Measurement 237 (2024) 115189

Contents lists available at ScienceDirect

Measurement

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

Steel surface defect detection algorithm in complex background scenarios

BaiTing Zhao [a], YuRan Chen [a,*], XiaoFen Jia [b,c,**], TianBing Ma [c]

[a] School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China
[b] School of Artificial Intelligence, Anhui University of Science and Technology, Huainan 232001, China
[c] State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines, Anhui University of Science and Technology, Huainan 232001, China

ARTICLE INFO

Keywords:
Defect detection
YOLOv8
Deep learning
Multi-scale feature extraction

ABSTRACT

Detecting surface defects on steel poses a significant challenge attributed to factors such as poor contrast, diverse defect types, complex background clutter, and noise interference present in images of steel surface defects. Current detection techniques face challenges in quickly and accurately identifying defects within complex backgrounds. To address the deployment of high-precision detection models on edge devices with limited resources, particularly for identifying steel surface defects, this study introduces a Multi-Scale Adaptive Fusion (MSAF) YOLOv8n defect detection algorithm designed for complex backgrounds. This algorithm effectively balances detection speed and accuracy. Firstly, a Multi-Scale Adaptive Fusion Block (MS-AFB) is proposed for the extraction of multi-scale features. Secondly, a Dynamic Coordinate Attention Ghostconv Space Pooling Pyramid-fast Cross-stage Partial Convolutional (DCA-GSPPFCSPC) is devised to significantly improve detection accuracy. Furthermore, the detection head has been redesigned utilizing Lightweight Multi-scale Convolutional (LMSC) approach, and an Adaptive Pyramid Receptive Field Block (AP-RFB) has been introduced to improve the receptive field efficiently. Meanwhile, Normalized Weighted Distance (NWD) and Weighted Intersection over Union (WIoU) are employed as the boundary box loss functions, serving as substitutes for Complete Intersection over Union (CIoU) loss function with a ratio of 2:8. The experimental results obtained from the improved Northeastern University Defect Dataset (NEU-DET) dataset demonstrate that MSAF-YOLOv8n model, despite having 40.4 % of the parameters and 28.8 % of Floating Point Operations (FLOPs) of YOLOv8s, achieves a mAP@.5 that is 0.9 % higher than that of YOLOv8s. Additionally, MSAF-YOLOv8n demonstrates robust generalization capabilities in Pascal VOC2007, self-constructed datasets, and various other datasets. Subsequently, the model is implemented on embedded systems, namely Jeston TX2 NX and Orange Pi 5+, both of which demonstrate real-time detection capabilities.

1. Introduction

Steel is the favored material for industrial applications [1] and is commonly referred to as the “backbone” of the industry. It finds extensive application in various sectors including construction, aerospace, national defense, mechanical manufacturing, and mine cage guidance. Various defects commonly manifest on steel surfaces as a result of complex background factors, including manufacturing processes and production environments. These defects not only impact the product’s aesthetics but also have enduring negative implications on its functionality and safety. An illustration of this is the steel cage guide, a crucial element of the vertical shaft lifting system, which relies on constant friction contact with the cage ears. If the steel quality is substandard or the surface is significantly worn, it will impact the safe functioning of the entire lifting system. Therefore, to guarantee the secure utilization of steel products, it is imperative to perform defect detection on the steel [2].

The traditional method for detecting surface defects in steel involves manual visual inspection, which is labor-intensive and time-consuming. This approach suffers from low detection efficiency, lacks real-time capabilities, and is susceptible to errors such as missed detections and false alarms, rendering it unreliable [3]. Machine vision techniques have been extensively employed for defect detection, supplanting conventional manual visual inspection due to their rapid detection speed, high

* Corresponding author at: School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China.
** Corresponding author at: School of Artificial Intelligence, Anhui University of Science and Technology, Huainan 232001, China.
E-mail addresses: 1014913959@qq.com (Y. Chen), jxfzbt2008@163.com (X. Jia).

https://doi.org/10.1016/j.measurement.2024.115189
Received 25 December 2023; Received in revised form 6 June 2024; Accepted 24 June 2024
Available online 25 June 2024

precision, and robust stability. Machine vision inspection techniques rely on defect detection through manually crafted feature extraction algorithms, with the accuracy of the detection outcomes impacted by the quality of the algorithms. However, algorithms that are manually designed typically exhibit lower robustness. This method may yield favorable detection outcomes in typical backgrounds, as identifying general background anomalies is more straightforward in simple backgrounds devoid of substantial noise, texture, and lighting fluctuations. The utilization of a lightweight method can facilitate detection, leading to improved real-time performance. In the identification of defects within complex settings such as steel surfaces and mine cage guides, the backdrop often comprises diverse elements such as noise, texture, uneven lighting, color variations, and other complex factors. These complexities can result in defects being obscured within the background or erroneously identified as noise. At present, the machine vision inspection method faces challenges in meeting the detection requirements. In recent years, the successful implementation of deep learning models, such as Convolutional Neural Networks (CNNs), in diverse areas of computer vision has led to the emergence of numerous defect detection algorithms based on deep learning. Deep learning techniques have the capability to utilize images as direct inputs to the neural network. This enables the automatic extraction of features, facilitating the classification and localization of defects. Furthermore, these methods offer a more streamlined approach to achieving end-to-end detection. In contrast to machine vision techniques, deep learning approaches offer superior detection accuracy and improved generalization capabilities. Therefore, deep learning algorithms have found extensive application in detecting defects within complex backgrounds. However, lightweight deep-learning networks encounter challenges in extracting effective features within complex backgrounds, hindering the achievement of optimal detection outcomes. As a consequence, deep learning networks are becoming increasingly deeper and wider to extract features and mitigate background interference. This expansion leads to larger and more complex models, significantly impacting the real-time performance of detection. Consequently, achieving a balance between detection accuracy and efficiency in defect detection under complex backgrounds becomes challenging.

Recently, developments in computer processing capabilities have facilitated extensive research on the application of deep learning algorithms for the detection of surface defects in challenging environments characterized by complex backgrounds. These environments include but are not limited to steel surfaces, textiles, road cracks, and mine cage guides. Li et al. [16] introduced the incorporation of a residual block containing both an encoder and decoder, along with a feature alignment module inspired by YOLOv4. This integration significantly improved the model's capacity for feature learning. Wei et al. [46] integrated the strengths of CNNs and Vision Transformers (ViTs) by introducing a novel local–global lightweight ViT model derived from Convmixer. This model was designed for defect recognition in micro/micro Light Emitting Diode (LED) chips. Chen et al. [12] implemented an effective defect detection method by incorporating histogram equalization and deformable convolution techniques to dynamically capture features. Yeung et al. [13] examined the difficulties associated with alterations in steel surface dimensions, changes in shape, and the effectiveness of defect detection. They also suggested improved methods for locating and categorizing defects. Yu et al. [14] introduced a progressively improved redistribution pyramid with supervisory considerations, built upon YOLOv5s, for complex defect detection. Their approach attained State-Of-The-Art (SOTA) performance across various datasets. Xing et al. [15] proposed a loss function, XIOU, tailored for steel surface defect detection with the aim of enhancing the accuracy of defect identification. Zhao et al. [36] incorporated multi-scale modules into the surface defects of steel and devised a dual-feature pyramid network to improve features, thereby increasing the depth of the entire network and facilitating feature reuse. The aforementioned review highlights the considerable research efforts dedicated to detecting surface defects in complex backgrounds in recent years, resulting in significant advancements. However, the majority of these studies are still confined to the laboratory setting, posing challenges for practical implementation due to two primary reasons. Firstly, the background of product testing is complex and multifaceted. Surface defects of underground cage guides in mines are typically captured under standard lighting conditions during data collection, potentially leading to favorable detection outcomes in the course of network training. However, real-world production settings often encounter significant environmental disruptions, including but not limited to uneven lighting, light reflections, noise, and motion blur. This results in a low contrast of background clutter and numerous false defects, thereby often compromising the recognition effectiveness [4]. Secondly, online inspection necessitates high real-time capabilities. Numerous studies have incorporated extensive parametric and computational requirements to achieve high levels of inspection accuracy. This poses significant challenges for embedding these technologies into devices with constrained resources, making it challenging to align with real-world industrial needs.

Therefore, ongoing research is being carried out on the contemporary challenges associated with detecting defects in complex backgrounds. In this study, the lightweight YOLOv8n [11] model is chosen as the foundational framework. Subsequently, a novel lightweight high-precision steel surface defect detection network named Multi-scale Adaptive Fusion (MSAF)-YOLOv8 is introduced with the objective of enhancing the capability to detect steel surface defects amidst complex backgrounds. The primary contributions of this research can be outlined as follows:

1) A Multi-Scale Adaptive Fusion Block (MS-AFB), has been developed to improve the accurate and efficient detection of defects with varying shapes. By initially coupling and subsequently decoupling the decoupling head in YOLOv8, and incorporating the suggested Lightweight Multi-scale Convolutional (LMSC) method, MS-AFB diminishes the parameters and computational complexity of the detection head by 33 %, while concurrently enhancing mean Average Precision (mAP) by 0.9 %.
2) The study introduces a novel approach named Dynamic Coordinate Attention GhostConv Space Pooling Pyramid-Fast Cross-Stage Partial Convolutional (DCA-GSPPFCSPC) to substitute the conventional SPPF module. This proposed method demonstrates a substantial improvement in detection accuracy. Meanwhile, an Adaptive Pyramid Receptive Field Block (AP-RFB) is developed, incorporating dilated convolutions to significantly improve the receptive field. This improvement aims to boost the model's feature extraction capabilities and generalization performance.
3) An innovative proposal of MSAF-YOLOv8 is introduced, aiming to improve the detection accuracy of defects that closely resemble the background. Experiments were conducted on multiple datasets, and the results indicate that the proposed approach demonstrates robust generalization performance.
4) Subsequently, the model was implemented on the embedded devices Jetson TX2 NX and Orange Pi 5+, resulting in successful real-time detection.

The subsequent sections of the paper are structured as follows. In Section 2, a review of relevant literature is conducted, and the limitations of the existing research are discussed. In Section III, a detailed presentation of the proposed model MSAF-YOLOv8n is provided. The research assess the proposed experimental results and compare them with other cutting-edge models in Section IV. In Section V, the model is implemented to evaluate its detection performance on embedded devices. In Section VI, a comprehensive summary of the entire paper is provided, along with an identification of its limitations and a discussion on potential avenues for future research endeavors.

2. Related work

2.1. Multi-scale block

The multi-scale representation of features plays a crucial role in various downstream tasks, such as computer vision. This is because the ability to perceive information at different scales improves the understanding of the overall target [17]. Neural networks that incorporate multi-scale information typically exhibit improved feature extraction and improved integration of contextual information. GoogLeNet [42], which is built on the Inception module, emerged as the victor in the 2014 ImageNet competition by leveraging convolutional kernels of varying sizes to capture multi-scale features. In Res2Net [17], a hierarchical residual connection is established within a single residual block, which replaces the conventional single 3×3 convolution kernel. It has the capability to depict multi-scale features at a more detailed granularity level, thereby enhancing the receptive field of each network layer. This has led to achieving SOTA performance in ImageNet image classification. Li et al. [44] proposed a multi-scale feature extraction module that leverages the model's feature extraction capacity, significantly enhancing its capability in this regard. Yuan et al. [38] introduced a "plug and play" multi-scale module called Hierarchical Split Block (HSB), demonstrating improvements across various computer vision tasks. Li [47] developed a multi-scale feature extraction module aimed at augmenting the model's feature extraction capacity. This module utilizes three branches with varying receptive fields to extract multi-scale features. Cheng et al. [37] introduced an Adaptive Spatial Feature Fusion module (ASFF) to improve the integration of multi-scale feature information for optimal feature fusion across different levels. Additionally, an Attention Enhanced Feature Fusion (AEFF) module was developed and incorporated into the Neck network to facilitate effective feature fusion [45].

