Sensors and Materials, Vol. 37, No. 6 (2025) 2257–2277
MYU Tokyo

S & M 4054

One Improved Small-object Detection You-only-look-once Network for Strip-steel Surfaces

Sijie Qiu,[1] Chi-Hsin Yang,[1*] Long Wu,[1] Hao Gao,[1] and Wenqi Song[2]

[1]School of Mechanical and Electric Engineering, Sanming University, Sanming, Fujian Province 365004, China
[2]School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350000, China

(Received November 11, 2024; accepted May 23, 2025)

Keywords: strip-steel surface defect detection, small-object samples, multiscale PA-Net, Coord-DH module

To address the challenges posed by limited sample sizes and varying defect sizes on strip-steel surfaces in industrial applications, in this paper, we introduce a small-object detection you-only-look-once (YOLO) network (SODY-Net) specifically designed for such surfaces by machine learning technology. Initially, we build upon the YOLOv5s framework and develop a multiscale path aggregation network that incorporates an attention mechanism to improve the model's capability to predict across multiple scales. Next, we present an adaptive coordinate-decoupled head module for resolving the conflict between the classification and regression tasks. Finally, we propose a bounding box regression loss function that integrates the Wasserstein distance to enhance detection accuracy for small defects. Experimental results indicate that our SODY-Net surpasses other small-object detection frameworks when evaluated on a few-shot dataset of strip-steel surface defects, making it particularly suitable for defect detection tasks in industrial settings.

1. Introduction

The challenges of detecting small-object samples, particularly in identifying small objects in aerial imagery, have been thoroughly explored in various studies.[1–3] Such studies include the identification of defects in solenoid connectors used in electrical systems[4] and surface imperfections on magnetic rotors[5] among others. The difficulties related to small-object sample detection arise from a limited number of samples, insufficient diversity in sample locations, and a lack of distinct features among the samples. These factors make it a challenging area in the research on object detection network models.

Hot-rolled steel, an essential industrial material, has widespread applications in manufacturing production, the aerospace industry, and various other fields. The hot rolling process leads to surface defects in steel, such as cracks, roll marks, and oxidized iron layers, which negatively affect wear resistance, corrosion resistance, and fatigue strength owing to

*Corresponding author: e-mail: 20190207@fjsmu.edu.cn
https://doi.org/10.18494/SAM5464

ISSN 0914-4935

manufacturing processes and equipment limitations.[6,7] Currently, many companies still rely on manual inspection methods to detect these defects, which are often costly and inefficient.[8]

As deep learning algorithms have advanced, the method for defect detection utilizing convolutional neural networks has increasingly replaced manual inspections combined with memory-based systems and traditional machine vision techniques.[7] To improve the detection network's capability to represent sample characteristics and enhance feature reuse, dense convolutional blocks were introduced in Ref. 9 for better feature extraction, leading to the development of a strip-steel surface defect detection model based on you-only-look-once (YOLO) using the YOLOv3 framework. Additionally, Yu *et al.* have proposed[10] a lightweight model for detecting defects on tile surfaces to address the inadequate identification capabilities of existing models for small surface defects, which significantly reduced false positives and missed detections of minor target defects in earlier models.

In previous studies, the superiority of models based on deep learning for defect detection tasks is highlighted. However, the impressive effectiveness of these models often relies on having a large amount of labeled training data. In industrial applications, defects in manufactured products are rare, leading to insufficient data that can result in poor model generalization and an increased risk of overfitting.

To tackle the problem of limited sample availability, in Ref. 11, a customized small-sample learning model specifically designed for target detection was introduced. This model incorporates a meta-feature learner within the YOLOv2 architecture to enable end-to-end training suited for small-sample scenarios. Wang *et al.* contended in Ref. 12 that merely including meta-learners into pre-existing models can lead to decreased memory efficiency when the support set's category count rises. As a remedy, they propose a two-stage detection framework for generalized small-object detection fine-tuning, which enhances accuracy for the reclassification dataset while preserving the dataset-based model's performance. Nevertheless, these general algorithms face challenges when dealing with variations in shape and size typical in industrial applications. To address this issue, an advanced small-object detection model that employs multirelation aggregation along with an adaptive learning approach was presented in Ref. 13. This model functions within a dual-branch meta-learning training structure and leverages hidden relationships between queries and their associated supporting images.

In Ref. 14, Chen *et al.* integrated a defect highlighting module within a two-stage detection framework to optimize the exploitation of defect-free samples, thereby enhancing the characterization of defect regions and achieving superior detection results. In Ref. 15, a training framework for small-sample models based on Faster R-CNN was presented to strengthen the robustness of detection capabilities. These approaches for defect identification in industrial contexts demonstrate that small-sample learning techniques are indeed applicable in such scenarios. Nevertheless, they all fall within the purview of two-stage detection frameworks. Despite the attainment of satisfactory levels of accuracy by these strategies, their detection rates in previous studies[11–15] frequently fail to meet the requisites of actual industrial automation processes.

The identification of defects encounters substantial impediments because of multiple factors affecting strip-steel surfaces, such as the meager contrast between the defects and the

background, the extensive size spectrum of the targets, and the profusion of small targets. Motivated by the findings of prior analyses, to overcome these aforementioned hindrances, in this study, we incorporate the small-sample learning technology. A specifically tailored small-object detection YOLO network (SODY-Net) for strip-steel surfaces has been established on the basis of the YOLOv5s single-stage detection architecture.

As a result, the significant contributions and innovative aspects of our study are clearly evident in multiple ways.

(1) We propose a bounding box regression loss function known as Wasserstein distance-intersection over union (WD-IoU) for improving detection accuracy for small target defects.
(2) Additionally, we have designed a multiscale path aggregation network (PA-Net) that incorporates an attention mechanism, serving as the model's neck to enhance its capabilities for multiscale predictions and feature extraction.
(3) Moreover, we introduce the adaptive coordinate-decoupled head (Coord-DH) module for target prediction, which seeks to effectively address the conflict between the classification and localization tasks when dealing with limited sample sizes.

This paper is structured into the ensuing sections. In Sect. 2, we present a succinct delineation of the original YOLOv5s model architecture and expound on the functionalities of its submodules. In Sect. 3, we introduce the SODY-Net architecture and proffer three approaches for improving the original YOLOv5s model, involving the characteristics of the WD-IoU loss function, the implementation of a multiscale PA-Net, and an adaptive Coord-DH module. In Sect. 4, we elaborate upon the training and efficacy verification of the proposed SODY-Net employing the NEU-DET dataset. Finally, a comprehensive summary is presented.

2. Architecture of YOLOv5s Model

The YOLOv5s model architecture, as illustrated in Fig. 1, is structured into four primary modules, that is, the Input, Backbone, Neck, and Head. Each module within YOLOv5s is further delineated in Table 1.

The comprehensive definitions and operational guidelines of these modules are available in Ref. 16. CSPDarknet53 is employed in the Backbone for feature extraction,[17] with the cross-stage partial (CSP) module effectively reducing network parameters and computational load while enhancing feature extraction efficiency. Additionally, the spatial pyramid pooling fast (SPPF) module[18] consolidates feature maps of different scales into a single scale for efficient feature fusion. The Neck module utilizes the PA-Net[19] to merge feature maps from different levels using top-down and bottom-up approaches to obtain more extensive feature information. Finally, the Head module comprises three detection layers that predict targets of various scales ($80 \times 80 \times 256$, $40 \times 40 \times 512$, and $20 \times 20 \times 1024$), culminating in generating detection results through loss function calculation to predict bounding box position, size, and category for each target on the feature map.

