Journal of Manufacturing Processes 114 (2024) 196–212

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

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

An integrated defect detection method based on context encoder and perception-enhanced aggregation for cylinder bores

Xujie He, Jing Jin *, Duo Chen, Yiyuan Feng

School of Astronautics, Harbin Institute of Technology, Harbin 150001, China

ARTICLE INFO

Keywords:
Defect detection
Cylinder bore
Automobile engine
Locally sensitive and globally covered integrated encoder
Perception-enhanced feature path aggregation

ABSTRACT

The detection of defects in cylinder bores is crucial for the industrial manufacturing of automobile engines. Current efforts which exhibit unstable accuracies, time inefficiencies and high-cost expenditures, have been mainly initiated by well-trained inspectors. In this paper, we build on the detection framework FasterRCNN to propose an integrated defect detection method based on context encoder and perception-enhanced aggregation for cylinder bores of automobile engines, named CBDetector. The improvements are twofold. The context encoder, which is a locally sensitive and globally covered integrated encoder composed of a CNN and an improved Transformer architecture used for well-rounded feature extraction, is proposed. To robustly perceive full scale cylinder bore defects, a perception-enhanced feature path aggregation unit is introduced. Extensive experiments conducted on our established dataset HIT-EngD and a public steel dataset NEU-DET demonstrate the SOTA performance of the CBDetector, with mAP^{50} increases of 22.7 and 7.8 compared to FasterRCNN integrated with Feature Pyramid Network on HIT-EngD and NEU-DET. Moreover, our method can run at a high frame rate (~10 FPS, Nvidia-A100).

1. Introduction

Defect detection is an important and indispensable aspect of industrial manufacturing. Current product defect detection is accomplished primarily by professionally trained inspectors. However, manual detection results can easily be impacted by subjective factors. Moreover, manual detection exhibits low automation, low productivity and high expenditure. With the rapid development of computer vision and industrial automation, exploiting computer vision to replace manual vision is gradually becoming a mainstream strategy for industrial defect detection. Therefore, studying defect detection task using visual image-based defect detection methods to analyze each current workpiece, which improves the detection accuracy while reducing the manual inspection cost, is currently particularly important.

To our knowledge, current research regarding defect detection can be categorized into two types based on the methods used: traditional defect detection methods and deep learning-based defect detection methods. Traditional defect detection methods typically utilize handcrafted features to detect defects. Three subtypes of traditional defect detection methods currently exist, including texture-based defect detection methods [1], shape-based defect detection methods [2] and color-based defect detection methods [3]. Traditional defect detection methods do not require very large datasets to learn features, but they exhibit poor adaptability. For example, textures are unstable to noise, shapes are determined by the templates used and colors are easily impacted by illumination. Due to the powerful feature extraction ability and model-object agnostic nature of neural networks, methods employing deep learning techniques are currently predominating other methods. Current deep learning-based defect detection methods include supervised defect detection methods, such as representation learning [4], which categorizes defect detection into classification [5], detection [6,7] and segmentation [8], such as anomaly detection [9] tasks, and metric learning [10], which learns to formulate the similarity between defective images and defect-free images; unsupervised defect detection methods [11], such as one-class learning [12]; and semisupervised defect detection methods [13,14], which use image-level annotation to achieve segmentation- and localization-level detection. However, these methods are still not adaptable for cylinder bore detection in automobile engines due to the particularity these bores, such as:

1) *Tiny defects*. Tiny defects, such as sandholes, appearing in ultrahigh-resolution images result in a smaller percentage of defects on an

* Corresponding author.
E-mail address: jinjinghit@hit.edu.cn (J. Jin).

https://doi.org/10.1016/j.jmapro.2024.02.006
Received 24 October 2023; Received in revised form 2 February 2024; Accepted 3 February 2024
Available online 10 February 2024

image, introducing additional detection difficulties, as shown in Fig. 1(a).

2) *Cluttered defects*. Splashes of oil and rusts mixed with crack defects on the surface of engine bores appear in a dense state, as shown in Fig. 1 (b), which makes accurately detecting all defect locations difficult.
3) *Illumination variation*. Illumination is a key factor in the performance of defect detection; for example, in overexposed and underexposed images, defects tend to be partially lost or the entire defect is lightened up or darkened down, as shown in Fig. 1 (c). Overcoming illumination variation is of great necessity for stable defect detection.

Given the peculiar problems illustrated above, we ultimately follow the detection-based defect detection paradigm based on deep learning techniques. Specifically, we present an integrated defect detection method based on context encoder and perception-enhanced feature path aggregation unit for cylinder bores of automobile engines. Our method starts with the basic FasterRCNN [15] but differs from this method. We first propose a locally sensitive and globally covered integrated encoder (LG-IE) to introduce global context for the image scenes of ultrahigh-resolution images. LG-IE enables the method to better capture long-range contextual information within and between defects, such as

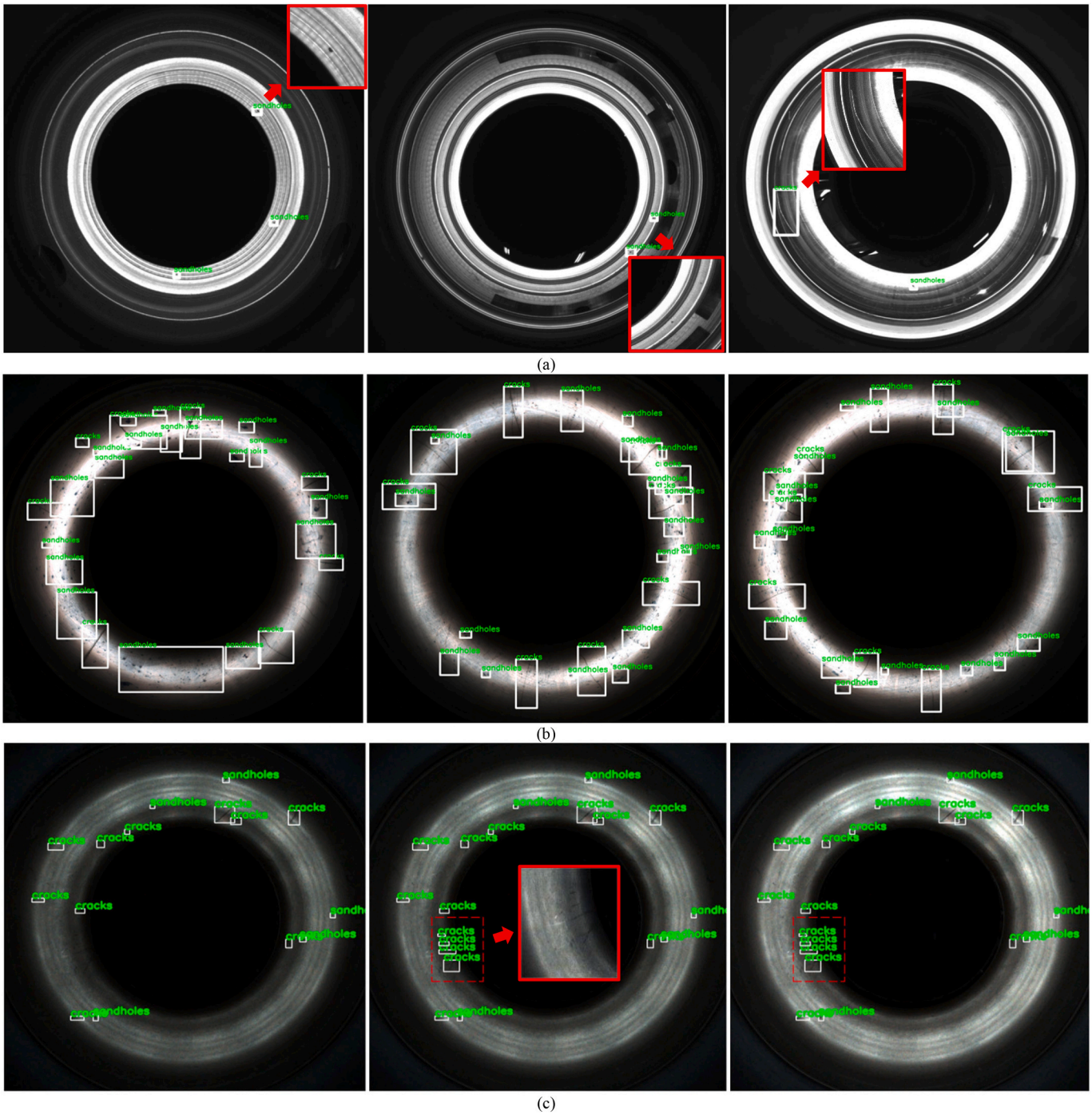

Fig. 1. Illustration of the unique challenges arising in defect detection for cylinder bores. The defects in the cylinder bores in (a), referred to as sandholes, are tiny; the defects in (b) are cluttered; and the defects in (c) can be obscured by overexposure or underexposure of the images caused by variations in illumination. For example, in the first image in (c), four crack defects located in the lower left corner of the image cannot be observed because of low illumination, making them difficult to detect. Best viewed in color.

clutter defects, to further augment the selection ability for valid features. Moreover, to further enhance the ability of the model to locate and detect defects of full scale, such as sandholes, we introduce a perception-enhanced feature path aggregation unit (PE-FPA) and utilize it to locate defects from multiscale features. Aiming to better balance performance and time latency while addressing tiny defects, optimization of the hierarchical structure of the LG-IE module is additionally conducted. To verify the effectiveness of the methods for cylinder bore defects, we collect the first visual image-based cylinder bore defect dataset of automobile engines using our *Product Condition Surveillance Systems,* as shown in Fig. 2, and execute careful annotation on it. The dataset, named HIT-EngD, contains three types of defects, including cracks, sandholes and bumps on all kinds of automobile engines. Experiments on HIT-EngD and a public steel defect dataset NEU-DET demonstrate the SOTA performance of our method, which yields mAP^{50} increases of 22.7 and 7.8 when compared to FasterRCNN.

To summarize, the contributions of our work include the following:

1) An integrated defect detection method for cylinder bores of automobile engines is proposed, integrating the proposed locally sensitive and globally covered integrated encoder (LG-IE) and perception-enhanced feature path aggregation (PE-FPA) unit into a single pipeline.
2) The first visual image-based cylinder bore defect dataset of automobile engines, HIT-EngD, is collected and annotated. The HIT-EngD will be made publicly available to encourage further research.
3) Extensive experiments are carried out, demonstrating the effectiveness of the method in detecting defects in not only cylinder bores but also steel surfaces.

The remainder of this article is organized as follows. In Section 2, we review the literature on defect detection. In Section 3, we provide the details of the CBDetector. In Section 4, we report the experimental and comparison results. We discuss these results in Section 5. Finally, in Section 6, we draw conclusions and suggest possible future work.

2. Related work

Detection-based and segmentation-based (anomaly) defect detection methods should be given priority when defect location is needed. Given the peculiar problems of cylinder bores illustrated in Section 1, the segmentation-based method is not suitable, as it has proven a typical letdown in outlining boundaries [16], especially for tiny sandholes and slender cracks, the segmentation-based method makes no profits under these circumstances. Therefore, we adopt a detection-based method for cylinder bores. Excellent defect detection methods for cylinder bore defects and other defect detection problems should always involve the most suitable method frameworks, powerful feature extractors and all-