The Inception module is characterized by a substantial number of parameters and significant computational complexity, posing challenges for its practical implementation. While acknowledging the significance of multi-scale information, it is essential to distinguish the importance of information across various scales. While Res2Net incorporates multi-scale information, it fails to differentiate between significant scales. HSB module exhibits high complexity due to a significant number of 'split' and 'concat' operations, which pose challenges for embedded deployment. Utilizing ASFF will lead to an increase in the quantity of network parameters, training duration, and training memory usage.

2.2. Spatial pooling Pyramid (SPP)

SPP [5] was initially developed to address issues related to neuron configuration arising from variations in the dimensions of image inputs between convolutional and fully-connected layers. Subsequently, it was integrated into You Only Look Once (YOLO) algorithm. The utilization of SPP demonstrates significant performance improvements, leading to improved model accuracy and generalization capabilities. In YOLOv5 [7], Spatial Pyramid Pooling − Fast (SPPF) was introduced as an improvement of SPP, aiming to achieve faster execution efficiency compared to SPP. In YOLOv6 [9], Simplified SPPF (SimSPPF) was proposed to improve the speed of inference. In the research conducted by Wang et al. [10], the utilization of SPPCSPC led to improved detection accuracy; however, it also resulted in an increase in both the number of parameters and computational requirements. In YOLOv6 3.0 [25], SPPFCSPC was untilized to improve speed performance while keeping the receptive field constant. While SPP and SPPF are characterized by their lightweight nature, their capacity for feature fusion is insufficient to enable the fusion of local and global features at the feature map level in complex industrial environments. SPPCSPC and SPPFCSPC exhibit robust feature fusion capabilities; however, their parameter and computational complexities are excessive for integration into lightweight networks.

2.3. Dilated convolution

Dilated convolution was initially introduced to address the challenge of image segmentation, aiming to expand the receptive field without altering the feature map size. It also serves as an efficient method for extracting multi-scale features without the need to increase the quantity of convolution kernels. DeepLabv2 [43] introduced ASPP module, which employs several parallel high expansion dilated convolutional layers to capture object information at multiple scales. While significantly enhancing the receptive field, the parameters are excessively large, leading to slow inference times. RFB [18] simulates the receptive field of human vision by leveraging the concept of Inception and integrating dilated convolutions based on the Inception model, thereby enhancing the receptive field effectively. Jiang et al. [39] implemented a cascaded dilated convolutional approach in conjunction with the backbone network. This strategy aims to mitigate information loss during the downsampling process and improve the receptive field. Fang et al. [40] used dilated convolutions with varying dilation rates to extract feature maps with diverse receptive fields. Subsequently, these feature maps were concatenated to improve the detection accuracy of mesoscale defects. However, Gao et al. [41] revealed that dilated convolutional layers may not consistently provide utility and could potentially introduce misleading information. When employing dilated convolutional layers, it is imperative to identify significant features while excluding redundant and ambiguous ones.

2.4. Defect detection on steel surfaces in complex backgrounds

Most of the existing research focused on addressing the detection of steel surface defects in complex backgrounds encounters two primary issues. When encountering defects such as 'crazing' that closely resemble the background, the network's feature extraction capabilities may not be robust enough to yield accurate detection outcomes. Zhang et al. [48] proposed an improvement of YOLOv5s, leading to an overall improvement in detection performance. However, it is important to note that the detection accuracy for 'crazing' deteriorates, achieving only 40.6 % accuracy. Secondly, numerous networks are intricately designed to achieve high accuracy, leading to challenges in network deployment and potentially compromising real-time performance. Zhao et al. [49] and Li et al. [50] demonstrated promising detection outcomes. However, the network's complexity poses challenges for deployment. The existing challenge encountered in defect detection within complex backgrounds pertains to the occurrence of issues such as large models, inefficient detection, and challenging deployment when the detection accuracy reaches a satisfactory level.

The defect 'crazing' bears a striking resemblance to the background, emphasizing the critical need for a breakthrough in its detection to improve the overall accuracy of detection. The practical implementation necessitates the hardware deployment of the detection model. To achieve this objective, the study began by leveraging a robust feature extraction capability, a sufficiently expansive sensing field, and a streamlined network structure. The research scope is to address the two primary challenges currently encountered in the detection of steel surface defects within complex backgrounds.

3. MSAF-YOLOv8n

3.1. Baseline

The defects on the surface of steel exhibit significant variations in size, aspect ratio, and other attributes. To attain optimal detection outcomes with conventional YOLO networks, it is necessary to perform manual clustering and adjustment of anchor values. In YOLO series of detection models, the detection network typically utilizes anchor-based technology. The concept involves utilizing a collection of anchor boxes that are clustered based on K-means on the training set prior to training.

Subsequently, during the inference process, the feature maps are slid to extract multiple anchor boxes for subsequent classification and regression [6]. The default clustering anchor box is derived from Common Objects in Context (COCO) dataset. This clustering approach may result in limited generalization capabilities of the model when applied to diverse datasets. Due to the fact that a significant portion of the anchor boxes generated post-training are not utilized, there exists a considerable redundancy in the computation process. This redundancy leads to heightened computational expenses and diminished detection speed [8]. In an effort to improve the model's generalization performance, YOLOX [8] incorporates anchor-free technology, which demonstrates superior generalization capabilities when compared to conventional YOLO algorithms, leading to an improvement in detection performance. YOLOv8 [11] implements an anchor-free architecture, demonstrating robust generalization capabilities across various datasets. It incorporates various SOTA technologies, thereby improving performance and flexibility. YOLOv8 was selected as the baseline benchmark network.

3.2. Design ideas and architecture

In defect detection, particularly in complex scenarios such as identifying defects in the underground cage guide within coal mines, delineating the boundaries of defects poses a significant challenge. The distinction between positive and negative samples is minimal, whereas the variation within the class is substantial. For instance, a single scratch can manifest differently under varying lighting conditions: it may appear white in certain lighting, black in others, and in certain cases, it may be visible only at the edges. The presence of ambiguous shapes, textures, colors, and positions can result in various manifestations of defects, thereby complicating the detection process. In this scenario, the extraction of features at multiple scales and the perception of contextual information with a large receptive field are of paramount importance. Therefore, the module was developed in accordance with the aforementioned key scientific issues.

The network architecture of MSAF-YOLOv8n is illustrated in Fig. 1. Similar to YOLOv8, the network is segmented into three components: Backbone, Neck, and Head. The network's input is defined as $H \times W \times 3$, where H and W represent the height and width of the input feature map, and 3 represents the three RGB channels. Specifically, the Backbone architecture is segmented into five convolutional blocks, each commencing with a 3×3 convolutional layer with a stride of 2. This design facilitates downsampling and results in a doubling of the channel number. After connecting MS-AFB following CBS to extract multi-scale features and broaden the receptive field, DCA-GSPPFCSPC method is employed to additionally extract and integrate the multi-scale features. Finally, an AP-RFB is incorporated into the link connecting the Backbone and Neck to improve the receptive field. In the Neck section, the architecture closely resembles that of the Neck in YOLOv8. The nearest neighbor interpolation in the upsampling process is replaced with bilinear interpolation to minimize information loss in the upsampling procedure. Furthermore, C2f module in the Neck is replaced with MS-AFB** to extract multi-scale information. In the Head section, a lightweight module LMSC is introduced with multi-scale extraction capability, which couples the detection head before decoupling the output.

3.3. MS-AFB

The identification of surface defects on steel poses a significant challenge. The steel surface exhibits a variation in defect scale, with certain defects being large and others small. Secondly, the diverse shapes of defects present on the surface of steel result in significant difference between intra-class and inter-class defects, thereby presenting considerable challenges for detection. Multi-scale modules have the capability to extract feature maps from various perspectives and levels

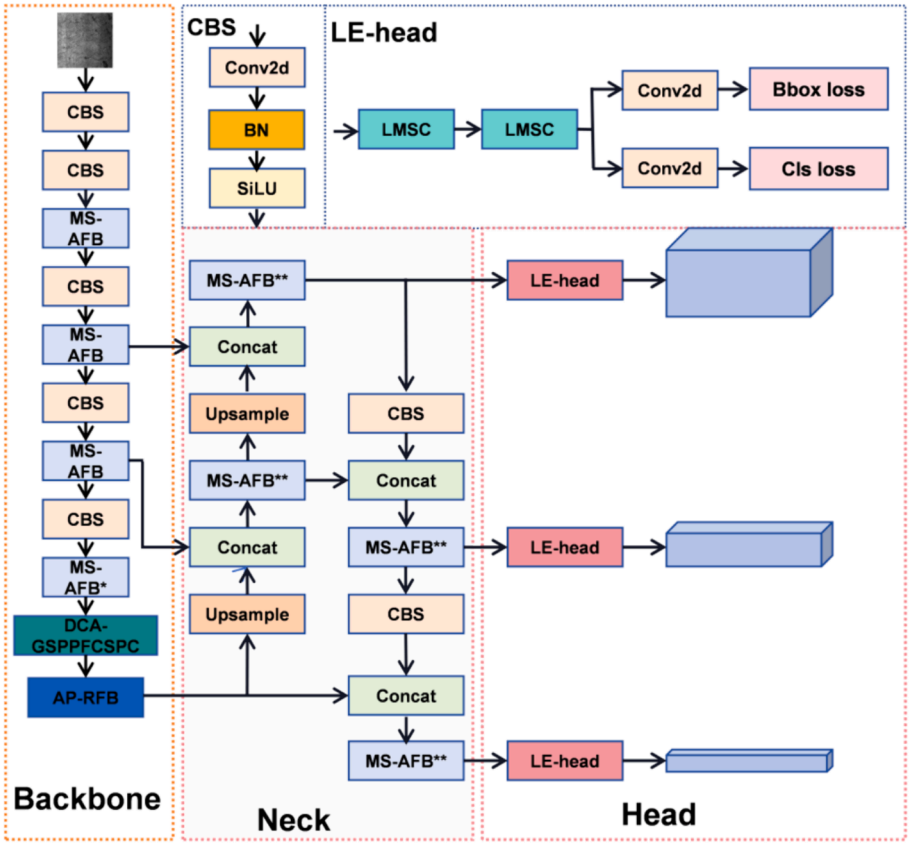

Fig. 1. MSAF-YOLOv8n overall network architecture.

of detail, thereby enhancing the detection of diverse defects. Therefore, the multi-scale module MS-AFB is depicted in Fig. 2(b) and incorporated into the respective locations of the Backbone and Neck (Fig. 1).