The loss function of the YOLOv5s model comprises three primary elements, that is, the positional regression's loss function L_{IoU}, the cross-entropy loss function for classification, L_{cls}, and the bounding box regression's loss function L_{reg}. L_{IoU} (intersection over union) is defined in

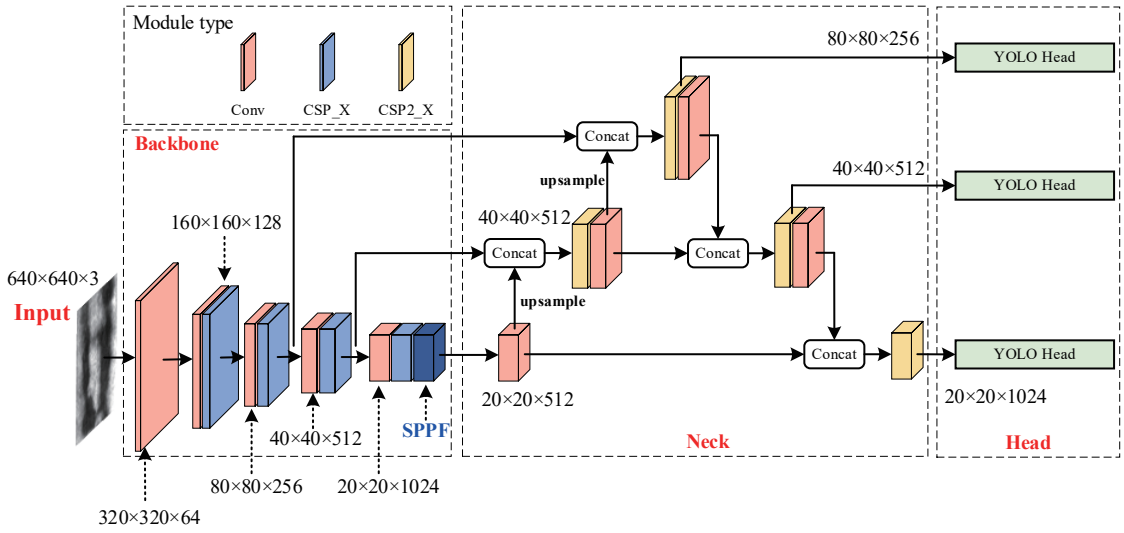

Fig. 1. (Color online) Architecture of YOLOv5s model.

Table 1
Modules within YOLOv5s model.

	Module	Detail structure of module
(1)	ConvBNSiLU	Conv → BN → SiLU
(2)	BottleNeck1	ConvBNSiLU → ConvBNSiLU → ⊕
(3)	BottleNeck2	ConvBNSiLU → ConvBNSiLU
(4)	CSP_X	ConvBNSiLU → BottleNeck1 × X → Concat → ConvBNSiLU; ConvBNSiLU → Concat
(5)	CSP2_X	ConvBNSiLU → BottleNeck2 × X → Concat → ConvBNSiLU; ConvBNSiLU → Concat
(6)	SPPF	ConvBNSiLU → Concat → ConvBNSiLU; ConvBNSiLU → MaxPool2d → MaxPool2d → MaxPool2d → Concat
(7)	YOLO Head	Conv → Conv → Anchor×C ; Cls loss; Anchor×4 ; Reg loss; Anchor×1 ; IoU loss

Eq. (1) as the ratio of the area of overlap between the predicted bounding box A_{pred} and the ground truth bounding box A_{gt} to their combined area.

$$L_{IoU} = \frac{A_{pred} \cap A_{gt}}{A_{pred} \cup A_{gt}} \tag{1}$$

Both L_{cls} and L_{reg} utilize the binary cross-entropy loss.[20] Additionally, L_{IoU} integrates the CIoU loss function.[21] The specific definition of the L_{CIoU} function is as follows.

$$L_{CIoU} = 1 - L_{IoU} + \frac{\rho^2(A_{pred}, A_{gt})}{D^2} + \alpha\upsilon \tag{2}$$

$$\alpha = \frac{\upsilon}{1 - L_{IoU} + \upsilon} \tag{3}$$

$$\upsilon = \frac{4}{\pi}\left(\arctan\frac{w_{gt}}{h_{gt}} - \arctan\frac{w_{pred}}{h_{pred}}\right)^2 \tag{4}$$

The symbol $\rho(A_{pred}, A_{gt})$ represents the Euclidean distance between the centroids of A_{pred} and A_{gt}, whereas D denotes the diagonal length of the minimum enclosing rectangle that encompasses both boxes. The variables w_{gt} and h_{gt} denote the width and height of the ground truth frame, and w_{pred} and h_{pred} represent those of the predicted frame, respectively. Additionally, α is a positive trade-off parameter used in optimization algorithms, whereas υ measures the aspect ratio consistency between w_{gt}/h_{gt} and w_{pred}/h_{pred}.

3. SODY-Net

Because of the imperfections present on strip-steel surfaces, several characteristics emerge. (1) There is a notably low contrast between the defects and their background. (2) The size range of these defects varies significantly. (3) Identifying these defects poses a challenging problem in terms of detecting small targets. Consequently, in academic applications, detecting defects on strip-steel surfaces represents a significant area of research.

To address the challenges mentioned earlier, we employ small-object learning methods based on the YOLOv5s one-stage detection framework in this study. A specialized network for defect detection with a limited sample size, named SODY-Net, has been carefully developed for inspecting strip-steel surfaces. The configuration of SODY-Net presented in this study is illustrated in Fig. 2. There are two key differences between SODY-Net and YOLOv5s models. (1) The Neck module of SODY-Net incorporates a multiscale PA-Net that features an attention mechanism. (2) The Head module of SODY-Net utilizes an adaptive decoupling detection framework known as the adaptive Coord-DH module, which is specifically designed for target prediction.

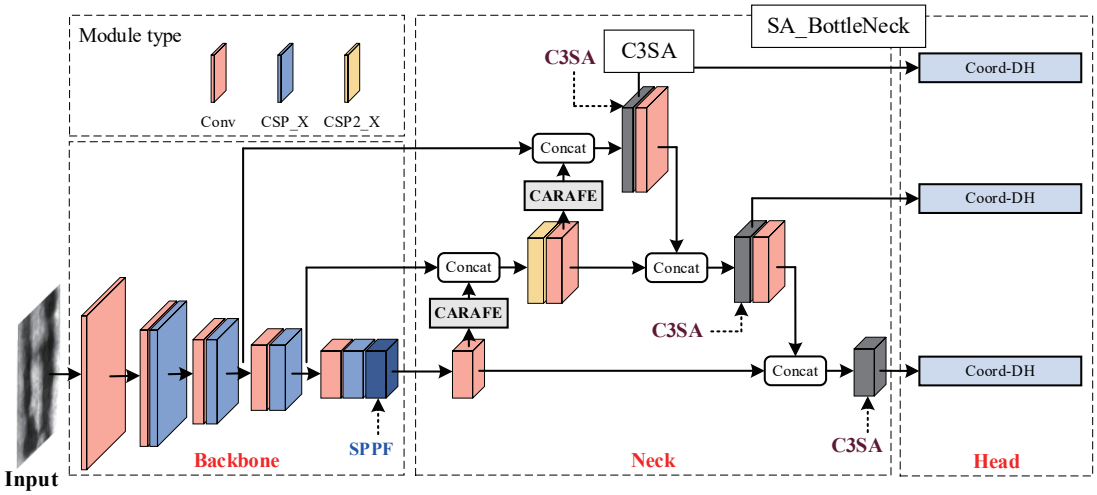

Fig. 2. (Color online) Architecture of SODY-Net.

SODY-Net operates as follows. First, the input image is processed by the Backbone module to generate feature maps at three different scales: 20 × 20, 40 × 40, and 80 × 80. These feature maps are then integrated into the Neck section for feature fusion. This architecture employs the content-aware re-assembly of feature (CARAFE) module to extract significant features from the Backbone module through upsampling, which helps retain detailed information. The shallow features are subsequently concatenated with these deeper features to achieve the multiscale integration of features. The resulting feature map is utilized by the C3SA module for global feature extraction, allowing for richer information capture. Finally, in the Head module, the detector predicts targets of various sizes (large, medium, and small) and calculates detection results using a loss function.

