Computing and Informatics, Vol. 45, 2026, 1100–1128, doi: 10.31577/cai_2026_5_1100

# DAE-YOLO: AN IMPROVED INDUSTRIAL SURFACE DEFECT DETECTION ALGORITHM BASED ON YOLOV11

Guangqiao Xie, Yan Zhou*, Chengshang Wu

*Department of Information Center*
*Mianyang Cigarette Factory of Sichuan China Tobacco Industry Company Limited*
*Mianyang, 621052, China*
*e-mail:* {xiegq, zhouyan}@sctobacco.com, 220232636@dlmu.edu.com

**Abstract.** The accurate identification of surface defects in industrial products is critical for ensuring consistent high-quality standards in modern manufacturing. Nevertheless, the prevalence of small-scale defects and pronounced cross-scale variability on industrial surfaces poses substantial challenges to achieving precise detection. To address these issues, this study introduces a high-precision neural network architecture, termed DAE-YOLO, which is developed based on the YOLOv11 framework and specifically tailored for industrial surface defect detection. To effectively mitigate the impact of cross-scale feature discrepancies, an efficient multi-scale attention mechanism (EMA) is embedded within the backbone network. This leads to the design of a novel feature extraction module, D-EMA (Deformable and Efficient Multi-scale Attention), which enhances the model's ability to capture both global contextual dependencies and local structural details. This module enhances the feature extraction capabilities of the convolutional neural network by fusing global context and local features, thereby enabling precise modeling of spatial information and pixel-level relationships. The design of D-EMA carefully considers the fusion of cross-spatial information, allowing the model to effectively process targets at varying scales while preserving accurate location information, significantly improving performance in defect detection tasks. Further, to address the challenge of detecting small targets, the latest deformable convolutional module was incorporated into the Neck component. A new bidirectional weighted feature pyramid network (BDFPN, bidirectional deformable feature pyramid network) was then designed, combining multi-scale features to enhance the representation of small defect features. Experiments conducted on two publicly available datasets (NEU-DET and PCB defect) demonstrated the effectiveness of the proposed network architecture.

* Corresponding author

The results indicate that the proposed method outperforms YOLOv11 by 3.88 % in detection accuracy on the NEU-DET dataset, and by 2.8 % on the PCB defect dataset.

**Keywords:** Surface defect detection, attention mechanism, deformable convolution, feature pyramid network, computer vision

**Mathematics Subject Classification 2010:** 68U10

## 1 INTRODUCTION

Defect detection is a critical component of modern industrial manufacturing, exerting a direct influence on both product quality and production efficiency. Timely and accurate identification of defects enables substantial reductions in rework and scrap rates, thereby enhancing the consistency and reliability of final products. With the increasing demand for stricter quality control standards in industrial production, surface defect detection has garnered growing attention. This rising importance has, in turn, catalyzed continuous advancements in defect detection technologies, particularly those leveraging intelligent vision and deep learning methodologies [1]. However, surface defect detection of industrial products presents numerous challenges. Not only do the variety of product types and materials, along with the different forms of defects and detection requirements, make it difficult for universal detection methods to meet all demands, but some defects are also small and hard to detect, such as micro-cracks and bubbles. Furthermore, environmental factors (e.g., lighting, temperature, and humidity) can affect detection accuracy, particularly in complex production environments [2, 3].

Traditional image processing algorithms in defect detection mainly rely on analyzing product surface images to identify and locate defects. Image preprocessing techniques, such as grayscale conversion, denoising, and edge enhancement, are first applied to improve image quality. Then, methods like gray-level co-occurrence matrix (GLCM) [4] and local binary patterns (LBP) [5] are used to analyze image textures and extract features. The processed image features are subsequently compared with preset standard models for defect detection and classification. Common methods include threshold segmentation [6], region growing and watershed algorithms [7], and machine learning techniques [8]. Threshold segmentation simplifies the problem by dividing the image into "defective" and "non-defective" regions based on a set threshold. Region growing and watershed algorithms are employed to segment defect regions within the image, enabling accurate defect localization. Machine learning approaches, such as support vector machines (SVM) [9], decision trees [10], and random forests [11], are used for automated classification and identification of defects.

In recent years, with the development of deep learning and improvements in hardware computing power, deep learning-based defect detection technologies have become increasingly prevalent. Convolutional Neural Networks (CNNs) [12] can automatically extract image features through convolutional layers without the need for manual feature design, demonstrating excellent performance in handling complex backgrounds and various defect types. Deep Convolutional Neural Networks (DCNNs), an extension of traditional CNNs, increase the network's depth, enabling it to learn more complex features. In defect detection, DCNNs [13] improve the ability to identify subtle defects, particularly those with complex shapes that shallow networks struggle to capture. However, both networks still have limitations when dealing with complex backgrounds and small defects. The Single Shot Multibox Detector (SSD) [14], an end-to-end object detection model, predicts targets at different scales, achieving both localization and classification in a single step. SSD relies on predefined anchor boxes for target detection, but these anchors may not be suitable for all defect types, particularly those with significant morphological variations or irregular shapes, leading to degraded performance.

Recent algorithm models have made significant advancements in defect detection. The two-stage algorithm, Faster R-CNN [15], introduces a Region Proposal Network (RPN) to automatically generate candidate regions and combines with Fast R-CNN for object classification and localization. Compared to R-CNN [16] and Fast R-CNN, Faster R-CNN shows notable improvements in both detection accuracy and speed; however, its performance is suboptimal for small objects or complex background scenes. YOLOv3 [17] enhances detection accuracy for targets of various sizes through multi-scale detection and an improved network structure (Darknet-53). YOLOv5 [18] further optimizes model lightweighting and inference speed, making it suitable for deployment on edge devices, and introduces improvements in data augmentation. However, YOLOv3 and YOLOv5 may suffer from missed detections in complex backgrounds or for small targets, especially when the defects have irregular shapes, leading to inaccurate localization. Additionally, the performance in processing high-resolution images is limited, making it challenging to detect fine and diverse defect types.

YOLOv11, as a widely adopted algorithm in object detection, achieves a balance between efficiency and accuracy. However, its general-purpose design limits its robustness and adaptability in domain-specific tasks such as industrial defect detection. This study critically examines the shortcomings of YOLOv11 when applied to surface defect detection and introduces targeted enhancements to improve its capability in identifying small, irregularly shaped defects with greater precision. The principal contributions of this work are outlined as follows:

1. Design of the D-EMA Module to Replace the C3K2 Module: To improve model performance in defect detection, a new module, D-EMA, is proposed, integrating deformable convolution (DCNv4, deformable convolutional network v4) and an efficient multi-scale attention mechanism (EMA). DCNv4 captures key information of small targets by flexibly adjusting sampling points, while EMA

explicitly models spatial and semantic dependencies, preserving the spatial location and fine-grained semantic information of small targets. The combined use of D-EMA enhances the model's ability to integrate multi-scale information, effectively replacing the traditional C3K2 module.

2. Introduction of the BDFPN Structure for Optimized Multi-Scale Feature Fusion: Drawing inspiration from the bi-directional feature pyramid network (BiFPN), a novel bi-directional deformable feature pyramid network (BDFPN) is designed. This structure integrates deformable convolution, residual connections, cross-scale connections, attention mechanisms, and weighted feature fusion methods, allowing for more efficient representation and processing of multi-scale features. BDFPN optimizes node topology, removes redundant nodes, and replaces standard convolutions with deformable convolutions, significantly reducing model parameters and computational overhead while maintaining effective feature extraction.
3. Introduction of Auxiliary Bounding Box Loss to Accelerate the Regression Process: To overcome the limitations of traditional Intersection over Union (IoU) loss in various detection tasks, this study introduces the use of auxiliary bounding boxes for loss calculation to accelerate the bounding box regression process. By incorporating a scaling factor to adjust the size of the auxiliary bounding boxes and distinguishing between different regression samples, the loss is computed using auxiliary bounding boxes of varying scales. This approach significantly enhances the regression speed and optimizes the model's convergence rate.

## 2 RELATED WORK

This section investigates the relevant technologies for surface defect detection based on image processing, including traditional detection methods and deep learning-based approaches, followed by a detailed discussion.

### 2.1 Traditional Detection Methods

Traditional surface defect detection techniques have been extensively studied and applied across various industrial scenarios. While these methods can deliver stable performance under specific conditions, they are often constrained by inherent limitations such as poor adaptability to complex backgrounds and limited generalization to diverse defect types. The following section reviews three representative conventional surface defect detection approaches and presents a comprehensive analysis of their respective strengths and limitations.

Statistical-based methods are among the most common techniques in traditional surface defect detection. These methods rely on surface images captured through imaging systems and utilize image processing algorithms for feature extraction and

analysis to identify defects. Key techniques include edge detection [19], thresholding, and morphological processing [20]. Although traditional statistical methods have been extensively studied, they fail to provide reliable and accurate results, particularly for defects with variations in lighting or slight deformations.