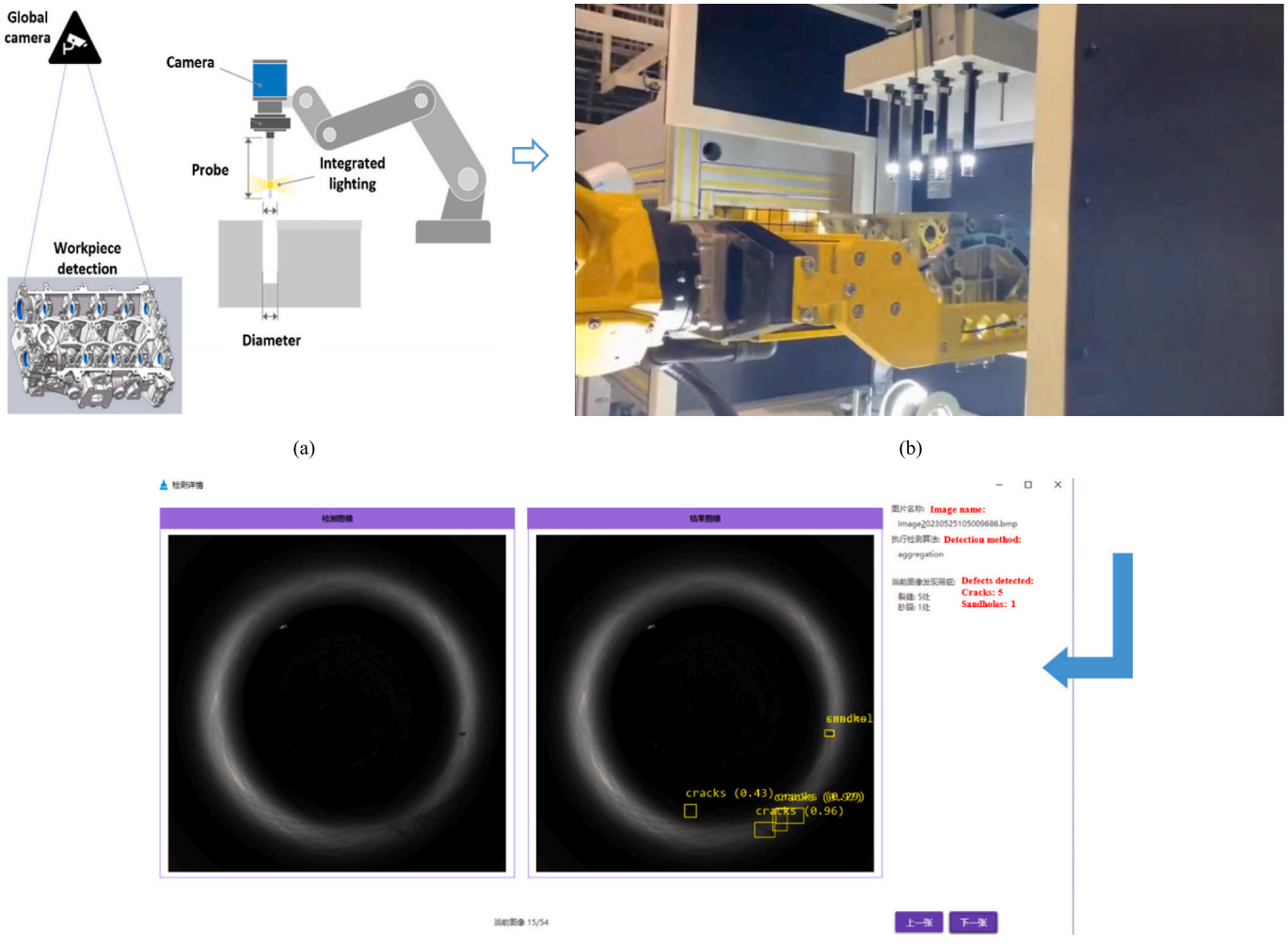

Fig. 2. Illustration of product condition surveillance systems.

sided defect localization methods. Moreover, since deep learning methods are currently the predominant methods [17,18] in many fields of computer vision, the above three segments, all revolving around deep learning processing, are detailed in this section.

2.1. Defect detection framework

Current frameworks that are widely used for defect detection can be categorized into two types: one-stage detectors and two-stage detectors. The SSD [19] and YOLO series [20–24] are one-stage detectors that are widely used because of their high efficiency. Both of these methods take the entire image as input and regress the location of the defects and the corresponding classes directly in the output layers. For example, [25] first proposed a new image adjustment technique to accurately reveal defects in a given radiographic image and subsequently performed detection using a proposed twin model-based defect detection method based on YOLO. In [26], YOLO integrated with a classifier is also deployed to detect defects in power system and exhibited promising accuracy. [27] addressed smaller defects on steel strip surfaces by using higher-resolution feature maps based on the YOLO framework. [28] optimized the backbone of SSD by using efficient MobileNet to detect the defects of catenary support devices. [29] introduced an attention module along with the fusion of multilayer features to detect PCB defects based on SSD. In contrast to one-stage detectors, two-stage detectors first deploy region proposal networks (RPNs) to obtain the regions containing defects and then apply ROI pooling and postmapping networks to regress their locations and classes. [30] was the first attempt to use FasterRCNN [15] for defect detection, exhibiting excellent performance in bridge surface defect detection. [31] integrated an improved Inception network together with a region proposal network and position-sensitive ROI pooling to detect lining defects in shield tunnels. [32] introduced the idea of domain adaptation into FasterRCNN to overcome the lack of available datasets and used the improved FasterRCNN to detect defects in spacecraft composite structures (SCSs). By using an existing public dataset with a large amount of data to represent the source domain and currently available SCS data to represent the target domain, the authors migrated information on the source domain to the target domain through domain adaptation, thus reducing the reliance of the model on the amount of available target-domain data. [33] used the Manhattan distance to obtain the optimal anchor sizes used in RPNs and adopted many data augmentation strategies to enhance the generalizability of the FasterRCNN in detecting defects of 3-D printed lattice structures. [34] also integrated Gabor kernels into the FasterRCNN to detect fabric defects. Because of its excellent detection ability, FasterRCNN is also widely applied in other defect detection fields, such as polarizer surfaces of LCD panels [35], steel rails [36] and thermal imaging insulator defect detection [37]. As the level of GPU hardware continues to evolve, defect detection accuracy is becoming a practical issue that should be prioritized during deployment. As a result, two-stage defect detection methods such as FasterRCNN are becoming more advantageous for addressing practical problems.

2.2. Feature extraction

Feature extraction as the first step is critical in defect detection, substantially influencing the final detection accuracy. Convolutional neural networks (CNNs) are being widely used as feature extractors in defect detection due to their powerful feature extraction capabilities of model and object agnostics. For example, [38] used an SVM and the VGG19 [39] network to detect defects in solar panels and demonstrated that the feature extraction ability of VGG19 outperformed that of many handcrafted feature descriptors, such as the scale-invariant feature transform (SIFT) and the Speeded-Up Robust Features (SURF) descriptor, as well as the SVM. [40] proposed a two-stage rail defect detection method, in which the rail region is first cropped out of each input image and InceptionV3 [41] is subsequently used to extract features for classifying the cropped rail images as defective or defect-free images. [42] used MobileNet [43–45], ResNet [46] and VGG16 [39] to detect defects in PCBs and extensively compared the defect detection performance of the different CNNs. [47] used ShuffleNetV2 [48] to classify defects of inkjet codes of bottles under complex scenes. Other lightweight CNNs have also been employed in defect detection, such as SqueezeNet [49]. The successful introduction of the transformer mechanism in computer vision has prompted some researchers to gradually introduce this mechanism to defect detection. For example, [50] deployed a SOTA transformer-based detection method named DETR [51] to detect defects in sewer pipes. [52] used the Swin Transformer [53] as a feature extractor and followed the FasterRCNN paradigm to detect defects in cylinder liners while addressing the overdetection issue with a masking mechanism. The work presented in [54] is similar to that in [52]; the difference is that [54] further utilized feature layers at 5 scales to detect defects on steel surfaces. [55] proposed a UNet-shaped model named LETNet based on the Transformer architecture to segment pavement cracks in images. Since CNNs are not equipped with long-range modeling capabilities and Transformers suffer from efficiency issues, a series of hybrid structures have been proposed [70,71]. [9] was the first attempt to stack 2D convolutions in front of each Transformer block to address anomaly detection. The convolved features were then reshaped into a one-dimensional form and input into subsequent attention modules, realizing the hybridization of the CNN and Transformer algorithms. This approach yielded promising performance, but processing a single image still required 0.17 s (P5000/GPU). The approach of placing convolutions ahead of Transformer blocks was also adopted in [55] for detecting salient pavement cracks. Nevertheless, relatively few attempts have been made in the field of defect detection to utilize hybrid structures for feature extraction. We argue that although Transformers have powerful feature extraction capabilities, their processing inefficiency and data-hungry nature limit their practical application. Therefore, considering how to effectively combine the strengths of CNNs and Transformers to detect defects in specific scenarios while addressing model efficiency concerns is a promising direction for designing future defect detectors.

2.3. Defect localization

Regressing defect locations from single-sized feature maps and aggregating defects from multiscale multifeatures are two currently prevailing strategies for addressing defect localization. Substantial current studies and experimental results reveal that defect locations from multiscale feature maps are more all-rounded than single-scale feature maps [18]. Due to the hierarchical structure of CNNs, the feature maps at the final output are often condensed with abundant semantic information, which is also referred to as advanced features. However, because of the low resolution of advanced feature maps, fewer details tend to be retained in advanced features than in other features; therefore, the performance for small defect detection is consequently unsatisfactory, and most of the time, the scale invariance of the model with respect to defects originates only from the training data. To make models scale-agnostic, [56] first attempted to detect objects of different scales by performing pooling operations with different stride sizes on the last layer of feature maps. [19] targeted objects in feature maps of different scales and directly output the object locations after convolutional layers instead of outputting location offsets. [57] proposed a top-down architecture to aggregate features from layers of different scales using elementwise addition for the purpose of detecting objects of different scales. In addition to [57,58] introduced detailed location information from the bottom layers back into the top layers in a bottom-up way and appended an adaptive pooling layer at the back end to aggregate the detection information of multiple features. [59] further optimized the structure of [58] by introducing learnable weights to classify the importance of different input features. However, because of the peculiarities of defects in cylinder bores, as mentioned in Section 1, off-the-

shelf defect detection methods are not entirely applicable to cylinder bore defect detection. Additionally, the following considerations are pertinent: 1) in two-stage detectors, the detection task is decomposed into two subtasks, implemented with a region proposal network and a region refinement module (location and category), making such detectors more advantageous than single-stage detection methods because they can obtain more accurate defect locations and categories in a coarse-to-fine manner; 2) two-stage detectors offer higher accuracy and precision than single-stage detectors, as not only reported in [72] but also experimentally demonstrated in Table 7; and 3) with increasing hardware capabilities, two-stage methods have also achieved excellent real-time performance (>10 FPS) and thus do not impact the overall speed of control in actual use. Therefore, many method developers prefer to deploy two-stage detectors in high-precision situations. Based on these considerations, we take FasterRCNN as the basic framework in this paper and improve it accordingly. The proposed CBDetector is the first robust defect detector for detecting defects in cylinder bores of automobile engines from multiple features (defect localization) using an integrated feature extraction structure.

3. CBDetector

3.1. Overview

CBDetector, which follows the two-stage framework of FasterRCNN, detects defects on the surface of cylinder bores in a coarse-to-fine manner. The proposed CBDetector comprises six components, including a preprocessing module (PpM), a preliminary feature extraction (PFE) module, a locally sensitive and globally covered integrated encoder (LG-IE), a perception-enhanced feature path aggregation (PE-FPA) unit, a region proposal network (RPN) and a region of interest align (ROIAlign) module. As shown in Fig. 3, the PpM is adopted to apply image transformations on the input image $I \in \mathbb{R}^{H\times W\times 3}$ to strengthen the model generalizability. Then, the transferred image $I_{trn} \in \mathbb{R}^{H\times W\times 3}$ is fed into the PFE module to project $\boldsymbol{I}_{trn}$ into high-dimensional space. After PFE, we introduce the LG-IE, which is stacked with four stages of E_1, E_2, E_3, and E_4 and is assembly integrated with CNN and improved Transformer to hierarchically encode the features of F_P obtained from PFE. To make our method scale-agnostic, the encoded features from multiple layers of multiple scales of the LG-IE are fed into the PE-FPA that is also composed of four tiers of H_1, H_2, H_3, and H_4 for feature aggregation. The RPN and ROIAlign module are jointly used to decode the encoded features of $F_{PE-FPA} \in \mathbb{R}^{M\times 256}$ of PE-FPA to regress and classify the defects in a coarse-to-fine manner.

In the following subsection, we elaborate on the newly proposed locally sensitive and globally covered integrated encoder (LG-IE) and perception-enhanced feature path aggregation (PE-FPA) unit.

3.2. Locally sensitive and globally covered integrated encoder

As mentioned above, feature extraction is a crucial step in defect detection. Therefore, to detect defects on the surface of cylinder bores, three improvements are first made regarding feature extraction:

1) *Integrated context encoder pipeline*. In most current high-performance defect detection methods, such as those in [25–27,29], CNNs such as ResNet50, or its variants obtained by adding bypass structures or fragments, are used for feature extraction. Transformer mechanisms

