scientific reports

Check for updates

OPEN

A high precision YOLO model for surface defect detection based on PyConv and CISBA

Shufen Ruan[1,2,4], Chenmei Zhan[1,4], Bo Liu[1,4], Quan Wan[1,4] & Kunfang Song[3,4]✉

Defect detection is vital for product quality in industrial production, yet current surface defect detection technologies struggle with diverse defect types and complex backgrounds. The challenge intensifies with multi-scale small targets, leading to significantly reduced detection performance. Therefore, this paper proposes the EPSC-YOLO algorithm to improve the efficiency and accuracy of defect detection. The algorithm first introduces multi-scale attention modules and uses two newly designed pyramid convolutions in the backbone network to better identify multi-scale defects; Secondly, Soft-NMS is introduced to replace traditional NMS, which can reduce information loss and improve multi-target detection accuracy by smoothing and suppressing the scores of overlapping boxes. In addition, a new convolutional attention module, CISBA, is designed to enhance the detection capability of small targets in complex backgrounds. In the end, we validate the effectiveness of EPSC-YOLO on NEU-DET and GC10-DET datasets. The experimental results show that, compared to YOLOv9c, mAP_{50}^{val} increases by 2% and 2.4%, and $mAP_{50:95}^{val}$ increases by 5.1% and 2.4%, respectively. Meanwhile, EPSC-YOLO demonstrates superior accuracy and significant advantages in real-time detection of surface defects on products compared to algorithms such as YOLOv10 and MSFT-YOLO.

Detection plays a crucial role in optimizing product quality[1], reducing production costs[2], improving production efficiency[3], ensuring safety[4], complying with regulatory standards, providing data analysis, and enhancing market competitiveness. It is an indispensable part of modern manufacturing and production processes.

Traditionally, the defect detection was completed by people. However, the manual defect inspection is slow, costly, and even dangerous. So, how to alleviate above problem, a simple structure, fast speed and high accuracy object detection algorithms is important. Nowadays, more and more excellent defect detection algorithms are emerging. These algorithms typically utilize machine learning and computer vision technologies to automatically or semi-automatically detect defects and anomalies in products or systems.

As research continues to deepen, the accuracy of these algorithms is also improving. These algorithms not only enhance the accuracy and efficiency of detection but also help companies reduce costs and improve production quality. Zeng et al. proposed a multi-scale feature fusion method called the atrous spatial pyramid pooling-balanced-feature pyramid network (ABFPN)[5], which used a skipped atrous spatial pyramid pooling (contains five D-ASPP blocks[6]) module and a balanced module (contains three blocks, which are the resize and average block, the space nonlocal block, and the residual block[7,8]). Rudolph et al. proposed an innovative fully convolutional cross-scale normalizing flow (CS-Flow) approach, capable of jointly processing multiple feature maps at different scales[9]. Lv et al. proposed a novel defect detection network (EDDN) based on the Single Shot Multi-Box Detector, which can deal with defects with different scales[10] . Cheng et al. proposed a new deep neural network, named DEA_RetinaNet, which used the adaptive spatial feature fusion (ASFF) module and a novel channel attention mechanism[11]. Rudolph proposed a novel network, DifferNet, which leverages the descriptive features extracted by convolutional neural networks to estimate their density using normalizing flows[12]. Xu et al. proposed an improved Mask R-CNN algorithm for tunnel surface defect detection and segmentation[13]. This algorithm introduced a Path Aggregation Feature Pyramid Network (PAFPN) and adds an edge detection branch. Cracks, alligator cracks, and potholes on the road surface can easily lead to traffic safety issues. Therefore, timely detection and identification of these defects are crucial for reducing hazards. Chen et al. proposed a network named MANet, which is based on multi-scale mobile attention, for road surface defect detection[14]. MANet not

[1]The School of Mathematical and Physical Sciences, Wuhan Textile University, Wuhan 430200, China. [2]Research Center for Applied Mathematics and Interdisciplinary Sciences, Wuhan Textile University, Wuhan 430200, China. [3]The School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200, China. [4]Shufen Ruan, Chenmei Zhan, Bo Liu, Quan Wan, and Kunfang Song contributed equally to this work. ✉email: skf@wtu.edu.cn