Template matching-based defect detection methods [21] identify defects by comparing the similarity between the image to be inspected and a predefined template. This approach is typically used for detecting defects with fixed shapes and structures. The core idea of template matching is to match one or more known defect patterns (i.e., templates) with the image to be inspected, determining the presence of defects by calculating the degree of similarity. A template is created based on specific application requirements, and then, using a sliding window technique, the template is compared to each position in the image to compute similarity. Common similarity measures include normalized cross-correlation (NCC) [22] and squared differences [23]. Classic surface defect detection models include the Markov Random Field (MRF) model [24], Weibull model [25], and active contour model [26]. These methods evaluate the degree of match between the template and the current window region to locate the defect. However, fixed templates perform poorly when dealing with image rotation, scale changes, or deformation, and variations in lighting can significantly reduce the similarity between the template and the actual image.

With the advancement of computer science and artificial intelligence technologies, machine learning-based defect detection methods have gradually been adopted. These methods rely on training datasets and learn defect features to automatically identify defects in new images. Classic machine learning techniques include support vector machines (SVM), decision trees, and k-nearest neighbors (KNN) [27].

### 2.2 Detection Methods Based on Deep Learning

Deep learning-based methods for surface defect detection in industrial products have been widely applied. Convolutional Neural Networks (CNNs) can effectively recognize complex image patterns and learn deep features of defects by automatically extracting local features from images. Their hierarchical structure makes them highly expressive and precise when handling defect detection tasks [28, 29, 30]. Hou et al. [31] established a deep convolutional neural network (DCNN)-based model to directly extract deep features from X-ray images. This method crops patches from X-ray images as the training dataset and addresses the class imbalance in the number of defect patches by applying three resampling methods on three balanced datasets. The results demonstrate that DCNNs significantly outperform traditional feature extraction methods in deep feature extraction.

To address the time-consuming and inefficient nature of traditional fabric defect detection, Wei et al. [32] employed an improved Faster R-CNN method based on the VGG architecture. To minimize the impact of input data on Faster R-CNN, the fabric defect dataset was expanded. Additionally, by utilizing fabric defect image features, the number of anchor points in Faster R-CNN was reduced. Experimental

results validated the improved Faster R-CNN model's capability and effectiveness in fabric defect detection.

In order to enhance defect detection rates and fabric product quality, Jing et al. proposed a more real-time fabric defect detection method based on an improved YOLOv3 model [17]. This approach involves two key steps: first, combining the fabric defect size with the k-means algorithm to perform dimensional clustering on the target frames in YOLOv3, determining the number and size of prior frames. Second, by integrating low-level features with high-level information, YOLO detection layers are added at different feature map sizes, which enhances the model's application in detecting defects in gray and plaid fabrics. The improved network model achieved error rates of less than 5 % in detecting plain and plaid fabric defects.

To address the low accuracy in current industrial defect detection, Li et al. [33] proposed a two-stage industrial defect detection framework based on Improved-YOLOv5 and Optimized-Inception-ResNetV2, which completes both localization and classification tasks using two specific models. To enhance the first-stage detection in effectively localizing small, unimportant defects on steel surfaces, the method improves YOLOv5 by refining the backbone network, feature fusion layer scales, and multi-scale detection layers. In the second stage, to better extract defect features and achieve accurate classification, a convolution block attention module (CBAM) is embedded into the Inception-ResNetV2 model. The model architecture and loss function are then optimized for precise performance.

However, research on capturing multi-scale contextual information in industrial surface defect detection remains insufficient. Compared to existing solutions, our approach can more effectively handle contextual information in images, enhancing the dynamism and expressiveness of spatial aggregation. The following section provides a detailed explanation of our approach and an analysis of the characteristics of the adopted technologies.

## 3 PROPOSED DAE-YOLO

### 3.1 D-EMA Module

Achieving high defect recognition accuracy while preserving spatial precision is vital for surface defect detection. A major challenge stems from the prevalence of small-scale defects in industrial products. These defects occupy limited pixel regions, leading to low spatial resolution and significant feature loss after multiple down-sampling operations in convolutional networks. Additionally, they are highly sensitive to scale variations, where even slight changes in viewpoint can further weaken or obscure their features.

To address these issues, a novel module, D-EMA, has been designed to replace the C3K2 module in the backbone. The specific structure of the D-EMA module is shown in Figure 1 (Figure 1 is drawn by the authors). The D-EMA module incorporates the Deformable Convolution (DCNv4) [34] and Efficient Multi-scale

Attention (EMA) [35]. The deformable convolution module adjusts sampling locations on low-resolution feature maps to efficiently capture key information from small targets. The efficient multi-scale attention mechanism (EMA) explicitly models spatial and semantic dependencies, retaining the spatial position and fine-grained semantic information of small targets during feature extraction, thus preventing feature degradation. Thanks to the combined effect of these two components, D-EMA demonstrates superior multi-scale information integration, effectively replacing the C3K2 module.

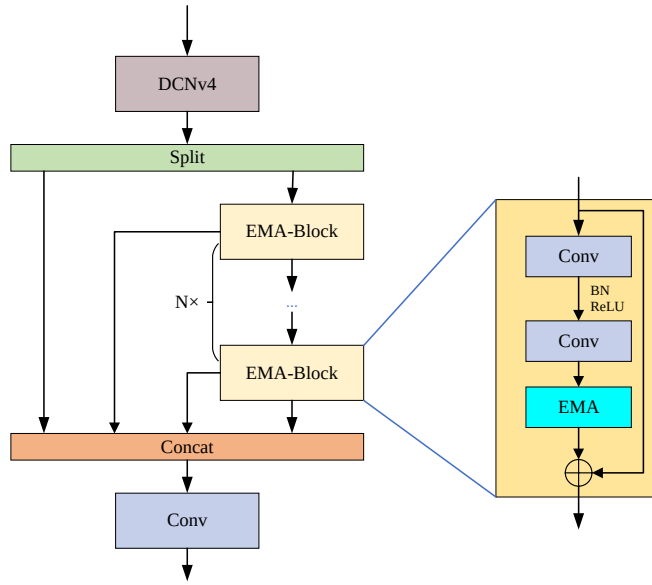



Figure 1. Overall structure of the D-EMA module

In the D-EMA module, DCNv4 is used to replace traditional convolution for two main reasons:

1. Stronger Spatial Perception: Unlike traditional convolution with fixed sampling grids, DCNv4 introduces learnable offsets that allow dynamic adjustment of sampling positions. This enables better adaptation to geometric variations such as rotation, scaling, and deformation, enhancing the model's ability to represent irregular surface defects.
2. Adaptability to Sparse and Non-Rigid Targets: For non-rigid or sparse targets, traditional convolution, with its fixed receptive field, struggles to capture global information. DCNv4's dynamic receptive field allows it to flexibly adjust sampling locations, better covering the target area and adapting to non-rigid structural changes.

Compared to its predecessor, DCNv3 [36], DCNv4 eliminates the Softmax normalization for spatial aggregation weights. This removes the restriction of keeping weight values within the range of 0 to 1, thereby enhancing dynamic properties and expressive capability. The unconstrained dynamic nature of the weights enables DCNv4 to model complex dependencies more effectively. Furthermore, by optimizing memory access and reducing redundant operations, DCNv4 achieves significant speed improvements, with its forward inference speed being more than three times faster than DCNv3. This improvement aligns better with the sparse computation principles inherent in deformable convolutions.

EMA is an improved version of Coordinate Attention (CA) [37], a mechanism that adaptively and dynamically adjusts the weights in feature maps based on the importance of each region. By introducing the EMA attention mechanism, key channel information is preserved while reducing computational costs. The specific implementation, shown in Figure 1, reorganizes certain channels into the batch dimension and divides the channel dimension into multiple sub-feature groups to ensure a uniform distribution of spatial semantic features within each group. This method not only maintains the correlation between channels but also significantly reduces computational overhead, allowing the module to focus more on key areas of the task. As a result, D-EMA demonstrates enhanced specificity and robustness in complex scenarios.

First, the features are grouped to learn different semantics. The input tensor $X \in \mathbf{R}^{C\times H\times W}$(where C is the number of channels, H is the height, and W is the width) is divided into G groups along the channel dimension.The groups are defined as $X = [X_0, X_i, \ldots, X_{G-1}], X_i \in \mathbf{R}^{C//G\times H\times W}$. Then, the EMA processes the grouped data through three parallel paths to capture spatial multi-scale details. Two of the paths use $1 \times 1$ convolutions, while one path employs a $3 \times 3$ convolution. In the two $1 \times 1$ branches, the channels are encoded along two spatial directions using two 1D global average pooling operations. After concatenating and applying a convolution, two S-shaped functions are used to adjust the outputs of the two $1 \times 1$ branches into a 2D bifurcation distribution. To implement different cross-channel interaction features between the two parallel paths in the $1 \times 1$ branches, the attention maps of the two paths are aggregated in each group through simple multiplication. In the $3 \times 3$ branch, a single $3 \times 3$ convolution kernel is stacked to capture multi-scale feature representations. Through this multi-branch structure, EMA not only processes cross-channel information but also retains precise spatial information in the channels, while encoding inter-channel information to adjust the importance of different channels.