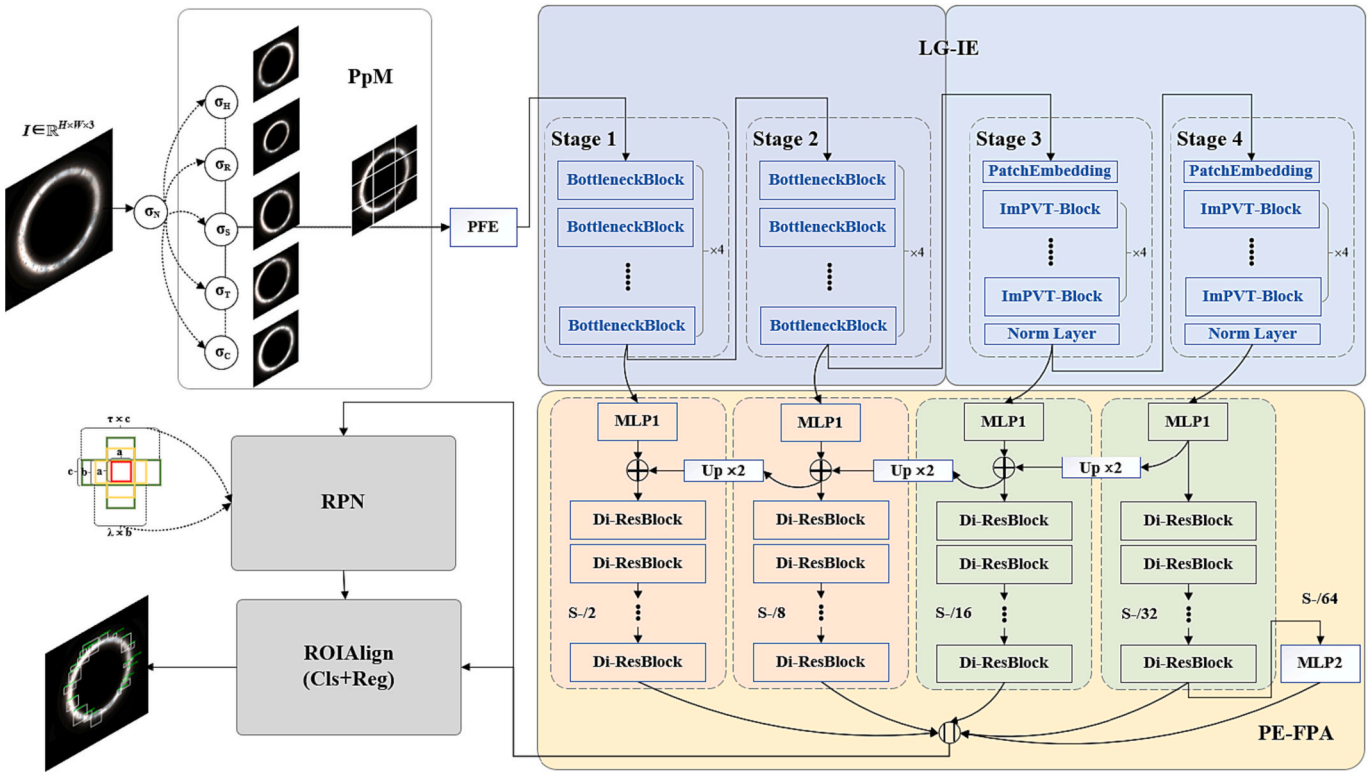

Fig. 3. Deployment of CBDetector. CBDetector takes one image $I \in \mathbb{R}^{H\times W\times 3}$ captured by the camera as input. $\boldsymbol{I}$ is passed into the preprocessing module (PpM) to randomly augment the input image. The augmented (transferred) image $I_{trn} \in \mathbb{R}^{H\times W\times 3}$ is then fed into the preliminary feature extraction (PFE) module to obtain the preliminary features of high-dimensional space F_P. The locally sensitive and globally covered integrated encoder ((LG-IE) is next adopted to encode the preliminary high-dimensional features. The encoded features of 4 scales (1/2, 1/8, 1/16 and 1/32) are passed into the perception-enhanced feature path aggregation (PE-FPA) unit to further aggregate the features from different scales to address the defects of different scales. Finally, we adopt the region proposal network (RPN) and ROIAlign to decode the encoded features $F_{PE-FPA} \in \mathbb{R}^{M\times 256}$ obtained from the PE-FPA and regress and classify the defect locations in a coarse-to-fine manner. Best viewed in color.

have also begun to be introduced into several excellent methods for defect detection; for example, ViT, Swin and PVT were adopted for this purpose in [50,52,54,55,60,61]. We argue that *i*) due to the inherent properties of CNNs, such as translation invariance, the locality property introduces inductive bias, which enables input images with unrestricted sizes and consequently CNNs them to be widely applied for computer vision tasks without introducing too much computational overhead. However, due to the limitation of the size of the convolutional kernels used, CNNs tend to capture the information of local regions and are unable to establish global long-distance connections of the image. Therefore, the perception of the entire image cannot be grasped. *ii*) In contrast, Transformer learns the interrelationship between features by the attention mechanism while retaining more spatial information with good general applicability; it is not completely dependent on the data itself. However, Transformer cannot use the prior knowledge of scale and translation invariance and the local traits of the image itself; it is therefore data-hungry and requires a large-scale dataset to learn high quality intermediate representations. We therefore propose the locally sensitive and globally covered integrated encoder (LG-IE) to jointly formulate context information of the local region and expand the connections to the entire image, as shown in Fig. 3.

The LG-IE module is hierarchically composed of four stages, E_1, E_2, E_3 and E_4, that jointly integrate CNNs and improved Transformer block into a single pipeline adhering to:

$$\widetilde{\mathbf{Y}} = \prod_{i=1}^{4} \circ E_i\left[\widetilde{\mathbf{X}}, \mathbf{W}_i^*\right] \tag{1}$$

where $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ are input and output, $\mathbf{W}_i^*$ indicate the learnable weights.

Considering that CNNs are more efficient at handling feature maps than Transformer is, we design E_1 and E_2 with CNNs and E_3 and E_4 with improved Transformer blocks. Specifically, E_1 and E_2 are employed with BottleneckBlock from ResNet to first locally encode features on large feature maps, which guarantees efficiency during feature extraction. Inspired by the efficient structure of PVT2 [62] and CBAM [63], E_3 and E_4 are eventually employed with a composite structure composed of patch embedding layers, IMPVT2 blocks and a normalization layer to make inferences on the relationships from pixels to defects and from defects to the global image background at long distances.

2) *IMPVT2 Block*. Feature sharpening can better benefit defect detection on some occasions, such as in small defect detection, than feature smoothing can. When smoothing is used to characterize features, small defect areas tend to be smoothed out due to their low proportion rate, but when sharpening is used, the continuity of the background raises semantic ruptures between the defect edge and background. These semantic ruptures should be captured as much as possible. We therefore propose the IMPVT2 Block with the improved multidimension multihead cross attention (MD-MHCA) module, as shown in Fig. 4.

Specifically, we enhance the effective feature extraction of the model for the regions of semantic ruptures by introducing the sharpening strategy. We still simultaneously adopt a feature smoothing strategy to maintain the feature integrity. Specifically, as shown in Fig. 4, to enhance the effective feature filtering over the semantic ruptures to capture the defect locations, we adopt average pooling to achieve smoothing and max pooling to achieve sharpening. At the same time, aiming to further strengthen the positional relations between features, we adopt channel average pooling and channel max pooling. Finally, we aggregate the features with positional information and the features

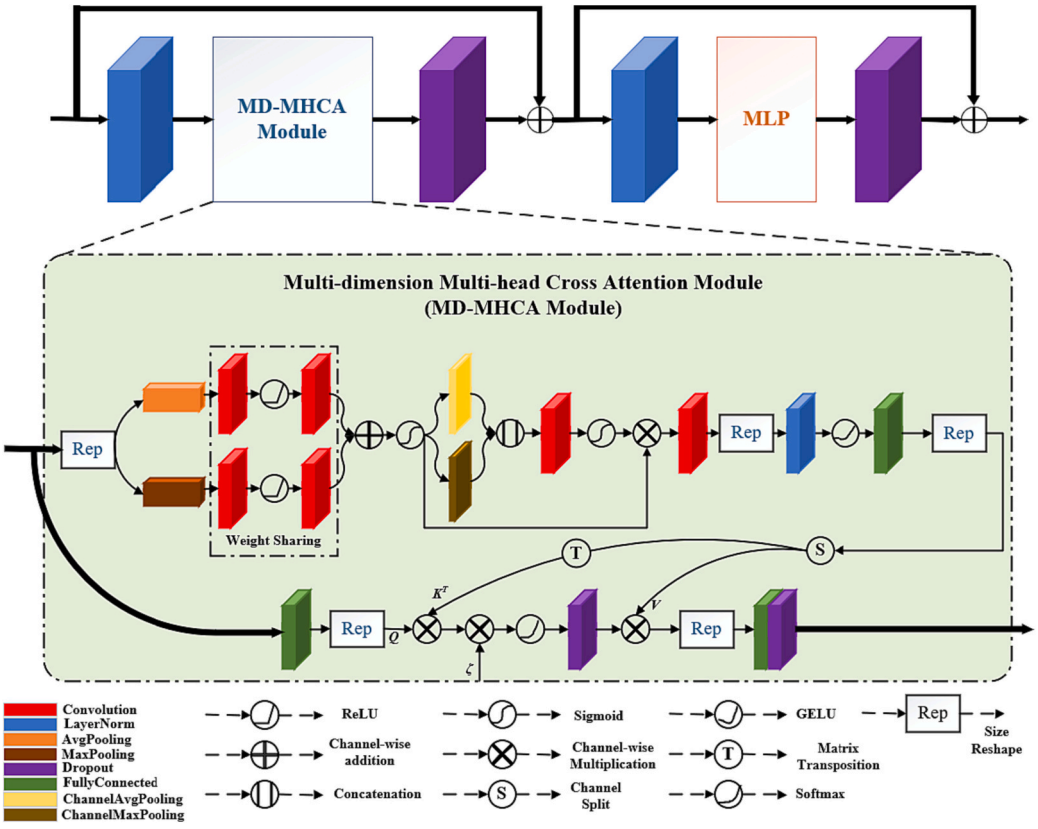

Fig. 4. Structure of IMPVT2. The IMPVT2 retains the pipeline of the original PVT2 block (shown in the upper part of the figure), but it differs in the MD-MHCA module used in the IMPVT2. The MD-MHCA takes the input as two separable streams and conducts feature smoothing and feature sharpening, extracting the representative features while also maintaining the integrity of the features. Best viewed in color.

smoothed and sharpened, and then pass these together to the later encoder units. The IMPVT2 can be formulated as:

$$\begin{aligned}\widetilde{U}^{i} &= \rho\{\mathrm{MD\_}MHCA[\mathrm{LN}(U^{i})]\} + U^{i}\\ U^{i+1} &= \rho\{\mathrm{MLP}[\mathrm{LN}(\widetilde{U}^{i})]\} + \widetilde{U}^{i}\end{aligned} \tag{2}$$

where MD_MHCA indicates the projecting function of the MD-MHCA module, ρ is the dropout operation that is used to overcome overfitting and LN denotes layer normalization. The MD-MHCA is formulated as follows:

$$\begin{aligned}&\mathrm{MD\_}MHCA(Q,K,V)\\ &\quad=\phi(Q,K,V)\times W^{\phi}\\ &\quad=\phi(Q_i,K_i,V_i)\,|_{\sum_{i=1}^{n}}\times W^{\phi}\\ &\quad=\varsigma\left(\frac{Q_iW_i^{Q}\times K_i^{T}(W_i^{K})^{T}}{\sqrt{\kappa}}\right)\times V_iW_i^{V}\Big|_{\sum_{i=1}^{n}}\times W^{\phi}\end{aligned} \tag{3}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ indicate the queries, keys and values that are encoded from the input and are used in the attention mechanism. ϕ is the attention mechanism, and $\boldsymbol{W}^{\phi}$ is its learnable weight. n is the number of heads used, and $\boldsymbol{WQ}\,i$, $\boldsymbol{WK}\,i$, and $\boldsymbol{WV}$ i are the learnable weights under different heads. ς indicates the *softmax* function, and κ is the clamping coefficient. Σ indicates the concatenation operation integrating the attentioned results of different heads.

The MLP used in Eq. (2) is stacked with two consecutive linear layers defined as follows:

$$\begin{aligned}\mathrm{MLP}(\overline{U}^{i}) &= W^{\mathrm{M}}\times\overline{U}^{i}+B^{\mathrm{M}}\\ &=\begin{bmatrix}\omega_{11}^{\mathrm{M}} & \cdots & \omega_{1k}^{\mathrm{M}}\\ \vdots & \ddots & \vdots\\ \omega_{k1}^{\mathrm{M}} & \cdots & \omega_{kk}^{\mathrm{M}}\end{bmatrix}\times\begin{bmatrix}\overline{u}_1^{i}\\ \vdots\\ \overline{u}_k^{i}\end{bmatrix}+\begin{bmatrix}\overline{b}_1^{\mathrm{M}}\\ \vdots\\ \overline{b}_k^{\mathrm{M}}\end{bmatrix}\end{aligned} \tag{4}$$

where $\boldsymbol{W}^{\mathrm{M}}$ is the learnable weight and $\boldsymbol{B}^{\mathrm{M}}$ indicates the bias used.

3) *Layer adjustments*. Considering model efficiency, two modifications are accordingly made regarding the overall number of layers of the LG-IE: *i*) we first shrink the expansion rate of the CNN bottleneck block to 2; *ii*) we compress the overall number of layers of the LG-IE from the commonly used [64, 256, 1024, 2048] to [64, 128, 320, 512], which greatly reduces the computational overhead of the model. For details of the LG-IE and PFE modules, please refer to Table 1.

3.3. Perception-enhanced feature path aggregation unit

As mentioned above, detecting defects in cylinder bores captured by ultrahigh-resolution images on differently scaled feature maps is of great necessity. Considering model efficiency, we propose the perception-enhanced feature path aggregation (PE-FPA) unit based on the FPN [57]. To detect defects at full scales, three adjustments are made, as shown in Fig. 5:

1) *Defects of different sizes are located from multiscale feature maps*. Specifically, feature maps of four different scales are adopted: 2×, 8×, 16×, and 32× downsampling feature maps, which are from PEN, E_2, E_3, and E_4, respectively. We allow small-scale feature maps to focus more on large defects and large-scale feature maps to focus more on small defects. We also further downsample the 32× downsampling feature maps by 2× to fully encompass large defects.
2) *Perception enhancement*. Small defects need a larger perception field to provide contextual information; otherwise, the missing alarming rate increases. We therefore further expand the perception field of the feature maps at different scales, accounting for efficiency. As reported in [64], the perception field matters in localization; therefore, we finally expand the perception field with dilated convolution without adding too much computational burden.