In Fig. 2(b), considering the *input* of MS-AFB as (bs, C, H, W), where bs denotes the batch size, C denotes the number of channels, and H and W represent the height and width of the feature map, respectively. MS-AFB utilizes a Res2Block [17] for the extraction of multi-scale feature information. The Res2block conducts a 1 × 1 convolution on the *input* feature map, subsequently dividing it into four parts x_i, $i = 1, 2, \ldots, 4$. The 3 × 3 convolution kernel associated with the *input* x_i ($3 \le i \le 4$, $i \in N^+$) can extract features not only from x_i ($3 \le i \le 4$, $i \in N^+$) but also from $y_{i\text{-}1}$ ($3 \le i \le 4$, $i \in N^+$), thereby expanding the receptive field. The combinatorial explosion effect enables y_i ($3 \le i \le 4$, $i \in N^+$) to encompass various combinations of receptive field sizes. Subsequently, y is derived through the concatenation operation for y_i, $i = 1, 2, \ldots, 4$. By applying a 1 × 1 convolution to adjust the channel numbers, a spatial adaptive weight of (bs, 4, H, W) can be obtained. After applying the softmax activation function, the data is divided into four segments along the channel axis. Each segment undergoes element-wise operations with the corresponding y_i ($1 \le i \le 4$, $i \in N^+$) to derive the adaptive output value out_i ($1 \le i \le 4$, $i \in N^+$), thereby achieving adaptive feature selection. Subsequently, a 1 × 1 convolutional operation is employed to transform the quantity of *input* channels into the desired number of output channels. Subsequently, incorporate the *input* value using a shortcut. Finally, the output should be passed through *Hardswish* activation function. Assuming the input is denoted as *input* and the output as *output*, the equation for MS-AFB can be expressed as follows:

$$x_1, x_2, x_3, x_4 = split(Conv_{1*1}(input)) \tag{1}$$

$$y_i = \begin{cases} x_i, i = 1 \\ Conv_{3*3}(x_i), i = 2 \\ Conv_{3*3}(x_i + y_{i-1}), 3 \le i \le 4 \end{cases} \tag{2}$$

$$out_i = y_i \otimes split(softmax(Conv_{1*1}(concat(y_i)))) 1 \le i \le 4 \tag{3}$$

$$output = Hardswish(Conv_{1*1}(concat(out_i)) + input) 1 \le i \le 4 \tag{4}$$

After each 3 × 3 convolution kernel in the Backbone and each Concatenation operation in the Neck, an MS-AFB is incorporated to extract multi-scale features and improve the receptive fields. The incorporation of residual connections in MS-AFB within the Backbone primarily serves to increase the network depth and address issues such as gradient vanishing and network degradation. In the Neck module, the primary objective is to accomplish feature fusion. The network is currently situated at a profound level, and the incorporation of residual connections is unlikely to provide significant assistance. Therefore, MS-AFB in Neck lacks an additional connection, denoted as MS-AFB**.

In an effort to optimize the receptive field during primary feature extraction, this study aims to configure the 3 × 3 convolution kernel in Res2block in MS-AFB for Backbone as dilated convolution (MS-AFB*). To mitigate the issue of the grid effect, the dilation rates were set to 1, 2, and 5, drawing inspiration from reference [22]. However, an excessive employment of dilated convolutions not only results in the loss of information continuity and correlation, thereby reducing detection accuracy, but also introduces substantial inference delays. Consequently, only the 3 × 3 convolution kernel is configured within the Res2block in the final MS-AFB of the Backbone as a dilated convolution. To assess the efficacy of this concept, three experiments were carried out: ① MS-AFB replaced four C2f in the BackBone of YOLOv8; ② MS-AFB replaced the initial three C2f in BackBone + MS-AFB* replaced the final C2f; ③ MS-AFB* replaced four C2f in BackBone. The experimental results are presented in Table 1. Experiment 3 demonstrates that replacing all with MS-AFB* does not improve detection accuracy but significantly diminishes speed. Although the frames per second (FPS) of Experiment 2 experienced a slight decrease, there was an improvement in the mAP@0.5 metric. Consequently, only the fourth MS-AFB of the Backbone is configured as MS-AFB*.

The introduction of MS-AFB module has improved the network's capacity to extract and filter significant features across various scales. The mAP at an Intersection over Union (IoU) threshold of 0.5 on the improved Northeastern University Defect Dataset (NEU-DET) shows an improvement of 1.9 %.

Table 1
MS-AFB experimental results.

Methods	mAP@0.5	mAP@0.5.0.95	FLOPs(G)	Params(M)	FPS
①	0.841	0.555	8.9	3.39	102
②	0.847	0.552	8.9	3.39	99
③	0.84	0.551	8.9	3.39	82

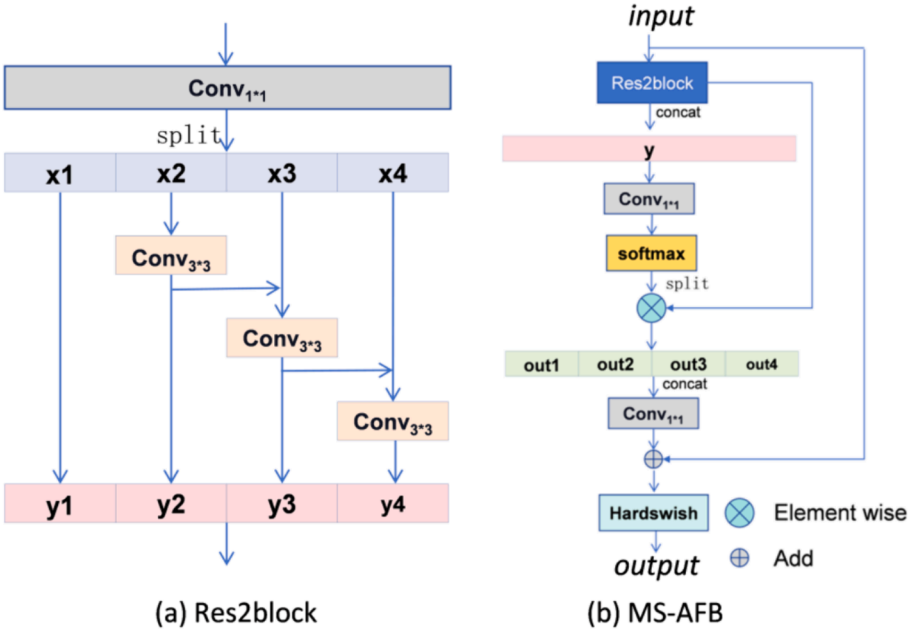

Fig. 2. Res2block and MS-AFB network structure.

3.4. DCA-GSPPFCSPC

SPPFCSPC exhibits a robust feature fusion capability that can improve accuracy; however, it also results in a substantial increase in the number of parameters. GhostConv [23] posits that certain feature maps exhibit a significant level of similarity, leading to a significant redundancy among them. The redundant information can be leveraged to acquire feature maps via a basic linear operation. When capturing cross-channel information, CA [24] has the capability to capture directional and positional information, thereby enhancing the model's ability to accurately locate and identify the target of interest. Meanwhile, CA demonstrates relative flexibility and lightweight properties, enabling its seamless integration into various positions through a plug-and-play approach. To improve the extraction of topological structures of diverse defects with varying shapes on the steel surface and to uncover the subtle distinctions among intra-class defects, while simultaneously optimizing the trade-off between accuracy and the number of parameters. DCA-GSPPFCSPC is formulated based on the architecture of SPPFCSPC, incorporating GhostConv and the improved CA (Fig. 3).

The input to DCA-GSPPFCSPC comprises two parallel branches. The initial branch involves passing through a 1 × 1 GhostConv(cv2) layer to facilitate the subsequent integration of features from both the output and input. The second branch includes a 1 × 1 GhostConv (cv1) layer to regulate the channel number. Subsequently, the output of cv1 undergoes processing by Dycaconv to incorporate an adaptive attention mechanism. Next, the preceding output undergoes sequential processing through three 5 × 5 maximally pooled downsampling layers. Next, the quantity of channels is altered using a 1 × 1 GhostConv (cv3), and feature extraction is improved through a 3 × 3 GhostConv (cv4). Finally, the outputs of cv2 and cv4 are combined in the channel direction, and the ultimate output is derived through a 1 × 1 GhostConv operation on cv5.

Fig. 4 depicts the structural diagram of Dycaconv, which introduces an intermediate branch (Adaptive pool + Linear + Softmax) building upon CA to enable dynamic adaptive fusion. Upon receiving the input {bs, c, h, w}, the Dycaconv module initiates the process by conducting average pooling on the height and width dimensions of the input feature map. This operation results in two outputs: {bs, c, 1, w} and {bs, c, h, 1}. Subsequently, the inputs are concatenated and subjected to a 1 × 1 convolution operation to adjust the channel number back to the original 1/32, yielding a tensor of dimensions {bs, c/32, h + w, 1}. Subsequently, the output is evenly distributed across the channels, and each channel undergoes a 1 × 1 convolution operation to adjust the number of channels to input channels. Subsequently, the application of the sigmoid function yields the outputs {bs, c, h, 1} and {bs, c, 1, w}. The middle branch undergoes adaptive average pooling, initially transforming the input dimensions from {bs, c, h, w} to {bs, c, 1, 1}. Subsequently, following the traversal of a fully connected layer and the application of the softmax function, the intermediate output {bs, 2} is acquired. The left branch functions as a residual join, directly multiplied with both the middle and right branches to yield a result that is subsequently convolved with a 1 × 1 matrix to obtain the final output {bs, c_out, h, w}, where c_out represents the quantity of output channels.

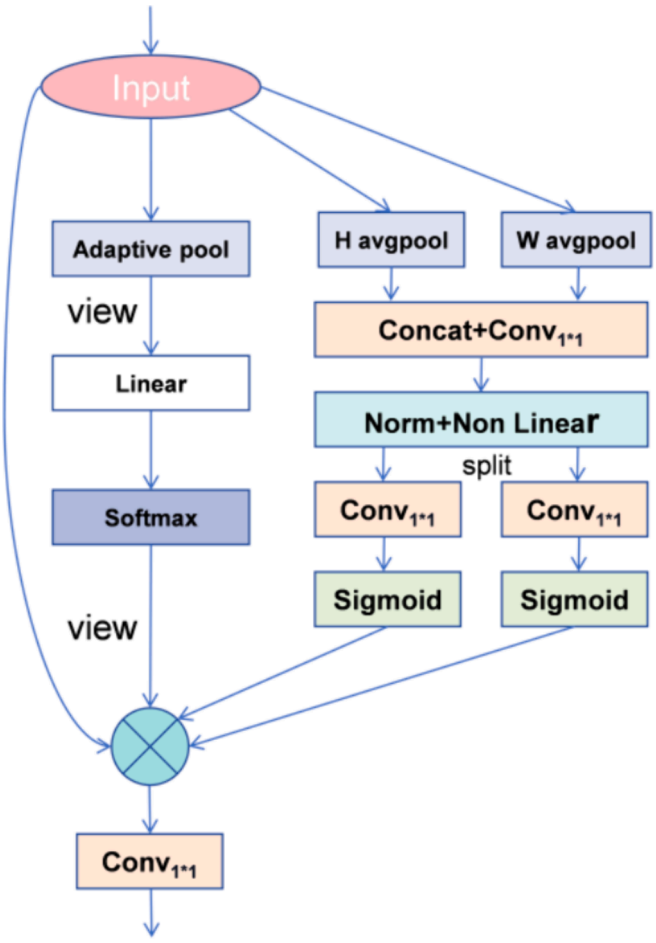

Fig. 4. DycaConv architecture.