Furthermore, EMA is inspired by PSA [38] and employs a cross-space information aggregation method across different spatial dimensions to achieve richer feature aggregation. To preserve the global spatial information from the $1 \times 1$ branches, a 2D global pooling method is used for encoding. Before the channel feature joint activation mechanism, the output from the smallest branch is directly transformed into the corresponding dimensional shape. The formula for the 2D global pooling

operation is:

$$z_c = \frac{1}{H \times W} \sum_{j}^{H} \sum_{i}^{W} x_c(i, j). \tag{1}$$

At the same time, the adjusted $3 \times 3$ branch features are incorporated into the joint activation mechanism. The results of these parallel processing stages are combined using matrix dot product operations to form a spatial attention map. This method generates two spatial attention maps that respectively preserve precise positional information and pixel-level pairwise relationships. Ultimately, by aggregating the attention weights and applying the Sigmoid function, effective capture of both pixel-level and global context is achieved, while maintaining the same feature map size as the input.

The attention factors of EMA are guided by the similarity between global and local features within each group, utilizing a cross-space information aggregation approach to embed precise positional information and long-range dependencies. Furthermore, EMA replaces the progressive limited receptive field operations with parallel $3 \times 3$ and $1 \times 1$ convolutions, maximizing the extraction of contextual information from intermediate feature maps. The multi-scale context fusion enables EMA to generate superior pixel-level attention, adapt to high-level feature maps, and express both global and local features more effectively, all while achieving high computational efficiency, making it well-suited for complex scene tasks.

### 3.2 The BDFPN

Feature Pyramid Networks (FPN) and Path Aggregation Networks (PAN) are essential architectures for multi-scale feature fusion. PAN is an improved version of FPN. Both aim to address the feature fusion challenge in multi-scale object detection; however, they differ significantly in terms of feature information propagation direction and fusion strategies.

The Feature Pyramid Network (FPN) constructs a pyramid by combining deep semantic features with shallow detailed features to improve detection accuracy. However, its unidirectional information flow limits the full exploitation of bottom-up detail information. To address this, the Path Aggregation Network (PAN) introduces a bidirectional propagation mechanism, adding a bottom-up path alongside the top-down pathway of FPN. Building on these ideas, BiFPN employs efficient bidirectional fusion, enabling more effective interaction between high-level semantics and low-level details, thereby further enhancing detection performance.

Assume that $L$ feature maps have been extracted from a backbone network: $\{P_1, P_2, \ldots, P_L\}$, where $P_i$ represents the feature map of the $i^{\text{th}}$ layer. Typically, as $i$ increases, the resolution of the feature map decreases, but the semantic information becomes richer. FPN uses a top-down path to gradually fuse high-level semantic information into low-level features. The fusion process can be represented as: $P_i' = \text{Upsample}\left(P_{i+1}'\right) + \text{Conv}\left(P_i\right)$ is the fused feature map of the $i^{\text{th}}$ layer, Upsample $(\bullet)$ refers to upsampling $P_{i+1}'$ to match the resolution of $P_i$, and $\text{Conv}\left(P_i\right)$ represents

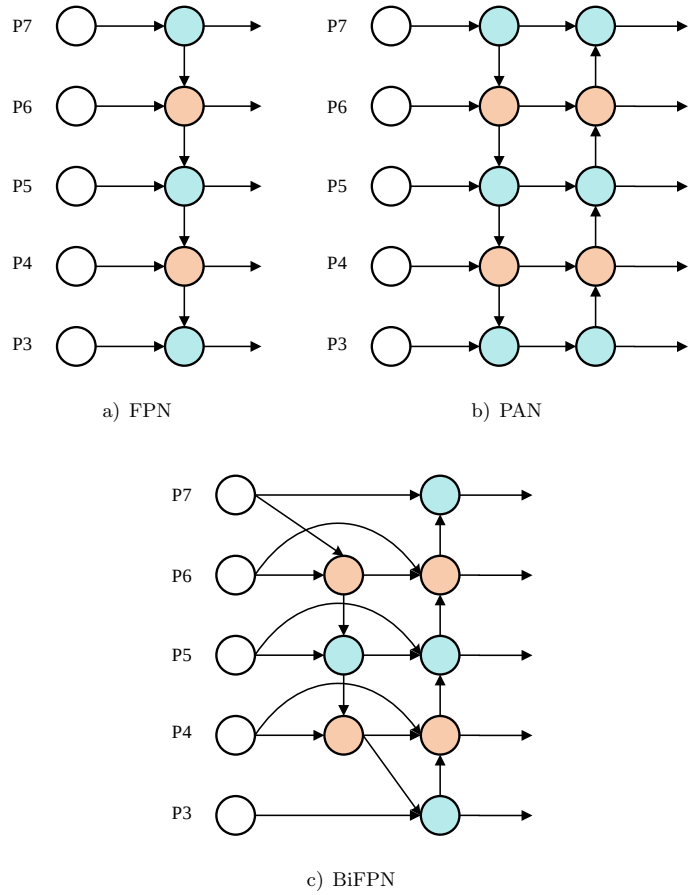



Figure 2. The structure of FPN, PANet, and BiFPN

a $1 \times 1$ convolution operation on $P_i$ to adjust the channels to match the dimensions. This approach gradually transmits global semantic information from higher layers to lower layers, enhancing the semantic representation of the lower-level features, as shown in Figure 2 a) (Figure 2 is drawn by the authors).

PAN adds an additional bottom-up path on top of FPN, used to gradually transfer low-level detail information into higher-level features. The formula is: $P_i'' = \text{Downsample}\left(P_{i-1}''\right) + \text{Conv}\left(P_i'\right)$, where $P_i''$ is the feature map of the $i^{\text{th}}$ layer after enhancement through the bottom-up path. Downsample $(\cdot)$ refers to a downsampling operation that reduces the resolution of $P_{i-1}''$ to match $P_i'$. With this step, higher-level features can regain low-level details, further enhancing their ability to perceive small targets, as shown in Figure 2 b).

BiFPN adds skip connections from the input layer to the output layer, enabling the network to learn to balance the proportion of low-level and high-level features during feature fusion. By stacking multiple BiFPN units to construct the final network, feature fusion becomes more comprehensive as the model deepens. Figure 2 a) shows the computation process of BiFPN. Inspired by this, this paper adopts a fast normalization fusion method for weighted fusion. The specific calculation process is shown in Equation (2):

$$F = \sum_i \frac{w_i}{\eta + \sum_j w_j} \cdot X_i. \tag{2}$$

In this context, $w_i$ and $w_j$ are learnable weights, and $\eta$ represents a fixed constant to prevent numerical instability. $X$ denotes each input feature, while $F$ represents the fused weighted feature. For example, in Figure 2 c), the two fused features at the fourth layer can be expressed as follows:

$$P_5^{td} = C\left(\frac{w_1 \cdot P_5^{\text{in}} + w_2 \cdot R\left(P_6^{\text{in}}\right)}{w_1 + w_2 + \eta}\right), \tag{3}$$

$$P_5^{\text{out}} = C\left(\frac{w_1' \cdot P_5^{\text{in}} + w_2' \cdot P_5^{td} + w_3' \cdot R\left(P_4^{\text{out}}\right)}{w_1' + w_2' + w_3' + \eta}\right). \tag{4}$$

Here, $P_l^{td}$ denotes the intermediate features of the $l^{\text{th}}$ layer, while $P_l^{\text{out}}$ represents the output features of the $l^{\text{th}}$ layer. The calculation process of $P_l^{td}$ is shown in Equation (3). The calculation process of $P_l^{\text{out}}$ is shown in Equation (4). The function $R$ typically denotes either an upsampling or downsampling operation, while $C$ represents a convolution operation used to process and enhance feature representation. These equations precisely describe how feature contributions are dynamically balanced across different layers during the feature fusion process, leading to more effective feature representation.

This paper adopts the concept of BiFPN and introduces a novel Bidirectional Deformable Feature Pyramid Network (BDFPN) structure. This structure efficiently represents and processes multi-scale features while integrating more comprehensive feature information. The proposed network incorporates deformable convolutions, residual connections, cross-scale connections, attention mechanisms, and weighted feature fusion methods, all of which play a crucial role in optimizing model performance.

The nodes of BDFPN are primarily constructed based on the D-EMA module, as shown in Figure 3 (Figure 3 is drawn by the authors). The brown arrows represent the downsampling of low-level features to high-level features through convolutional layers. The blue arrows indicate the upsampling of high-level features via deconvolution layers. The pink arrows represent skip connections. At each layer, features from different levels are fused and converted to the same size via weighted summation.

BDFPN significantly reduces the model's parameter count and computational overhead by extensively replacing standard convolutions with the latest deformable

convolution (DCNv4), while maintaining feature extraction effectiveness. Additionally, by removing redundant nodes and optimizing the network topology, the information flow becomes more efficient.