only adopts an encoder-decoder architecture but also incorporates multi-scale convolutions and hybrid attention modules. Fan et al. proposed an improved algorithm for detecting defective apples[15]. This algorithm is based on YOLOv4 and incorporates channel pruning and layer pruning methods, along with a new L1-norm based non-maximum suppression (NMS) technique. The algorithm successfully implements defect detection in the apple sorting process, but it is significantly affected by lighting variations, struggles with detecting small defects in complex backgrounds, and faces accuracy issues when distinguishing between different types of fruits. Hu et al. proposed an improved YOLOv5 algorithm for detecting surface defects in oranges[16]. This algorithm integrates the CBAM attention mechanism and replaces the loss function with DIoU. The method improves the detection accuracy of citrus epidermis defects, but there are still challenges in distinguishing small defects from epidermis noise, especially when multiple surface defects coexist. Xiong et al. proposed an improved YOLOv5 deep learning algorithm for license plate defect detection[17]. This method introduces a novel attention module that combines ECA-net and CBAM, adopts a simplified BottleneckCSP structure, and modifying the loss function as Alpha-IoU. This algorithm struggles with detecting low-contrast and small defects, and has a higher false detection rate in complex backgrounds. Xu et al. proposed an improved YOLOv7 algorithm for defecting the weld surface[18]. This algorithm introduces a coordinate attention mechanism and integrates the newly designed Le-HorBlock module into the last four CBS modules of the backbone network to fully extract information. Additionally, the CIoU loss function is replaced with the SIoU loss function. This algorithm has made some progress in pipeline weld surface defect detection, but there are still challenges in balancing real-time performance and high accuracy for large-scale weld defect detection. Zhang et al. proposed a YOLOv8-CM framework, which combines Convolutional Block Attention Module (CBAM), Segment Anything Model (SAM) and U-shaped Network (U-net)[19]. This algorithm improves the accuracy of automatic detection and segmentation of tunnel defects and objects, but there are issues with segmenting complex-shaped defects, especially in areas with overlapping or similar structures, where the algorithm may suffer from misidentification or missed detection. Lu et al. proposed an improved algorithm based on YOLOv9, named WSS-YOLO[20]. The WSS-YOLO combines the C2f-DSC (integrates the dynamic snake convolution into C2f), GSConv and VOV-GSCSP modules. It has improved the steel surface defect detection network, but the model's performance remains limited when dealing with irregular and minute defects, especially when the surface has oxides, corrosion, and other complex textures, leading to unstable defect detection results.

Based on the research on improved YOLO-series algorithms, it can be observed that although these algorithms have made significant progress in the field of defect detection, several issues remain that urgently need to be addressed. These can be summarized in the following three points: (1) Detecting small defects in complex backgrounds, especially when multiple types of defects coexist, remains challenging. (2) In industrial application scenarios requiring extremely high real-time performance, achieving rapid detection while ensuring high accuracy continues to be a challenge that must be overcome. (3) Existing models demonstrate limited accuracy and robustness in detecting low-contrast or small defects, which restricts their applicability in certain complex environments.

In summary, to address the issue of detecting small and multi-scale objects in complex backgrounds, this paper proposes an improved EPSC-YOLO algorithm.

The main contributions in this paper are as follows:

1. Introduced the efficient multi-scale attention module (EMA) , which enhance the attention to important multi-scale objects.
2. Designed two types of pyramidal convolution (PyConv) module, which improve the multi-scale feature extraction capability and the accuracy of the network.
3. Proposed a new convolutional attention mechanism module, named CISBA module, which enhance the detection capability of small targets in complex backgrounds.
4. Used Soft-NMS instead of NMS, which reduce information loss and improve multi-target detection accuracy by smoothing and suppressing the scores of overlapping boxes.

The remaining structure of this paper is as follows: Section "Related works" discusses related works. In "Methods" section, we provide a detailed description of the proposed method in this paper, named EPCS-YOLO. Section "Experiments and results" presents the experimental results of proposed method on two public datasets, comparing them with the baseline model, some classical object detection algorithms and improved object detection algorithms. Section "Conclusion" is dedicated to discussion and conclusion.

Related works

Object detection algorithms

The development of object detection algorithms has gone through two periods: the era of traditional object detection algorithms before 2014 and the era of deep learning-based object detection algorithms since 2014. Traditional object detection algorithms such as Viola Jones (VJ) Detector[21], Histogram of Oriented Gradient (HOG) Detector[22] and Deformable Part Model (DPM)[23], primarily relied on manually designed features and conventional computer vision techniques, making it difficult to adapt to complex scenes and diverse features. It wasn't until the emergence of Region-based Convolutional Neural Network (R-CNN) in 2014, which introduced deep learning to object detection, that deep learning-based object detection algorithms began to develop rapidly. R-CNN algorithm first gains a lot of region proposals through selective search algorithm, then extracts features using a convolutional neural network, finally predicts whether these region proposals contain objects and their categories using a linear Support Vector Machine (SVM) classifier[24]. Although R-CNN has gain significant improvements compared to conventional object detection algorithms, its shortcomings are also quite evident: computational redundancy. The selective search algorithm generates a large number of overlapping region