3) *Projection combination of bottleneck and anti-bottleneck layers*. To balance the relations between feature magnitudes, we use a combination of bottleneck and anti-bottleneck layers to further expand the perception field of feature maps for different scales. Specifically, we use a layer adjustment coefficient $\varepsilon = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4]$ to modulate intermediate layers. As shown in Fig. 6, we use anti-bottleneck layers of $\varepsilon_i < 1$ for large feature maps and bottleneck layers of $\varepsilon_i > 1$ for small feature maps to encode features.

Let $\boldsymbol{F} = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \boldsymbol{F}_3, \boldsymbol{F}_4\}$ be the features obtained from PFE, E_2, E_3 and E_4. Therefore, the output $\widetilde{U}_{F_j}^{3}$ for each feature map after the PE-FPA is formulated as follows:

$$\begin{cases}\widetilde{U}_{F_j}^{i} = \vartheta\left\{\mathrm{BN}\left[\hbar_{i,j}\left(\widetilde{X}_j^{i-1}\right)+B_j^{i}\right]\right\}, & i\in[1,3]\ and\ j\in[1,4]\\ \widetilde{U}_{F_j}^{3} = \max p\left(\widetilde{U}_{F_j}^{3}\right), & j=5\end{cases} \tag{5}$$

where:

$$\widetilde{X}_j^{i-1} = \begin{cases}\widetilde{U}_{F_j}^{i-1}+\ell\left(\widetilde{U}_{F_{j+1}}^{i-1}\right), & i=1\ and\ j\in[1,3]\\ \widetilde{U}_{F_j}^{i-1}, & i\neq 1\ or\ j=4\end{cases} \tag{6}$$

$$\hbar_{i,j}(Y) = \begin{cases}\mathrm{conv}(Y), & i=2\\ \mathrm{d\_}conv_{|d=\varepsilon_j}(Y), & i\neq 2\end{cases} \tag{7}$$

$$\widetilde{U}_{F_j}^{0} = \mathrm{conv}\left(U_{F_j}\right)+B_j^{0} \tag{8}$$

where ϑ, ℓ, conv and d_conv denote ReLU activation, interpolation, convolution and dilated convolution, respectively. For details of the PE-FPA, please refer to Table 2.

3.4. Preprocessing module

As shown in Fig. 3, to overcome the overfitting problem, the preprocessing module (PpM) used to augment images is also integrated before feature extraction. In this paper, we adopt normalization, rotation, translation, horizontal flipping, shear and color jitter strategies. All augmenting strategies except for normalization are randomly performed according to a uniform distribution on [0,1] (performing the corresponding operation when the sampling value is >0.5). The normalization strategy used is given by:

$$\sigma_N = \frac{I - I_{min}}{I_{max} - I_{min}} \tag{9}$$

where $\boldsymbol{I}_{min}$ and $\boldsymbol{I}_{max}$ are the minimum and maximum pixel values of input

Table 1
Details of the PFE and LG-IE Modules. Note that C, BN, R, and LN indicate convolution, batch normalization, ReLU and layer normalization, respectively. Subscripts of C indicate kernel size.

Layer name	Operation	In channels	Out channels
PFE	C_7-BN-R	3	64
E_1	Bottleneck Block×3	64	128
E_2	Bottleneck Block×4	128	256
	PatchEmbedding Layer	256	320
E_3	IMPVT2 Block	320	320
	LN Layer×6	320	320
	PatchEmbedding Layer	320	512
E_4	IMPVT2 Block	512	512
	LN Layer×3	512	512

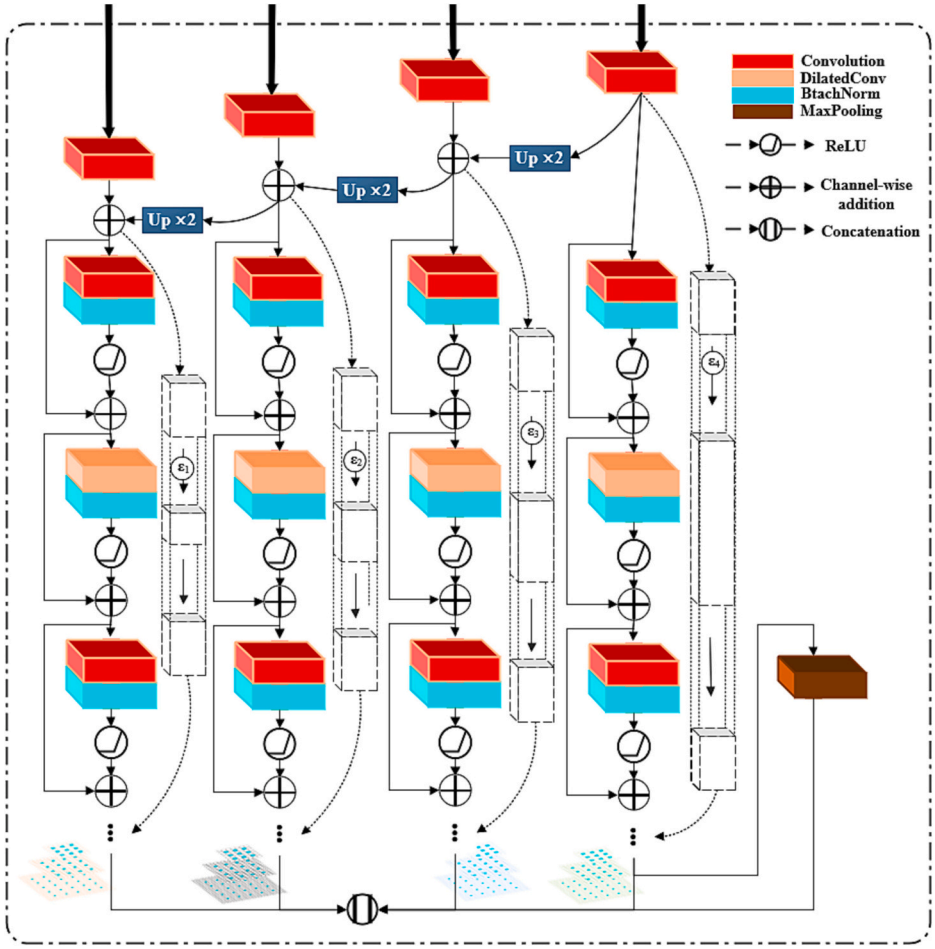

Fig. 5. Structure of the PE-FPA. The PE-FPA takes the features of four scales of the LG-IE as input. Features of different scales are aggregated and further encoded using different perception fields in a combination of bottleneck and anti-bottleneck methods. Best viewed in color.

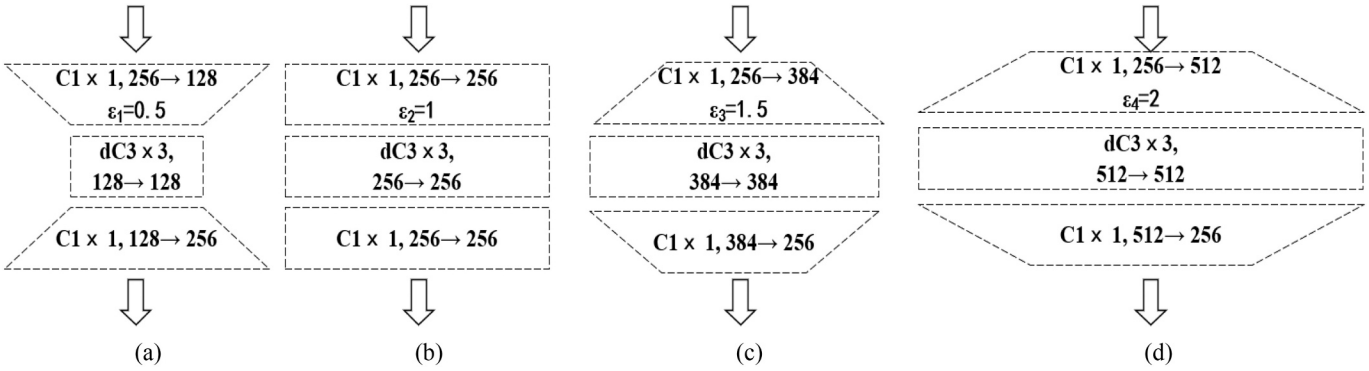

Fig. 6. Illustration of four necks used in PE-FPA. The formation of the necks is determined by the adjustment coefficient ε; when ε is >1, the neck is rendered in an expansion manner, while when ε is <1, the neck is rendered in a compression manner. Best viewed in color.

image $\boldsymbol{I}$, respectively. The rotation strategy used is given by:

$$\sigma_R = \begin{cases} x_2 = x_1 \times cos(\Delta\alpha) - y_1 sin(\Delta\alpha) \\ y_2 = x_1 \times sin(\Delta\alpha) + y_1 cos(\Delta\alpha) \end{cases} = \begin{bmatrix} cos(\Delta\alpha) & -sin(\Delta\alpha) \\ sin(\Delta\alpha) & cos(\Delta\alpha) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{10}$$

where (x_2, y_2) are the output positions after rotating (x_1, y_1) by $\Delta\alpha$. The translation strategy used is given by:

$$\sigma_T = \begin{cases} x_2 = x_1 + \Delta x \\ y_2 = y_1 + \Delta y \end{cases} = \begin{cases} x_2 = 1 \times x_1 + 0 \times y + 1 \times \Delta x \\ y_2 = 0 \times x + 1 \times y_1 + 1 \times \Delta y \end{cases} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{11}$$

where Δx and Δy are the horizontal and vertical offsets, respectively. The horizontal flipping strategy used is given by:

Table 2
Details of the PE-FPA module. Note that DC indicates the dilated convolution, and subscripts indicate kernel size.

Layer name	Operation	In channels	Out channels
	C_1	64	256
2× DS	C_1-BN-R	256	128
($\varepsilon_1 = 0.5$)	DC_3-BN-R	128	128
	C_1-BN-R	128	256
	C_1	256	256
8× DS	C_1-BN-R	256	256
($\varepsilon_2 = 1$)	DC_3-BN-R	256	256
	C_1-BN-R	256	256
	C_1	320	256
16× DS	C_1-BN-R	256	384
($\varepsilon_3 = 1.5$)	DC_3-BN-R	384	384
	C_1-BN-R	384	256
	C_1	512	256
32× DS	C_1-BN-R	256	512
($\varepsilon_4 = 2$)	DC_3-BN-R	512	512
	C_1-BN-R	512	256
64× DS	M_1	256	256

$$\sigma_H = \begin{cases} x_2 = W - x_1 \\ y_2 = y_1 \end{cases} = \begin{cases} x_2 = -1 \times x_1 + 0 \times y_1 + W \\ y_2 = 0 \times x_1 + 1 \times y_1 + 0 \times W \end{cases} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ W \end{bmatrix} \tag{12}$$

where W is the width of the input image. The shear strategy used is given by:

$$\sigma_S = \begin{cases} x_2 = x_1 + y_1 \times sin(\Delta\eta) \\ y_2 = y_1 \times cos(\Delta\eta) \end{cases} = \begin{cases} x_2 = 1 \times x_1 + sin(\Delta\eta) \times y_1 \\ y_2 = 0 \times x_1 + cos(\Delta\eta) \times y_1 \end{cases} = \begin{bmatrix} 1 & sin(\Delta\eta) \\ 0 & cos(\Delta\eta) \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{13}$$

where $\Delta\eta$ is the shear angle. For color jitter, the image to be detected in RGB space is first augmented using gamma transformation, and then the augmented image is converted to HSV space. We next rectify the H channel and S channel using the correction coefficients λ_H and λ_S, respectively. The rectified image in HSV space is converted back to RGB space. Finally, we execute gamma transformation again over the reconverted image and obtain the final output image after color jitter.

3.5. Loss function

The overall loss function is $\mathscr{L} = \mathscr{L}_{\text{RPN}} + \mathscr{L}_{\text{ROI}}$ penalizing over RPN and final location regression in ROIAlign (as ROIAlign is integrated with regression head and classification head at the end). The $\mathscr{L}_{\text{RPN}}$ and $\mathscr{L}_{\text{ROI}}$ are all composed of a classification loss and a regression loss. In implementation, the cross entropy loss is employed as the classification loss and smooth L1 loss is employed as the regression loss. The calculation method adheres to:

$$\mathscr{L}_{\text{RPN/ROI}} = \lambda_{cls} \sum_i \mathscr{L}_{cls}(q_i, \widetilde{q}_i) + \lambda_{reg} \sum_i \mathbf{1}_{q_i} \cdot \mathscr{L}_{reg}(c_i, \widetilde{c}_i) \tag{14}$$

where q_i and $\widetilde{q}_i$ indicate the class predicted and class annotated, c_i and $\widetilde{c}_i$ are the location $\{x_i, y_i, w_i, h_i\}$ predicted and annotated, respectively. λ_{cls} and λ_{reg} are two balancing coefficients. $\mathbf{1}_{qi}$ indicates that only predictions belonging to foreground objects are penalized. The $\mathscr{L}_{cls}$ adheres to:

$$\mathscr{L}_{cls}(q_i, \widetilde{q}_i) = -\sum_{q_i}^{Q} \widetilde{q}_i \cdot log\left(\frac{exp(q_i)}{\sum_{q_i}^{Q} exp(q_i')}\right) \tag{15}$$

The $\mathscr{L}_{reg}$ adheres to:

$$\mathscr{L}_{reg}(c_i, \widetilde{c}_i)2 = Smooth_{L1}(c_i - \widetilde{c}_i) = \begin{cases} 0.5(c_i - \widetilde{c}_i)^2 \Big/ \varepsilon^2, & \text{if } |c_i - \widetilde{c}_i| < 1 \Big/ \varepsilon^2 \\ |c_i - \widetilde{c}_i| - 0.5, & \text{otherwise} \end{cases} \tag{16}$$

where $|\bullet|$ indicates taking absolute magnitude, ε is a coefficient set to 1 in this paper.