3.1 WD-IoU loss function

The NEU-DET dataset,[22,23] commonly utilized for algorithm model training and validation, is a publicly available compilation specifically curated for the analysis of surface defects on steel materials. By examining the steel datasets, we computed the ratio of the surface defect area to the total image area for each picture, as detailed in Table 2. Results show that roughly 44.8% of all defects have an area ratio ≤10%, whereas about 24.2% of all defects have an area ratio ≤5%. This underscores the significance of small surface defects and small to medium-sized imperfections, such as inclusions, patches, roll scraps, and scratches in steel data.

Utilizing the surface defect data of diverse steel materials in the NEU-DET dataset as detailed in Table 2, which includes defect targets of varying dimensions, we here introduce a calculation method for the WD-IoU loss function L_{WDIoU} of the predicted bounding box. This approach is specifically tailored to concurrently accommodate targets of diverse sizes, thereby enhancing the regression accuracy of SODY-Net and optimizing the efficiency of detecting and identifying defect targets.

Table 2
Ratio of defect area in NEU-DET dataset.

Ratio of area (Defect area / Total image area)	Image count
Ratio of area ≤ 5%	1014
5% < Ratio of area ≤ 10%	863
Ratio of area > 10%	2312

The WD-IoU loss function L_{WDIoU} is computed as

$$L_{WDIoU} = 1 - WD_{IoU}, \tag{5}$$

$$WD_{IoU} = \lambda_1 \cdot NWD(N_{gt}, N_{pred}) + \lambda_2 \cdot Dis_{IoU} - Asp. \tag{6}$$

In Eq. (6), the WD_{IoU} metric is constrained within the range of $[0, 1]$, where $\lambda_1, \lambda_2 \in [0, 1]$ denote the scale coefficients, which are fine-tuned using the NEU-DET dataset. The specific process for selecting λ_1 and λ_2 is further expounded upon in Sect. 4. The computation of L_{WDIoU} consists of three primary components, whose detailed calculation procedures are outlined below.

(1) Normalized Wasserstein distance $NWD(N_{gt}, N_{pred})$

The rectangular box, defined by the vector $R = [cx, cy, h, w]^T$ representing its center coordinates (cx, cy), height h, and width w in the image map, can be accurately characterized as a bivariate Gaussian distribution. This model effectively captures the varying pixel intensities within the box, with the central pixel (cx, cy) exhibiting maximum weights with h and w, gradually decreasing towards the boundary along both axes. Mathematically, this representation aligns with a two-dimensional Gaussian distribution, $N(\mu, \Sigma)$:

$$\mu = \begin{bmatrix} c_x & 0 \\ 0 & c_y \end{bmatrix}, \ \Sigma = \begin{bmatrix} h^2/4 & 0 \\ 0 & w^2/4 \end{bmatrix}, \tag{7}$$

where μ and Σ represent the covariance matrix of the mean vector and the Gaussian distribution, respectively.

The Wasserstein distance is a metric derived from optimal transport theory.[24,25] The Gaussian–Wasserstein distance between the two-dimensional Gaussian distributions $m_1 = N(\mu_1, \Sigma_1)$ and $m_2 = N(\mu_2, \Sigma_2)$ can be formally defined as

$$W_2^2(m_1, m_2) = \left\| \mu_1 - \mu_2 \right\|_2^2 + \left\| \Sigma_1^{1/2} - \Sigma_1^{1/2} \right\|_2^2. \tag{8}$$

Thereby, the vector R representations of the ground truth and predicted bounding boxes are denoted as $R_{gt} = [cx_{gt}, cy_{gt}, h_{gt}/2, w_{gt}/2]^T$ and $R_{pred} = [cx_{pred}, cy_{pred}, h_{pred}/2, w_{pred}/2]^T$, respectively. In this context, the coordinates (cx_{gt}, cy_{gt}) are used to represent the central point of the ground truth bounding box, whereas h_{gt} and w_{gt} describe its size. Similarly, the coordinates (cx_{pred}, cy_{pred}) indicate the central point coordinates of the predicted bounding box, with h_{pred} and w_{pred} representing its dimensions. After calculating Eq. (7), $N_{gt}(\mu_{gt}, \Sigma_{gt})$ is obtained for

modeling the multivariate Gaussian distribution of the ground truth bounding box as well as $N_{pred}(\mu_{pred}, \Sigma_{pred})$ for the predicted bounding box. Subsequently, using Eq. (8), we compute $W_2^2(N_{\text{gt}}, N_{\text{pred}}) \in [0, \infty)$, which represents their dissimilarity obtained by Gaussian–Wasserstein distance measurements. *NWD* in Eq. (8) is further adjusted by applying an exponential function.

$$NWD(N_{gt}, N_{pred}) = \exp\left(-\sqrt{W_2^2(N_{gt}, N_{pred})/\gamma^2}\right) \in [0, 1] \tag{9}$$

Here, γ is a constant determined by the average absolute magnitude of the target in the dataset.

(2) Ratio of height difference to width difference, *Asp*

As stated in Ref. 26, an effective bounding box regression loss should take into consideration the IoU values of the predicted and ground truth bounding boxes, along with their centroid distance and aspect ratio. The NWD introduced in Eq. (9) addresses both the IoU and centroid distance of two boundary boxes simultaneously. Therefore, to incorporate the aspect ratio, in this study, we adopt the formula for calculating the height-to-width difference ratio of ground truth and predicted bounding boxes from the EIoU loss function.[21]

$$Asp = \frac{\rho^2(w_{pred}, w_{gt})}{C_w^2} + \frac{\rho^2(h_{pred}, h_{gt})}{C_h^2} \tag{10}$$

The variables w_{pred} and h_{pred} represent the dimensions of the predicted bounding box, whereas w_{gt} and h_{gt} denote the measurements of the ground truth bounding box. The terms $\rho^2(w_{pred}, w_{gt})$ and $\rho^2(h_{pred}, h_{gt})$ indicate the squared discrepancies between the widths and heights of the predicted and ground truth bounding boxes, respectively. C_w and C_h refer to the dimensions of the minimum enclosing rectangle that encompasses both sets of bounding boxes.

(3) Reassessment of positional regression for large predicted bounding boxes in object detection

The concentration of the foreground within the central region of the bounding box for small targets, along with the predominant distribution of background along the periphery of the bounding box, necessitates utilizing *NWD* to represent the overlap area between the ground truth and predicted bounding boxes. This approach also takes into account their distance from the centroid, facilitating a gradual reduction in distribution weight from the center to the edge, thereby aligning with characteristics specific to small targets. Note that large targets may not necessarily adhere to this distribution pattern. Given that large targets constitute a significant proportion of our dataset as indicated by Table 3, further consideration is required for elements such as actual overlapping areas between two bounding boxes and distances between their respective centroids. The relevant calculation formula is provided below.

$$Dis_{IoU} = \frac{A_{pred} \cap A_{gt}}{A_{pred} \cup A_{gt}} - \frac{\rho^2(A_{pred}, A_{gt})}{D^2} \tag{11}$$

The first term in Eq. (11) represents the ratio of IoU, whereas $\rho(A_{pred}, A_{gt})$ denotes the Euclidean distance between the centroid of the predicted bounding box A_{pred} and that of the ground truth

Table 3
Proposed novel modules.