proposals, leading to extensive repeated computations by the convolutional neural network. This results in very slow detection speeds for the R-CNN algorithm, making it unable to satisfy industrial demands. In order to reduce the computational redundancy brought by lots of overlapping region proposals, He et al. proposed Spatial Pyramid Pooling Network (SPP-Net)[25]. SPP-Net directly feeds the entire image into a convolutional neural network (CNN), then extracts image features using Spatial Pyramid Pooling (SPP), and finally utilizes a fully connected neural network to produce the final output. SPP-Net not only fine-tuned only the fully connected layers, ignoring the parameters of other layers in the network, but also suffers from drawbacks such as a complex training process. Fast Region-based Convolutional Neural Network (Fast R-CNN) is an improved version based RCNN and SPP-Net[26]. Fast R-CNN solved the problem that exist in SPP-Net. Although Fast R-CNN processes an image in approximately two seconds, which is faster than R-CNN, this speed is still not ideal for large real-world datasets. Faster R-CNN significantly improves the generation speed of detection boxes by using a Region Proposal Network, making it the first end-to-end object detection algorithm with near real-time performance[27]. It offers higher accuracy and faster speed compared to previous object detection algorithms. The deep learning-based object detection algorithms mentioned above are all two-stage object detection algorithms. These two-stage detection algorithms typically consist of two phases: the first phase generates candidate regions (such as Selective Search or Region Proposal Network), and the second phase performs object classification and precise localization. This multi-stage design results in high overall computational complexity, requiring more computational resources and time, making it difficult to meet the demands of practical applications. Therefore, single-stage object detection algorithms such as the YOLO series and SSD series emerged, with the YOLO object detection algorithm becoming the most widely used object detection algorithms in the industrial field due to their advantages of high speed and computational efficiency.

YOLOv1 is the first single-stage deep learning object detection algorithm, known for its very fast detection speed. The main idea of YOLOv1 algorithm is to divide the image into multiple grids and simultaneously predict the bounding boxes, classes and probabilities for each grid[28]. Compared to two-stage object detection algorithms, although YOLOv1 has significantly improved detection speed, it comes at the cost of some accuracy. The SSD object detection algorithm uses a multi-branch structure to simultaneously predict bounding boxes, classes, and probabilities on feature maps at different scales, thereby achieving higher detection accuracy and speed compared to YOLOv1[29]. By replacing the feature extraction network VGG-16 in YOLOv1 with DarkNet19, YOLOv2 achieved significant improvements in accuracy, speed, and the number of detectable classes[30]. The RetinaNet object detection algorithm, based on FPN, addresses the issue of class imbalance by introducing Focal Loss, which automatically adjusts weights according to the value of the loss, achieving high-precision and high-speed single-stage object detection[31].By replacing the backbone DarkNet19 in YOLOv2 with DarkNet53 to extract feature, classifying the objects using the logistic function instead of the softmax function, using triple-branch to detect objects of different scales. YOLOv3 achieved significant improvements in speed[32]. EfficientDet uses EfficientNet as the backbone network, combined with BiFPN (Bi-Directional Feature Pyramid Network) and Automated Machine Learning (AutoML) techniques (including neural architecture search and hyperparameter optimization), to achieve efficient feature fusion and model optimization[33]. YOLOv4 introduces several techniques such as feature enhancement and data augmentation, significantly improving the algorithm's detection accuracy and speed[34]. Compared to YOLOv3, YOLOv4 better balances the precision and speed of object detection. Scaled-YOLOv4 introduces a series of improvements based on YOLOv4, such as scaling the model's depth, width, and input resolution[35]. These enhancements improve the algorithm's performance across different hardware environments and increase the accuracy and speed of object detection in various application scenarios, addressing the performance optimization limitations of YOLOv4. DETR (DEtection TRansformer) introduces the Transformer architecture into object detection, replacing traditional convolutional neural networks (CNNs) and region-based methods to achieve end-to-end object detection[36]. By reformulating object detection as a set prediction problem and using self-attention mechanisms to directly predict object bounding boxes and class labels, DETR simplifies complex processes in traditional algorithms, such as manually designed anchor boxes and non-maximum suppression (NMS), offering a more streamlined and effective object detection solution. YOLOX introduces an anchor-free approach and integrates technologies like decoupled head and SimOTA, addressing the complexities in anchor box design and label assignment of traditional YOLO models, thus enhancing detection accuracy and efficiency[37]. YOLOR is a multi-task learning method based on a unified network that enhances object detection accuracy and efficiency by integrating explicit and implicit knowledge[38]. YOLOF is a simple and efficient object detection framework that relies on a single feature layer for detection. By incorporating dilated encoders and balanced matching strategies, it enhances the algorithm's detection efficiency and accuracy[39]. YOLOv5 addressed YOLOv4's issues with model complexity and training efficiency by introducing a more lightweight model architecture and improved training strategies[40]. PP-YOLOE is an object detection algorithm based on the PaddlePaddle framework that uses an anchor-free approach. By incorporating technologies such as CSPRepResNet, PAN (Path Aggregation Network), ET-head (Efficient Transformer head), and TAL (Task Adaptive Learning), it enhances both detection efficiency and accuracy[41]. YOLOv6 introduces efficient convolution layers, depthwise separable convolutions, and adaptive anchor box strategies, optimizing the balance between detection accuracy and computational speed, achieving faster and more precise object detection[42]. YOLOv7 builds on ELAN with a new network architecture, E-ELAN, to enhance gradient flow, and explores several trainable "bag-of-freebies" methods, significantly improving detection accuracy and efficiency while addressing gradient stability and feature fusion issues[43]. YOLOv8 introduces the C2f (Cross-Stage Feature Fusion) module for effective feature extraction and fusion, addressing limitations in feature integration and enhancing both detection accuracy and computational efficiency[44]. Gold-YOLO is an improved model based on YOLOv8. The algorithm enhances multi-scale feature fusion by introducing an advanced Gather-and-Distribute mechanism, significantly improving detection accuracy[45]. YOLOv9 introduces GELAN for improved architecture and PGI to enhance the training process, addressing limitations in feature extraction and model