4. Experiments

4.1. Dataset

Applying industrial computer vision technology to the defect detection of cylinder bores of automobile engines is an emerging and significant task. However, no dataset is currently available for this task, so we collect data on cylinder bore defects using the defect detection systems developed by our research team (Fig. 2). The collected data are carefully annotated by our team fellows. We name the collected and annotated dataset HIT-EngD (Harbin Institute of Technology Engine Dataset), as shown in Fig. 7. To our knowledge, this is the first visual image-based defect detection dataset of cylinder bores of automobile engines. To encourage further research, we plan to make this dataset publicly available to the community. Moreover, to further validate the effectiveness of our method, we also conduct experiments on a public steel surface defect dataset named NEU-DET [65]. The details of these two datasets are as follows:

HIT-EngD contains 8800 images with a superhigh resolution of 4000 × 4000 collected by *an optical* lens, of which 8175 images are defective and 625 are defect-free. All of the images are collected from nearly 200 different kinds of cylinder bores, which can overlay the current common cylinder bore types. HIT-EngD currently contains 3 types of defects. 61,736 defects have been annotated, including 38,045 cracks, 23,146 sandholes and 545 bumps, as shown in Table 3. HIT-EngD covers common but challenging scenarios, such as tiny defects (sandholes) under ultrahigh-resolution images, dense defects, and blurring defects resulting from illumination variation, as previously mentioned in Section 1. Therefore, conducting experiments on HIT-EngD is challenging.

NEU-DET encompasses defects on the steel surface captured from hot-rolled steel plates. This dataset provides examples of six types of steel defects on hot-rolled steel plates. 300 images exist of each defect, and each image has a resolution of 200 × 200. The defect types include pitted surface, scratches, crazing, rolled-in scales, patches and inclusions. Nearly 5000 defects exist in NEU-DET.

4.2. Evaluation metrics

Four widely used metrics, including precision, recall, AR (average recall) and the comprehensive metric mAP (mean average precision), are adopted to evaluate the performance of the methods in detecting defects. The precision and recall are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{17}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{18}$$

where TP, FP, and FN indicate the number of true positives, false positives, and false negatives, respectively. Therefore, the AR is given by:

$$\text{AR} = 2 \times \int_{0.5}^{1} R_i d(R_i) \tag{19}$$

The mAP is given by:

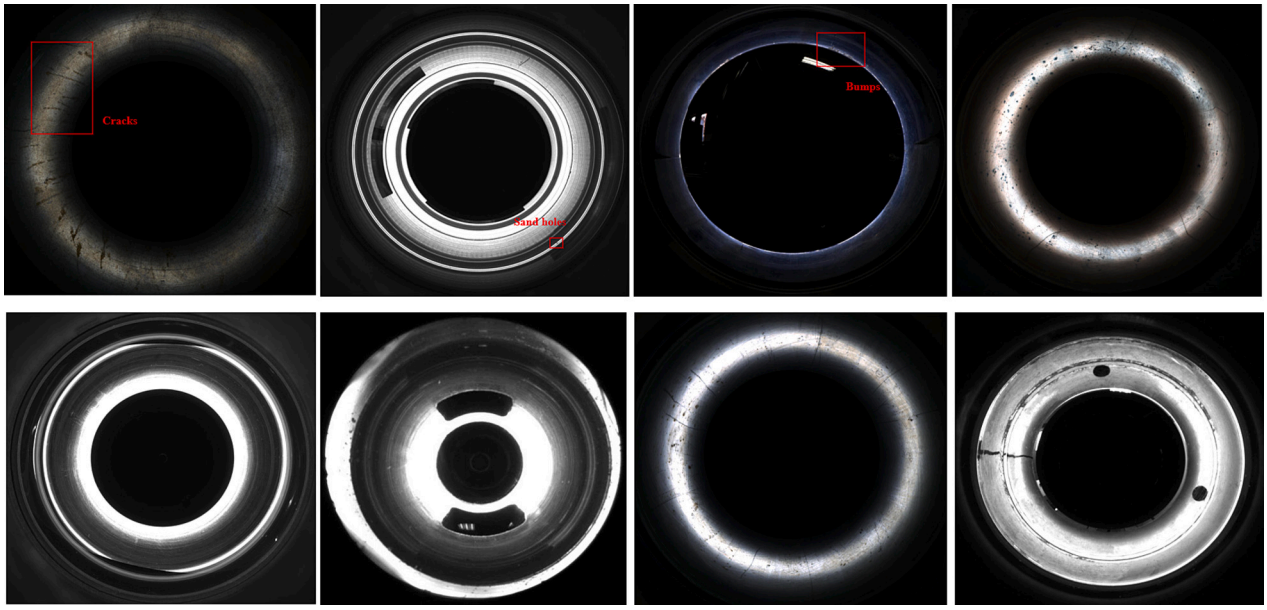

Fig. 7. Eight sample images from HIT-EngD. As shown in this figure, there are currently three types of defects represented in the dataset, namely, *cracks* (the first image in the first row), *sandholes* (the second image in the first row) and *bumps* (the third image in the first row). Because of the ultrahigh resolution and illumination variations, defect detection in these cylinder bore images is challenging. Best viewed in color.

Table 3
Details of HIT-EngD.

Type	Samples	Totality	Cracks	Sand holes	Bumps
Train	7040	49,390	30,436	18,518	436
Test	1760	12,346	7609	4628	109

$$\mathrm{mAP} = \frac{\sum_{i=1}^{N} \mathrm{AP}_i}{N} = \frac{\sum_{i=1}^{N} \int_0^1 P_i(R_i) d(R_i)}{N} \tag{20}$$

where R denotes the metric recall and $P(R)$ indicates the curves formed by the metric precision and metric recall. The integral intervals are acquired by setting different thresholds using IoU similarity from 0 to 0.5 and 0 to 1. N indicates the number of defect classes.

4.3. Ablation study on integrated encoder

4.3.1. Experimental design

We first conducted a group of ablation experiments on the integrated encoder over our established HIT-EngD. To further validate the effectiveness of the methods, we performed the same ablation study on NEU-DET. Specifically, we started with the pure CNN architecture (ResNet50) and gradually replaced the CNN block with the proposed Transformer mechanism.

4.3.2. Configurations

All the methods involved were trained with 100 epochs. We used AdamW to optimize the model. The initial learning rate (lr_0) was set to 2e-4 and was decayed at the 75th epoch. The batch size was set to 32 with GPU memory. We also adopted the loss penalty used in FasterRCNN [15] to enable a fair comparison.

4.3.3. Results

1) The results conducted on HIT-EngD are shown in Table 4. We also plotted these results into a line graph, as shown in Fig. 8(a). When using the integrated encoder to extract features, *i*) In terms of comprehensive metric mAP^{50}, Model_3 exhibits the best performance of 93.9, with an increase of 18.9 compared to the baseline (FasterRCNN integrated with FPN); Model_2 exhibits a competitive performance of 91.2, with an increase of 16.2 compared to the baseline; and Model_1 yields an increase of 2 compared to the baseline. *ii*) When the number of IMPVT2 modules is <3, the defect detection performance is positively correlated with the number of IMPVT2 modules. However, when the four ResNet blocks are entirely replaced with IMPVT2 modules, the defect detection performance characterized by mAP^{50} starts to decrease (85.5), but the model still outperforms the baseline, as shown in Fig. 8(a). *iii*) The increasing rate slows after two IMPVT2 modules are used.
2) Moreover, as the results conducted on NEU-DET in Table 4 and in Fig. 8(a) show, *i*) In terms of mAP^{50}, Model_2 also yields the most competitive and best defect detection performance of 78.0, with an increase of 6.1 compared to the baseline (FasterRCNN integrated with FPN). *ii*) When a single IMPVT2 module is adopted to replace a ResNet block, Model_1 yields a result of 74.2, with an increase of 2.3 compared to the baseline. When two IMPVT2 modules are adopted, this result continues increasing and reaches its maximum value (Fig. 8(a)), with an increase of 6.1 compared to the baseline. When 3 IMPVT2 modules are added, the results begin decreasing, but Model_1 still outperforms the baseline. However, when the ResNet

Table 4
Ablation experimental results on the integrated encoder. The baseline adopted is the FasterRCNN integrated with the FPN network. Higher results are expected in metric mAP^{50}. √ indicates that the current block is adopted, and the number after '×' indicates the number of current blocks used.

Data	Models	ResNet	IMPVT2	mAP^{50}
HIT-EngD	Baseline	√, ×4		75.0
	Model_1	√, ×3	√, ×1	77.0
	Model_2	√, ×2	√, ×2	91.2
	Model_3	√, ×1	√, ×3	93.9
	Model_4		√, ×4	85.5
NEU-DET	Baseline	√, ×4		71.9
	Model_1	√, ×3	√, ×1	74.2
	Model_2	√, ×2	√, ×2	78.0
	Model_3	√, ×1	√, ×3	76.1
	Model_4		√, ×4	60.1

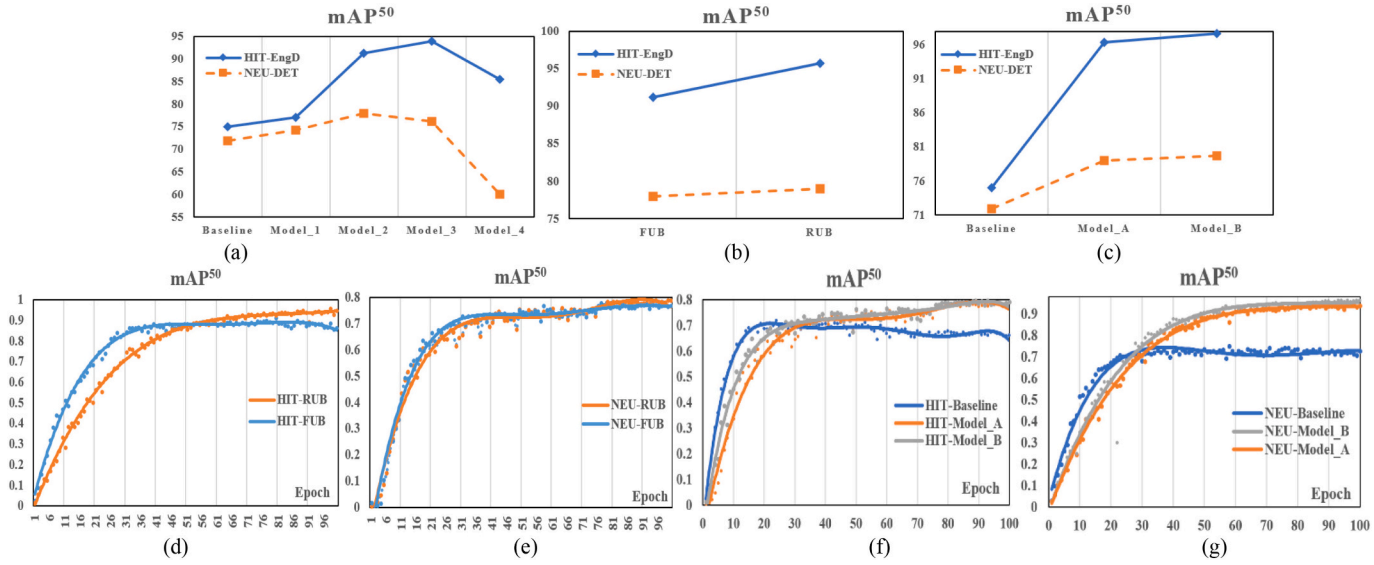

Fig. 8. Illustration of the ablation studies. Figure (a) plots the mAP^{50} results of the ablation study on the integrated encoder, (b) plots the mAP^{50} results of the ablation study on scaling strategies, and (c) plots the mAP^{50} results of the ablation study on CBDetector. (d) ~ (g) further plot the trends of each model of each ablation study throughout the training process in terms of the mAP^{50} results. Best viewed in color.

blocks are entirely replaced with 4 IMPVT2 modules, the results of Model_1 become inferior to those of the pure CNN structure. Therefore, after considering the defect detection performance and efficiency, we select the integrated encoder with two ResNet blocks and two IMPVT2 modules; we refer to this encoder as the LG-IE module.

