95.7	97.1	95.4	95.7	95.9	95.4	94.9
ES-Net	CSPDarknet53	147.98 M	53	97.5	99.5	98.1	99.5	96.8	95.2	95.9
OURS	ResNet18	11.65 M	46.1	**99.6**	**99.7**	**99.6**	**99.7**	**99.6**	**99.4**	**99.3**

4.3. Experimental results and analysis

4.3.1. Comparison with state-of-the-art methods

(1) Results on PVEL_AD: We compare our method with the state-of-the-art methods including RetinaNet (Lin, Goyal, et al., 2017), FCOS (Tian et al., 2019), YOLOv3 (Redmon & Farhadi, 2018), YOLOv5 (Ultralytics, 2020), YOLOX (Ge, Liu, Wang, Li, & Sun, 2021), YOLOv6 (Li, Li, et al., 2022), and YOLOv7 (Wang et al., 2023). The results, as shown in Table 1, indicate that the proposed method achieved the highest mAP of 92.0%, surpassing the second-best by 2.2%. In the individual class evaluations, our method achieved an AP of 81.3% for the crack, leading the second-best by 5.9%. For the black_core, our method recorded an AP of 99.3%, closely matching the performance of the best. In the thick_line, our method's AP was 92.9%, outperforming the second-best by 1.9%. Finally, for the finger, the proposed method achieved an AP of 94.3%, exceeding the second-best by 1.0%. These results clearly demonstrate the effectiveness of the proposed method. Furthermore, all methods demonstrate good performance on black_core due to its distinct features. Notably, the proposed method exhibits pronounced superiority in crack detection over other methods. This is largely attributed to the abundant background noise in crack images. While other methods are easily affected by this noise, our approach effectively suppresses it, underscoring our method's advantage in such a challenging scenario. It can be observed that while the proposed method does not reach the maximum FPS, its parameter count is only 11.65M, which is significantly less than that of other methods, implying that it has a lower cost of deployment.

(2) Results on PCB defect dataset: We compared our method with the state-of-the-art methods including FCOS (Tian et al., 2019), YOLOv3 (Redmon & Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), YOLOv5 (Ultralytics, 2020), YOLOX (Ge et al., 2021) and ES-Net (Yu et al., 2022). The comparison results are shown in Table 2. It can be seen that the proposed method achieved a mAP of 99.6%. For each class, the proposed method obtained an AP of 99.7% for missing hole, an AP of 99.6% for mouse bite, an AP of 99.7% for open circuit, an AP of 99.6% for short, an AP of 99.4% for spur, and an AP of 99.3% for spurious copper. Our method not only achieved the highest mAP but also led the AP scores in each class. The AP for each class approach a near-perfect 100%. With these results, it outperformed the second-best method, ES-Net, by 2.1% in mAP and by 0.2%, 1.5%, 0.2%, 2.8%, 4.2%, and 3.4% for each respective class. Particularly, the enhancements in the classes of short, spur, and spurious copper were quite evident, which demonstrated the effectiveness of our method in detecting small size defects, highlighting its superior detection capability for such defects.

(3) Results on NEU-DET: We compared our method with state-of-the art methods including FCOS (Tian et al., 2019), YOLOv3 (Redmon & Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), YOLOv5 (Ultralytics, 2020), YOLOX (Ge et al., 2021), DEA-RetinaNet (Cheng & Yu, 2020), CABF-FCOS (Yu et al., 2021) and ES-Net (Yu et al., 2022). The comparison results are shown in Table 3. It can be seen that the proposed method achieved a mAP of 81.6%. For each class, the proposed method obtained an AP of 57.3% for Cr, an AP of 63.1% for RS, an AP of 95.2% for Pa, an AP of 89.9% for PS, an AP of 88.5% for In, and an AP of 95.4% for Sc. The proposed method achieved the highest mAP. Meanwhile, it outperformed other methods in the Pa, In, and Sc classes and obtained the second-best scores in the Cr, RS, and PS classes. This implies that our method achieved balanced results across all classes, which is attributed to our thorough consideration of the characteristics of industrial surface defects.

4.3.2. Ablation study

(1) Effectiveness of each module: We investigated the contributions of GFEM, LFEM, and BRM on the PVEL_AD dataset, and the results are presented in Table 4. From the table, it can be observed that the baseline without the introduction of GFEM, LFEM, and BRM achieves an mAP of 86.6%. Specifically, the AP for crack is 71.2%, for black_core is 98.4%, for thick_line is 86.1%, and for finger is 90.6%. After introducing GFEM, the overall mAP increased by 2.3% to 88.9%. Additionally, the AP values for crack, black_core, thick_line, and finger rose to 74.5%, 98.6%, 90.5%, and 91.8%, reflecting gains of 3.3%, 0.2%, 4.4%, and 1.2% respectively. This advancement can be attributed to GFEM's ability to enhance $F5$'s capability of global information modeling . After incorporating LFEM, the mAP reached 90.6%, making an increase of 1.7%. Specifically, the AP for crack rose to 78.9%, black_core to 98.9%, thick_line to 91.5%, and finger to 93.0%, with respective improvements of 4.4%, 0.3%, 1.0%, and 1.2%. This improvement can be attributed to LFEM amplifying the local peaks in $F3$ and $F4$, thereby enhancing the feature representation of small-sized defects. Meanwhile, by weakening the response of regions with sub-peak values, it effectively reduces background noise interference as well.