optimization to boost overall detection performance[46]. YOLOv10 introduces a consistent dual assignment strategy and eliminates the dependence on Non-Maximum Suppression (NMS), thereby achieving high accuracy and low latency[47].

Although the aforementioned YOLO series algorithms achieve a good balance between efficiency and accuracy in surface defect detection, there is still room for improvement in detecting defects within complex backgrounds and in identifying multiple types of defects. Therefore, in this paper, we use YOLOv9c (shown in Fig. 1) as the baseline model for our improvements.

Attention mechanism

In defect detection tasks, traditional object detection methods typically involve an exhaustive search over the entire image to locate potential defect positions. These approaches are computationally expensive, especially when processing high-resolution industrial images. Searching for every possible defect location and scale is not only time-consuming but also prone to reduced detection accuracy, resulting in false positives and false negatives. In contrast, defect detection methods that incorporate attention mechanisms can more intelligently focus on key areas of the image where defects are likely to occur. By assigning more weight to these critical regions, attention mechanisms reduce the computational burden associated with irrelevant areas, significantly improving both the accuracy and efficiency of defect detection and ensuring more precise localization of defects. Xuan et al. introduce different attention mechanism to improve the ability of small object detection[48–51]. Ma et al. designed a parallel dual-channel attention module to enhance the effect of different channels on the feature map[52]. Chen et al. introduced the SE module into local importance pooling module to enhance the sensitivity of the locally important pooling to channel characteristics. Chen et al. introduced the coordination attention (CA) module, replacing the backbone network's spatial pyramid pooling (SPP) structure[53]. This change further factorizes the pooling operation and effectively enhances the network's ability to locate defects. Xiao et al. introduced the coordinate attention mechanism (CA) into YOLOv7 to enhance the feature extraction capability of the model[54]. Zheng add the dynamic head block (Dyhead Block) to the detection head, resulting in a target detection head with attention to perform classification and regression tasks[55].

In this paper, we used an efficient multi-scale attention (EMA) module[56], placing it after the second convolutional (Conv) module in the backbone. Secondly, we proposed a new convolutional attention mechanism module, named CISBA module, placing it between backbone and head. Above operations further enhance the model's feature extraction capability and small object detection.