	Module	Detailed structure of module
(1)	SA_BottleNeck	ConvBNSiLU → ConvBNSiLU → ShuffleAttention
(2)	C3SA	ConvBNSiLU → SA_BottleNeck × **X** → Concat → ConvBNSiLU; ConvBNSiLU → Concat

bounding box A_{gt}. In this context, D indicates the diagonal length of the minimum enclosing rectangle formed by A_{pred} and A_{gt}. This measure is commonly employed in object detection applications to assess spatial alignment between predicted bounding boxes and their corresponding ground truth annotations.

3.2 Multiscale PA-Net incorporating attention module

Generally, the effective feature extraction of strip-steel surface defects is crucial for successful defect identification in input feature images because of the low contrast with the background and varying defect sizes. In this study, to meet practical application needs, we enhance the network's feature extraction capability by integrating the CARAFE module[27] and the dual shuffle attention (SA) module[28] based on the multiscale PA-Net in the Backbone module. This allows for a more accurate distinction of defects in input images and improved recognition capability for different scales of strip-steel surface defect targets. Additionally, the integration of these modules provides the network with enhanced context awareness and the capability to better handle varying defect sizes, ultimately leading to more accurate and efficient defect identification.

3.2.1 CARAFE module

The nearest-neighbor and bilinear interpolations are used in the YOLOv5s upsampling process. A reduction in computing complexity, a simple algorithm, and quick processing are the advantages of this approach. Instead of taking into account the values of other nearby pixels, the nearest-neighbor interpolation method just takes into account the gray-level binary values of the pixels closest to the sample point. As a result, there are some shortcomings of the first YOLOv5s upsampling technique, including a lack of sufficiently rich semantic information recorded and a significant amount of computation resulting from a larger number of parameters. Loss of image data and a reduction in object detection precision are possible outcomes of these deficiencies.[29,30]

Although the nearest-neighbor interpolation only considers the immediate pixel region surrounding the sample point, the CARAFE module aggregates continuous information from neighboring regions to enable upsampling across a wider receiving range.[27] More appropriate sampling for strip-steel surface defect characteristics and finer feature maps with low detail loss

are achieved by the CARAFE module through the use of adaptive and optimized recombination cores at different places.

The CARAFE module is a content-aware kernel reassembly operator that consists of two steps. Firstly, it predicts a reassembly kernel for each target location by analyzing its content, then it reassembles features using the anticipated kernels. The following formulas delineates the operating processes:

$$W_{l'} = \psi(N(X_l, k_{encoder})), \tag{12}$$

$$X'_{l'} = \phi(N(X_l, k_{up}), W_{l'}), \tag{13}$$

where the kernel prediction module is represented by ψ and the content-aware reassembly module by ϕ. The original feature map is denoted by X in this instance and the new feature map produced by upsampling is denoted by X'. The pre-upsampled target location is represented by the variable l, whereas the post-upsampled target position is indicated by l'. In essence, $N(X_l, k)$ represents the domain of X_l by symbolizing the $k \times k$ subregion of X centered at position l. The term W_l refers to the kernel prediction module ψ, which uses contextual data taken from X_l to predict a suitable kernel for each location l'.

Moreover, $k_{encoder}$ is a convolution layer with a certain kernel size in Eqs. (12) and (13), whereas k_{up} denotes the re-assembly kernel size. In this situation, an empirical formula such as $k_{encoder} = k_{up} - 2$ has been found to provide the best possible trade-off between efficiency and performance (see Ref. 27 for further information about the CARAFE module operation).

3.2.2 Fusion of the dual-path SA module

The attention module primarily operates on diverse channels and spatial positions within the feature graph to attenuate less salient features. The integration of the attention module can facilitate the algorithm in efficiently discerning points of focus, particularly in scenarios with limited sample data. To minimize computational power consumption, we introduce a lightweight and efficient SA module,[28] which integrates a dual-path attention mechanism to fully leverage the relationship between spatial and channel attention. This enables a quicker convergence of loss values for the corresponding Coord-DH module in the algorithm model of the Head module.

As demonstrated in Table 3(1), we propose the fusion of the SA module with the BottleNeck2 module, as shown in Table 1(3), to form a novel SA_BottleNeck module. This integrated module is then incorporated into the CSP2 module to create the C3SA module, as shown in Table 3(2), which is subsequently positioned before the input of the three Coord-DH modules in the Head module. Following the feature fusion operation, the feature map undergoes global feature extraction through passage via the C3SA module prior to transmission to the Head's detector, as shown in Fig. 2. Within the C3SA module, the integration of the SA_BottleNeck module into an attention module primarily aims at reducing computational load and obtaining more comprehensive gradient flow information than previously achieved. This not only ensures rich

scale feature information for various defects within a feature graph but also mitigates interference from redundant information on detectors within the Head module.

Figure 3 illustrates the process of the SA module, which initially divides the feature graph F into g groups along its channel dimension, that is, $F = [F_1, \ldots, F_g]$, $F_k \in R^{(c/g)\times h\times w}$. The subfeature F_k is then split into F_{k1} and F_{k2} along the channel dimension and introduced into the channel attention branch and spatial attention branch, respectively. In Eq. (14), within the channel attention branch, F_{k1} undergoes global average pooling to obtain its channel statistic s. Subsequently, as per Eq. (15), s undergoes weight convolution with the sigmoid function to yield the channel attention weight matrix, which is then multiplied by F_{k1} to produce F'_{k1}. Here, W_1 and b_1 respectively represent the weight matrix and vector in channel attention.

$$s = F_{gp}(F_{k1}) = \frac{1}{h\times w}\sum_{i=1}^{h}\sum_{j=1}^{w} F_{k1}(i,j) \tag{14}$$

$$F'_{k1} = \sigma(F_c(s))\cdot F_{k1} = \sigma(W_1 s + b_1)\cdot F_{k1} \tag{15}$$

In the spatial attention branch, the acquisition of spatial dimension information involves the utilization of group norm (GN)[31] for F_{k2}. Subsequently, the feature F'_{k2} is obtained through a process that weights different spatial positions using Eq. (16).

$$F'_{k2} = \sigma(W_2\cdot GN(F_{k2}) + b_2)\cdot F_{k2} \tag{16}$$

Upon the completion of consolidating the results from both branches through designated communication channels, the features are aggregated into F', followed by cross-group information exchange facilitated through channel shuffling.

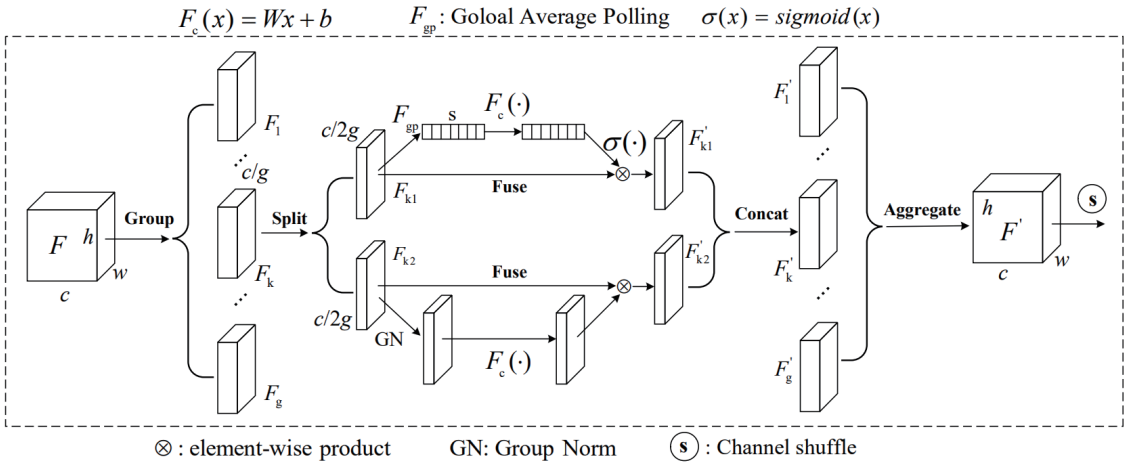

Fig. 3. SA module.