After replacing SPPF with DCA-GSPPFCSPC and SPPFCSPC in YOLOv8, DCA-GSPPFCSPC not only results in a 1 % mAP improvement compared to SPPFCSPC on the improved NEU-DET but also decreases the parameter number by 30 %. Upon the integration of DCA-GSPPFCSPC into YOLOv8, replacing SPPF, the network's feature fusion capability improved, resulting in a 2.9 % increase in mAP on the improved NEU-DET.

3.5. Lightweight efficient detection head (LE Head)

In the field of object detection, classification and regression tasks are inherently conflicting challenges. Coupling network outputs not only impacts the convergence speed of the network but also significantly impairs its performance. Furthermore, YOLOX [8] conducted comprehensive experiments demonstrating that decoupling heads offer a

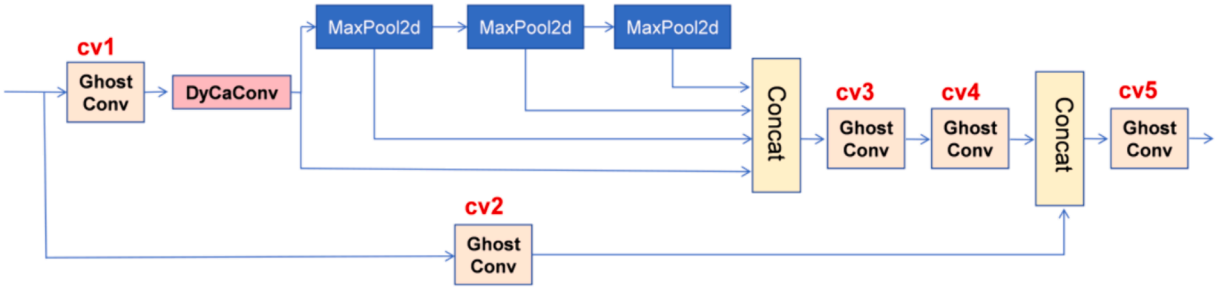

Fig. 3. DCA-GSPPFCSPC architecture.

significant advantage in end-to-end detection.

YOLOv8 modified its coupling head to a decoupling head inspired by YOLOv5. While the experiment revealed performance improvements, it was observed that YOLOv8 head imposes a significant computational burden. The decoupling head of YOLOv8n contributes approximately 40 % to the network computation load, and its share of the overall network parameters is approximately 35 times greater than that of YOLOv5n's coupling head in the total network parameters. Consequently, to address the challenges encountered with YOLOv8, improvements have been implemented to its header, leading to the introduction of the Lightweight Effective Detection Head (LE head). The configuration of LE head is illustrated in Fig. 1.

In LE Head, a LMSC module is proposed (Fig. 5). The design concept of this module is derived from GhostNet [23]. The input feature map of LMSC approach is initially partitioned into three segments along the channel axis. Specifically, one-half of the channel number undergoes the Cheep operation, one-quarter is subjected to 3 × 3 convolution, and the remaining one-quarter is processed through 5 × 5 convolution. Subsequently, these three components are merged in the channel dimension, culminating in the generation of the output via a 1 × 1 convolution. In the header section, LMSC method is employed to substitute the 3 × 3 convolution that is decoupled in YOLOv8 header. Subsequently, the output of LMSC module undergoes decoupling through two 1 × 1 convolutions to generate the final output.

Assuming the input channel number of LMSC method is *Cin*, the output channel number as *Cout*, and the width and height of the feature map as *W* and *H*. The parameter quantities for normal convolution and LMSC approach are presented in Equations (5) and (6) (excluding bias), and the computational costs are detailed in Equations (7) and (8) (excluding bias), respectively.

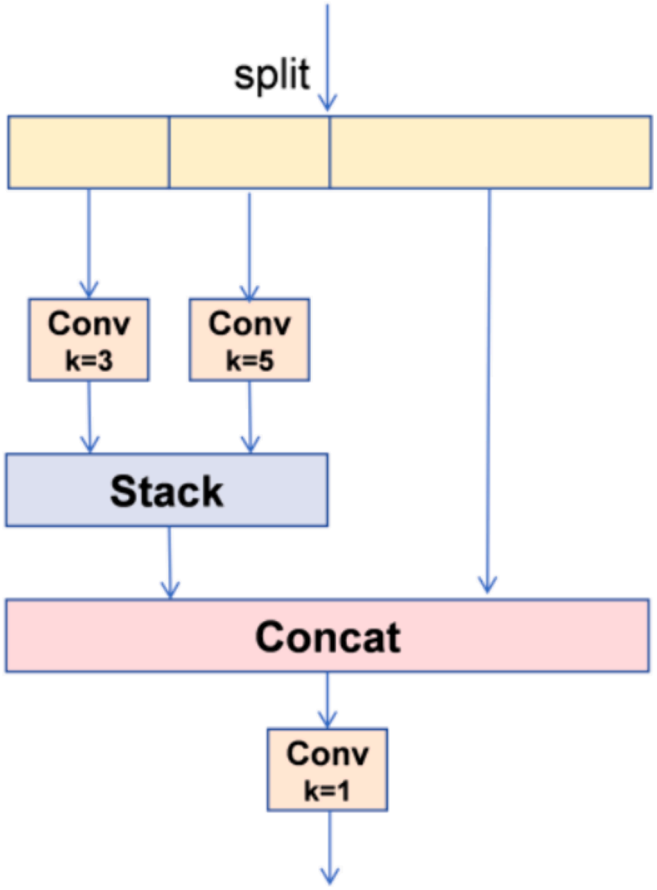

Fig. 5. LMSC architecture.

$$para1 = Cout*(3*3*Cin) \tag{5}$$

$$para2 = \frac{Cout}{4}*(5*5+3*3)*\frac{Cin}{4} \tag{6}$$

$$Flops1 = Cout*(3*3*Cin)*W*H \tag{7}$$

$$Flops2 = \frac{Cout}{4}*(5*5+3*3)*\frac{Cin}{4}*W*H \tag{8}$$

In Eqs. (5) and (6), the quantity of parameters and computational load of LMSC method amounts to 23.6 % of a standard 3 × 3 convolution. Thus, in comparison to standard 3 × 3 convolutions, this module is more lightweight and incorporates multi-scale information.

The incorporation of LE-Head in the system has led to the inclusion of multi-scale information in the detection header. This integration has resulted in a 33 % decrease in the number of header parameters, a 66 % reduction in Floating Point Operations (FLOPs), and a 0.9 % improvement in map performance on the improved NEU-DET.

3.6. AP-RFB

In the deep layers of the network, the output feature maps exhibit low resolution yet possess a robust ability to represent semantic information. However, challenges arise, including the propensity for feature loss and the limitation in extracting profound information. For defect detection against complex backgrounds, this situation can lead to potential confusion between defects and backgrounds, consequently elevating the likelihood of false positives and missed detections. RFB neural network exhibits robust generalization and parallel information processing capabilities by employing multidimensional nonlinear mapping. Additionally, it demonstrates strong clustering analysis proficiency, making it well-suited for both intra-class aggregation and inter-class separation of surface defects in steel. The SE-block channel attention mechanism [19,34], has the capability to selectively assign weights to individual channels, consequently enhancing the information output. Therefore, AP-RFB illustrated in Fig. 6 has been developed. It can offer an expanded receptive field for the network, thereby enriching the network's output with more global features. This, in turn, helps in mitigating false positives and missed detections in complex backgrounds, ultimately enhancing the model's capability for defect detection in complex backgrounds. Simultaneously, emphasis is placed on identifying defect characteristics and integrating them adaptively, consequently decreasing both the false detection rate and missed detection rate. This approach improves the defect detection model's generalization capability in complex backgrounds.

In Fig. 6, AP-RFB method concurrently undergoes three 1 × 1 convolutions on the input in parallel, resulting in a reduction of the channel number to 1/8 of the original. The second 3 × 3 convolution on the left and middle is employed for feature extraction and to increase the channel number twofold. The second 3 × 3 convolution on the right augments the channel number by 1.5 times, while the third 3 × 3 convolution doubles the number of channels. To mitigate information loss during downsampling, the three 3 × 3 yellow convolutions employ dilated convolutions to increase the receptive field. The dilated convolutions from left to right correspond to dilation rates of 1, 3, and 5, respectively. The subsequent procedure involves passing them through an SE-block and subsequently adjusting weights adaptively through the utilization of the softmax function. Further conduct element-wise operations on the results with the previous z_0, z_1, and z_2, and subsequently concatenate the results along the channel axis to derive *z*. To modify the number of output channels, one can achieve the desired channel number by executing a 1 × 1 convolution operation on them. Finally, the residual connection links z with the input and outputs the result using the *Silu* activation function. Assuming the input is *i* and the output as *o*, the calculation equation for AP-RFB can be expressed as follows:

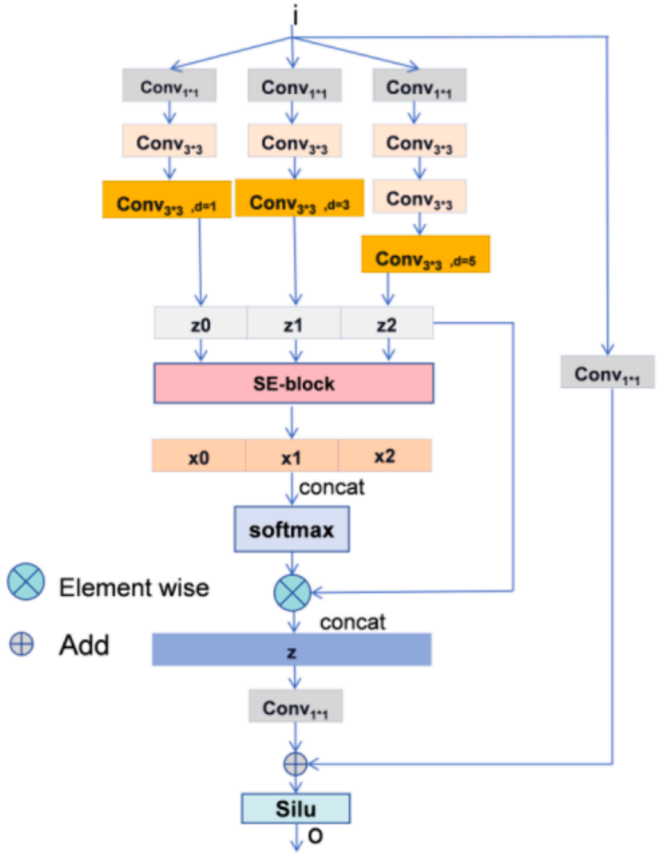

Fig. 6. AP-RFB architecture.

$$z_0 = Conv_{3*3}(Conv_{3*3}(Conv_{1*1}(i)), d = 1) \tag{9}$$

$$z_1 = Conv_{3*3}(Conv_{3*3}(Conv_{1*1}(i)), d = 3) \tag{10}$$

$$z_2 = Conv_{3*3}(Conv_{3*3}(Conv_{3*3}(Conv_{1*1}(i))), d = 5) \tag{11}$$

$$z = concat(softmax(concat(SE - block(z_i)))\otimes z_i), 0 \le i < 3 \tag{12}$$

$$o = Silu(Conv_{1*1}(i) + Conv_{1*1}(z)) \tag{13}$$

Upon the integration of AP-RFB into the final layer of the backbone, there was an improvement in the network's receptive field, resulting in a 0.9 % improvement in mAP on the improved NEU-DET.