Furthermore, after incorporating BRM, the mAP increased to 92.0% from a 1.4% rise. The AP values for crack, black_core, thick_line and finger rose to 81.3%, 99.3%, 92.9% and 94.3%, with increases of 2.4%, 0.4%, 1.4% and 1.3% respectively. This positive change is due to BRM's ability to capture defect shape's information, leading to more precise predictions. Obviously, the combination of GFEM, LFEM, and BRM improved the baseline's mAP by 5.4%, and the AP improvements were as follows: cracks by 10.1%, black_core by 0.9%, thick_line by 6.8%,

Table 3
Comparison with state-of-the-art methods on NEU-DET.

Method	Backbone	Params	FPS	mAP@0.5 (%)	Cr (%)	RS (%)	Pa (%)	PS (%)	In (%)	Sc (%)
FCOS	ResNet50	31.89 M	64.9	75.7	47.7	58.3	91.5	87.8	81.1	87.9
YOLO3	Darknet53	61.54 M	76.1	69.1	34.2	54.2	87.4	79.1	76.2	83.6
YOLO4	CSPDarknet53	27.60 M	79.8	74.3	43.4	57.5	89.1	86.2	83.3	86.3
YOLO5	CSPDarknet53	21.20 M	81.5	76.2	46.2	59.7	88.8	88.9	84.8	88.7
YOLOX	CSPDarknet53	25.30 M	75.2	77.0	44.7	57.6	92.2	89.3	85.3	92.8
DEA-RetinaNet	ResNet50	42.20 M	12.2	79.1	**60.9**	**67.2**	94.3	**95.8**	82.5	74.1
CABF-FCOS	ResNet50	56.30 M	18	76.7	55.4	62.9	93.5	88.9	75.0	84.4
ES-Net	CSPDarknet53	147.98 M	–	79.1	56.0	60.4	88.3	87.4	87.6	94.9
OURS	ResNet18	11.65 M	69.3	**81.6**	57.3	63.1	**95.2**	89.9	**88.5**	**95.4**

Table 4
Ablation study results of the proposed method.

Baseline	GFEM	LFEM	BRM	mAP@0.5 (%)	Crack (%)	Black_core (%)	Thick_line (%)	Finger (%)
✓	–	–	–	86.6	71.2	98.4	86.1	90.6
✓	✓	–	–	88.9	74.5	98.6	90.5	91.8
✓	✓	✓	–	90.6	78.9	98.9	91.5	93.0
✓	✓	✓	✓	92.0	81.3	99.3	92.9	94.3