3.3 Adaptive Coord-DH module

The feature map obtained after feature fusion is abundant in intricate and detailed location features, significantly enhancing the information utilized for object detection. In the original YOLOv5s algorithm model, the Head module is responsible for two pivotal tasks, target classification and location, both of which rely on the extracted features.

The classification task necessitates that the Head module accurately discerns the category of the target object while disregarding its spatial coordinates. This requirement ensures that categorization is solely based on intrinsic characteristics rather than the position within the image. Conversely, the location task demands a comprehensive understanding of the target's spatial information, including precise positioning and size. This aspect holds particular significance in real-world scenarios where positional data plays a critical role.

The inherent conflict between these tasks arises from their divergent objectives, that is, accuracy in classification versus precision in location The former focuses on correct identification, whereas the latter aims to pinpoint exact locations. This conflict becomes more pronounced in scenarios with limited training samples as the learning complex relationship between features and tasks becomes challenging, potentially leading to suboptimal performance in both classification and location. Further insights into this issue can be found in Ref. 32.

Hence, in this study, the adaptive Coord-DH module is proposed for target prediction. Differing from the coupled detection structure of the original YOLOv5s algorithm model's Head module, which integrates various information on a feature graph, the adaptive Coord-DH module presented in this study primarily separates classification and location tasks into two distinct branches. Furthermore, an adaptive learning mechanism is designed to enable these branches to independently address target classification and location requirements, thereby alleviating task conflicts and enhancing detection accuracy.

The structure of the Adaptive Coord-DH module is depicted in Fig. 4. The input feature map undergoes dimensionality reduction through the 1 × 1 × Conv convolution module and is subsequently partitioned into two branches. The classification branch maintains translation invariance via the Coordinate-convolution (Coord-Conv) module, followed by a 1 × 1 × Conv convolution for classification operations. In contrast, the regression branch captures translation variability through the Coord-Conv module before being decomposed into two parallel branches for 1 × 1 × Conv convolution, one dedicated to positioning and the other to confidence detection.

In the aforementioned statements, the Adaptive Coord-DH module can acquire diverse levels of translation invariance and variability, contingent upon distinct tasks, under the effect of the Coord-Conv module. The structure of the Coord-Conv module is delineated in Fig. 5. Although traditional convolution operations are well suited for classification tasks because of their translation invariance, they lack positional information, resulting in suboptimal localization effects. In this study, the adopted Coord-Conv module incorporates x- and y-coordinate channels from the original input into feature graph F, facilitating spatial information perception during the convolution process. Consequently, when the coordinate channel does not acquire any information, it behaves akin to traditional convolution with translation invariance. However, when it acquires specific information, it demonstrates discernible translation variability that

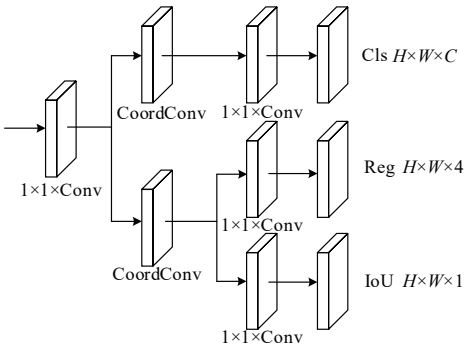

Fig. 4. Structure of Adaptive Coord-DH module.

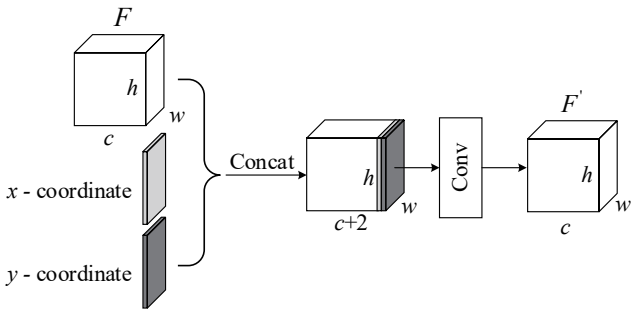

Fig. 5. Structure of Coord-Conv module.

enables different branches within a decoupled detection structure to adaptively learn various tasks.

4. Experimental Design and Analysis of Results

To verify the efficiency of the proposed method, we evaluated the model's detection performance using the NEU-DET public dataset provided by Northeastern University.[23] The experimental setup utilized the Ubuntu 20.04 LTS operating system with 16 GB of memory, an AMD Ryzen 5 5600X CPU, and an NVIDIA GeForce RTX3060 GPU with 12 GB of VRAM. PyTorch version 1.10.1 and CUDA version 11.2 were employed for software implementation, and Python version 3.7 served as the primary programming language.

The training strategy remained consistent across all experiments, utilizing a batch size of 16 and a fixed input image size of 640 × 640 pixels. The training process involved a total of 120 epochs, commencing with an initial learning rate set at 0.01 and a momentum value at 0.937 using stochastic gradient descent (SGD) as the optimizer with a regression coefficient set to optimize the model at a value of 0.0005.

4.1 Construction of experimental dataset

The NEU-DET dataset covers six distinct defect categories, namely, cracks (Cr), inclusions (In), patches (Pa), pittings (Ps), rolling scraps (Rs), and scratches (Sc). Examples of these six

defect categories are shown in Fig. 6. Each category consists of a total of 300 high-resolution images measuring 200 × 200 pixels.

In the field of steel defect databases, where sample sizes are limited, there is currently no established representative data repository. Therefore, in this study, we utilize the existing NEU-DET dataset for reclassification and develop a training database specifically designed for network models aimed at detecting small-object sample defects on steel surfaces.

In Ref. 15, a research methodology for the detection of surface defects in a few samples of strip-steel surface defects was introduced in terms of the NEU-DET dataset. In this work, the NEU-DET dataset is divided into two categories, that is, the base class, which includes In, Rs, and Sc, and the novel class, which comprises Cr, Pa, and Ps. The base class dataset D_{Base} consists of D^{T}_{Base}, which has 720 training set images, and D^{V}_{Base}, which has 180 validation set images. D^{T}_{Base} and D^{V}_{Base} are utilized for model pre-training and validation. Similarly, the novel class dataset D_{Novel} includes D^{T}_{Novel}, which has 720 training set images, and D^{V}_{Novel}, which has 180 validation set images. During the network fine-tuning stage, the k count of images from each category is randomly selected from D^{T}_{Novel} for training and validation on D^{V}_{Novel}. Figure 7 illustrates the construction process for the dataset for small-object sample defects of strip-steel surfaces.

In contrast to the approach outlined in Ref. 15, in this study, we employ a fine-tuning process on the pre-trained model using a composite dataset comprising D^{T}_{Base} and D^{T}_{Novel}, achieved by the random sampling of k instances from each category. Validation is subsequently conducted on the combined $D^{T}_{Base} + D^{T}_{Novel}$ dataset at various k values of 5, 10, and 30.

Figure 8 illustrates the training flow during the fine-tuning stage of SODY-Net. To enhance the model's generalization capability, we apply the following strategies:

(1) During each training iteration in both the network base training stage and the fine-tuning stage, an 80% probability of random scale change to the input image is made. This is done to improve the model's scale invariance.
(2) Additionally, a 50% probability of rotation is used to enhance rotational invariance.