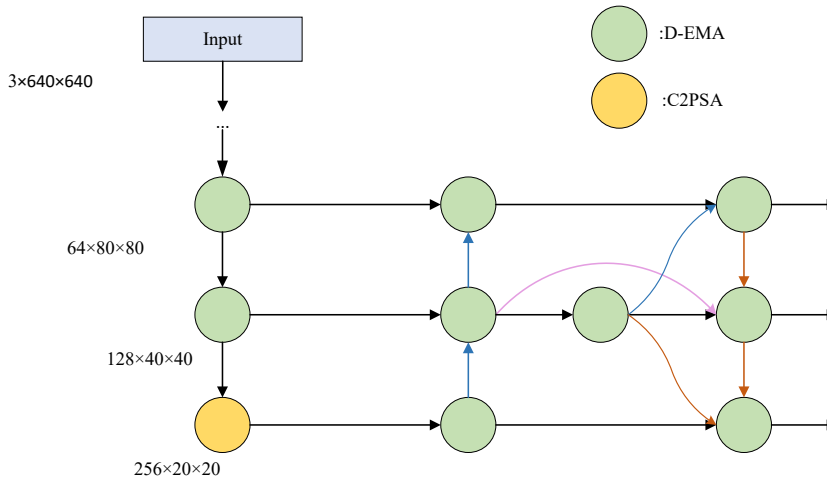



Figure 3. BDFPN structure diagram

### 3.3 Inner-CIoU

The existing IOU loss function demonstrates limitations in terms of poor generalization ability and slow convergence speed across various detection tasks. To address these issues, this paper introduces an innovative approach that accelerates the boundary box regression (BBR) process by utilizing auxiliary bounding boxes for loss calculation. In the Inner IOU [39] method, a scaling factor (ratio) is employed to control the size of the auxiliary bounding boxes. The specific calculation process is illustrated in Figure 4 (Figure 4 is drawn by the authors). By distinguishing between different regression samples and using auxiliary bounding boxes of varying scales for loss computation, this method significantly speeds up the boundary box regression process. The specific calculation formula is as follows:

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} * \text{ ratio}}{2}, \tag{5}$$

$$b_r^{gt} = x_c^{gt} + \frac{w^{gt} * \text{ ratio}}{2}, \tag{6}$$

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} * \text{ ratio}}{2}, \tag{7}$$

$$b_b^{gt} = y_c^{gt} + \frac{h^{gt} * \text{ ratio}}{2}, \tag{8}$$

$$b_l = x_c - \frac{w^* \text{ ratio}}{2}, \tag{9}$$

$$b_r = x_c + \frac{w^* \text{ ratio}}{2}, \tag{10}$$

$$b_t = y_c - \frac{h * \text{ ratio}}{2}, \tag{11}$$

$$b_b = y_c + \frac{h * \text{ ratio}}{2}. \tag{12}$$

The variables $b_l^{gt}, b_r^{gt}, b_t^{gt}, b_b^{gt}$ represent the left, right, top, and bottom boundaries of the GT box, respectively. These coordinates are calculated based on the GT box center and the scaling factor ratio. Similarly, the variables $b_l, b_r, b_t, b_b$ represent the left, right, top, and bottom boundaries of the anchor box, respectively, and are calculated using the anchor box center and the scaling factor ratio.

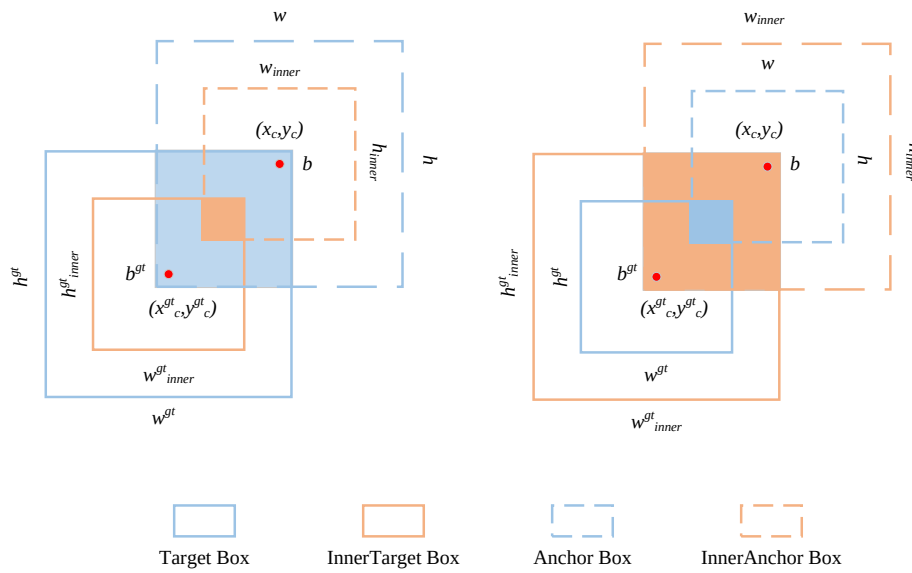



Figure 4. Inner-CIoU

The formula for calculating the intersection area (Intersection) is given by Equation (13):

$$\text{inter} = \left(\min\left(b_r^{gt}, b_r\right) - \max\left(b_l^{gt}, b_l\right)\right) * \left(\min\left(b_b^{gt}, b_b\right) - \max\left(b_t^{gt}, b_t\right)\right). \tag{13}$$

The formula for calculating the union area (Union) is given by Equation (14):

$$\text{union } = \left(w^{gt} * h^{gt}\right) * (\text{ ratio })^2 + (w * h) * (\text{ ratio })^2 - \text{ inter }, \tag{14}$$

$$IoU^{\text{inner}} = \frac{\text{inter}}{\text{union}}. \tag{15}$$

The intersection area (inter) represents the overlapping region between the GT box and the anchor box, while the union area (union) represents the combined area of the GT box and the anchor box. By combining Inner IoU and CIoU(Complete Intersection over Union), the Inner CIoU loss introduces a new penalty term to further optimize the shape of the predicted bounding boxes, going beyond the limitations of traditional DIoU loss. This penalty term ensures the consistency of the shapes of the two bounding boxes. The mathematical expression for the Inner CIoU loss is given by Equation (16) as follows:

$$L_{\text{inner-CloU}} = 1 - \text{loU}_{\text{inner}} + \frac{\rho^2}{C^2} + \alpha_2 v, \tag{16}$$

where:

$$\alpha_2 = \frac{v}{1 - \text{loU} + v}, \tag{17}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w_g}{h_g} - \arctan\frac{w_p}{h_p}\right)^2. \tag{18}$$

In the above formula, $\rho$ represents the Euclidean distance between the center points of the predicted box and the ground truth box, while $C$ denotes the diagonal length of the minimal enclosing box that contains both the predicted and ground truth boxes. $w_g, h_g$ and $w_p, h_p$ represent the width and height of the ground truth box and the predicted box, respectively.

The design goal of the Inner CIoU loss is to achieve better alignment between the shapes of the two bounding boxes by balancing their aspect ratio and diagonal tilt angle. This mechanism ensures precise alignment of the bounding box shapes.

Moreover, Inner CIoU not only preserves the theoretical validity of the IoU but also significantly accelerates model convergence by incorporating an auxiliary bounding box mechanism. Additionally, during the optimization process, the Inner CIoU loss considers both the intersection area and the center distance between the bounding boxes, while a shape loss term ensures the consistency of the predicted bounding box shape, thereby correcting the bias in the bounding box predictions.

## 4 EXPERIMENTS AND RESULTS

This section presents experiments conducted to validate the effectiveness of the proposed defect detection method. Two publicly available datasets, representing

diverse defect detection tasks, were selected to comprehensively evaluate the model's adaptability and robustness. Relevant evaluation metrics are introduced to quantify detection accuracy and performance across various scenarios. Finally, several widely used deep learning models serve as baselines for comparison, with experimental results and analyses provided.

### 4.1 Datasets and Evaluation Metrics

#### 4.1.1 Datasets

This experiment utilized two publicly available datasets: the NEU-DET dataset and the PCB defect dataset, provided by Northeastern University and Peking University, respectively. These datasets are widely used in defect detection.

The NEU-DET dataset contains a large collection of steel surface defects and is specifically focused on detecting defects on steel surfaces. It is commonly used for the development and evaluation of deep learning and image processing algorithms. The dataset includes various common types of steel surface defects, such as crazing (Cr), inclusion (In), patches (Pa), pitted surface (Ps), rolled-in scales (Rs), and scratches (Sc). For each defect type, there are 300 images, each of size $200 \times 200$ pixels, with each image containing one of the aforementioned defect types. The dataset provides clear annotations for each defect type, and the images have relatively high resolution, making it suitable for defect detection tasks. Examples of the different defect types in the dataset are shown in Figure 5 (Figure 5 contains images sourced from public datasets).

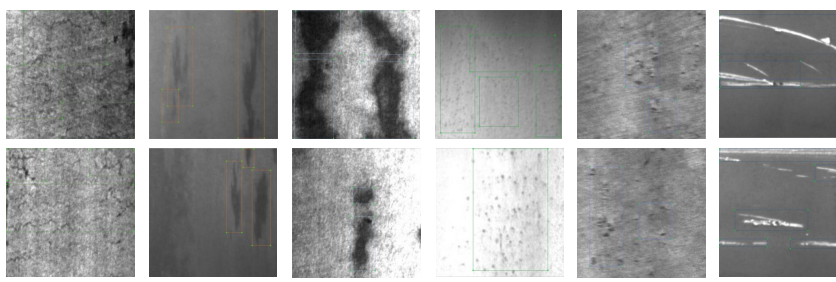



Figure 5. Partial presentation of the NEU-DET dataset

The PCB defect dataset is provided by Peking University and focuses on defect detection in industrial PCB production. This dataset includes six defect types: missing hole (MH), mouse bite (MB), open circuit (OC), short (SH), spur (SP), and spurious copper (SC). For each defect type, there are 115 images, with pixel dimensions ranging from $3\,034 \times 2\,464$ to $2\,904 \times 1\,521$. The image quality varies, including both high-resolution clear images as well as those that may contain noise

or low contrast, reflecting a more realistic application scenario. Each image is annotated with the defect type and location, making it suitable for defect detection tasks. Examples of the different defect types in the dataset are shown in Figure 6 (Figure 6 contains images sourced from public datasets).