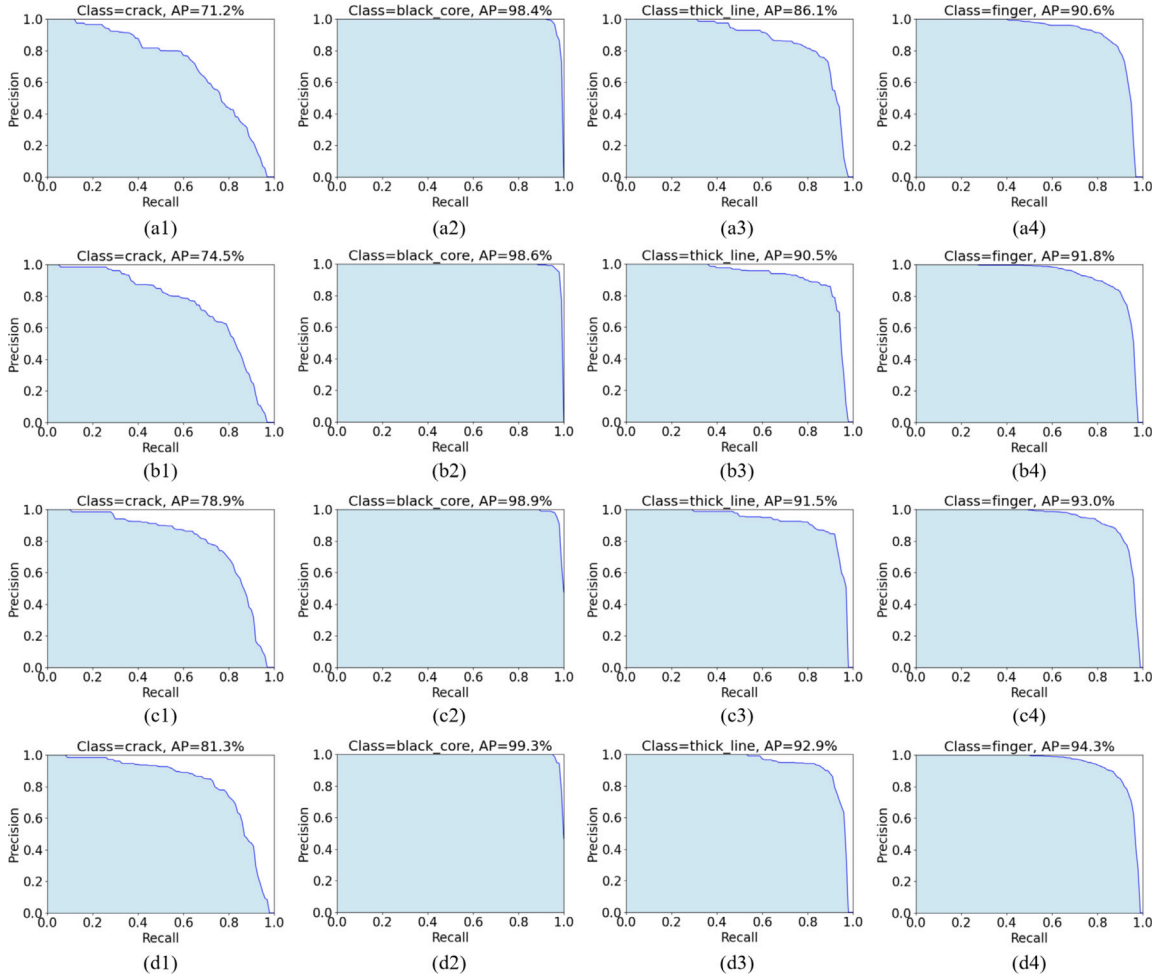

Fig. 9. P-R curves for each class from the ablation study. a1–a4 represent the baseline. b1–b4 represent the introduction of GFEM. c1–c4 represent the introduction of both GFEM and LFEM. d1–d4 represent the introduction of GFEM, LFEM, and BRM.

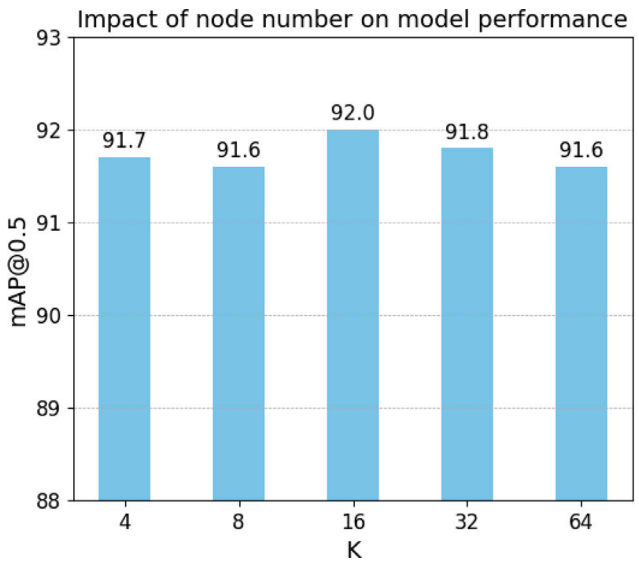

Fig. 10. The impact of node number in GFEM on model performance.

Table 5
Comparison result of model parameters and FPS across different modules.

Baseline	GFEM	LFEM	BRM	Params	FPS
✓	–	–	–	11.49 M	99.8
✓	✓	–	–	11.62 M	64.9
✓	✓	✓	–	11.64 M	51.7
✓	✓	✓	✓	11.65 M	46.1

Table 6
The impact of GAP and GMP on GFEM's performance.

Baseline	GAP	GMP	mAP@0.5	Params	FPS
✓	–	–	86.6	11.49 M	99.8
✓	✓	–	88.1	11.62 M	79.6
✓	–	✓	87.9	11.62 M	77.3
✓	✓	✓	88.9	11.62 M	64.9

and fingers by 3.7%, demonstrating their effectiveness in industrial surface defect detection. Besides, Fig. 9 shows the P-R curves for each class from the ablation study.

(2) Comparison of model parameters and FPS: We compared model parameters and FPS across different modules on PVEL_AD and the results are shown in Table 5. It can be observed that the baseline model has a parameter count of 11.49M and an FPS of 99.8. After introducing GFEM, LFEM, and BRM, the model's parameter count saw a modest increase from 11.49M to 11.65M, while the FPS dropped from 99.8 to 46.1. These three modules did not significantly increase the number of parameters, and although there was some loss in FPS, the detection performance still remains at real-time level (FPS greater than 30).

(3) The impact of GAP and GMP on GFEM's performance. We conduct separate experiment to analyze the impact of GAP and GMP on GFEM's performance and the results are shown in Table 6. It can be observed that the introduction of both GAP and GMP enhances the model's performance, with GAP increasing the model's mAP from 86.6% to 88.1%, and GMP increasing it from 86.6% to 87.9%. This demonstrates that GAP and GMP are effective, and their combination can further enhance the model's performance because they capture both the overall context and peak activations of feature maps, providing a holistic feature representation.

(4) The impact of node number in GFEM on model performance. To analyze the impact of node number on model performance, we set the values of K to 4, 8, 16, 32, and 64 respectively. The test result is shown in Fig. 10. The larger the value of K, the greater the number of vertices in the node graph. It can be seen that changing the values of K has a negligible effect on the model's performance.