3.7. Loss function

The loss function of YOLOv8 comprises two components: category classification loss and bounding box regression loss. The category classification loss is determined by cross-entropy loss, whereas the bounding box regression loss is calculated using a combination of Distribution Focal Loss (DFL) and Complete Intersection over Union (CIOU) loss functions, with weights assigned in a ratio of 0.5:1.5:7.5. CIOU takes into account the intersection area, the distance between the center points of the predicted box and the actual box, and the aspect ratio of the predicted box.

Zhang et al. [26] identified the irrationality of CIOU in prediction frame regression. They observed that the width and height of the prediction box exhibit an inverse relationship, where an increase in one dimension results in a decrease in the other. This phenomenon hinders the convergence speed of CIOU. Zhang et al. [26] introduced a proficient EIoU loss along with a regression version that emphasizes the loss. The regression process is centered on high-quality anchors; however, the full potential of non-monotonic focusing mechanisms remains underutilized due to the staticity of the current focusing mechanism. Tong et al. [21] proposed an outlier parameter to characterize the anchor's quality. They also developed a WIoU loss function that incorporates a dynamic non-monotonic focusing mechanism. This mechanism dynamically assigns the optimal gradient gain to the anchor frame. WIoU does not incorporate additional metrics such as aspect ratio, resulting in faster processing speeds in comparison to CIOU.

In Fig. 7, the visualization of NEU-DET reveals variations in the size of defects on the steel surface, indicating the presence of small targets. While WIoU demonstrates strong performance, it is important to note that its evaluation based on IOU [29] makes it highly sensitive to deviations in small objects. Even minor positional discrepancies can result in a reduction in IOU. Consequently, the utilization of IOU metrics may result in a decrease in detection performance. However, Normalized Weighted Distance (NWD) [20] lacks sensitivity to the scale of objects and is better suited for assessing similarity among small objects. To address this issue, the research integrated Weighted Intersection over Union (WIoU) loss with NWD loss in a 2:8 ratio, thereby replacing the initial CIOU loss.

After incorporating WIoU and NWD, the equation for the bounding box loss function can be expressed as follows:

$$L_{loc} = 7.5*((1-\beta)*NWD + \beta*WloU) + 1.5*DFL \tag{14}$$

where L_{loc} is the bbox regression loss; DFL, NWD, and WloU are DFL loss, NWD loss, and WloU loss, respectively; and β is a weight proportion coefficient used to control the proportion of WIoU in the bounding box regression loss, with a value range from 0 to 1. In subsequent experiments, it is set to 0.8.

NWD employs the Gaussian distribution associated with the predicted target and the actual target to determine the similarity between the bounding box and the actual target box. The incorporation of NWD loss into the bounding box regression loss function may address the limitations of WIoU loss for small object detection. By preserving WIoU loss, the algorithm demonstrates accelerated convergence in predicting bounding box localization and improves the overall performance of the model. The experimental results demonstrate that the inclusion of WIoU and NWD leads to a 1.4 % improvement in mAP.

3.8. Bilinear interpolation

Nearest neighbor interpolation involves assigning the value of the new pixel to the value of the nearest original pixel. While it offers a low computational cost and rapid calculation speed, this method may result in discontinuities in the grayscale values of pixels within the interpolated image. In defect detection, the texture and details of the image play a crucial role in localizing defects. The utilization of nearest neighbor interpolation may introduce considerable noise and artifacts to the image. When identifying defects such as "cracking" that closely resemble the background, the introduction of noise and artifacts due to nearest neighbor interpolation can significantly disrupt the network. This interference may lead the network to erroneously incorporate noise and artifacts while identifying such defects. Consequently, the detection efficacy of these defects is inadequate.

Bilinear interpolation considers the correlation effect of the four immediate neighboring points surrounding the sampled point under examination. With a marginal rise in computational complexity, bilinear interpolation improves smoothness and edge characteristics, diminishes the discontinuity of sampled grayscale values, consequently reducing noise and artifacts. This improvement improves defect detection, making it akin to the background. In the experiments conducted on the improved NEU-DET, the incorporation of bilinear interpolation resulted in a 1 % improvement in mAP. However, when encountering defects resembling the background, such as "cracking", the improvement rose to 5 %.

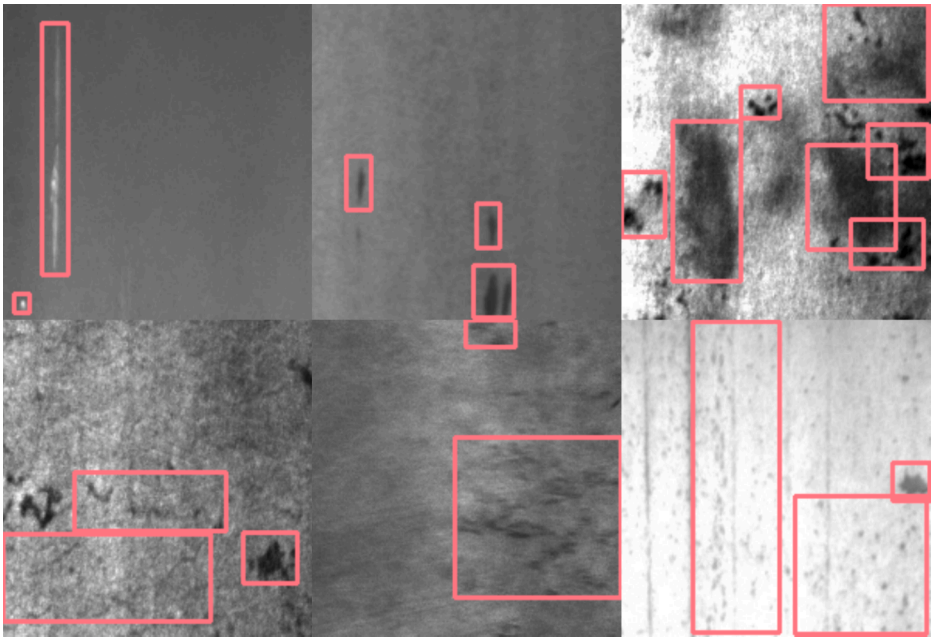

Fig. 7. NEU-DET partial image visualization.

4. Experiments and results

4.1. Datasets

The study utilized various datasets including NEU-DET [30], PASCAL VOC2007 [32], Crack defect dataset [35], and a self-constructed cage guide dataset to evaluate the overall effectiveness of the proposed model. NEU-DET has been improved through the utilization of data augmentation methods such as scaling, distortion, rotation, and color gamut transformation. The augmented dataset comprises a total of 5,372 images. PASCAL VOC2007 is not directly relevant to this study. However, Real-Time Detection Transformer (RT-DETR), YOLOv8, and other networks have conducted experiments using COCO dataset, which bears similarities to PASCAL VOC2007. To ensure fairness and to more accurately represent the generalization capabilities of the proposed network, this study utilized PASCAL VOC2007 dataset for the experiments. The self-compiled dataset comprised 900 defect images of cage guides. These images were captured by an industrial camera installed on the vertical shaft simulation platform within State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines. To simulate the three prevalent defects in the cage guide, namely abrasion, crack, and displacement, a total of 100 images were gathered for each defect category across three light intensities (10 lx, 20 lx, and 50 lx). Furthermore, the dataset was augmented through rotation, mirroring, and cropping techniques, resulting in an expansion to 3,000 instances. The datasets are partitioned into training, validation, and testing sets following a ratio of 7:1:2.

4.2. Experimental platforms

To accurately depict the process of model deployment and operation, the research developed both a training platform and a deployment platform to facilitate the training and deployment of models.

4.2.1. Training platform

This experiment utilizes YOLOv8 code library and employs Stochastic Gradient Descent (SGD) optimizer for network training. The initial learning rate is established at 0.01, and it follows the cosine decay strategy for learning rate adjustment. The momentum decay and weight decay parameters are specified as 0.937 and 0.0005, respectively. The training process involves running 200 epochs with a batch size of 16. All images are standardized to a resolution of 640 × 640 before being fed into the neural network. The network has been implemented utilizing Python programming language. Conducting training and testing on NVIDIA RTX3080 GPU. All experiments are carried out on the Linux operating system. Python version utilized is 3.8, CUDA version is 11.8, and Torch version is 2.0. To validate the results, metrics such as mAP, parameter number, and computational complexity are employed for evaluation purposes.

4.2.2. Deployment platform

In the deployment platform of Jetson Tx2 nx,Python version utilized is 3.6, CUDA version is 10.2, Torch version is 1.10, JetPack version is 4.5.1, and TensorRT version is 7.2.0. The device utilized in this study is RAM 16G Orange Pie 5+, featuring Rexchip RK3588 as the primary controller. It is integrated with a quad-core A76 and quad-core A55 octa-core CPU, along with a 6TOPs arithmetic NPU. The high arithmetic NPU is capable of supporting INT4, INT8, INT16, and FP16 hybrid computations, thereby enhancing the speed of network model inference. In Fragrant Orange Pie 5+, Ubuntu 20.04 operating system is utilized.

4.3. Ablation experiment

To evaluate the efficacy of individual components within the proposed network, ablation experiments were performed on the improved NEU-DET, utilizing YOLOv8 as the benchmark network. Conduct experiments on each of the seven improvement points and their associated combinations.

Table 2 presents the results of the ablation experiment. The replacement of the C2f module with MS-AFB results in an improvement of the model's receptive field. Additionally, the incorporation of multi-scale information allows the network to capture more complex details. The network evaluation parameters mAP@0.5 and mAP@0.5:0.95 demonstrated an increase of 1.9 % and 1.6 %, respectively. Replacing the SPPF module with DCA-GSPPFCSPC improves the network model's capacity to assimilate and integrate surface defect characteristics of strip steel effectively. The mAP at IoU threshold of 0.5 and mAP between IoU thresholds of 0.5 and 0.95 demonstrated an improvement, showing a rise of 2.9 % and 3.9 %, respectively. After the integration of LE Head to substitute the detection head in YOLOv8, there has been a reduction in

Table 2
Experimental results comparing proposed method with baseline network ablation.