The two datasets complement each other in terms of structure and content. The NEU-DET dataset is well-suited for fine-grained defect detection, while the PCB defect dataset aids in developing more robust detection algorithms, particularly in complex industrial environments. Together, these datasets provide rich testing resources for this study, enabling a comprehensive evaluation and optimization of the defect detection model's performance.

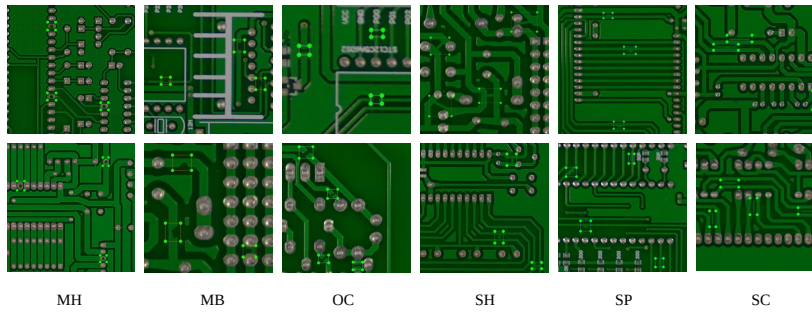



Figure 6. Partial presentation of the PCB defect dataset

### 4.1.2 Evaluation Metrics

In the field of object recognition, the Precision-Recall (PR) curve is a crucial metric for evaluating the performance of classification models, particularly in cases of class imbalance. Precision refers to the proportion of samples predicted as positive that are actually positive, while Recall indicates the proportion of actual positive samples that are correctly identified as positive by the model. The formulas for calculating Precision and Recall are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{20}$$

In the above formulas, TP (True Positive) represents the number of samples correctly predicted as positive, FP (False Positive) refers to the number of negative samples incorrectly predicted as positive, and FN (False Negative) indicates the number of positive samples incorrectly predicted as negative.

The Precision-Recall Curve (PR Curve) is commonly used to assess a model's performance at different thresholds, reflecting the trade-off between precision and

recall. However, the PR curve has certain limitations, especially in multi-class or high-complexity tasks, as a single PR curve may not fully capture the model's performance across multiple classes. To address this, Average Precision (AP), which quantifies the PR curve, has gradually become a core metric for evaluating model performance. AP is calculated by computing the area under the PR curve, providing a comprehensive score that evaluates precision and recall at various thresholds.

mAP (mean Average Precision) is the average of AP across multiple classes. It is calculated by determining the AP for each class and averaging them to evaluate the model's overall performance across all categories. The formula is given by Equation (21) as follows:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i. \tag{21}$$

In the formula, $N$ represents the number of classes, and $AP_i$ is the Average Precision for the $i^{\text{th}}$ class. mAP provides a balanced evaluation criterion across multiple classes.

In this paper, when the IoU exceeds a predefined threshold (set to 0.5 in this experiment), the predicted box is considered to match the ground truth box and is classified as True Positive (TP); otherwise, it is classified as False Positive (FP). IoU (Intersection over Union) is used to measure the overlap between the predicted box and the ground truth box. The IoU calculation formula is given by Equation (22) as follows:

$$IoU_{A,B} = \frac{|A \cap B|}{|A \cup B|}. \tag{22}$$

### 4.2 Experimental Environment and Confguration

All experiments were conducted on a PC running the Ubuntu operating system, with an Intel(R) Core(TM) i9-9940X CPU @ 3.30 GHz processor, an NVIDIA RTX2080Ti GPU, and 64 GB of RAM. The source files were compiled using Python 3.8.20, with PyTorch 1.12.0 as the deep learning framework. A batch size of 32 was used. Experiments were also conducted on a Windows platform. The initial learning rate was set to 0.01, and the minimum learning rate was 0.1 times the initial rate. A cosine annealing scheduler was used to adaptively control the learning rate. The Adam optimizer was selected, with weight decay and momentum set to 0.0005 and 0.937, respectively. The images in the NEU-DET and PCB defect datasets were resized to $640 \times 640$. The NEU-DET dataset was randomly split into training, validation, and test sets in a 0.8:0.1:0.1 ratio, while the PCB dataset was similarly split into 0.8:0.1:0.1.

### 4.3 Ablation Study and Analysis

In this section, a series of ablation experiments based on the PCB defect dataset will be conducted to systematically evaluate and analyze the key components of the

proposed method. The impact of each component on performance will be explored to further validate the effectiveness and advantages of the proposed approach.

### 4.3.1 Ablation Study of D-EMA Module

To validate the effectiveness of the proposed D-EMA structure, experiments are conducted by replacing D-EMA with different modules. The selected modules are the C3 module from YOLOv5, the C2F module from YOLOv8, and the C3K2 module from YOLOv11. The experimental results are shown in Table 1. As indicated in Table 1, under the same network architecture, the D-EMA module achieves the highest mAP on the PCB defect dataset, thereby confirming the effectiveness of the D-EMA module.

| Model in Backbone | Neck | Loss Function | mAP(%) |
|---|---|---|---|
| C3 | BDFPN | Inner-CIoU | 90.26 |
| C2F | BDFPN | Inner-CIoU | 93.57 |
| C3K2 | BDFPN | Inner-CIoU | 92.83 |
| **D-EMA** | **BDFPN** | **Inner-CIoU** | **95.70** |

Table 1. Ablation study results of the D-EMA module

### 4.3.2 Ablation Study of BDFPN

| Model in Backbone | Neck | Loss Function | mAP(%) |
|---|---|---|---|
| D-EMA | FPN | Inner-CIoU | 87.92 |
| D-EMA | PAN | Inner-CIoU | 93.41 |
| **D-EMA** | **BDFPN** | **Inner-CIoU** | **95.70** |

Table 2. Ablation study results of BDFPN

A comparison between FPN, PAN, and BDFPN is conducted on the PCB defect dataset, with the experimental results presented in Table 2. FPN enhances small object detection through a top-down pyramid structure, but its unidirectional information flow limits the effective guidance of high-level features to low-level features. PAN, built upon FPN, adds bottom-up path aggregation to strengthen the utilization of low-level features, although it may introduce redundant computations in some tasks. BDFPN employs bidirectional information flow, enhancing feature fusion across different levels and reducing interference from irrelevant information. As shown in Table 2, under the same experimental conditions, BDFPN achieves a significant improvement in mAP compared to FPN and PAN, thus validating the effectiveness of BDFPN.

### 4.3.3 Ablation Study of Inner-CIoU

A comparison is made between IoU, DIoU, and CIoU on the PCB defect dataset, with the experimental results presented in Table 3. The traditional IoU loss only focuses on the overlapping region between the predicted and ground truth boxes, failing to directly optimize the non-overlapping areas. This results in weaker gradients and reduced optimization efficiency at low IoU levels. The DIoU loss (Distance IoU) introduces a penalty term based on the distance between the centers of the boxes, improving convergence speed, but it does not account for aspect ratio impacts on the regression.

| Model in Backbone | Neck | Loss Function | mAP(%) |
|---|---|---|---|
| D-EMA | BDFPN | IoU | 92.61 |
| D-EMA | BDFPN | DIoU | 94.69 |
| D-EMA | BDFPN | CIoU | 95.28 |
| **D-EMA** | **BDFPN** | **Inner-CIoU** | **95.70** |

Table 3. Ablation study results of Inner-CIoU

The CIoU loss (Complete IoU) further combines DIoU with aspect ratio constraints, leading to more accurate boundary box shape matching, though its optimization effect decreases in the high IoU regime. The Inner-CIoU loss used in this study combines Inner-IoU with CIoU, allowing the predicted boxes to converge more stably when close to the ground truth boxes, resulting in more precise detection. As shown in Table 3, under the same experimental conditions, Inner-CIoU outperforms IoU, DIoU, and CIoU in terms of ground truth box matching accuracy, thereby confirming its effectiveness.

## 4.4 Comparative Experiments and Analysis

In this section, experiments were conducted using the two open-source datasets mentioned in Section 4.1.1, and the results were compared with those of mainstream defect detection methods to validate the effectiveness of the proposed approach. Based on the comparison of experimental results, a corresponding analysis is provided below.In the following experiments, the backbone using the D-EMA module is referred to as D-EMA Net.

### 4.4.1 Experiments on the PCB Dataset

The experimental results comparing the proposed method with mainstream approaches on the PCB defect dataset are shown in Table 4. The proposed method outperforms existing methods in terms of defect detection accuracy across all categories. Specifically, the overall mAP of the proposed method reached 95.70 %, which is an improvement of 2.8 % over YOLOv11, and an increase of 2.05 % and 2 % compared to YOLOv5 and YOLOv8, respectively. Notably, for categories where

YOLOv11 performs poorly, such as spur, the proposed method shows a particularly significant improvement, with a mAP increase of 5.2 %. This demonstrates that the proposed method exhibits stronger robustness when dealing with complex defect shapes. Additionally, for small-sized, irregularly shaped defects like mouse bite, the proposed method achieved a 4.9 % mAP improvement over YOLOv11, indicating its superior performance in detecting small-object defects.