4.4. Ablation study on image scaling strategies

4.4.1. Experimental design

To better capture defects of different sizes, we introduce two different scaling strategies performed on the input images: the fixed upper bound (FUB) scaling strategy and the random upper bound (RUB) scaling strategy. The FUB scaling strategy fixes the upper bound of the shorter side of the input image to μ_1, and the RUB scaling strategy randomly samples μ_1 from 384 to 800 with a step of 32. Algorithm 1 details the scaling strategy.

Algorithm 1: Scaling strategy

Input: Current image $\boldsymbol{I}$,
initialized upper bound of shorter side μ_1,
initialized upper bound of longer side μ_2
Output: Scale ratio ν

min_side $\chi_1 \leftarrow \min(\boldsymbol{I}.height, \boldsymbol{I}.width)$
max_side $\chi_2 \leftarrow \max(\boldsymbol{I}.height, \boldsymbol{I}.width)$
scale ratio $\nu \leftarrow \mu_1 / \chi_1$
if $\nu \times \chi_2 > \mu_2$ **then**
    scale ratio $\nu \leftarrow \mu_2 / \chi_2$
end if
return scale ratio ν

4.4.2. Configurations and results

Experiments on the two scaling strategies are also conducted on HIT-EngD and NEU-DET. The model used is Model_2 as described in Table 4. Configurations remain consistent with those of the previous section. The experimental results are shown in Table 5 and are also plotted in a line graph in Fig. 8(b). When the RUB scaling strategy is used, the defect detection performance is quantitatively better on both datasets than when the FUB scaling strategy is used, yielding increases of 4.5 and 1, respectively, on HIT-EngD and NEU-DET. We also plotted the changing trend of the mAP^{50} with increasing epochs, as shown in Fig. 8(d) and (e). The FUB strategy results in an overall faster converging speed than the RUB strategy does on both datasets. Nonetheless, for the final optimal performance, RUB achieves a higher mAP^{50} on both datasets. Therefore, the RUB scaling strategy is used for all of the following experiments if not otherwise stated.

Table 5
Ablation experimental results on scaling strategies. The model adopted is the FasterRCNN integrated with two CNN blocks and two IMPVT2 blocks. Higher results are expected in metric mAP^{50}.

Data	Strategies	μ_1	μ_2	mAP^{50}
HIT-EngD	FUB	800	1333	91.2
	RUB	*randomm*	1333	95.7
NEU-DET	FUB	800	1333	78.0
	RUB	*randomm*	1333	79.0

4.5. Ablation study on CBDetector

4.5.1. Experimental design

To investigate the effectiveness of the two proposed modules, we conducted a group of ablation experiments on the structure of the CBDetector on HIT-EngD. Moreover, to validate the effectiveness of the proposed methods, we also performed the same experiments on NEU-DET. Specifically, we gradually added the LG-IE and the PE-FPA modules into the baseline (FasterRCNN).

4.5.2. Configurations

All the involved methods were trained with 200 epochs on HIT-EngD and 100 on NEU-DET because of the discrepancy of the dataset capacity. The initial learning rate (lr_0) was set to 2e-4 and was decayed to 2e-5 at the 175th epoch for HIT-EngD and the 75th epoch for NEU-DET. The other configurations were kept the same as in the previous sections.

4.5.3. Results

The results are listed in Table 6 and are also plotted in a line graph, as

Table 6
Ablation experimental results on CBDetector. PR indicates a pure ResNet structure.

Data	Models	PR	LG-IE	FPN	PE-FPA	mAP^{50}
HIT-EngD	Baseline	√		√		75.0
	Model_A		√	√		96.4
	Model_B		√		√	97.7
NEU-DET	Baseline	√		√		71.9
	Model_A		√	√		79.0
	Model_B		√		√	79.7

shown in Fig. 8(c). The experimental results demonstrate that 1) The performance of Model_A, the structure integrated with LG-IE and FPN, is substantially increased when compared to that of the baseline FasterRCNN integrated with FPN, with mAP^{50} increases of 21.4 on HIT-EngD and 7.1 on NEU-DET; when further using the proposed two modules, 2) Model_B yields the best results of 97.7 on HIT-EngD and 79.7 on NEU-DET, with increases of 22.7 and 7.8, respectively. Likewise, the changing trend of mAP^{50} with increasing epochs is also plotted in Fig. 8(f) and (g). FasterRCNN integrated with FPN has an overall faster convergence speed but an inferior final performance on both datasets, but the improved Model_A and Model_B achieve better mAP^{50} values than the baseline. Furthermore, Model_B, which jointly integrates LG-IE and PE-FPA, exhibits not only a faster converging speed but also a higher performance than Model_A.

4.5.4. Configurations

For a fair comparison, all of the methods involved adopt our preprocessing module (Section 3.4) to perform the same preprocessing operations on the input images. The remaining protocols are the same as those described in the previous sections.

4.5.5. Quantitative results

The configurations remain consistent with those of previous sections. The experimental results are listed in Table 7. Notably, 1) our proposed CBDetector yields the best performance among SOTA detection methods on HIT-EngD, with an mAP^{50} increase of 22.7 compared to FasterRCNN integrated with FPN and mAP^{50} increases of 23.2, 81.2, and 12.5 compared to YOLO5, deformable DETR and DINO, respectively. CBDetector also yields substantial increases in AR and AP under different defect scales (S, M and L indicate that the defect areas are $<32^2$, (32^2, 96^2], and $> 96^2$). 2) On the NEU-DET dataset, the CBDetector also yields competitive results of 79.7 in mAP^{50}, with an increase of 7.8 compared to the FasterRCNN with FPN. Furthermore, DINO achieves the best AR^S and AR^L results on NEU-DET.

We also list the defect detection results of FasterRCNN + FPN and CBDetector regarding each defect class in Table 8 and present the results in bar charts, as shown in Fig. 9, for better observation. CBDetector has achieved substantial improvements in each class of defect types on both HIT-EngD and NEU-DET.

4.5.6. Qualitative results

The qualitative results of FasterRCNN+FPN and our CBDetector on HIT-EngD and NEU-DET are visualized in Fig. 10 and Fig. 11, respectively. We visualize two groups of qualitative results for HIT-EngD, as shown in Fig. 10, and one group of qualitative results for NEU-DET, as shown in Fig. 11. All of the defects missed by FasterRCNN when applied to HIT-EngD are annotated with red dashed boxes. On HIT-EngD, 1) CBDetector achieves significantly improved performance in tiny defect detection in comparison to FasterRCNN+FPN, as shown in Fig. 10(a)∼(e), (h) and (j); 2) CBDetector possesses a better defect capture capability than FasterRCNN for cluttered scenes, as shown in Fig. 10(f) and (l); and 3) CBDetector has better illumination adaptability than FasterRCNN, whether the image is underexposed or overexposed, enabling CBDetector to accurately locate defects. The experimental results on the NEU-DET dataset also further demonstrate the ability of CBDetector to locate defects accurately and comprehensively.

5. Discussion

Combining Table 4 and Fig. 8(a) demonstrates that when the proposed IMPVT2 module is used to replace all the ResNet blocks, the defect detection performance decreases on both datasets (HIT-EngD, NEU-DET) and is even inferior to that of the FasterRCNN tested on NEU-DET. Furthermore, when the number of IMPVT2 modules used is greater than or equal to three, the increasing speed of mAP^{50} on HIT-EngD starts to decrease. Such results arise from the characteristics of the model structures [69]. The Transformer mechanism makes few assumptions about the structural information of the data, making the Transformer mechanism a general and flexible architecture. However, the lack of inductive bias [69] also causes Transformer to perform strongly on datasets of large scales, whereas CNNs strengthen the translation invariance and local inductive bias of the network by locally sharing convolutional kernels to incorporate prior knowledge. The experiments in this section also reveal that since our newly established HIT-EngD has 8800 samples while NEU-DET has only 1800, a substantial improvement is yielded on both HIT-EngD and NEU-DET when one or two IMPVT2 modules are used. However, when the number of IMPVT2 modules is increased, the defect detection performance on HIT-EngD further increases but at a slower rate, while the performance on the NEU-DET dataset begins decreasing. When IMPVT2 is used to replace all ResNet blocks, the defect detection performance decreases on both datasets. Therefore, we adopted the '2 + 2' integrated mode to preserve both the prior information and the long-distance dependency modeling capability.

The ablation experiments for image scaling strategies on both

Table 7
Comparison experimental results. The metrics used in this section are average recall (AR) and mAP^{50}. The superscripts S, M and L indicate small, medium and large defects, respectively. '*' indicates that models are pretrained using other large-scale datasets, such as ImageNet.

Data	Models	AR^S	AR^M	AR^L	mAP^S	mAP^M	mAP^L	mAP^{50}
HIT-EngD	FRCNN(+FPN)	25.1	38.4	47.9	35.4	45.8	54.9	75.0
	YOLO5*							74.5
	[51]	7.3	9.7	15.5	2.1	3.5	8.7	16.5
	[68]	48.3	60.0	71.2	40.7	47.4	59.7	85.2
	CBDetector	55.7	73.1	83.1	51.3	68.8	79.2	**97.7**
NEU-DET	FRCNN(+FPN)	52.0	44.2	45.7	33.1	28.2	31.8	71.9
	YOLO5*							68.9
	[51]	0.2	1.1	5.1	0	0	0	0
	[68]	51.5	48.8	69.8	7.3	8.0	13.5	14.5
	[22]*							78.1
	[66]*							72.4
	[60]*							**80.5**
	[67]*							79.1
	CBDetector	46.1	49.8	65.9	34.9	34.9	55.7	**79.7**

Table 8
Detection results regarding each class.

Data	Models	mAP^{50}						
		Cracks		Sandholes		Bumps		Average
HIT-EngD	FRCNN	77.9		70.7		76.3		75.0
	CBDetector	97.7		96.4		98.9		97.7
		pi	sc	cr	ro	pa	inc	average
NEU-DET	FRCNN	83.7	91.9	37.6	55.9	86.7	75.7	71.9
	CBDetector	88.2	95.8	48.2	71.1	91.3	83.8	f79.7

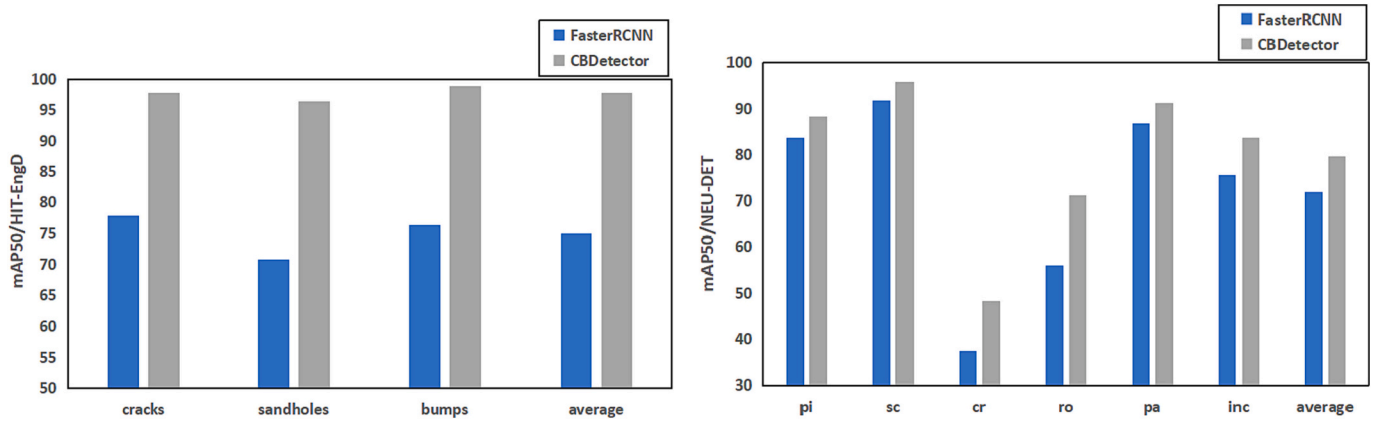

Fig. 9. Bar charts of the detection results of each defect class. Best viewed in color.

datasets (Table 5) demonstrate that the random upper bound (RUB) scaling strategy has a greater advantage than the fixed upper bound (FUB) scaling strategy, especially on HIT-EngD. We argue that this is because the image size varies in a small range after resizing according to Algorithm 1 when FUB is used, which benefits defect detection when the size variation is smooth. However, when the defect sizes change drastically or the defects tend to be tiny, such as the defects of HIT-EngD, scaling adjustment within a small range will not improve performance. At this point, scaling adjustment of a large range should be accounted for, such as through the RUB strategy. Moreover, Fig. 8(d), (e) demonstrate that the convergence speed of the model using the RUB becomes slower than that using the FUB. This is because the input of the model has fluctuated widely, and the model is gradually learning more comprehensive and accurate features under a wide range of size fluctuations, so the final performance is better than that of the FUB strategy, as depicted in Fig. 8(f) and (g).