MS-AFB	DCA-GSPPFCSPC	LE Head	AP-RFB	NWD	WIoU	Bilinear	mAP@.5(%)	mAP@.5:.95(%)	Params(M)	FLOPs(G)
							0.828	0.536	3	8.2
√							0.847	0.552	3.39	8.9
	√						0.857	0.575	4	8.9
		√					0.837	0.543	**2.8**	**6.2**
			√				0.837	0.545	3.17	8.2
				√			0.837	0.537	3	8.2
					√		0.835	0.539	3	8.2
						√	0.838	0.541	3	8.2
√						√	0.857	0.567	3.3	8.9
√					√		0.859	0.572	3.39	8.9
√				√			0.856	0.567	3.39	8.9
√		√					0.857	0.579	3.2	7
				√	√		0.842	0.546	3	8.2
√	√	√					0.868	0.593	4.2	7.8
		√		√	√		0.842	0.549	**2.8**	**6.2**
		√	√	√	√		0.856	0.536	3.1	6.5
√	√	√		√	√	√	0.874	0.599	4.2	7.8
√	√	√	√	√	√	√	**0.882**	**0.603**	4.5	8.1

computational and parameter demands, along with the incorporation of additional multi-scale information. Consequently, the metrics mAP@0.5 and mAP@0.5:.95 have shown increments of 0.9 % and 0.7 %, respectively. Upon the implementation of AP-RFB, the receptive field is expanded, resulting in a 0.9 % increase in both mAP@0.5 and mAP@0.5:.95. Upon joining NWD, there has been an improvement in the detection capability for small targets, resulting in a 0.9 % improvement in mAP@0.5. Following enrollment in the WIoU program, the mAP@0.5 metric exhibited a 0.7 % improvement. The inclusion of bilinear interpolation resulted in a 1 % improvement in mAP at an IoU threshold of 0.5.

In summary, the utilization of MS-AFB, DCA-GSPPFCSPC, LE Head, AP-RFB, NWD, WIoU, and bilinear interpolation techniques collectively contribute to enhancing the accuracy of the model. Among these methods, DCA-GSPPFCSPC demonstrates the greatest improvement, albeit at the cost of introducing a higher number of additional parameters. In Table 2, the improvements demonstrate effectiveness. Upon incorporating all the suggested improvements, there was a slight increase in the number of parameters, a slight decrease in computational workload, and a significant improvement in the model's detection accuracy. Specifically, the mAP@0.5 and mAP@0.5:.95 increased by 5.4 % and 6.7 %, respectively.

4.4. Comparative experiments

To validate the effectiveness of the proposed MSAF-YOLOv8n, a comparison was conducted with established object detection networks, including Faster R-CNN (R50) [27], YOLOv3 [6], YOLOv5 [7], YOLOv7 [10], Transformer Prediction Head (TPH)-YOLOv5 [33], RT-DETR-R18 [28], and LF-YOLO [31]. Notably, Faster R-CNN (R50) utilized pre-trained weights, while the remaining models were trained from scratch. The evaluation of accuracy employed metrics such as mAP@0.5, mAP@0.5:.95, Params, and FLOPs. The comparative experiments were carried out using the improved NEU-DET, and the results are presented in Table 3. Analysis of the comparative experimental results indicates that the proposed method attained the highest MAP value. Compared to other networks, MSAF-YOLOv8n demonstrates an improvement in detection accuracy. MSAF-YOLOv8n demonstrates superior performance compared to YOLOv3-tiny across all metrics. It achieves a 21 % higher mAP while utilizing 51 % fewer parameters and exhibiting 62 % lower computational complexity.

MSAF-YOLOv8n exhibits a superiority in detecting complex defects such as 'cr'. In comparison to the benchmark network, mAP has shown a 15.8 % improvement, while in comparison to YOLOv7-tiny, the mAP has demonstrated a 38.8 % improvement. The observed phenomenon is attributed to the inadequate feature extraction capability and restricted receptive field of alternative networks when confronted with a substantial volume of extraneous noise and complex features. The reduction of effective features extracted by neural networks can lead to network fitting being impacted by noise rather than relevant features, thereby impacting the accuracy of detection.

In order to conduct a comprehensive assessment of the detection capabilities of MSAF-YOLOv8n, it is compared against algorithms known for superior detection outcomes, such as TPH-YOLOv5 [33], YOLOv8n, and YOLOv8s, using the augmented NEU-DET. Subjective indicators were chosen, and the results of the detection are presented in Fig. 8. When addressing four types of defects (e.g., Rs, Pa, In, Sc), the algorithms mentioned above demonstrate effective detection performance, attributed to their significant difference from the background. However, the detection accuracy of most networks is inadequate when targeting defects such as Cr and Ps, often resulting in false positives and missed detections. This complexity arises from the complex textures and

Table 3
Comparative experiments of various methods on improved NEU-DET.

Methods	mAP@.5(%)	mAP@.5:.95(%)	Params(M)	FLOPs(G)	Cr(%)	In(%)	Pa(%)	Ps(%)	Rs(%)	Sc(%)
Fasetr R-CNN(R50)[27]	0.692	0.323	41.4	71.7	–	–	–	–	–	–
YOLOv3-tiny	0.672	0.33	8.67	12.9	0.577	0.786	0.884	0.441	0.678	0.667
YOLOv5n	0.757	0.422	1.76	4.2	0.4	0.849	0.902	0.818	0.661	0.914
YOLOv7-tiny	0.734	0.394	6.02	13.1	0.37	0.828	0.89	0.81	0.60	0.90
YOLOv5s	0.817	0.499	7.0	15.8	0.555	0.886	0.918	0.869	0.736	0.937
LF-YOLO[31]	0.726	0.381	7.25	16.2	0.376	0.78	0.918	0.878	0.541	0.861
TPH-YOLOv5[33]	0.803	0.466	7.18	30.0	0.541	0.845	0.922	0.85	0.736	0.923
RT-DETR-R18[28]	0.853	0.578	19.8	57.0	0.703	0.893	0.933	0.84	0.78	0.967
YOLOv8n	0.828	0.536	3.0	8.2	0.6	0.885	0.927	0.867	0.73	0.957
YOLOv8s	0.873	0.604	11.2	28.4	0.732	0.902	0.942	0.9	0.809	0.955
MSAF-YOLOv8n	0.882	0.603	4.5	8.1	0.758	0.896	0.946	0.923	0.802	0.971

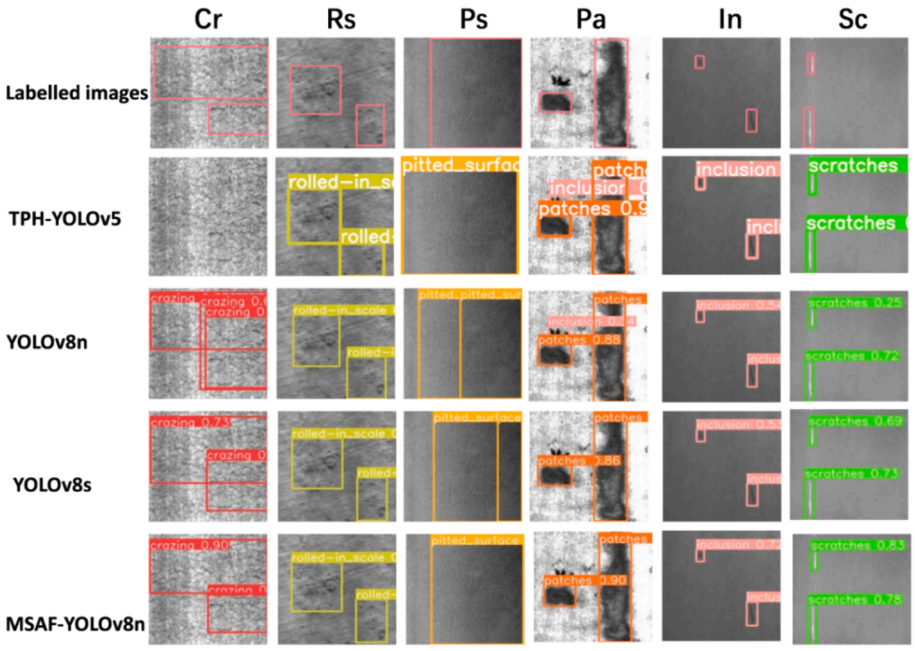

Fig. 8. Detection performance comparison of several algorithms on dataset.

the presence of significant irrelevant noise in these two types of defects. Networks with limited extraction capabilities and small receptive fields tend to capture noise rather than features, leading to subpar detection performance in these specific categories. MSAF-YOLOv8n improves feature extraction and fusion capabilities through the incorporation of MS-AFB, which offers multi-scale feature extraction, and DCA-GSPPFCSPC, known for its robust feature fusion capabilities. AP-RFB technique improves the receptive field of the model, while bilinear interpolation helps to mitigate the impact of background and noise resembling defects. Consequently, this approach improves the detection capabilities of MSAF-YOLOv8n for complex defects such as Cr and PS.

To improve the visualization of the comparison results, six types of defects were represented using a heatmap on NEU-DET. In Fig. 9, the experimental results indicate that alternative networks extract a lower number of effective features and tend to overly emphasize noise, potentially leading to the misclassification of defects as part of the background. MSAF-YOLOv8n model possesses a substantial receptive field along with multi-scale feature extraction and fusion capabilities. These attributes enable the model to capture texture details more accurately and comprehensively, accurately locate defect positions, and exhibit effective detection performance across various types of defects. Furthermore, MSAF-YOLOv8n is characterized by a modest number of parameters and computational complexity, rendering it suitable for deployment on embedded devices with limited resources. Thus, in comparison to other networks, MSAF-YOLOv8n demonstrates higher competitiveness and is better suited for defect detection tasks on steel surfaces.

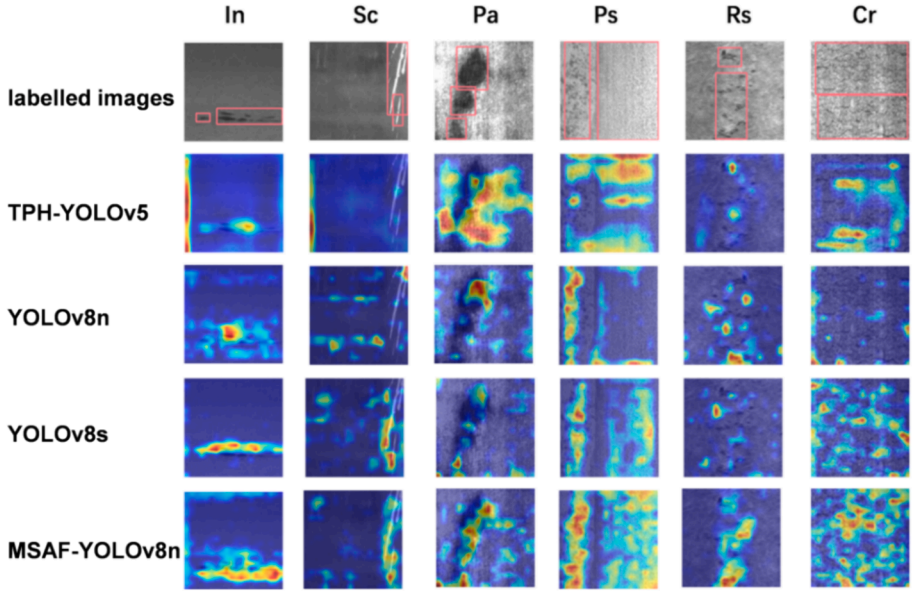

Fig. 9. Heat map visualization.

4.5. Performance on various datasets

To assess the generalization performance of MSAF-YOLOv8n, comparative experiments were carried out with Fasetr R-CNN (R50) [27], YOLOv7-tiny [10], YOLOv8n [11], LF-YOLO [31], RT-DETR-R18 [28], and TPH-YOLOv5 [33] on PASCAL VOC2007 dataset, Crack dataset, and a self-constructed cage guide dataset. The experimental results are presented in Tables 4-6.