(5) Robustness and generalization ability of GFEM. To validate the robustness and generalization ability of GFEM, we test it on other baseline methods. We select Faster-RCNN, RetinaNet, and FCOS as our baselines, with their backbones being ResNet34 and ResNet50, respectively. Table 7 presents the test results, where the application of GFEM has led to increased mAP values for all baseline methods. Specifically, Faster-RCNN's mAP has risen from 83.4% to 85.8% and from 84.3% to 86.6%. RetinaNet has seen an increase from 86.5% to 89.0% and from 87.2% to 89.4%. FCOS's mAP has improved from 85.8% to 88.1% and from 86.4% to 88.5%. These improvements demonstrate GFEM's robustness and its good generalization ability.

Table 7
Comparison of different baseline methods with and without GFEM.

Baseline	Backbone	GFEM	mAP@0.5
Faster-RCNN	ResNet34	–	83.4
Faster-RCNN	ResNet34	✓	**85.8**
Faster-RCNN	ResNet50	–	84.3
Faster-RCNN	ResNet50	✓	**86.6**
RetinaNet	ResNet34	–	86.5
RetinaNet	ResNet34	✓	**89.0**
RetinaNet	ResNet50	–	87.2
RetinaNet	ResNet50	✓	**89.4**
FCOS	ResNet34	–	85.8
FCOS	ResNet34	✓	**88.1**
FCOS	ResNet50	–	86.4
FCOS	ResNet50	✓	**88.5**

4.3.3. *Visualization analysis*

(1)Visualization of detection results. To further demonstrate the effectiveness of the proposed defect detector, we visualize its detection results on PVEL_AD, the PCB defect dataset, and NEU-DET. Fig. 11 displays the detection results for PVEL_AD. Despite the presence of defects of small size and complex backgrounds, the proposed defect detector effectively detects the defects. Figs. 12 and 13 present the detection results for the PCB defect dataset and NEU-DET. These detection results further illustrate the effectiveness of the proposed method in detecting defects, regardless of their small size or irregular shape. While there are instances of mistaken and missing detection, the overall performance of the proposed method remains superior.

(2) Visualization of extracted features. To further demonstrate the impact of GFEM and LFEM on feature maps, we visualized feature maps $F4$ from the PVEL_AD dataset both with and without using GFEM and LFEM in Fig. 14. As shown in Fig. 14(b), although the extracted feature maps without GFEM and LEFM can reflect defects in the image, there is a distinct presence of background noise in these feature maps, potentially affecting the accuracy of detection. As shown in Fig. 14(c), after applying GFEM, the background noise in the feature maps is distinctly suppressed, highlighting the defect features, which can be attributed to the strong global semantic information present in the enhanced $F5$, that is fused top-down into $F4$. As shown in Fig. 14(d), after applying LFEM, the defect features are strengthened because local peaks in feature maps are amplified. Fig. 14(e) displays the prediction results based on the feature maps. The effectiveness of GFEM and LFEM is further demonstrated by visualizing the extracted feature maps.

5. Conclusion and future work

In this paper, we proposed a novel real-time anchor-free defect detector tailored for industrial surface defect detection. Specifically, we primarily focus on addressing three main challenges inherent in industrial surface defect detection: complex background, small sized defects, and irregular defect shapes. Firstly, we proposed the GFEM