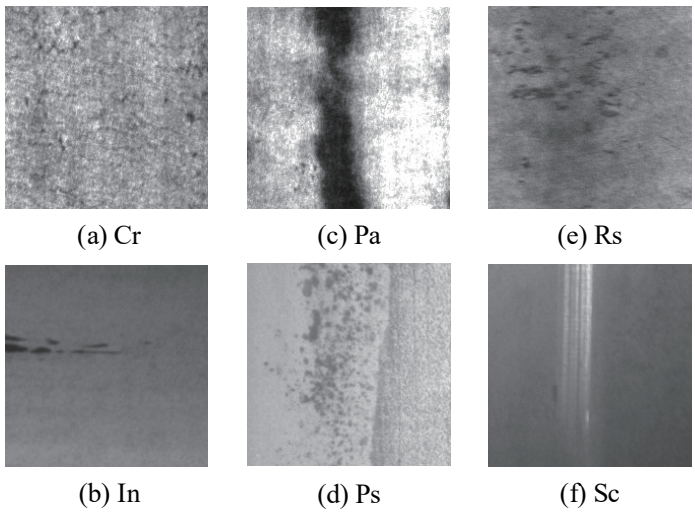

Fig. 6. Examples of defects in the NEU-DET dataset. (a) Cr, (b) In, (c) Pa, (d) Ps, (e) Rs, and (f) Sc.

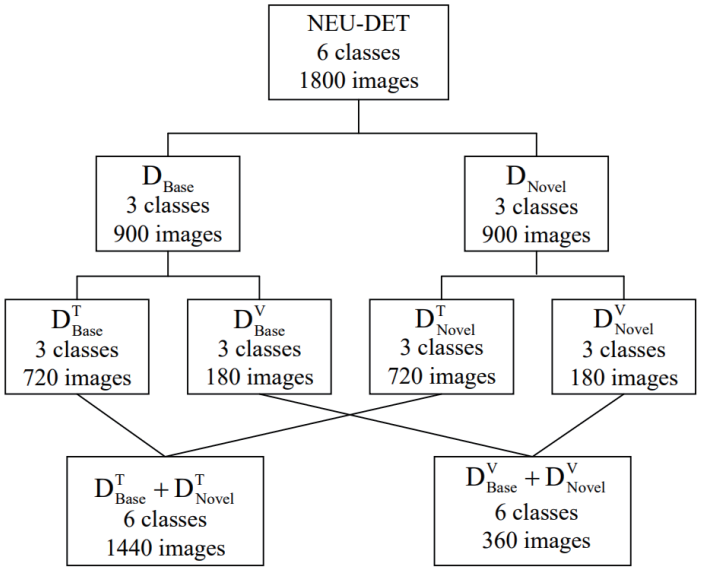

Fig. 7. Construction of dataset for small-object sample defects of strip-steel surfaces.

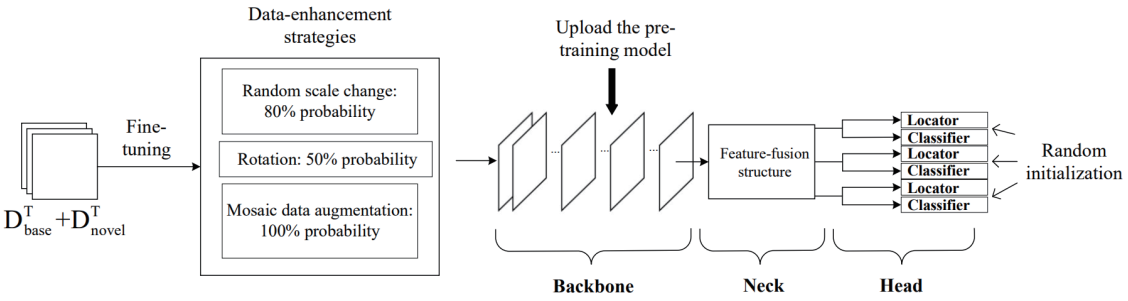

Fig. 8. Fine-tuning and training steps of SODY-Net.

(3) A 100% probability of mosaic data augmentation[33] is employed to enrich the training dataset.

Moreover, according to Fig. 8, the training procedure for the network model proposed in this study is outlined as follows.

(1) In the initial phase of training, the images from the training set D^{T}_{Base} are employed to train the model, whereas those from the verification set D^{V}_{Base} are used for validation to acquire a pretrained model.

(2) The number of categories in the classifier of the randomly initialized detector aligns with that during the fine-tuning stage.

(3) The Backbone module of the pretrained model remains unchanged and undergoes fine-tuning on the combined dataset $D^{T}_{Base} + D^{T}_{Novel}$, with validation also performed on the same combined dataset.

4.2 Ablation experiments

To evaluate the impact of the refined modules (C3SA and Coord-DH) introduced in the study on network model performance, ablation experiments were conducted under various shot conditions using the small-object sample defect dataset derived from the NEU-DET dataset. The results of these experiments are presented in Table 4. In this table, the non-SA&CARAFE representation model employs the original PA-Net configuration of YOLOv5s for feature fusion, whereas all other parameters align with SODY-Net. The term 'non-DH' indicates that the standard detection header from YOLOv5s is used instead of the Coord-DH architecture, and all other configurations adhere to SODY-Net standards. Furthermore, the non-WD representation model applies the original bounding box regression loss function for the YOLOv5s model, ensuring consistency with SODY-Net in every other aspect.

A thorough examination of the results presented in Table 4 reveals several important observations when compared with SODY-Net.

(1) The absence of C3SA and CARAFE modules within the feature fusion framework significantly reduces detection accuracy for the non-SA&CARAFE network model by 4.11, 3.74, and 3.54% under 5-, 10-, and 30-shot conditions, respectively. This highlights the essential contribution these two modules make towards enhancing the network's feature extraction capabilities and their effectiveness in learning critical features.

(2) Furthermore, not including Coord-DH in the detection setup leads to a notable decline in detection accuracy for the non-DH network model by 3.16, 4.88, and 5.12% across all three shot scenarios, indicating that Coord-DH effectively addresses conflicts between the classification and localization tasks, thus improving overall detection performance.

(3) Additionally, failing to apply the WD-IoU loss function for optimizing prediction frame regression results in a drop in detection accuracy for the non-WD network model by 0.73, 0.59, and 0.51% under the three shot conditions. This suggests that utilizing WD-IoU aids in achieving a more accurate identification of small targets.

4.3 Experimental analysis of WD-IoU loss functions

In this section, we clarify the WD-IoU loss function utilized in this study and evaluate the effectiveness of the small-object sample defect dataset introduced in Sect. 4.1, focusing on the experimental analysis of the predicated bounding box loss function WD_{IoU}. SODY-Net employs the WD-IoU loss functions as outlined in Eqs. (5) and (6). In this context, λ_1 and λ_2 represent the coefficients linked to the size distribution ratio among training samples. When small and

Table 4
Results of ablation experiments.

Modules	5-shot	10-shot	30-shot
non-SA&CARAFE	58.13	63.18	69.07
non-DH	59.08	62.04	67.49
non-WD	61.51	66.33	72.10
SODY-Net (ours)	**62.24**	**66.92**	**72.61**

medium-sized targets predominate within these samples, increasing the value of λ_1/λ_2 can amplify the effect of the NWD index. Conversely, decreasing the value of λ_1/λ_2 can enhance the significance of the intersection ratio Dis_{IoU}. Since fine-tuning requires training with various random samples, it implies that multiple parameter combinations (λ_1, λ_2) must be established for comparison. To expedite the efficacious determination of the appropriate (λ_1, λ_2), in this study, we impose the condition that $\lambda_1 + \lambda_2 = 1$ to streamline the exploration and deliberation.

In Table 5, the impacts of the CIoU loss function applied to the YOLOv5s model and the effects of the NWD and WD-IoU loss functions applied to SODY-Net are compared in detail using the small-object sample defect dataset under the 5-shot test. In this case, n is the defect area divided by the whole picture area, N is the total number of defects, and N_S is the number of defect objects whose area percentage is less than n, and $\lambda_1/\lambda_2 = N_S/(N - N_S)$. The associated loss function at this point is labeled WD_{IoU_n}. Utilizing the NEU-DET dataset shown in Table 3, we further categorize WD_{IoU_n} into three different scenarios: n = 0.05, 0.1, and 0.2.