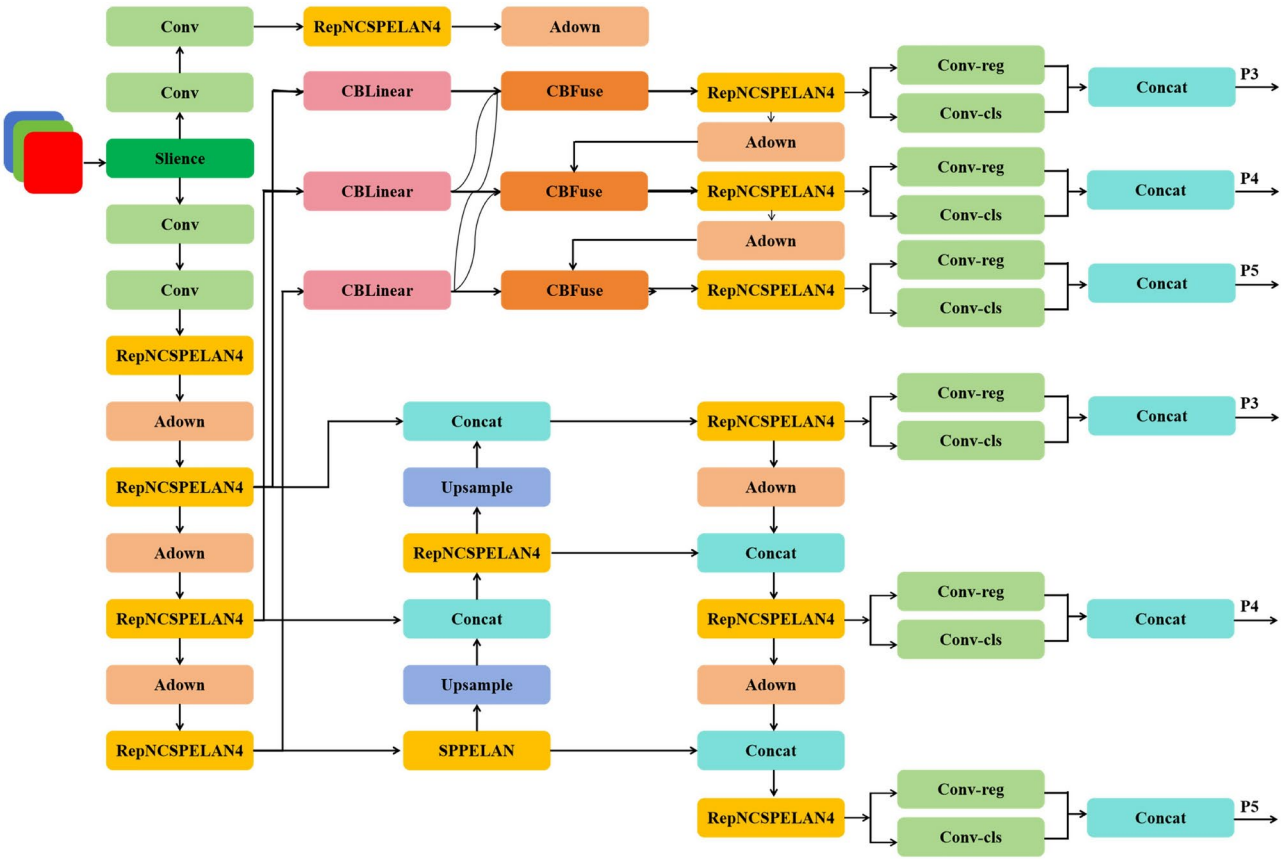

Fig. 1. The structure of YOLOv9c.

Convolution

Convolution-based object detection algorithms show a high level of automation in defect detection but still face challenges such as low accuracy in detecting small defects and interference from complex backgrounds. The convolutional modules in these algorithms are limited in addressing these issues, mainly due to their sensitivity to local features and insufficient capture of contextual information. Therefore, improving these convolutional modules or introducing new convolutional modules can effectively enhance the model's ability to identify critical defect features, thus increasing detection accuracy and robustness. Su et al. combined the depthwise separable convolution and Resnet-34 network structure to improve the feature extraction backbone network. The introduction of the depthwise separable convolution is to reduce the network parameters[57,58]. Li et al. adopted the PConv-based Fusion Faster module as a fundamental operator, which enhances the feature-extraction ability of shallow networks while maintaining speed[59].

In this paper, we designed two types of pyramid convolution (PyConv) modules[60]: replacing the second convolutional (Conv) module in the backbone with a three-layer PyConv (PyConv3) module and the second Conv module in the head with a two-layer PyConv (PyConv2) module. The model's ability of multi-scale object detection can further improve by using different pyramid convolution with different kernel size.