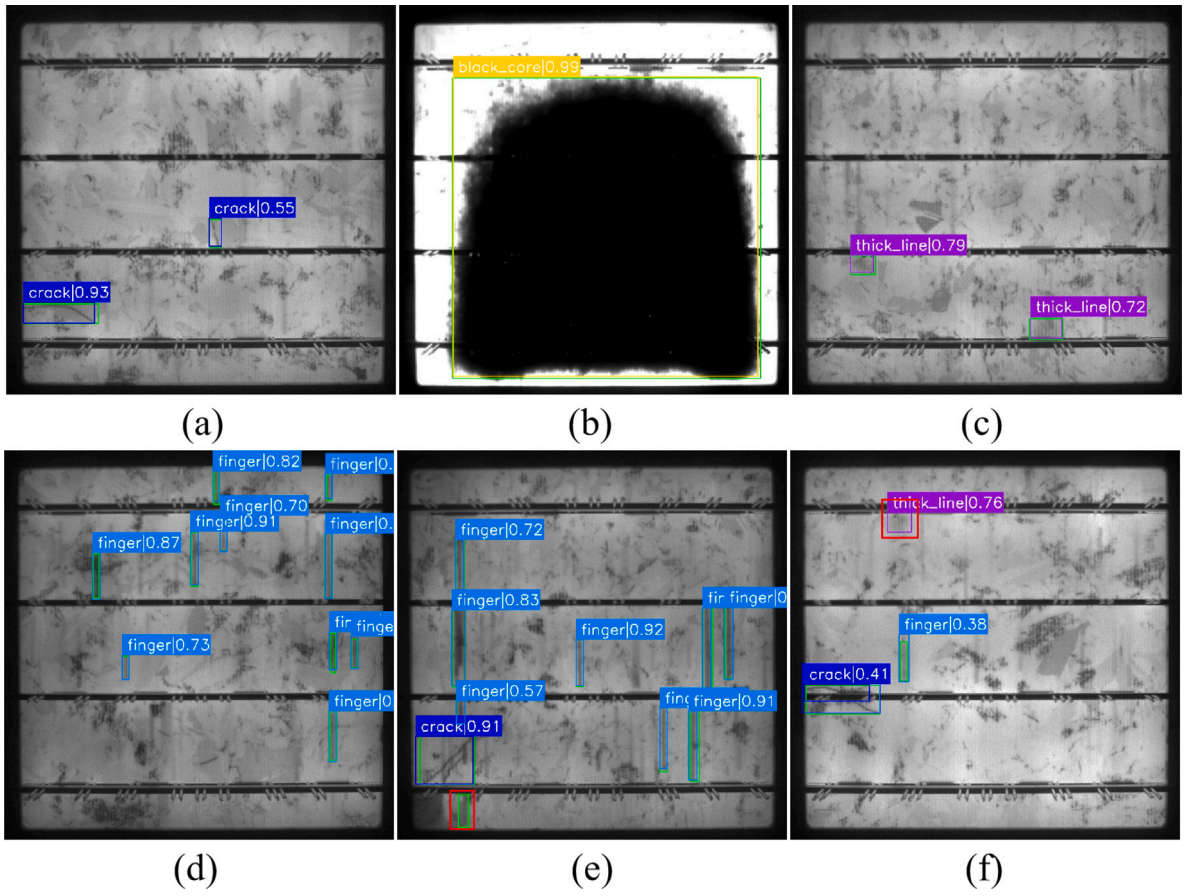

Fig. 11. The visualization detection results of the proposed detector on PVEL_AD. (a) crack. (b) black_core. (c) thick_line. (d) finger. (e) missing detection in red box. (f) mistaken detection in red box.

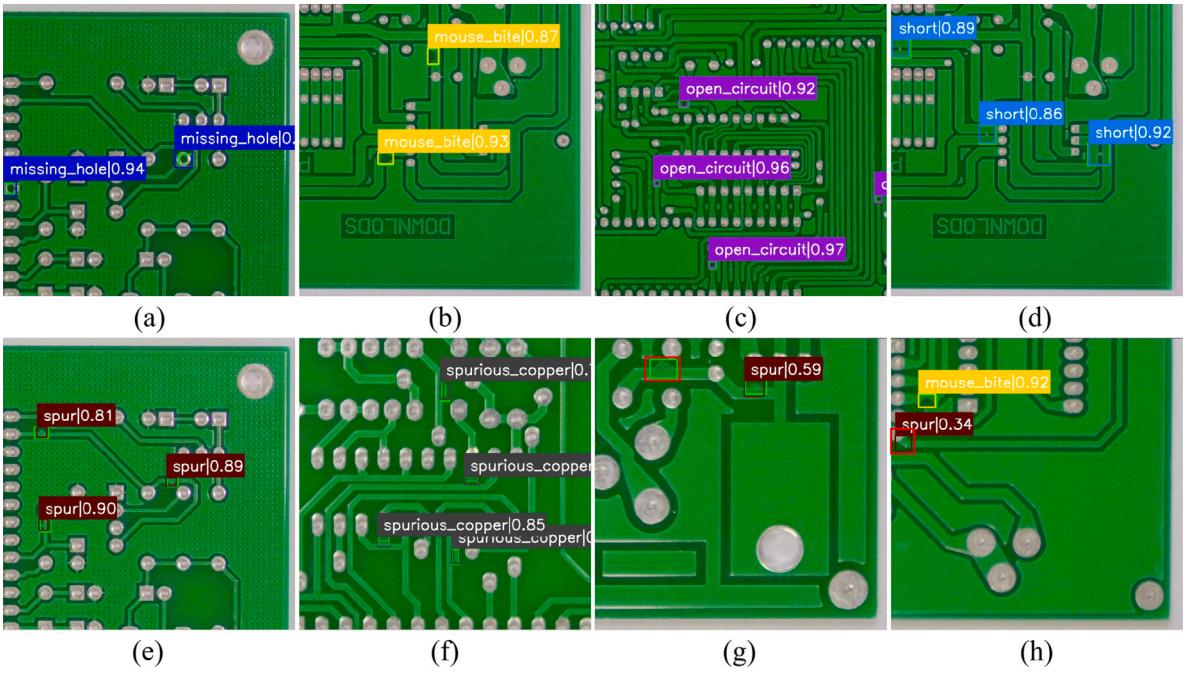

Fig. 12. The visualization detection results of the proposed detector on PCB defect dataset. (a) missing_hole. (b) mouse_bite. (c) open_circuit. (d) short. (e) spur. (f) spurious_copper. (g) missing_detection in red box. (h) mistaken detection in red box.