Table 5 illustrates that the NWD loss function significantly outperforms the CIoU loss function used in the YOLOv5s model, resulting in an increase of 0.81% in AP_{50}. Furthermore, it demonstrates better detection performance for small and medium-sized objects, with AP_S and AP_M improving by 3.45 and 0.59%, respectively. On the other hand, the CIoU loss function shows improved detection capabilities for larger targets, achieving an AP_L value that surpasses that of the NWD loss function by 1.01%.

The results presented in Table 5 indicate that the $WD_{IoU\_0.1}$ loss function can provide optimal detection performance. Specifically, the λ_1/λ_2 ratio was calculated from the count of targets whose area ratio is below 10% of the total number of other targets, resulting in an AP_{50} score of 55.87%. This marks a 1.76% improvement over the CIoU loss function. Additionally, both small- and large-target detection precision peaked, with AP_S and AP_L values recorded at 15.38 and 36.21%, respectively. Importantly, the gap between AP_S and AP_L when using the $WD_{IoU\_0.1}$ loss function was reduced to only 20.83%. These findings illustrate that the WD_{IoU} loss function effectively refines target boundary box regression while enhancing model detection capabilities across various object sizes.

4.4 Compared experiments

To assess the detection capability of the SODY-Net presented in this study, the YOLOv5s model is employed as the fundamental framework. Moreover, with the small-object sample defect dataset retrieved from the NEU-DET dataset, SODY-Net is compared with the prevalent two-stage fine-tuning approach (TFA)[12] and the Faster R-CNN incremental few-shot defect detection (IFDD)[15] model for the identification of small-object sample defects in industrial application scenarios. The experimental findings are illustrated in Table 6.

As shown in Table 6, SODY-Net demonstrates superior detection performance under various conditions, that is, 5-, 10-, and 30-shot. Several important observations can be made.

(1) Specifically, under the training approach outlined in Sect. 3.1, the mean average precision (mAP) of SODY-Net improved by 7.36, 5.27, and 7.34% in comparison with that of the YOLOv5s model for each of the three shot conditions.

Table 5
Detection results of different loss functions under 5-shot condition.

Loss Function	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CIoU	27.23	54.11	24.53	10.47	19.74	36.17
NWD	27.69	54.92	24.88	13.92	20.33	35.16
$WD_{IoU\_0.05}$	27.84	55.16	25.17	13.15	20.16	36.01
$WD_{IoU\_0.1}$	**28.33**	**55.87**	**25.96**	**15.38**	**20.87**	**36.21**
$WD_{IoU\_0.2}$	28.06	55.48	25.62	15.02	19.98	35.45

Table 6
Results of performance evaluation on the small-object sample defect dataset.

Models	mAP = AP_{50}		
	5-shot	10-shot	30-shot
YOLOv5s	56.47	60.50	63.69
TFA[12]	36.55	38.89	43.22
Faster R-CNN (IFDD)[15]	52.64	58.36	64.29
SODY-Net (ours)	**63.83**	**65.77**	**71.03**

(2) In the 5-, 10-, and 30-shot situations, the mAP of SODY-Net increased by 27.28, 26.88, and 27.28%, respectively, in comparison with that of the TFA model.
(3) In comparison with that of the Faster R-CNN (IFDD) model, the mAP of SODY-Net rose by 11.19, 7.41, and 6.74% under the 5-, 10-, and 30-shot conditions, respectively.

The results of the experiments strongly support the superior detection performance of SODY-Net in industrial settings. Furthermore, the mAP of the TFA model is significantly lower than those of the other models, indicating that the commonly used small-object sample defect identification method, TFA, is poorly suited for defect detection tasks that involve various scales and shapes, making it unsuitable for direct application in industrial contexts.

The mAP metric for SODY-Net shows a significant increase from 62.24 to 72.61% under 5- to 30-shot conditions. This trend is also observed in both the YOLOv5s and TFA models. This indicates that the training methodology proposed in Sect. 3.1 aligns well with small-object sample defect detection models, thereby improving the effectiveness of existing defect identification models (YOLOv5s and TFA) when dealing with limited sample defects.

Moreover, the results of tests conducted on the YOLOv5s model, TFA model, and SODY-Net using the small-object sample defect dataset introduced in Sect. 4.1 are presented in Fig. 9. The results illustrated in Fig. 9 clearly indicate that SODY-Net demonstrates superior performance, whereas the detection capabilities of the other models are comparatively less effective.

(1) The YOLOv5s model identified a sole defect in Cr samples yet miscategorized multiple defects in Ps samples as a solitary one. In contrast, SODY-Net precisely recognized the multiple defects as discrete defects, manifesting its preeminent capacity for extracting global information.
(2) Furthermore, the YOLOv5s model was unable to detect small defects in Pa samples, but SODY-Net was able to detect these slight irregularities with high sensitivity, highlighting its ability to detect anomalies on a small scale.
(3) Additionally, the TFA model failed to identify several defects during the detection of Pa samples and did not detect any anomalies in Cr and Ps samples, which is an inadmissible

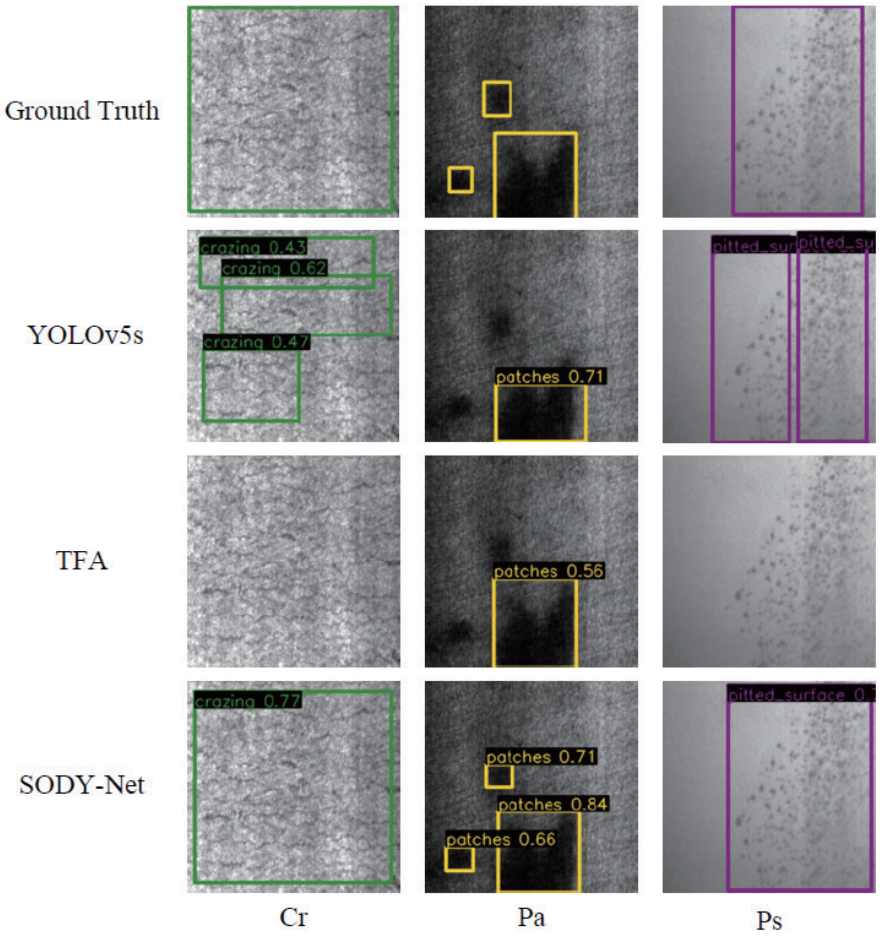

Fig. 9. (Color online) Detection results of three small-object sample defect detection models.

situation in industrial applications. Consequently, SODY-Net demonstrates a more reliable detection performance and is eminently suitable for the task of detecting defects on strip-steel surfaces in industrial scenarios.