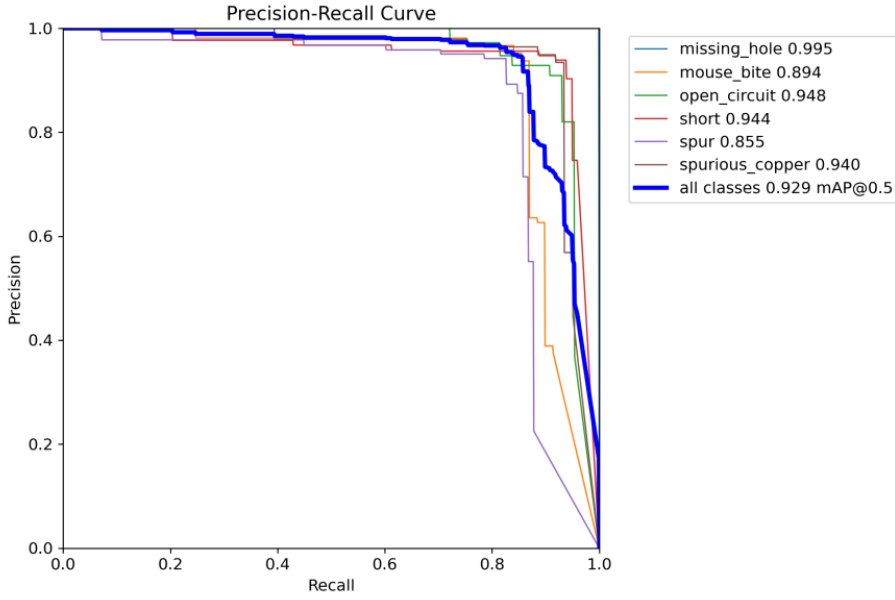
| Model | Input size | Backbone | mAP( %) |
|---|---|---|---|
| YOLOv3 | $640^2$ | DarkNet | 80.07 |
| YOLOv4 | $640^2$ | CSPDarkNet | 91.41 |
| YOLOv5n | $640^2$ | CSPDarkNet | 88.13 |
| YOLOv5m | $640^2$ | CSPDarkNet | 93.37 |
| YOLOv5l | $640^2$ | CSPDarkNet | 93.65 |
| YOLOv8n | $640^2$ | CSPDarkNet | 88.05 |
| YOLOv8m | $640^2$ | CSPDarkNet | 92.02 |
| YOLOv8l | $640^2$ | CSPDarkNet | 93.70 |
| YOLOv11n | $640^2$ | CSPDarkNet | 88.43 |
| YOLOv11m | $640^2$ | CSPDarkNet | 92.90 |
| YOLOv11l | $640^2$ | CSPDarkNet | 90.86 |
| Proposed method | $640^2$ | D-EMA Net | 95.70 |

Table 4. Comparison results on the PCB defect dataset

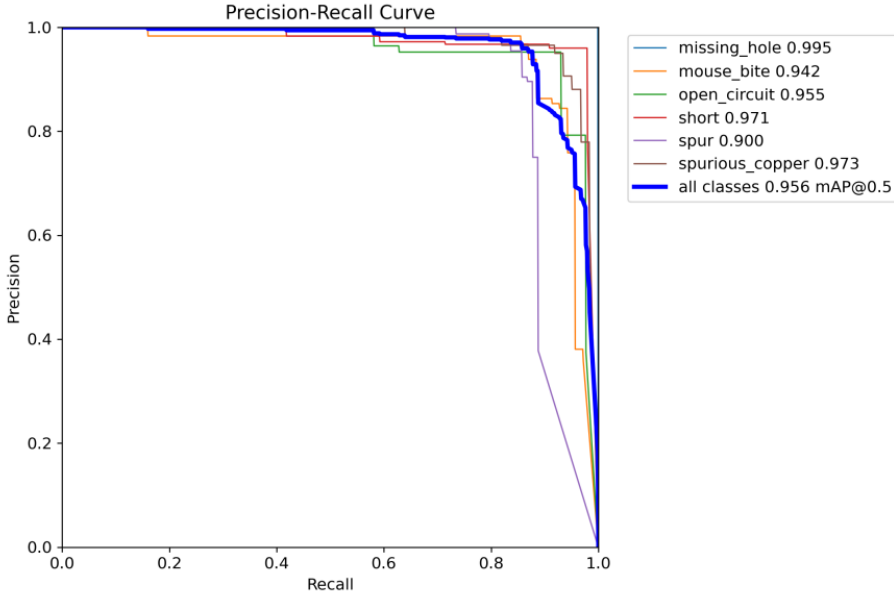
In addition, the comparison of the PR curves between the proposed method and YOLOv11, as shown in Figure 7 (Figure 7 is plotted based on experimental results), highlights the superior balance between precision and recall achieved by the proposed method. From the experimental results, it is evident that the proposed method outperforms the comparison methods across all six defect types in the PR curves, especially in the "spur" type, represented by the purple curve, where the improvement is most significant. Specifically, the PR curve for the "spur" defect type shows a noticeable trend toward the top-right corner, indicating that the proposed method effectively improves precision while maintaining a high recall rate.

These improvements can be attributed to the optimized design of specific modules in the proposed method, which enhances the feature extraction ability for small-sized and subtle structures. At the same time, the model demonstrates higher adaptability in multi-scale context information fusion. Additionally, through background noise suppression and feature space optimization, the model significantly reduces false positives and false negatives, leading to higher detection accuracy for "spur" type defects.

In Figure 8 (Figure 8 is generated from the model's detection results), the detection performance of YOLOv11 and DAE-YOLO on the PCB defect dataset is compared. The experimental results show that, although YOLOv11 is generally capable of accurately detecting defects on PCBs, it falls short when handling small defects and fails to effectively identify certain minor flaws. In contrast, DAE-YOLO




a) PR curve variation chart of YOLOv11 on the PCB defect dataset




b) PR curve variation chart of our proposed method on the PCB defect dataset

Figure 7. Comparison of PR curve variations between DAE-YOLO and YOLOv11

exhibits a stronger advantage in this regard, being better at detecting small or blurred defects.

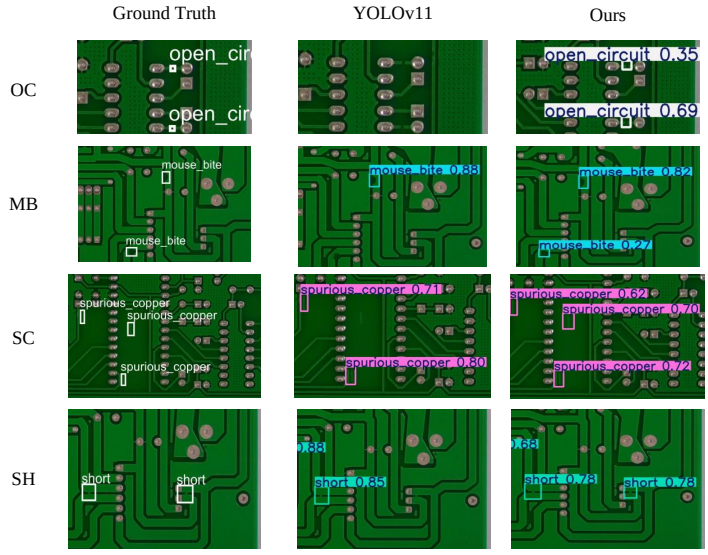



Figure 8. Visualization of the result comparison on the PCB defect dataset

### 4.4.2 Experiments on the NEU-DET Dataset

To further validate the performance of the proposed model under scale variations, the NEU-DET dataset was used to evaluate its effectiveness. Six object detectors, including Faster R-CNN, RetinaNet [40], YOLOv3, YOLOv5, YOLOv8, and YOLOv11, were selected for comparison, with the corresponding test results shown in Table 5.

YOLO series models, such as YOLOv3, YOLOv5, YOLOv8, and YOLOv11, due to their focus on generalization rather than specific optimization for defect detection data, achieved a maximum mAP that was 3.88 % lower than that of the proposed method on the NEU-DET dataset. YOLOv11 achieved a relatively low AP score of only 32.8 % for the "crazing" class, which is significantly lower than the 39.57 % AP obtained by the proposed method for the same class.