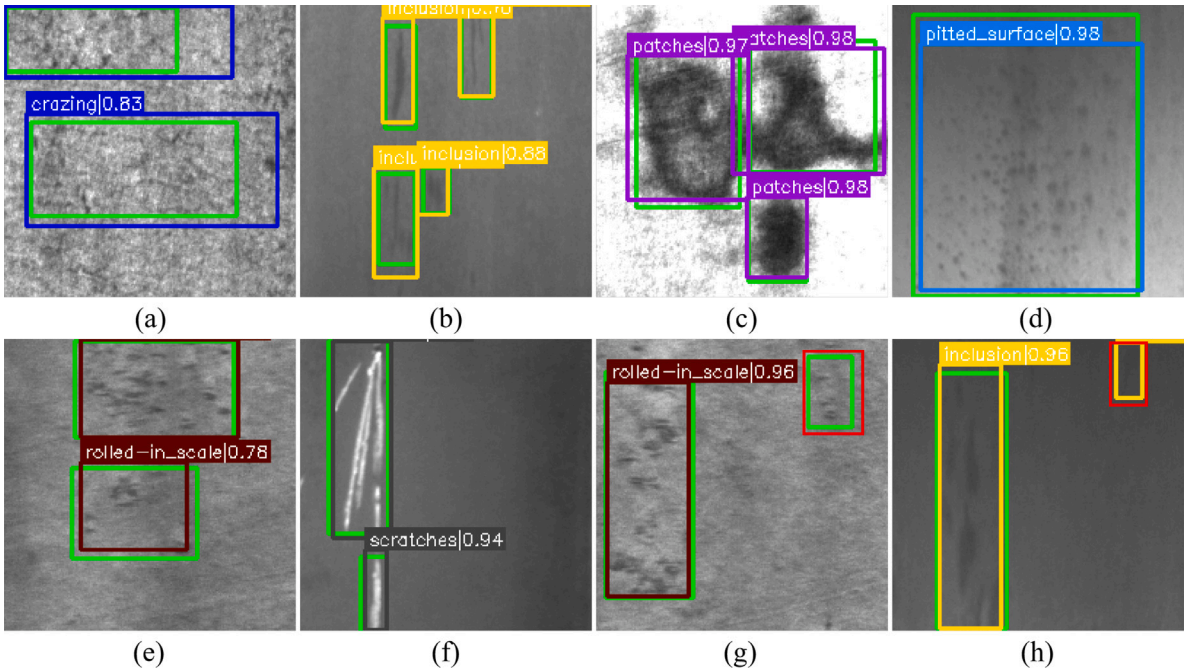

Fig. 13. The visualization detection results of the proposed detector on NEU-DET. (a) Cr. (b) RS. (c) Pa. (d) PS. (e) In. (f) Sc. (g) missing detection in red box. (h) mistaken detection in red box.

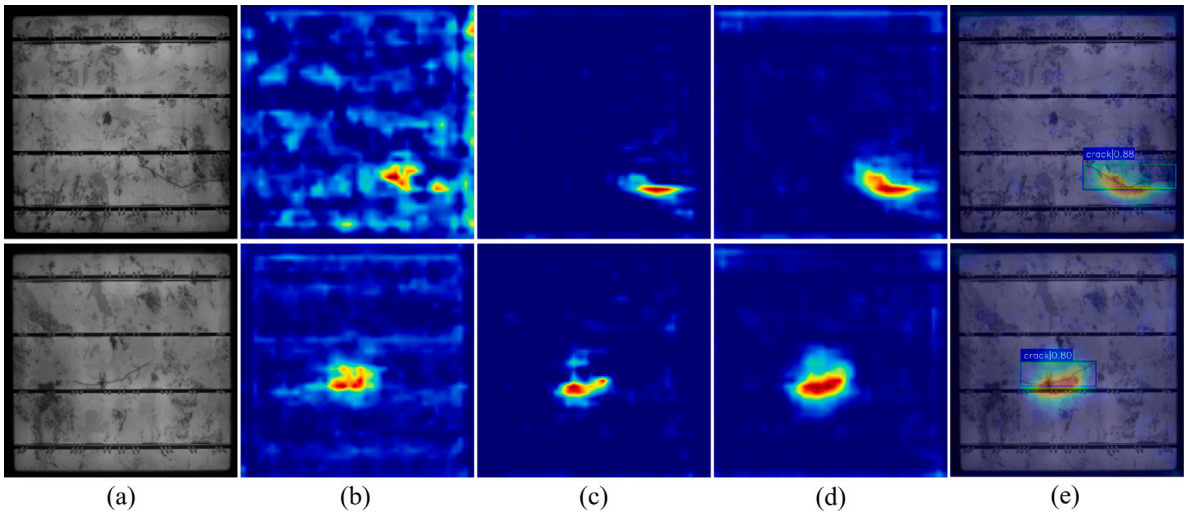

Fig. 14. Visualization of the impact of GFEM and LFEM on feature maps: (a) the original input. (b) feature maps without GFEM and LFEM. (c) feature maps with GFEM. (d) feature maps with both GFEM and LFEM. and (e) the final prediction results.

to reduce interference from background noise, it enhances high-level feature's ability in modeling global information by a channel-level NL operation. Secondly, we introduced the LFEM to enhance small size defect's features. It strengthen small size defect's feature by amplifying local peaks in low-level features. Thirdly, we introduced BRM to capture shape information of defects, thereby providing more accurate prediction results. Lastly, the proposed defect detector is evaluated on three public industrial surface defect datasets. The experimental results show that the proposed method reaches real-time detection level and outperforms state-of-the-art methods, highlighting the superiority of the proposed defect detector.