The ablation experimental results of the CBDetector structure (Table 6) demonstrate that when the LG-IE module is integrated into the FasterRCNN (Model_A), the mAP^{50} value is significantly improved on both HIT-EngD and NEU-DET datasets, with increases of 21.4 on HIT-EngD and 7.1 on NEU-DET. When the PE-FPA module is further integrated (Model_B), the defect detection precision is further improved by 1.3 on HIT-EngD and 0.7 on NEU-DET. Experimental results validate the effectiveness of the proposed modules for the following reasons: 1) the integrated LG-IE module first possesses self-adaptation ability to the input images and enables the model to consider the long-distance dependencies between features within the high dimensional space, allowing the model to better extract the global perception of current scenes and achieve higher-order spatial interactions; and 2) the integrated LG-IE has the excellent trait of feature translation invariance and the ability to extract local key features, so this model performs better than the structure that uses only CNNs. Further introducing the PE-FPA module overcomes the accuracy degradation due to defect scale variation by fusing different size feature maps and applying defect localization on them. This further improves the accuracy of the model, especially on HIT-EngD, because the scale variation of defects in HIT-EngD is more drastic than in NEU-DET.

The comparison results with SOTA methods are shown in Table 7. 1) The performance of the proposed CBDetector is substantially improved on both datasets, with the best performance being on HIT-EngD, with an mAP^{50} increase of 22.7 compared to FasterRCNN+FPN, and competitive performance on NEU-DET, with an mAP^{50} increase of 7.8 compared to FasterRCNN+FPN. 2) Both deformable DETR and DINO performed poorly on both datasets, especially the deformable DETR, with mAP^{50} values of only 16.5 (HIT-EngD) and 14.5 (NEU-DET). We argue that two reasons exist for this performance. The first is the reason for commonality: the pure Transformer mechanism is data hungry; therefore, to achieve better performance, a large-scale dataset is required (e.g., ImageNet, ~100 G) as a prerequisite for training. However, in real industrial scenarios, no such large-scale dataset is available for training; moreover, conducting training on such a large-scale dataset is time consuming and hardware demanding, usually requires several days to accomplish training, and the dataset has poor maintainability and reusability. Therefore, we propose using the integrated encoder structure described in this paper. This structure has a powerful long-distance modeling capability while also overcoming the drawbacks of massive data training. The second reason is the reason for specialty: the deformable DETR [51] improves the DETR by introducing a deformable attention module to overcome the tedious training epochs. Nevertheless, the convergence ability of the deformable DETR still needs many improvements for use in industrial applications.

The qualitative experimental results shown in Fig. 10 and Fig. 11 demonstrate that the proposed CBDetector has significantly great advantages over FasterRCNN+FPN on both datasets. As mentioned previously, FasterRCNN, a two-stage detection method, is still not completely competent for detecting the tiny defects of cylinder bores. This incompetence originates in two aspects of the task: one is the ultrahigh resolution images, and the other is the size of the defects themselves, such as sandholes. Therefore, we try to integrate FPN into FasterRCNN, but the results (Fig. 10) show that such integration is still not suitable enough for defect detection of cylinder bores. Considering that illumination variation is also a key factor affecting the defect detection of cylinder bores, we use the multiscale CNN and Transformer integrated structure and the corresponding preprocessing module to

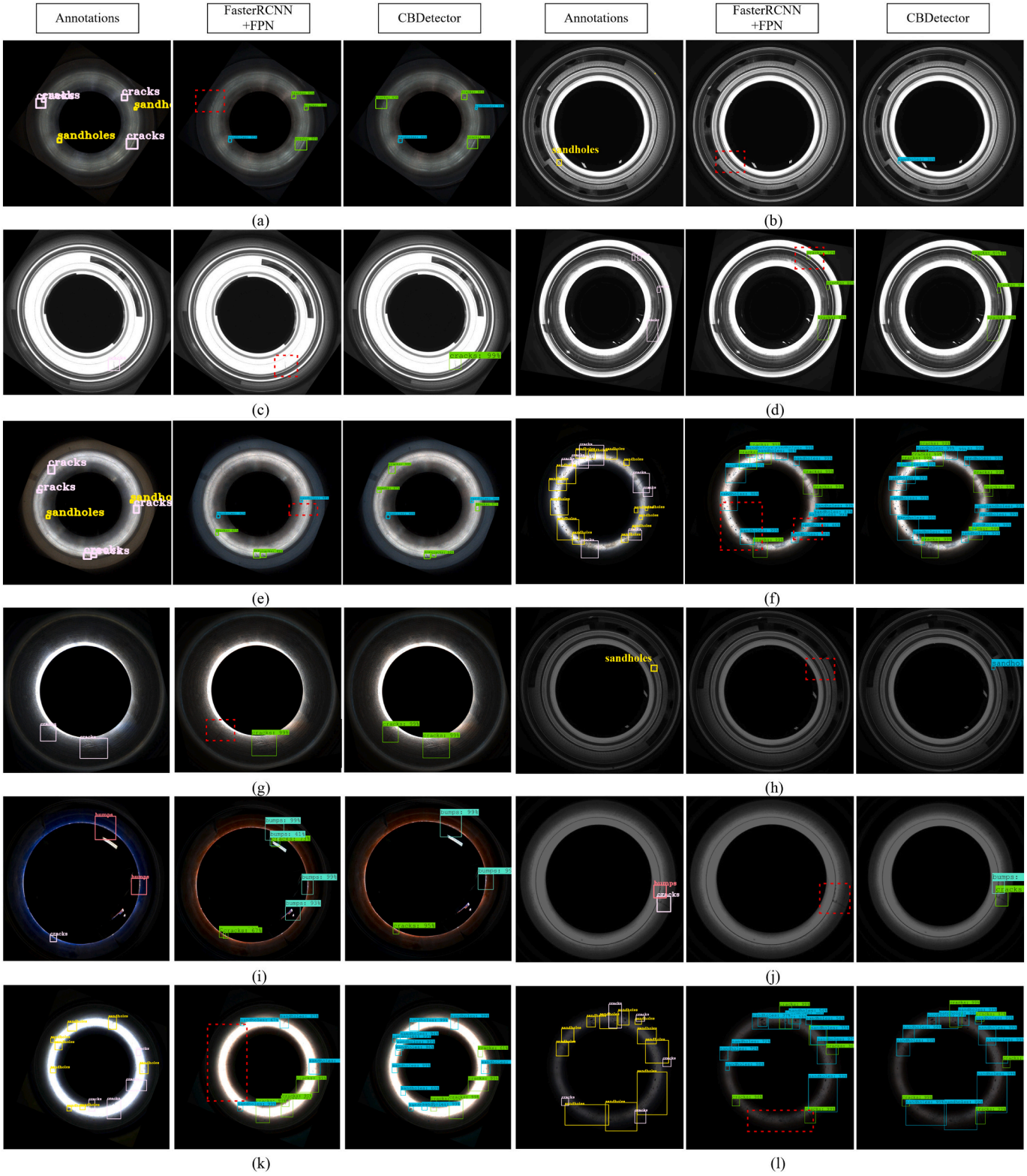

Fig. 10. Visualization of defect detection results on HIT-EngD dataset. For each group, the first row presents the corresponding annotated defect locations on the original image, the second row presents the detection results of FasterRCNN+FPN, and the third row presents the detection results of CBDetector. The red dashed boxes in (a) to (l) indicate the defects that are missed by FasterRCNN+FPN. Best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

address these issues. By modeling the interrelationship between global features of the images while retaining the ability to capture local key features and adapting the illumination of input images, the final defect detection performance is substantially improved, as shown in Fig. 10. The proposed CBDetector is more robust to tiny defects (Fig. 10(a)~(d)), cluttered defects (Fig. 10(f), (l)), and the degradation of defect detection due to illumination variation (Fig. 10(g)~(k)).

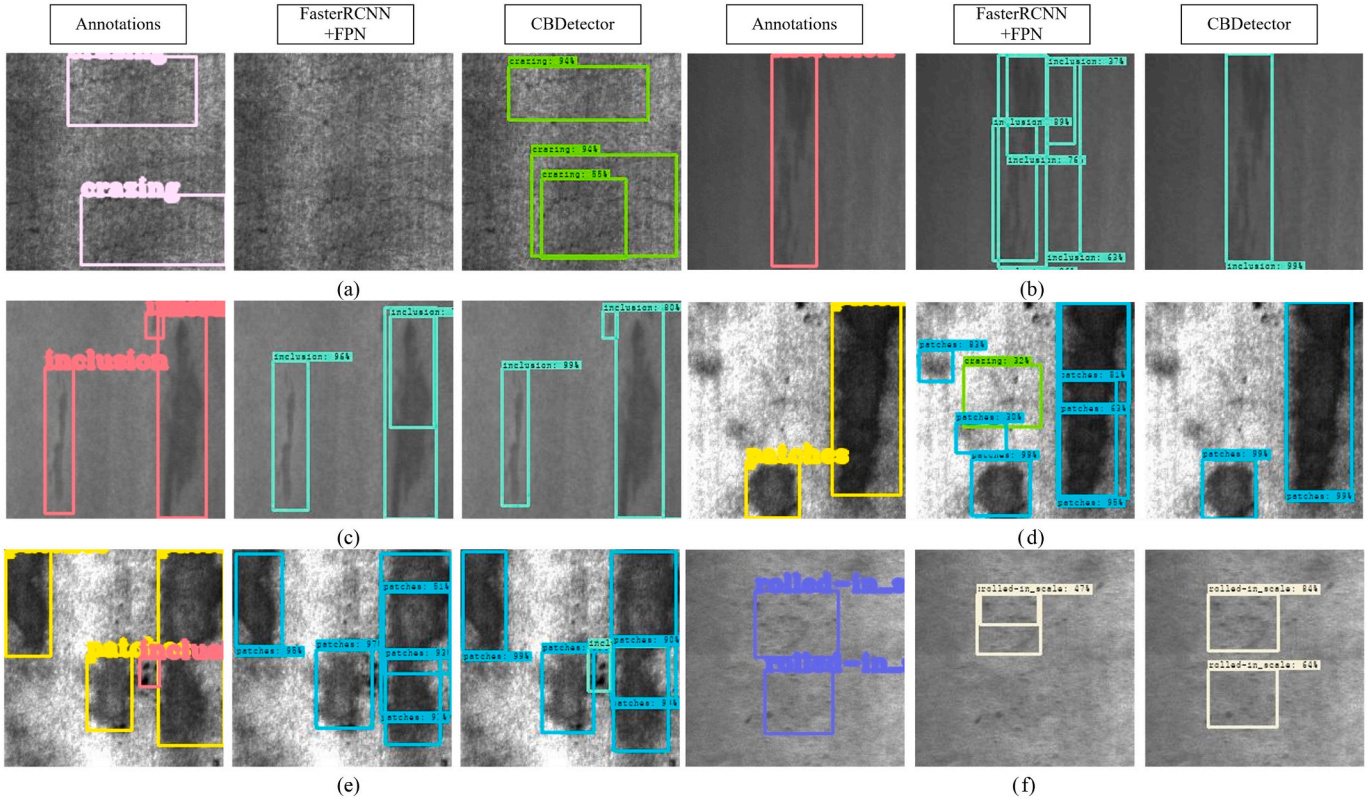

Fig. 11. Visualization of defect detection results on NEU-DET dataset. For each group, the first row presents the corresponding annotated defect locations on the original image, the second row presents the detection results of FasterRCNN+FPN, and the third row presents the detection results of CBDetector. Best viewed in color.

6. Conclusion

In this paper, a first defect detection method for cylinder bores of automobile engines, referred to as CBDetector, which integrates the locally sensitive and globally covered integrated encoder (LG-IE) and the perception-enhanced feature path aggregation (PE-FPA) unit into a single pipeline, is presented. Moreover, a novel visual image-based defect detection dataset of cylinder bores of automobile engines referred to as HIT-EngD is collected and carefully annotated. This dataset contains common yet challenging scenarios for defect detection of cylinder bores and will be made publicly available to encourage future research. Extensive experiments demonstrate the SOTA performance of CBDetector for defect detection not only on cylinder bores of automobile engines (HIT-EngD) but also on steel surfaces (NEU-DET), with mAP^{50} increases of 22.7 and 7.8 compared to the FasterRCNN integrated with FPN. CBDetector can also run at a high frame rate (Nvidia A100, ~10 FPS). We hope to provide a new structure layout for designing upcoming defect detection models from our work. Future work will focus on further optimizing the running efficiency.

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.

Acknowledgments

This work was supported in part by a project of the National Natural Science Foundation of China on Intelligent Recognition and Phase Estimation of X-ray Pulsar Observation Signals on Orbit [Grant No. 11973021], in part by the Research and Development Project of Visual Defect Detection Imaging Systems and Software Tools for Complex Workpieces of Heilongjiang Province, China [Grant No. 2023ZXJ01A01] and in part by Heilongjiang Province's "Million and Ten Million" Major Project in Science and Technology, China [Grant No. 2021ZX10A01].