Figure 9 (Figure 9 is generated from the model's detection results) compares the detection performance of YOLOv11 and DAE-YOLO on the NEU-DET dataset. YOLOv11 shows reduced detection accuracy, particularly for defects with low contrast or irregular shapes, with increased missed and false positive detections, especially in images with significant noise. In contrast, DAE-YOLO performs significantly better, especially when handling small defects and noise interference.

| Model | Input size | Backbone | mAP(%) |
|---|---|---|---|
| Faster R-CNN | $640^2$ | VGG | 71.07 |
| RetinaNet | $640^2$ | ResNet50 | 64.18 |
| YOLOv3 | $640^2$ | DarkNet | 75.43 |
| YOLOv4 | $640^2$ | CSPDarkNet | 76.62 |
| YOLOv5n | $640^2$ | CSPDarkNet | 68.78 |
| YOLOv5m | $640^2$ | CSPDarkNet | 69.90 |
| YOLOv5l | $640^2$ | CSPDarkNet | 70.06 |
| YOLOv8n | $640^2$ | CSPDarkNet | 66.10 |
| YOLOv8m | $640^2$ | CSPDarkNet | 71.80 |
| YOLOv8l | $640^2$ | CSPDarkNet | 72.26 |
| YOLOv11n | $640^2$ | CSPDarkNet | 70.17 |
| YOLOv11m | $640^2$ | CSPDarkNet | 70.74 |
| Proposed method | $640^2$ | D-EMA Net | 74.62 |

Table 5. Comparison results on the NEU-DET dataset

Through analysis, it can be observed that the superior performance of the proposed method on the PCB defect dataset can be attributed to the following factors:

1. The D-EMA module proposed in this paper introduces a deformable convolution module on low-resolution feature maps. By flexibly adjusting sampling points, it effectively captures key information of small targets, thereby enhancing detection performance. Additionally, the efficient multi-scale attention mechanism explicitly models spatial and semantic dependencies, allowing it to preserve the spatial positions and fine-grained semantic information of small targets during the feature extraction process, significantly reducing feature attenuation.
2. The bidirectional feature fusion path in BDFPN, which includes both top-down and bottom-up connections, fully integrates the detailed information from high-resolution features with the semantic information from low-resolution features, thereby enhancing the model's ability to detect multi-scale targets. Furthermore, BDFPN incorporates a learnable weighted feature fusion mechanism, allowing the network to adaptively allocate the importance of features from different layers and optimize the efficiency of information flow. This design not only improves the model's detection accuracy but also significantly reduces computational complexity.

## 5 CONCLUSIONS

This paper addresses the poor performance of mainstream object recognition algorithms in the field of defect detection by proposing an optimized detection method to tackle the specific challenges of defect detection tasks. Through an in-depth analysis of domain characteristics and task requirements, the paper designs and implements three innovations that significantly enhance detection performance.

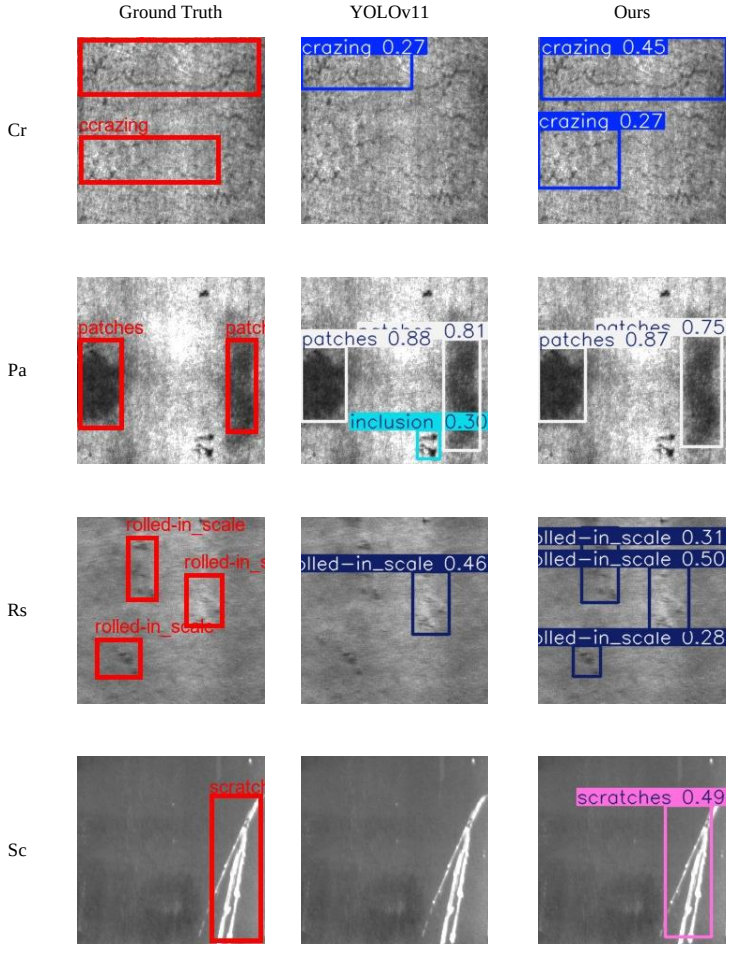



Figure 9. Visualization of the result comparison on the NEU-DET dataset

First, the D-EMA module was developed, combining the latest deformable convolution DCNv4 and the EMA attention mechanism. This module performs excellently when handling defects with irregular shapes and subtle differences. DCNv4 enhances the model's ability to flexibly adapt to target shapes, while the EMA attention mechanism strengthens the focus on key areas, significantly improving the model's ability to discern defect targets.

Second, in response to the challenges of feature fusion, the paper proposes the BDFPN (Bi-directional Dynamic Feature Pyramid Network) as the Neck component. Drawing on the idea of BiFPN, BDFPN dynamically adjusts the importance of features from different layers to achieve more efficient multi-scale information fu-

sion. This is particularly advantageous when handling defects with diverse sizes and complex backgrounds.

Finally, the paper improves the CIoU (Complete Intersection over Union) by introducing Inner CIoU, integrating Inner IOU into the boundary box regression loss function. This further enhances the model's accuracy in target boundary fitting, especially for detecting small or overlapping defects.

Experimental results demonstrate the effectiveness of the proposed method. Compared to mainstream detection models such as YOLO series, Faster R-CNN, and RetinaNet, the method achieves notable improvements across multiple metrics, including accuracy, recall, and robustness to small targets and complex backgrounds. While conventional models often struggle with abundant, irregular, and low-contrast defects, the proposed approach exhibits superior adaptability and resilience.

Nonetheless, certain limitations remain. Detection performance for rare defect patterns and ultra-low contrast cases requires further enhancement. Additionally, balancing accuracy with inference speed and model compactness is necessary to meet industrial application demands. The model's generalization also warrants validation on more diverse and complex datasets. Future work will focus on improving detection precision and efficiency, exploring innovative techniques, and optimizing the model to better serve practical defect detection needs.

**Acknowledgements**

Funded project: Research and Application of Intelligent Appearance Inspection Instrument in Mianyang Cigarette Factory, Sichuan China Tobacco Industry Co., Ltd. (SGYK2024018)

## REFERENCES

[1] Saberironaghi, A.—Ren, J.—El-Gindy, M.: Defect Detection Methods for Industrial Products Using Deep Learning Techniques: A Review. Algorithms, Vol. 16, 2023, No. 2, Art. No. 95, doi: 10.3390/a16020095.

[2] Khanam, R.—Hussain, M.—Hill, R.—Allen, P.: A Comprehensive Review of Convolutional Neural Networks for Defect Detection in Industrial Applications. IEEE Access, Vol. 12, 2024, pp. 94250–94295, doi: 10.1109/ACCESS.2024.3425166.

[3] Singh, S. A.—Desai, K. A.: Automated Surface Defect Detection Framework Using Machine Vision and Convolutional Neural Networks. Journal of Intelligent Manufacturing, Vol. 34, 2023, No. 4, pp. 1995–2011, doi: 10.1007/s10845-021-01878-w.

[4] Raheja, J. L.—Kumar, S.—Chaudhary, A.: Fabric Defect Detection Based on GLCM and Gabor Filter: A Comparison. Optik, Vol. 124, 2013, No. 23, pp. 6469–6474, doi: 10.1016/j.ijleo.2013.05.004.

[5] Liu, X.—Xue, F.—Teng, L.: Surface Defect Detection Based on Gradient LBP. 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), 2018, pp. 133–137, doi: 10.1109/ICIVC.2018.8492798.

[6] Liang, Y.—Xu, K.—Zhou, P.: Mask Gradient Response-Based Threshold Segmentation for Surface Defect Detection of Milled Aluminum Ingot. Sensors, Vol. 20, 2020, No. 16, Art. No. 4519, doi: 10.3390/s20164519.

[7] Rathod, V. R.—Anand, R. S.: A Comparative Study of Different Segmentation Techniques for Detection of Flaws in NDE Weld Images. Journal of Nondestructive Evaluation, Vol. 31, 2012, No. 1, pp. 1–16, doi: 10.1007/s10921-011-0116-6.

[8] Fu, Y.—Downey, A. R. J.—Yuan, L.—Zhang, T.—Pratt, A.—Balogun, Y.: Machine Learning Algorithms for Defect Detection in Metal Laser-Based Additive Manufacturing: A Review. Journal of Manufacturing Processes, Vol. 75, 2022, pp. 693–710, doi: 10.1016/j.jmapro.2021.12.061.

[9] Mustaqeem, M.—Saqib, M.: Principal Component Based Support Vector Machine (PC-SVM): A Hybrid Technique for Software Defect Detection. Cluster Computing, Vol. 24, 2021, No. 3, pp. 2581–2595, doi: 10.1007/s10586-021-03282-8.

[10] Aghdam, S. R.—Amid, E.—Imani, M. F.: A Fast Method of Steel Surface Defect Detection Using Decision Trees Applied to LBP Based Features. 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2012, pp. 1447–1452, doi: 10.1109/ICIEA.2012.6360951.