5. Conclusions

To tackle the issue of insufficient defect samples in industrial contexts, in this study, we introduce SODY-Net, a model designed for detecting defects on strip-steel surfaces. A multiscale PA-Net module called C3SA was developed to enhance the model's focus on defect characteristics and improve its capability to predict defects at various scales. Additionally, an adaptive decoupling detection framework named Coord-DH separates the model's detection functions into a classifier and a locator that can flexibly handle their respective tasks, thus reducing conflicts between the classification and location tasks. We also propose a bounding box regression loss function, WD-IoU, which incorporates the Wasserstein distance to improve precision in detecting small target defects. Using a small-object sample defect dataset derived from the NEU-DET dataset, we conducted comparative experiments and ablation studies. The

results demonstrate both the complexity of SODY-Net and the effectiveness of each enhanced module within it.

Acknowledgments

This work was carried out as part of the Joint Innovation Project of Industry University Research of Fujian Province (Grant No. 2023H6036), Major Science and Technology Projects of Fujian Province (Grant Nos. 2023T5001 and 2022HZ026025), the Program for Innovative Research Team in Science and Technology in Fujian Province University, the Production and Research Collaboration with Innovative in Key Scientific and Technological Project of Sanming City (Grant No. 2022-G-17), and the Operational Funding of the Advanced Talents for Scientific Research (Grant no. 19YG04) of Sanming University. The authors also acknowledge the support from the School of Mechanical and Electric Engineering, Sanming University.

References

1 H. Shi, W. Yang, D. Chen, and M. Wang: PLoS One **19** (2024) e0298698. https://doi.org/10.1371/journal.pone.0298698
2 W. Cai, X. Wang, X, Jiang, Z. Yang, X. Di, and W. Gao: Electronics **12** (2023) 4133. https://doi.org/10.3390/electronics12194133
3 X. Chen, T. Chen, H. Meng, Z. Zhang, D. Wang, J. Sun, and J. Wang: Front. Plant Sci. **15** (2024) 1360419. https://doi.org/10.3389/fpls.2024.1360419
4 M. Chen, Y. Liu, X. Wei, Z. Zhang, O. Gaidai, and H. Sui: PLoS One **19** (2024) e0297059. https://doi.org/10.1371/journal.pone.0297059
5 B. Yu, Q. Li, W. Jiao, S. Zhang, and Y. Zhu: Mathematics **12** (2024) 957. https://doi.org/10.3390/math12070957
6 L. Guo and B. Luo: J. Mech. Electr. Eng. **32** (2015) 352. https://doi.org/10.3969/j.issn.1001-4551.2015.03.011
7 B. Tang, L. Chen, W. Sun, and Z. K. Lin: IET Image Proc. **17** (2023) 303. https://doi.org/10.1049/ipr2.12647
8 L. Liu, C. Wang, S. Zhao, and H. Li: J. Electron. Meas. Instrum. **10** (2018) 47. https://doi.org/10.13382/j.jemi.2018.10.007
9 X. Kou, S. Liu, K. Cheng, and Y. Qian: Measurement **182** (2021) 109454. https://doi.org/10.1016/j.measurement.2021.109454
10 S. Yu, M. Zhang, and H. Yang: Comput. Syst. Appl. **32** (2023) 151. https://doi.org/10.15888/j.cnki.csa.009185
11 B. Kang, Z. Liu, and X. Wang: Proc 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV, 2019) 8419. https://doi.org/10.1109/ICCV.2019.00851
12 X. Wang, T. E. Huang, and T. Darrell: Proc 37th Int. Conf. Mach. Learn. (2020) 9919. https://doi.org/10.48550/arXiv.2003.06957
13 Y. Deng and Y. Song: ISIJ Int. **63** (2023) 1727. https://doi.org/10.2355/isijinternational.ISIJINT-2023-118
14 Z. Chen, Z. Liu, G. Li, and T. Peng: Comput. Eng. Appl. **58** (2022) 108. https://doi.org/10.3778/j.issn.1002-8331.2111-0470
15 H. Wang, Z. Li, and H. Wang: IEEE Trans. Instrum. Meas. **71** (2022) 5003912. https://doi.org/10.1109/TIM.2021.3128208
16 H. K. Jooshin, M. Nangir, and H. Seyedarabi: IET Image Proc. **18** (2024) 1985. https://doi.org/10.1049/ipr2.13077
17 M. Mahasin and I. A. Dewi: Int. J. Eng. Sci. Inf. Technol. **2** (2022) 64. https://doi.org/10.52088/ijesty.v1i4.291
18 H. Tang, S. Liang, D. Yao, and Y. Qiao: Opt. Express **31** (2023) 2628. https://doi.org/10.1364/OE.480816
19 S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia: Proc. 2018 IEEE Conf. Comput. Vision Pattern Recognition (2018) 8759. https://doi.org/10.48550/arXiv.1803.01534
20 J. Z. Pan, C. H. Yang, L. Wu, and W. H. Tang: Sens. Mater. **35** (2023) 4653. https://doi.org/10.18494/SAM4589
21 Y. Sun, J. Wang, and H. Wang: IEEE Access **12** (2024) 37363. https://doi.org/10.1109/ACCESS.2024.3359433
22 Y. He, K. Song, Q. Meng, and Y. Yan: IEEE Trans. Instrum. Meas. **69** (2019) 1493. https://doi.org/10.1109/TIM.2019.2915404
23 K. Song and Y. Yan: Appl. Surface Sci. **285** (2013) 858. https://doi.org/10.1016/j.apsusc.2013.09.002

24 J. Wang, C. Xu, W. Yang, and L. Yu: Comput. Vision Pattern Recognit. (2021) arXiv:2110.13389. https://doi.org/10.48550/arXiv.2110.13389
25 K. Sun, Z. Wu, M. Wang, J. Shang, Z. Liu, D. Zhao, and X. Luo: J. Mar. Sci. Eng. **12** (2024) 333. https://doi.org/10.3390/jmse12020333
26 K. Su, L. Cao, B. Zhao, N. Li, D. Wu, and X. Han. Neural Comput. Appl. **36** (2024) 3049. https://doi.org/10.1007/s00521-023-09133-4
27 J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin: Proc. 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV, 2019) 3007. https://doi.org/10.1109/ICCV.2019.00310
28 Q. L. Zhang and Y. B. Yang: Proc. 2021 IEEE Int. Conf. Acoustic. Speech Signal Process (2021) 2235. https://doi.org/10.48550/arXiv.2102.00240
29 J. Lin, M. Jiang, Y. Pang, H. Wang, Z. Chen, C. Yan, Q. Liu, and Y. Wang: Proc. 2022 Int. Conf. Sustainable Comput. Data Commun. Syst. (ICSCDS, 2022) 1455. https://doi.org/10.1002/cpe.6320
30 W. Yang, X. Ma, and H. An: Agronomy **13** (2023) 1613. https://doi.org/10.3390/agronomy13061613
31 Y. Wu and K. He: Int. J. Comput. Vis. **128** (2020) 742. https://doi.org/10.1007/s11263-019-01198-w
32 L. Qiao, Y. Zhao, and Z. Li: Proc. 2021 IEEE/CVF Int. Conf. Computer Vision (ICCV, 2021) 8681. https://doi.org/10.1109/ICCV48922.2021.00856
33 Y. Li, R. Cheng, C. Zhang, M. Chen, H. Liang, and Z. Wang: Math. Biosci. Eng. **20** (2024) 7193. https://doi.org/10.3934/mbe.2023311