In our future work, we plan to proceed in two primary directions. On one hand, we will explore the use of model compression methods, such as pruning and knowledge distillation, to further enhance the detection efficiency of our model, with the goal of achieving real-time detection performance on CPU, facilitating broader and more economical deployments. On the other hand, recognizing the challenges in industrial settings, such as rare defects, expensive data annotation, and the need for swift adaptability, we will explore few-shot defect detection methods.

Funding information

This work was supported by the National Natural Science Foundation of China under Grant 62273261.

CRediT authorship contribution statement

Qing Liu: Conceptualizaton, Methodology, Software, Validaton, Writing – original draft. **Min Liu:** Supervision, Funding acquisiton. **Q.M. Jonathan:** Supervision. **Weiming Shen:** Supervision.

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 authors do not have permission to share data.

References

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. http://dx.doi.org/10.48550/arXiv.2004.10934, arXiv preprint arXiv:2004.10934.

Chen, P., Chen, M., Wang, S., Song, Y., Cui, Y., Chen, Z., et al. (2023). Real-time defect detection of TFT-lcd displays using a lightweight network architecture. *Journal of Intelligent Manufacturing*, 1–16. http://dx.doi.org/10.1007/s10845-023-02110-7.

Cheng, X., & Yu, J. (2020). RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11. http://dx.doi.org/10.1109/tim.2020.3040485.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 764–773). http://dx.doi.org/10.1109/iccv.2017.89.

Ding, R., Dai, L., Li, G., & Liu, H. (2019). TDD-net: a tiny defect detection network for printed circuit boards. *CAAI Transactions on Intelligence Technology*, *4*(2), 110–116. http://dx.doi.org/10.1049/trit.2019.0019.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6569–6578). http://dx.doi.org/10.1109/iccv.2019.00667.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*, 303–338. http://dx.doi.org/10.1007/s11263-009-0275-4.

Gao, L., Zhang, J., Yang, C., & Zhou, Y. (2022). Cas-VSwin transformer: A variant swin transformer for surface-defect detection. *Computers in Industry*, *140*, Article 103689. http://dx.doi.org/10.1016/j.compind.2022.103689.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. http://dx.doi.org/10.48550/arXiv.2107.08430, arXiv preprint arXiv:2107.08430.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448). http://dx.doi.org/10.1109/iccv.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587). http://dx.doi.org/10.1109/cvpr.2014.81.

He, Y., Song, K., Meng, Q., & Yan, Y. (2019). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Transactions on Instrumentation and Measurement*, *69*(4), 1493–1504. http://dx.doi.org/10.1109/tim.2019.2915404.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). http://dx.doi.org/10.1109/cvpr.2016.90.

Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13713–13722). http://dx.doi.org/10.1109/cvpr46437.2021.01350.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141). http://dx.doi.org/10.1109/cvpr.2018.00745.

Huang, F., Wang, B.-w., Li, Q.-p., & Zou, J. (2021). Texture surface defect detection of plastic relays with an enhanced feature pyramid network. *Journal of Intelligent Manufacturing*, 1–17. http://dx.doi.org/10.1007/s10845-021-01864-2.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. http://dx.doi.org/10.48550/arXiv.1609.02907, arXiv preprint arXiv:1609.02907.

Li, S., He, C., Li, R., & Zhang, L. (2022). A dual weighting label assignment scheme for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9387–9396). http://dx.doi.org/10.1109/cvpr52688.2022.00917.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). YOLOv6: A single-stage object detection framework for industrial applications. http://dx.doi.org/10.48550/arXiv.2209.02976, arXiv preprint arXiv:2209.02976.

Li, X., Yang, Y., Zhao, Q., Shen, T., Lin, Z., & Liu, H. (2020). Spatial pyramid based graph reasoning for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8950–8959). http://dx.doi.org/10.1109/cvpr42600.2020.00897.

Liang, W., & Sun, Y. (2022). ELCNN: a deep neural network for small object defect detection of magnetic tile. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–10. http://dx.doi.org/10.1109/tim.2022.3193175.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125). http://dx.doi.org/10.1109/cvpr.2017.106.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988). http://dx.doi.org/10.1109/iccv.2017.324.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6-12, 2014, proceedings, part V 13* (pp. 740–755). Springer, http://dx.doi.org/10.1007/978-3-319-10602-1_48.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). Ssd: Single shot multibox detector. In *Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, proceedings, part I 14* (pp. 21–37). Springer, http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022). http://dx.doi.org/10.1109/iccv48922.2021.00986.

Liu, Q., Liu, M., Zhou, H., Yan, F., Ma, Y., & Shen, W. (2022). Intelligent manufacturing system with human-cyber-physical fusion and collaboration for process fine control. *Journal of Manufacturing Systems*, *64*, 149–169. http://dx.doi.org/10.1016/j.jmsy.2022.06.004.