In Table 4, while MSAF-YOLOv8n has not been fine-tuned for PASCAL VOC2007 dataset, it demonstrated comparable performance to other high-performing networks. In comparison to LF-YOLO, MSAF-YOLOv8n demonstrates a reduction of 62 % in network parameters and 50 % in computational complexity. Furthermore, its mAP@0.5 and mAP@0.5.0.95 metrics have shown improvements of 15.6 % and 19.9 %, respectively. Compared to TPH-YOLOv5, MSAF-YOLOv8n exhibits a reduction of 63 % in network parameters and 27 % in computational complexity. Additionally, its mAP at IoU thresholds of 0.5 and 0.5–0.95 have shown improvements of 1.3 % and 5.9 %, respectively. The utilization of pre-trained weights in Faster R-CNN results in a 1.5 % higher mAP@0.5 compared to the proposed approach. However, in terms of mAP@0.5:0.95, the proposed network demonstrates a significantly superior performance compared to it. The training results of RT-DETR-R18 indicate a dependency of transformer-based detection models on pre-trained models, with direct training resulting in suboptimal performance on small and medium-sized datasets. The aforementioned results illustrate the robust performance of the proposed network.

Table 5 illustrates the commendable performance of MSAF-YOLOv8n on the Crack dataset. In comparison to other prominent networks, MSAF-YOLOv8n exhibits an improvement in mAP@0.5:0.95. In comparison to LF-YOLO, MSAF-YOLOv8n demonstrated a 34.3 % increase in mAP@0.5:0.95. Compared to TPH-YOLOv5, MSAF-YOLOv8n demonstrates a 26.9 % improvement in mAP@0.5:0.95. The mAP at IoU thresholds of 0.5 and 0.95 for MSAF-YOLOv8n has shown a 42.2 % improvement compared to YOLOv7-tiny. Compared to the benchmark network, MSAF-YOLOv8n's mAP at IoU thresholds of 0.5 and 0.95 showed a 6.3 % improvement.

The experimental results presented in Table 6 demonstrate that MSAF-YOLOv8n attains the highest mAP value on the self-constructed dataset. Furthermore, the mAP@0.5:0.95 metric significantly surpasses that of other networks. Meanwhile, the identification of abrasions, cracks, and dislocations is more effective compared to alternative techniques. In Fig. 10, the detection performance of MSAF-YOLOv8n and YOLOv8n on the custom dataset is presented. The first row displays the annotated image, the second row shows the detection outcome of MSAF-YOLOv8n, and the third row presents the detection outcome of YOLOv8n. According to the detection results, YOLOv8n performed various inspections on a defect in the initial column. In the subsequent column, it was observed that the annotated image failed to identify a defect. However, both the proposed model and YOLOv8n detected the defect, although YOLOv8n produced a false positive. In the final column of images with extremely low brightness levels, the proposed approach fails to detect one highly concealed defect, whereas YOLOv8n overlooks two defects. The results demonstrate that the proposed approach is better suited for detecting steel surface defects in complex backgrounds.

Based on the information presented in Tables 4-6, it can be inferred that MSAF-YOLOv8n mAP@0.5.0.95 outperforms other methodologies significantly. This suggests that the proposed network demonstrates superior performance in high-confidence predictions compared to other networks and exhibits improved capabilities in addressing complex and challenging detection tasks. In summary, MSAF-YOLOv8n model demonstrates robust generalization performance. Through targeted optimization tailored to specific scenarios, it is believed that improved detection outcomes can be attained.

Table 4
Comparative experimental results of MSAF-YOLOv8n with other high-performing networks on PASCAL VOC2007 dataset.

Methods	mAP@0.5	mAP@0.5.0.95	FLOPs (G)	Params (M)
Fasetr R-CNN(R50) [27]	0.71	0.398	71.7	41.4
LF-YOLO[31]	0.539	0.278	16.2	7.25
TPH-YOLOv5[33]	0.682	0.418	30.0	7.17
RT-DETR-R18[28]	0.593	0.407	57.0	19.8
YOLOv7-tiny	0.685	0.428	13.1	6.02
YOLOv8n	0.669	0.456	8.2	3.00
MSAF-YOLOv8n	0.695	0.477	8.1	4.56

5. Deployment in embedded devices

The model trained on NEU-DET was implemented on the Jetson TX2 NX and the Orange Pi 5 + platforms, with the image input size set to 640 x 640 by default. On Jetson TX2 NX, the pt->ONNX->engine conversion path was utilized, which is a widely employed method for converting a PyTorch (pt) file into an engine file for inference purposes. The model was implemented on NVIDIA Jetson TX2 NX platform using C++ [51]. The onnx model was exported directly, and FPS of FP32 and FP16 were measured at a batch size of 1 on the deployment platform. The specific experimental results are outlined in Table 7. The results indicate that the FPS is 27 when utilizing FP32 inference model and can reach up to 37 when employing FP16 inference model. Both FPS values meet the real-time target detection requirement of 24 FPS.

Meanwhile, deployment experiments were conducted on the Orange Pie 5 + . The process involved utilizing pt->onnx->rknn conversion to transform pt files into rknn files for inference purposes. In contrast to deploying Jetson, the time-consuming post-processing step was separated from ONNX during the export process and reimplemented the post-processing phase using C++ for inference purposes. The utilization of thread pooling for inference [52] maximizes the use of the 3 NPUs in the Orange Pie 5+, leading to an improvement in the speed of inference. The experimental results of implementing MSAF-YOLOv8n on Orange Pie 5 + are summarized in Table 8. The values "5″, "7″, up to "15″ denote the quantity of threads utilized when employing 3 npu reasoning. The experimental results demonstrate that the model achieves an FPS of over 30 when employing multi-threaded reasoning, thereby satisfying the real-time target detection requirement. In the experiment, it was observed that beyond 11 threads, the speed of reasoning became highly unstable. Therefore, it is more optimal to utilize 9 threads.

6. Conclusion and outlook

This study explored steel surface defect detection in complex backgrounds and introduced an algorithm named MSAF-YOLOv8n for steel surface defect detection. MS-AFB method improves the capability of multi-scale feature extraction by generating feature maps from various perspectives and granularities. Simultaneously, it adaptively chooses significant scale features to improve detection performance. DCA-GSPPFCSPC achieves an improved equilibrium between accuracy and parameter quantity through the effective fusion of features at various scales. AP-RFB method improves the network's receptive field through the utilization of dilated convolutions. It improves the capacity to extract detailed features by acquiring additional contextual information. LE Head employs LMSC method to significantly decrease the quantity of head references. In comparison to the advanced technology LF-YOLO, MSAF-YOLOv8 demonstrates superior performance on the augmented dataset NEU-DET. The mAP at IoU threshold of 0.5 (mAP@0.5) increased by 21.49 %, and mAP at IoU thresholds of 0.5 and 0.95 (mAP@0.5:0.95) increased by 58.27 %. In contrast, Params (M) decreased by 37.93 %, and FLOPs (G) decreased by 50 %. The experimental results from four datasets demonstrate that MSAF-YOLOv8n exhibits robust generalization performance. The designed model was deployed on embedded devices, resulting in real-time detection. This

Table 5
Comparative experimental results of MSAF-YOLOv8n with other high-performing networks on Crack dataset.

Methods	mAP@0.5	mAP@0.5.0.95	FLOPs(G)	Params(M)	Severe Crack	Crack
Fasetr R-CNN[27]	0.876	0.635	71.7	41.4	–	–
LF-YOLO[31]	0.796	0.393	16.2	7.25	0.825	0.673
TPH-YOLOv5[33]	0.799	0.467	30.0	7.17	0.823	0.775
RT-DETR-R18[28]	0.850	0.575	56.9	19.8	0.871	0.83
YOLOv7-tiny	0.688	0.314	13.1	6.02	0.742	0.634
YOLOv8n	0.895	0.673	8.2	3.00	0.905	0.895
MSAF-YOLOv8n	0.922	0.736	8.1	4.56	0.913	0.931

Table 6
Comparative experimental results of MSAF-YOLOv8n with other high-performing networks on self-built dataset.

Methods	mAP@0.5	mAP@0.5.0.95	FLOPs(G)	Params(M)	abrasion	crack	dislocation
Fasetr R-CNN[27]	0.828	0.553	71.7	41.4	–	–	–
LF-YOLO[31]	0.812	0.496	16.2	7.25	0.711	0.828	0.897
TPH-YOLOv5[33]	0.862	0.562	30.0	7.17	0.812	0.856	0.919
RT-DETR-R18[28]	0.861	0.646	56.9	19.8	0.807	0.837	0.938
YOLOv7-tiny	0.832	0.553	13.1	6.02	0.794	0.836	0.867
YOLOv8n	0.848	0.592	8.2	3.0	0.808	0.835	0.901
MSAF-YOLOv8n	0.875	0.649	8.1	4.5	0.824	0.857	0.945

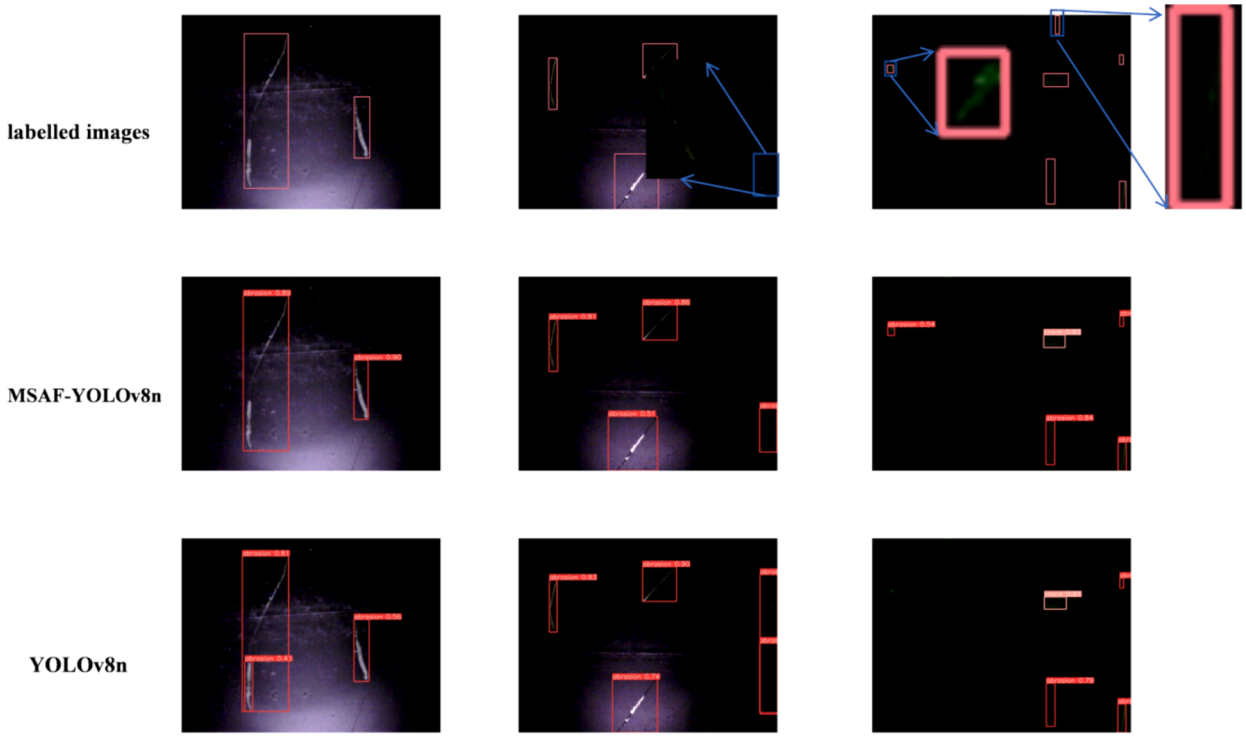

Fig. 10. Comparative demonstration of MSAF-YOLOv8n and YOLOv8n detection performance on self-built datasets.