Non-maximum suppression, NMS

Non-Maximum Suppression (NMS)[61] has very important applications in the field of computer vision, such as object tracking, data mining, 3D reconstruction, object recognition, and others. NMS can be understood as a local maximum search method: it retains the maximum elements by supressing non-maximum elements. In the YOLO series of object detection algorithms, the purpose of NMS is to retain the optimal detection results from multiple possible detected objects and suppress other suboptimal overlapping detection boxes, thereby reducing redundant detections. The specific steps are as follows:

1. For each category, sort the predicted boxes by confidence scores and select the one with the highest confidence as the reference.
2. Remove the reference box, then select the box with the largest overlap area with the reference box from the remaining predicted boxes. If the overlap area exceeds a certain threshold, delete it.
3. Repeat step 2 for the remaining predicted boxes until all overlap areas are below the threshold or there are no boxes left to delete.

The NMS algorithm improves the accuracy of object detection algorithms by suppressing highly overlapping boxes. However, NMS algorithm has two issues: When two object boxes are close to each other, the box with the lower score may be removed due to a large overlap area with the higher-scoring box; We need manually set the threshold for NMS, if the threshold too low, it may miss detections, and if the threshold too high, it may result in false positives. To alleaviate above problems, we used Soft-NMS[3] to replace traditional NMS.

Based on the aforementioned four strategies, we propose an algorithm named EPSC-YOLO. The detailed structure of EPSC-YOLO is illustrated in Fig. 2.

Methods

Overall structure

The diagram in Fig. 2 shows the overall structure of the EPSC-YOLO algorithm. Firstly, the efficient multi-scale attention was introduced into the backbone after original Conv module. Secondly, we designed two types of pyramidal convolution instead of oringal Conv module. Thirdly, we proposed a new convolutional attention mechanism module, named CISBA module, to enhance the detection capability of small targets in complex backgrounds by placing it between backbone and neck. Lastly, we used Soft-NMS instead of NMS to further improve the accuracy of object detection.

Efficient multi-scale attention (EMA) module

Traditional convolution, due to its fixed size of receptive field, struggles to simultaneously capture the details of small objects and the global features of large objects, which impacts its effectiveness in multi-scale object detection. Besides, convolution primarily relies on local feature extraction and has difficulty capturing long-range dependencies and global contextual information between objects, limiting its ability to detect occluded or dense targets in complex scenes. In contrast, attention mechanisms dynamically adjust the weights of features, enhancing the capture of global information and long-range dependencies, adaptively focusing on important features, and effectively handling multi-scale objects, thereby addressing the shortcomings of convolution in capturing global information and processing multi-scale features. So, we introduced an attention mechanism, named EMA (Efficient Multi-Scale Attention Module). EMA module can avoid more sequential processing and large depth by parallel substructures. The structure is shown in Fig. 3.

According to Fig. 3, the EMA module first applies parallel horizontal and vertical pooling operations to perform adaptive average pooling on the height and width of the input feature map, thereby extracting multi-scale spatial information. Next, a 1×1 convolution is used to fuse the pooled features, and a decomposition operation separates the fused features into components related to height and width. This parallel processing approach allows the model to capture information from different scales at the same level, avoiding the complexity of sequential processing and thereby improving computational efficiency. To further stabilize the training process, the EMA module uses Group Normalization (GN) to normalize the features within each group, ensuring that even with a shallow network, effective training can still be achieved. The module then uses a 3×3 convolution to further refine the features, capture local spatial information, and enhance the model's local perception capabilities. Next, the EMA module computes global contextual information for each feature map