References

[1] Zhang NX, Zhong Y, Dian S. Rethinking unsupervised texture defect detection using PCA. Opt Lasers Eng 2023;163:107470. https://doi.org/10.1016/j.optlaseng.2022.107470.
[2] Nakajima R, Yamamoto R, Hida T, Matsumoto T. A study on the effect of defect shape on defect detection in visual inspection. Procedia Manuf 2019;39:1641–8. https://doi.org/10.1016/j.promfg.2020.01.277.
[3] Zhang K, Yan Y, Li P, Jing J, Liu X, Wang Z. Fabric defect detection using salience metric for color dissimilarity and positional aggregation. IEEE Access 2018;6:49170–81. https://doi.org/10.1109/ACCESS.2018.2868059.
[4] Ericsson L, Gouk H, Loy CC, Hospedales TM. Self-Supervised Representation Learning: Introduction, advances, and challenges. 2022. p. 42–62.
[5] Liu J, Guo F, Gao H, Li M, Zhang Y, Zhou H. Defect detection of injection molding products on small datasets using transfer learning. J Manuf Process 2021;70:400–13. https://doi.org/10.1016/j.jmapro.2021.08.034.
[6] Zhang S, He M, Zhong Z, Zhu D. An industrial interference-resistant gear defect detection method through improved YOLOv5 network using attention mechanism and feature fusion. Meas J Int Meas Confed 2023;221:113433. https://doi.org/10.1016/j.measurement.2023.113433.
[7] Zhou F, Chao Y, Wang C, Zhang X, Li H, Song X. A small sample nonstandard gear surface defect detection method. Meas J Int Meas Confed 2023;221:113472. https://doi.org/10.1016/j.measurement.2023.113472.
[8] Li W, Li B, Niu S, Wang Z, Wang M, Niu T. LSA-net: location and shape attention network for automatic surface defect segmentation. J Manuf Process 2023;99:65–77. https://doi.org/10.1016/j.jmapro.2023.05.001.
[9] Wang J, Xu G, Yan F, Wang J, Wang Z. Defect transformer: An efficient hybrid transformer architecture for surface defect detection. Meas J Int Meas Confed 2023;211:112614. https://doi.org/10.1016/j.measurement.2023.112614.
[10] Kaya Mahmut, Bilge Hasan Sakir. Deep metric learning : a survey. Symmetry (Basel) 2019;11(9):1066.
[11] Taherkhani K, Eischer C, Toyserkani E. An unsupervised machine learning algorithm for in-situ defect-detection in laser powder-bed fusion. J Manuf Process 2022;81:476–89. https://doi.org/10.1016/j.jmapro.2022.06.074.

[12] Béthune L, Paul-sabatier U. Certifiable metric one class learning with adversarially trained Lipschitz Classifier. 2022. p. 1–13.
[13] Di H, Ke X, Peng Z, Dongdong Z. Surface defect classification of steels with a new semi-supervised learning method. Opt Lasers Eng 2019;117:40–8. https://doi.org/10.1016/j.optlaseng.2019.01.011.
[14] Napoletano P, Piccoli F, Schettini R. Semi-supervised anomaly detection for visual quality inspection. Expert Syst Appl 2021;183:115275. https://doi.org/10.1016/j.eswa.2021.115275.
[15] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2017;39:1137–49. https://doi.org/10.1109/TPAMI.2016.2577031.
[16] Kirillov A, Wu Y, He K, Girshick R. Pointrend: image segmentation as rendering. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2020:9796–805. https://doi.org/10.1109/CVPR42600.2020.00982.
[17] B MK, Danelljan M, Pflugfelder R, Drbohlav O, He L. VOT2020 Challenge Results vol. 1. 2020. doi:https://doi.org/10.1007/978-3-030-68238-5.
[18] Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, et al. Deep learning for generic object detection: a survey. Int J Comput Vis 2020;128:261–318. https://doi.org/10.1007/s11263-019-01247-4.
[19] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. SSD: Single shot MULTIBOX detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Comput. Vis. – ECCV. 2016. Cham: Springer International Publishing; 2016. p. 21–37.
[20] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. 2018.
[21] Bochkovskiy A, Wang C-Y, Liao H-YM. YOLOv4: Optimal speed and accuracy of object detection. 2020.
[22] Li Z, Tian X, Liu X, Liu Y, Shi X. A two-stage industrial defect detection framework based on improved-YOLOv5 and optimized-inception-ResnetV2 models. Appl Sci 2022:12. https://doi.org/10.3390/app12020834.
[23] Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: A single-stage object detection framework for industrial applications. http://arxiv.org/abs/2209.02976; 2022.
[24] Wang C-Y, Bochkovskiy A, Liao H-YM. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors 2022:1–15. http://arxiv.org/abs/2207.02696.
[25] Kwon JE, Park JH, Kim JH, Lee YH, Cho SI. Context and scale-aware YOLO for welding defect detection. NDT E Int 2023:139. https://doi.org/10.1016/j.ndteint.2023.102919.
[26] Souza BJ, Stefenon SF, Singh G, Freire RZ. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. Int J Electr Power Energy Syst 2023;148:108982. https://doi.org/10.1016/j.ijepes.2023.108982.
[27] Li J, Su Z, Geng J, Yin Y. Real-time detection of steel strip surface defects based on improved YOLO detection network. IFAC-PapersOnLine 2018;51:76–81. https://doi.org/10.1016/j.ifacol.2018.09.412.
[28] Liu Z, Liu K, Zhong J, Han Z, Zhang W. A high-precision positioning approach for catenary support components with multiscale difference. IEEE Trans Instrum Meas 2020;69:700–11. https://doi.org/10.1109/TIM.2019.2905905.
[29] Jiang W, Li T, Zhang S, Chen W, Yang J. PCB defects target detection combining multi-scale and attention mechanism. Eng Appl Artif Intell 2023;123:106359. https://doi.org/10.1016/j.engappai.2023.106359.
[30] Cha YJ, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput Civ Infrastruct Eng 2018;33:731–47. https://doi.org/10.1111/mice.12334.
[31] Xue Y, Li Y. A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. Comput Civ Infrastruct Eng 2018;33:638–54. https://doi.org/10.1111/mice.12367.
[32] Gong Y, Luo J, Shao H, Li Z. A transfer learning object detection model for defects detection in X-ray images of spacecraft composite structures. Compos Struct 2022;284:115136. https://doi.org/10.1016/j.compstruct.2021.115136.
[33] Zhang Y, Zhang Z, Fu K, Luo X. Adaptive defect detection for 3-D printed lattice structures based on improved faster R-CNN. IEEE Trans Instrum Meas 2022:71. https://doi.org/10.1109/TIM.2022.3200362.
[34] Chen M, Yu L, Zhi C, Sun R, Zhu S, Gao Z, et al. Improved faster R-CNN for fabric defect detection based on Gabor filter with genetic algorithm optimization. Comput Ind 2022;134:103551. https://doi.org/10.1016/j.compind.2021.103551.
[35] Lei HW, Wang B, Wu HH, Wang AH. Defect detection for polymeric polarizer based on faster R-CNN. J Inf Hiding Multimed Signal Process 2018;9:1414–20.
[36] Zhang T, Wang Z, Li F, Zhong H, Hu X, Zhang W, et al. Automatic detection of surface defects based on deep random chains. Expert Syst Appl 2023;229:120472. https://doi.org/10.1016/j.eswa.2023.120472.
[37] Zhao Z, Zhen Z, Zhang L, Qi Y, Kong Y, Zhang K. Insulator detection method in inspection image based on improved faster R-CNn. Energies 2019:12. https://doi.org/10.3390/en12071204.
[38] Deitsch S, Christlein V, Berger S, Buerhop-Lutz C, Maier A, Gallwitz F, et al. Automatic classification of defective photovoltaic module cells in electroluminescence images. Sol Energy 2019;185:455–68. https://doi.org/10.1016/j.solener.2019.02.067.
[39] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc 2015:1–14.
[40] Shang L, Yang Q, Wang J, Li S, Lei W. Detection of rail surface defects based on CNN image recognition and classification. Int Conf Adv Commun Technol ICACT2018-Febru; 2018. p. 45–51. https://doi.org/10.23919/ICACT.2018.8323642.
[41] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. In: Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2016-Decem; 2016. p. 2818–26. https://doi.org/10.1109/CVPR.2016.308.
[42] Zheng J, Sun X, Zhou H, Tian C, Qiang H. Printed circuit boards defect detection method based on improved fully convolutional networks. IEEE Access 2022;10:109908–18. https://doi.org/10.1109/ACCESS.2022.3214306.
[43] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. http://arxiv.org/abs/1704.04861; 2017.
[44] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2018:4510–20. https://doi.org/10.1109/CVPR.2018.00474.
[45] Howard A, Sandler M, Chen B, Wang W, Chen LC, Tan M, et al. Searching for mobileNetV3. In: Proc IEEE Int Conf Comput Vis. 2019-Octob; 2019. p. 1314–24. https://doi.org/10.1109/ICCV.2019.00140.
[46] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2016-Decem; 2016. p. 770–8. https://doi.org/10.1109/CVPR.2016.90.
[47] Liang Q, Zhu W, Sun W, Yu Z, Wang Y, Zhang D. In-line inspection solution for codes on complex backgrounds for the plastic container industry. Meas J Int Meas Confed 2019;148:106965. https://doi.org/10.1016/j.measurement.2019.106965.
[48] Ma N, Zhang X, Zheng HT, Sun J. Shufflenet V2: Practical guidelines for efficient cnn architecture design. In: Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 11218 LNCS; 2018. p. 122–38. https://doi.org/10.1007/978-3-030-01264-9_8.
[49] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. 2016. p. 1–13. http://arxiv.org/abs/1602.07360.
[50] Dang LM, Wang H, Li Y, Nguyen TN, Moon H. DefectTR: end-to-end defect detection for sewage networks using a transformer. Constr Build Mater 2022;325:126584. https://doi.org/10.1016/j.conbuildmat.2022.126584.
[51] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-End Object Detection with Transformers. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)12346 LNCS; 2020. p. 213–29. https://doi.org/10.1007/978-3-030-58452-8_13.
[52] Liu Q, Huang X, Shao X, Hao F. Industrial cylinder liner defect detection using a transformer with a block division and mask mechanism. Sci Rep 2022;12:1–14. https://doi.org/10.1038/s41598-022-14971-8.
[53] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin transformer: hierarchical vision transformer using shifted windows. Proc IEEE Int Conf Comput Vis 2021:9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986.
[54] Li Y, Xiang Y, Guo H, Liu P, Liu C. Swin transformer combined with convolution neural network for surface defect detection. Machines 2022;10:1083. https://doi.org/10.3390/machines10111083.
[55] Xu Z, Guan H, Kang J, Lei X, Ma L, Yu Y, et al. Pavement crack detection from CCD images with a locally enhanced transformer network. Int J Appl Earth Obs Geoinf 2022;110:102825. https://doi.org/10.1016/j.jag.2022.102825.
[56] He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 8691 LNCS; 2014. p. 346–61. https://doi.org/10.1007/978-3-319-10578-9_23.
[57] Lin T, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition 2017:2117–25.
[58] Liu S, Qi L, Qin H, Shi J, Jia J. PANet: Path Aggregation Network for Instance Segmentation. (arXiv:1803.01534v3 [cs.CV] UPDATED). Cvpr 2019:8759–68. http://arxiv.org/abs/1803.01534.
[59] Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2020:10778–87. https://doi.org/10.1109/CVPR42600.2020.01079.
[60] Gao L, Zhang J, Yang C, Zhou Y. Cas-VSwin transformer: a variant swin transformer for surface-defect detection. Comput Ind 2022;140:103689. https://doi.org/10.1016/j.compind.2022.103689.
[61] An K, Zhang Y. LPViT: a transformer based model for PCB image classification and defect detection. IEEE Access 2022;10:42542–53. https://doi.org/10.1109/ACCESS.2022.3168861.
[62] Wang W, Xie E, Li X, Fan DP, Song K, Liang D, et al. PVT v2: improved baselines with pyramid vision transformer. Comput Vis Media 2022;8:415–24. https://doi.org/10.1007/s41095-022-0274-8.
[63] Woo S, Park J, Lee JY, Kweon IS. CBAM: Convolutional block attention module. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics)11211 LNCS; 2018. p. 3–19. https://doi.org/10.1007/978-3-030-01234-2_1.
[64] Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J. You only look one-level feature n.d.
[65] Song K, Yan Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 2013;285:858–64. https://doi.org/10.1016/j.apsusc.2013.09.002.
[66] Lv X, Duan F, Jiang JJ, Fu X, Gan L. Deep metallic surface defect detection: the new benchmark and detection network. Sensors (Switzerland) 2020:20. https://doi.org/10.3390/s20061562.
[67] Cheng X, Yu J. RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. IEEE Trans Instrum Meas 2020;70:1–11.
[68] Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. http://arxiv.org/abs/2203.03605; 2022.

[69] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. http://arxiv.org/abs/2010.11929; 2020.
[70] Wu H, Xiao B, Codella N, Liu M, Dai X, Yuan L, et al. CvT: introducing convolutions to vision transformers. Proc IEEE Int Conf Comput Vis 2021:22–31. https://doi.org/10.1109/ICCV48922.2021.00009.
[71] Maaz M, Shaker A, Cholakkal H, Khan S, Zamir SW, Anwer RM, et al. EdgeNeXt: Efficiently amalgamated CNN-transformer architecture for Mobile vision applications13807. LNCS. Springer Nature Switzerland; 2023. https://doi.org/10.1007/978-3-031-25082-8_1.
[72] Rane N. YOLO and faster R-CNN object detection for smart industry 4.0 and industry 5.0: applications, challenges, and opportunities. SSRN Electron J 2023. https://doi.org/10.2139/ssrn.4624206.