Liu, Y., Shao, Z., & Hoffmann, N. (2021). Global attention mechanism: Retain information to enhance channel-spatial interactions. http://dx.doi.org/10.48550/arXiv.2112.05561, arXiv preprint arXiv:2112.05561.

Lu, H., Fang, M., Qiu, Y., & Xu, W. (2022). An anchor-free defect detector for complex background based on pixelwise adaptive multiscale feature fusion. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–12. http://dx.doi.org/10.1109/tim.2022.3229728.

Lu, B., & Huang, B. (2023). A texture-aware one-stage fabric defect detection network with adaptive feature fusion and multi-task training. *Journal of Intelligent Manufacturing*, 1–14. http://dx.doi.org/10.1007/s10845-023-02105-4.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). http://dx.doi.org/10.1109/cvpr.2016.91.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. http://dx.doi.org/10.48550/arXiv.1804.02767, arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*, http://dx.doi.org/10.1109/tpami.2016.2577031.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666). http://dx.doi.org/10.1109/cvpr.2019.00075.

Shim, S.-H., Hyun, S., Bae, D., & Heo, J.-P. (2022). Local attention pyramid for scene image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7774–7782). http://dx.doi.org/10.1109/cvpr52688.2022.00762.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. http://dx.doi.org/10.1109/slt.2016.7846307, arXiv preprint arXiv:1409.1556.

Singh, S. A., & Desai, K. (2023). Automated surface defect detection framework using machine vision and convolutional neural networks. *Journal of Intelligent Manufacturing*, *34*(4), 1995–2011. http://dx.doi.org/10.1007/s10845-021-01878-w.

Song, Q., Li, J., Guo, H., & Huang, R. (2023). Denoised non-local neural network for semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, http://dx.doi.org/10.1109/tnnls.2022.3214216.

Su, B., Zhou, Z., & Chen, H. (2022). PVEL-AD: A large-scale open-world dataset for photovoltaic cell anomaly detection. *IEEE Transactions on Industrial Informatics*, *19*(1), 404–413. http://dx.doi.org/10.1109/tii.2022.3162846.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR, http://dx.doi.org/10.48550/arXiv.1905.11946.

Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627–9636). http://dx.doi.org/10.1109/iccv.2019.00972.

Ultralytics (2020). YOLOv5. [Online]. Available: https://github.com/ultralytics/yolov5.

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. http://dx.doi.org/10.48550/arXiv.1607.08022, arXiv preprint arXiv:1607.08022.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7464–7475). http://dx.doi.org/10.1109/cvpr52729.2023.00721.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803). http://dx.doi.org/10.1109/cvpr.2018.00813.

Wang, X., Zhang, Z., Xu, Y., Zhang, L., Yan, R., & Chen, X. (2022). Real-time terahertz characterization of minor defects by the YOLOX-MSA network. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–10. http://dx.doi.org/10.1109/tim.2022.3201945.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision* (pp. 3–19). http://dx.doi.org/10.1007/978-3-030-01234-2_1.

Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., et al. (2020). Disentangled non-local neural networks. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XV 16* (pp. 191–207). Springer, http://dx.doi.org/10.1007/978-3-030-58555-6_12.

Yu, J., Cheng, X., & Li, Q. (2021). Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–10. http://dx.doi.org/10.1109/tim.2021.3136183.

Yu, X., Lyu, W., Zhou, D., Wang, C., & Xu, W. (2022). ES-Net: Efficient scale-aware network for tiny defect detection. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–14. http://dx.doi.org/10.1109/tim.2022.3168897.

Zhang, L., Chen, M., Arnab, A., Xue, X., & Torr, P. H. (2022). Dynamic graph message passing networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(5), 5712–5730. http://dx.doi.org/10.1109/cvpr42600.2020.00378.

Zhang, S., Chi, C., Yao, Y., Lei, Z., & Li, S. Z. (2020). Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9759–9768). http://dx.doi.org/10.1109/cvpr42600.2020.00978.

Zhang, H., Wang, Y., Dayoub, F., & Sunderhauf, N. (2021). Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8514–8523).

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4203–4212).

Zhang, Q.-L., & Yang, Y.-B. (2021). Sa-net: Shuffle attention for deep convolutional neural networks. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 2235–2239). IEEE, http://dx.doi.org/10.1109/icassp39728.2021.9414568.

Zhi, Z., Jiang, H., Yang, D., Gao, J., Wang, Q., Wang, X., et al. (2023). An end-to-end welding defect detection approach based on titanium alloy time-of-flight diffraction images. *Journal of Intelligent Manufacturing*, *34*(4), 1895–1909. http://dx.doi.org/10.1007/s10845-021-01905-w.

Zhu, B., Wang, J., Jiang, Z., Zong, F., Liu, S., Li, Z., et al. (2020). Autoassign: Differentiable label assignment for dense object detection. http://dx.doi.org/10.48550/arXiv.2007.03496, arXiv preprint arXiv:2007.03496.