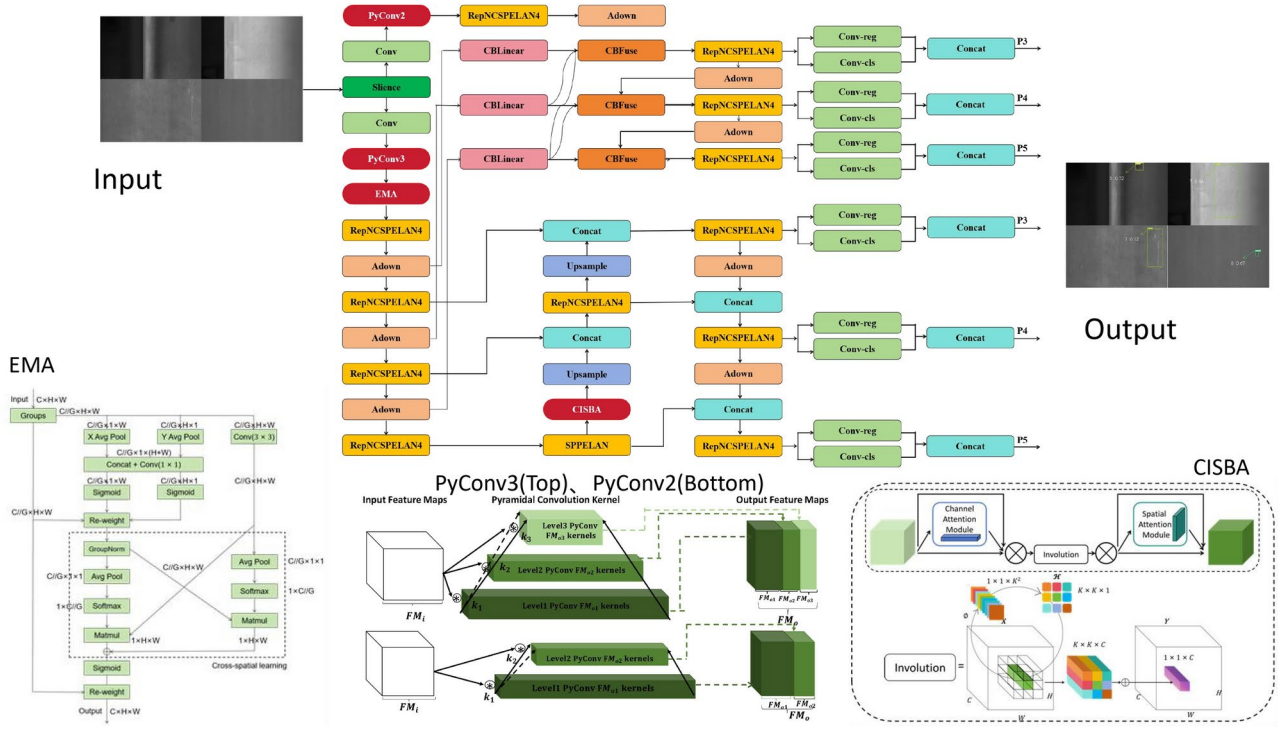

Fig. 2. The structure of EPSC-YOLO.

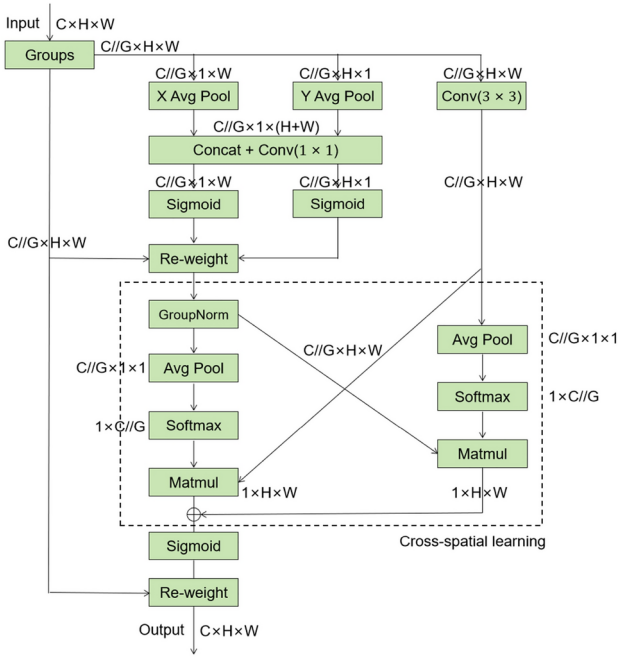

Fig. 3. The structure of EMA, where "G" means the divided groups, "X Avg Pool" represents the 1D horizontal global pooling and "Y Avg Pool" represents the 1D vertical global pooling.

through adaptive average pooling, allowing it to weight the spatial information of each channel. Then, Softmax is applied to calculate spatial attention weights, enabling the model to automatically learn how to weight spatial positions across multi-scale and multi-channel levels. Through this series of designs, the EMA module efficiently extracts multi-scale features with fewer network layers, improving computational efficiency and enhancing the model's expressive power. By combining YOLOv9 with EMA module for defect detection, the model becomes

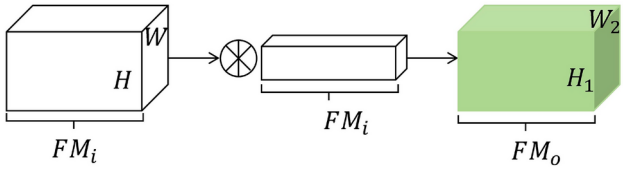

Fig. 4. Standard Convolution.