[11] Rustam, Z.—Saragih, G. S.: Predicting Bank Financial Failures Using Random Forest. 2018 International Workshop on Big Data and Information Security (IWBIS), IEEE, 2018, pp. 81–86, doi: 10.1109/IWBIS.2018.8471718.

[12] Wang, H.—Liu, C.—Yu, L.—Zhao, J.: Research on Target Detection and Recognition Algorithm Based on Deep Learning. 2019 Chinese Control Conference (CCC), IEEE, 2019, pp. 8483–8487, doi: 10.23919/ChiCC.2019.8865560.

[13] Cha, Y. J.—Choi, W.—Suh, G.—Mahmoudkhani, S.—Büyüköztürk, O.: Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. Computer-Aided Civil and Infrastructure Engineering, Vol. 33, 2018, No. 9, pp. 731–747, doi: 10.1111/mice.12334.

[14] Li, Y.—Huang, H.—Xie, Q.—Yao, L.—Chen, Q.: Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD. Applied Sciences, Vol. 8, 2018, No. 9, Art. No. 1678, doi: 10.3390/app8091678.

[15] Chen, M.—Yu, L.—Zhi, C.—Sun, R.—Zhu, S.—Gao, Z.—Ke, Z.—Zhu, M.—Zhang, Y.: Improved Faster R-CNN for Fabric Defect Detection Based on Gabor Filter with Genetic Algorithm Optimization. Computers in Industry, Vol. 134, 2022, Art. No. 103551, doi: 10.1016/j.compind.2021.103551.

[16] Xie, X.—Cheng, G.—Wang, J.—Yao, X.—Han, J.: Oriented R-CNN for Object Detection. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3500–3509, doi: 10.1109/ICCV48922.2021.00350.

[17] Jing, J.—Zhuo, D.—Zhang, H.—Liang, Y.—Zheng, M.: Fabric Defect Detection Using the Improved YOLOv3 Model. Journal of Engineered Fibers and Fabrics, Vol. 15, 2020, Art. No. 1558925020908268, doi: 10.1177/1558925020908268.

[18] Li, X.—Wang, C.—Ju, H.—Li, Z.: Surface Defect Detection Model for Aero-Engine Components Based on Improved YOLOv5. Applied Sciences, Vol. 12, 2022, No. 14, Art. No. 7235, doi: 10.3390/app12147235.

[19] Hocenski, Z.—Vasilic, S.—Hocenski, V.: Improved Canny Edge Detector in

Ceramic Tiles Defect Detection. IECON 2006 – 32nd Annual Conference on IEEE Industrial Electronics, 2006, pp. 3328–3331, doi: 10.1109/IECON.2006.347535.

[20] Mak, K. L.—Peng, P.—Yiu, K. F. C.: Fabric Defect Detection Using Morphological Filters. Image and Vision Computing, Vol. 27, 2009, No. 10, pp. 1585–1592, doi: 10.1016/j.imavis.2009.03.007.

[21] Nagata, F.—Tokuno, K.—Mitarai, K.—Otsuka, A.—Ikeda, T.—Ochi, H.—Watanabe, K.—Habib, M. K.: Defect Detection Method Using Deep Convolutional Neural Network, Support Vector Machine and Template Matching Techniques. Artificial Life and Robotics, Vol. 24, 2019, No. 4, pp. 512–519, doi: 10.1007/s10015-019-00545-x.

[22] Annaby, M. H.—Fouda, Y. M.—Rushdi, M. A.: Improved Normalized Cross-Correlation for Defect Detection in Printed-Circuit Boards. IEEE Transactions on Semiconductor Manufacturing, Vol. 32, 2019, No. 2, pp. 199–211, doi: 10.1109/TSM.2019.2911062.

[23] Nakajima, R.—Yamamoto, R.—Hida, T.—Matsumoto, T.: A Study on the Effect of Defect Shape on Defect Detection in Visual Inspection. Procedia Manufacturing, Vol. 39, 2019, pp. 1641–1648, doi: 10.1016/j.promfg.2020.01.277.

[24] Li, S. Z.: Markov Random Field Modeling in Image Analysis. Springer, 2009, doi: 10.1007/978-1-84800-279-1.

[25] Liu, K.—Wang, H.—Chen, H.—Qu, E.—Tian, Y.—Sun, H.: Steel Surface Defect Detection Using a New Haar–Weibull-Variance Model in Unsupervised Manner. IEEE Transactions on Instrumentation and Measurement, Vol. 66, 2017, No. 10, pp. 2585–2596, doi: 10.1109/TIM.2017.2712838.

[26] Yang, J.—Li, X.—Xu, J.—Cao, Y.—Zhang, Y.—Wang, L.—Jiang, S.: Development of an Optical Defect Inspection Algorithm Based on an Active Contour Model for Large Steel Roller Surfaces. Applied Optics, Vol. 57, 2018, No. 10, pp. 2490–2498, doi: 10.1364/AO.57.002490.

[27] Lu, S.—Yue, Y.—Liu, X.—Wu, J.—Wang, Y.: A Novel Unbalanced Weighted KNN Based on SVM Method for Pipeline Defect Detection Using Eddy Current Measurements. Measurement Science and Technology, Vol. 34, 2023, No. 1, Art. No. 014001, doi: 10.1088/1361-6501/ac9545.

[28] Luo, J.—Yang, Z.—Li, S.—Wu, Y.: FPCB Surface Defect Detection: A Decoupled Two-Stage Object Detection Framework. IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021, pp. 1–11, doi: 10.1109/TIM.2021.3092510.

[29] Li, F.—Li, F.—Xi, Q.: DefectNet: Toward Fast and Effective Defect Detection. IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021, pp. 1–9, doi: 10.1109/TIM.2021.3067221.

[30] Xie, L.—Xiang, X.—Xu, H.—Wang, L.—Lin, L.—Yin, G.: FFCNN: A Deep Neural Network for Surface Defect Detection of Magnetic Tile. IEEE Transactions on Industrial Electronics, Vol. 68, 2021, No. 4, pp. 3506–3516, doi: 10.1109/TIE.2020.2982115.

[31] Hou, W.—Wei, Y.—Jin, Y.—Zhu, C.: Deep Features Based on a DCNN Model for Classifying Imbalanced Weld Flaw Types. Measurement, Vol. 131, 2019, pp. 482–489, doi: 10.1016/j.measurement.2018.09.011.

[32] Wei, B.—Hao, K.—Tang, X.—Ren, L.: Fabric Defect Detection Based on Faster RCNN. In: Wong, W. K. (Ed.): Artificial Intelligence on Fashion and Textiles (AITA 2018). Springer, Cham, Advances in Intelligent Systems and Computing, Vol. 849, 2019, pp. 45–51, doi: 10.1007/978-3-319-99695-0_6.

[33] Li, Z.—Tian, X.—Liu, X.—Liu, Y.—Shi, X.: A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResNetV2 Models. Applied Sciences, Vol. 12, 2022, No. 2, Art. No. 834, doi: 10.3390/app12020834.

[34] Xiong, Y.—Li, Z.—Chen, Y.—Wang, F.—Zhu, X.—Luo, J.—Wang, W.—Lu, T.—Li, H.—Qiao, Y.: Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 5652–5661, doi: 10.1109/CVPR52733.2024.00540.

[35] Ouyang, D.—He, S.—Zhang, G.—Luo, M.—Guo, H.—Zhan, J.—Huang, Z.: Efficient Multi-Scale Attention Module with Cross-Spatial Learning. ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10096516.

[36] Wang, W.—Dai, J.—Chen, Z.—Huang, Z.—Li, Z.—Zhu, X.—Hu, X.—Lu, T.—Lu, L.—Li, H.—Wang, X.—Qiao, Y.: InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 14408–14419, doi: 10.1109/CVPR52729.2023.01385.

[37] Hou, Q.—Zhou, D.—Feng, J.: Coordinate Attention for Efficient Mobile Network Design. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13708–13717, doi: 10.1109/CVPR46437.2021.01350.

[38] Liu, H.—Liu, F.—Fan, X.—Huang, D.: Polarized Self-Attention: Towards High-Quality Pixel-Wise Mapping. Neurocomputing, Vol. 506, 2022, pp. 158–167, doi: 10.1016/j.neucom.2022.07.054.

[39] Zhang, H.—Xu, C.—Zhang, S.: Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box. 2023, doi: 10.48550/arXiv.2311.02877 (arXiv Preprint arXiv:2311.02877).

[40] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.



**Guangqiao Xie** works at the Mianyang Cigarette Factory. He graduated from the Southwest University of Science and Technology. He participated in multiple innovation projects at the factory, published 9 papers, and 1 national invention patent, obtained 2 software copyrights and 10 patents.



**Yan Zhou** works at the Mianyang Cigarette Factory of Sichuan China Tobacco Industry Company Limited, Mianyang, China. She graduated in electronic information engineering at the Changsha University of Science and Technology. She entered Mianyang Cigarette Factory of Sichuan China Tobacco Industry Company Limited in 2008 and has been engaged in information technology work since then. She participated in multiple innovation projects at the factory, published 3 papers, and obtained 6 software copyrights.



**Chengshang Wu** is currently studying civil engineering with a minor in artificial intelligence at the Dalian Maritime University. He has participated in numerous science and technology innovation competitions and won a gold medal at the International Robot Olympiad in China.