Table 7
Performance of MSAF-YOLOv8n deployment on Jeston TX2 NX.

Methods	**FP32**	**FP16**
MSAF-YOLOv8n	27	37

Table 8
Performance of MSAF-YOLOv8n deployment on Orange Pie 5+.

Methods	**FP32**	**5**	**7**	**9**	**11**	**13**	**15**
MSAF-YOLOv8n	6	30	33	35	37	39	39

result also suggests that MSAF-YOLOv8n strikes a more optimal balance between speed and accuracy, demonstrating significant practical implications and improved competitiveness.

During the experimental investigation of steel surface defect detection, it was observed that multi-scale feature extraction and sensory field play a crucial role. While the proposed model improves detection accuracy, it also results in a decrease in detection speed. The network will undergo further optimization in the future to improve detection speed while upholding detection accuracy. Meanwhile, the approach outlined in this paper is a supervised method that necessitates the consideration of the quantity and accessibility of datasets. It typically involves manual preprocessing, a task that can be challenging due to the limited availability and high cost of labeled data. Consequently, semi-supervised or

unsupervised techniques should be employedfor industrial defect detection to address the challenges associated with data collection and labeling in the industrial domain.

CRediT authorship contribution statement

BaiTing Zhao: Writing – review & editing, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **YuRan Chen:** Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Data curation. **XiaoFen Jia:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Resources, Supervision, Validation, Writing – review & editing. **TianBing Ma:** Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

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

Data will be made available on request.

Acknowledgments

This work was supported in part by The National Natural Science Foundation of China under Grant 52174141, in part by The Natural Science Foundation of Anhui Province Grant 2108085ME158, and in part by Anhui University of Science and Technology Introduction Talent Research Initiation Fund under Grant 2022yjrc44.

References

[1] N. Neogi, D.K. Mohanta, P.K. Dutta, Review of vision-based steel surface inspection systems[J], EURASIP J. Image and Video Processing 2014 (2014) 1–19.
[2] S. Kim, W. Kim, Y.K. Noh, et al., Transfer learning for automated optical inspection [C]//2017 international joint conference on neural networks (IJCNN), IEEE (2017:) 2517–2524.
[3] Y. He, K. Song, Q. Meng, et al., An end-to-end steel surface defect detection approach via fusing multiple hierarchical features[J], IEEE Trans. Instrum. Meas. 69 (4) (2019) 1493–1504.
[4] X. Wen, J. Shan, Y. He, et al., Steel surface defect recognition: a survey[J], Coatings 13 (1) (2022) 17.
[5] K. He, X. Zhang, S. Ren, et al., Spatial pyramid pooling in deep convolutional networks for visual recognition[J], IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[6] Redmon J, Farhadi A. (2018) Yolov3: An incremental improvement[J]. arxiv preprint arxiv:1804.02767.
[7] G. Jocher. YOLOv5-Master. Accessed: Mar. 1, 2021. [Online]. Available:https://github.com/ultralytics/yolov5.
[8] Ge Z, Liu S, Wang F, et al. (2021) Yolox: Exceeding yolo series in 2021[J]. arxiv preprint arxiv:2107.08430.
[9] Li C, Li L, Jiang H, et al. (2022) YOLOv6: A single-stage object detection framework for industrial applications[J]. arxiv preprint arxiv:2209.02976.
[10] C.Y. Wang, A. Bochkovskiy, H.Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C], Proce. IEEE/CVF Conference on Comp. Vision and Pattern Recognition (2023) 7464–7475.
[11] Jocher Glenn. Ultralytics yolov8. https://github.com/ultralytics/ultralytics,2023.
[12] H. Chen, Y. Du, Y. Fu, et al., DCAM-Net: a rapid detection network for strip steel surface defects based on deformable convolution and attention mechanism[J], IEEE Trans. Instrum. Meas. 72 (2023) 1–12.
[13] C.C. Yeung, K.M. Lam, Efficient fused-attention model for steel surface defect detection[J], IEEE Trans. Instrum. Meas. 71 (2022) 1–11.
[14] X. Yu, W. Lyu, C. Wang, et al., Progressive refined redistribution pyramid network for defect detection in complex scenarios[J], Knowl.-Based Syst. 260 (2023) 110176.
[15] Xing, M. Jia, A convolutional neural network-based method for workpiece surface defect detection, Measurement 176 (2021) 109185.J.
[16] S. Li, F. Kong, R. Wang, et al., EFD-YOLOv4: a steel surface defect detection network with encoder-decoder residual block and feature alignment module[J], Measurement 220 (2023) 113359.
[17] S.H. Gao, M.M. Cheng, K. Zhao, et al., Res2net: a new multi-scale backbone architecture[J], IEEE Trans. Pattern Anal. Mach. Intell. 43 (2) (2019) 652–662.
[18] S. Liu, D. Huang, Receptive field block net for accurate and fast object detection [C], Proce. European Conference on Comp. Vision (ECCV) (2018) 385–400.
[19] H. Zhang, K. Zu, J. Lu, et al., EPSANet: an efficient pyramid squeeze attention block on convolutional neural network[C], Proce. Asian Conference on Comp. Vision (2022) 1161–1177.
[20] Wang J, Xu C, Yang W, et al. (2021) A normalized Gaussian Wasserstein distance for tiny object detection[J]. arxiv preprint arxiv:2110.13389.
[21] Tong Z, Chen Y, Xu Z, et al. (2023) Wise-IoU: bounding box regression loss with dynamic focusing mechanism[J]. arxiv preprint arxiv:2301.10051.
[22] P. Wang, P. Chen, Y. Yuan, et al., Understanding convolution for semantic segmentation[C], 2018 IEEE Winter Conference on Applications of Comp. Vision (WACV) Ieee (2018) 1451–1460.
[23] K. Han, Y. Wang, Q. Tian, et al., Ghostnet: More features from cheap operations[C], Proce. IEEE/CVF Conference on Comp. Vision and Pattern Recognition (2020) 1580–1589.
[24] Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design [C], Proce. IEEE/CVF Conference on Comp. Vision and Pattern Recognition (2021) 13713–13722.
[25] Li C, Li L, Geng Y, et al. (2023) Yolov6 v3. 0: A full-scale reloading[J]. arxiv preprint arxiv:2301.05586.
[26] Y.F. Zhang, W. Ren, Z. Zhang, et al., Focal and efficient IOU loss for accurate bounding box regression[J], Neurocomputing 506 (2022) 146–157.
[27] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
[28] Zhao Y, Lv W, Xu S, et al. Detrs beat yolos on real-time object detection[J]. arxiv preprint arxiv:2304.08069, 2023.
[29] Yu Z, Huang H, Chen W, , et al. Yolo-facev2: A scale and occlusion aware face detector[J]. arXiv preprint arXiv:2208.02019, 2022.
[30] K. Song, Y. Yan, A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects[J], Appl. Surf. Sci. 285 (2013) 858–864.
[31] Liu M, Chen Y,.et al. LF-YOLO: A lighter and faster yolo for weld defect detection of X-ray image[J]. IEEE Sensors Journal, 2023, 23(7): 7430-7439.
[32] M. Everingham, , L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. The PASCAL Visual Object Classes Challenge (VOC2007), 2007, [online] Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[33] X. Zhu, S. Lyu, X. Wang, et al., TPH-Yolov5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios[c], Proce. IEEE/CVF Int. Conference on Computer Vision. (2021:) 2778–2788.
[34] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks[c], Proce. IEEE Conference on Comp. Vision and Pattern Recognition. (2018:) 7132–7141.
[35] https://universe.roboflow.com/iiti/2class_crack.
[36] C. Zhao, X. Shu, X. Yan, et al., RDD-YOLO: a modified YOLO for detection of steel surface defects[J], Measurement 214 (2023) 112776.
[37] X. Cheng, J. Yu, RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection[J], IEEE Trans. Instrum. Meas. 70 (2020) 1–11.
[38] Yuan P, Lin S, Cui C, et al. (2020) HS-ResNet: Hierarchical-split block on convolutional neural network[J]. ar**v preprint ar**v:2010.07621.
[39] W. Jiang, M. Liu, Y. Peng, et al., HDCB-Net: a neural network with the hybrid dilated convolution for pixel-level crack detection on concrete bridges[J], IEEE Trans. Ind. Inf. 17 (8) (2020) 5485–5494.
[40] H. Fang, et al., Automatic zipper tape defect detection using two-stage multi-scale convolutional networks, Neurocomputing 422 (2021) 34–50.
[41] Y. Gao, et al., A real-time defect detection method for digital signal processing of industrial inspection applications, IEEE Trans. Ind. Inf. 17 (5) (2020) 3450–3459.
[42] C. Szegedy, W. Liu, Y. Jia, et al., Going Deeper with Convolutions[c], Proce. IEEE Conference on Comp. Vision and Pattern Recognition. (2015:) 1–9.
[43] L.C. Chen, G. Papandreou, I. Kokkinos, et al., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J], IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[44] Z. Li, X. Wei, M. Hassaballah, et al., A deep learning model for steel surface defect detection[J], Complex & Intelligent Systems 10 (1) (2024) 885–897.
[45] Z. Li, X. Wei, M. Hassaballah, et al., A one-stage deep learning model for industrial defect detection[J], Adv. Theory and Simulations 6 (7) (2023) 2200853.
[46] L. Wei, J. Cai, K. Wen, et al., Local–global lightweight ViT model for mini/micro-LED-chip defect recognition[J], Eng. Appl. Artif. Intel. 123 (2023) 106247.
[47] Li Z, Wei X, Jiang X. (2023) SSDD-Net: A Lightweight and Efficient Deep Learning Model for Steel Surface Defect Detection[C]//Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Singapore: Springer Nature Singapore: 237-248.
[48] L. Zhang, Z. Fu, H. Guo, et al., Multiscale local and global feature fusion for the detection of steel surface defects[J], Electronics 12 (14) (2023) 3090.
[49] S.L. Zhao, G. Li, M.L. Zhou, et al., ICA-Net: industrial defect detection network based on convolutional attention guidance and aggregation of multiscale features [J], Eng. Appl. Artif. Intel. 126 (2023) 107134.
[50] G. Li, S. Zhao, M. Li, et al., IDP-Net: industrial defect perception network based on cross-layer semantic information guidance and context concentration enhancement [J], Eng. Appl. Artif. Intel. 130 (2024) 107677.
[51] https://github.com/shouxieai/infer.
[52] https://github.com/leafqycc/rknn-cpp-Multithreading.