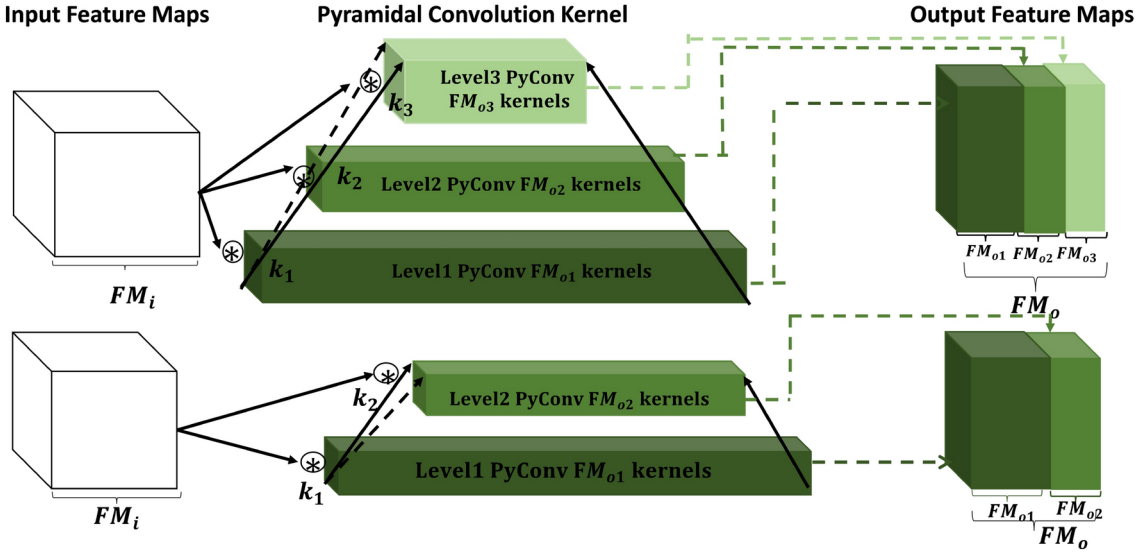

Fig. 5. Pyramidal convolution with two layers(bottom) and three layers(top).

more robust in complex industrial environments, especially in cases where defects vary significantly in size and position, providing more accurate detection results.

Pyramidal convolution (PyConv)

The Standard Conv module (Conv2d + Batch Normalization + Activation Function) is usually used for feature extraction. The structure shown in Fig. 4.

Standard convolution (Conv2d) is limited by fixed kernel size and stride, restricting its receptive field and making it difficult to fully capture multi-scale information, while also being less efficient. Therefore, we used the pyramid convolution, pyramid convolution utilizes the principle of grouped convolution, dividing the input features into multiple groups, with each group using different sizes of convolutional kernels to extract multi-scale features, the kernel size gradually increases from the bottom to the top, while the depth decreases, which design could better capture of both fine details and global features.

Compared to standard convolution, pyramid convolution has fewer parameters and lower computational complexity. For standard convolution, the parameters are $K_1 \cdot K_1 \cdot FM_o$, the FLOPs are $K_1 \cdot K_1 \cdot FM_i \cdot W_1 \cdot H_1 \cdot FM_o$. For pyramidal convolution with layers, the parameters are $k_1^2 \cdot FM_i \cdot FM_{o1} + k_2^2 \cdot \left(\frac{FM_i}{\left(\frac{k_2^2}{k_1^2}\right)} \right) \cdot FM_{o2} + \cdots + k_n^2 \cdot \left(\frac{FM_i}{\left(\frac{k_2}{k_2^2}\right)} \right) \cdot FM_{on}$, the FLOPs are $k_1^2 \cdot FM_i \cdot FM_{o1} \cdot (W \cdot H) + k_2^2 \cdot \left(\frac{FM_i}{\left(\frac{k_2^2}{k_1^2}\right)} \right)$.

In this paper, we designed two types of pyramid convolution to address the limitations of receptive field coverage and computational efficiency. By leveraging their flexibility and parallelism, these convolutions enhanced the model's feature extraction capabilities and improved detection performance. The structure of two types of pyramid convolutions shown in Fig. 5.

Soft-NMS

Soft-NMS is an improved algorithm based on NMS, it adopt a soft method, see the green box in Fig. 6. Soft-NMS does not directly delete all boxes with an IoU greater than the threshold; instead, it reduces the confidence scores of all boxes with an IoU greater than the threshold[62].

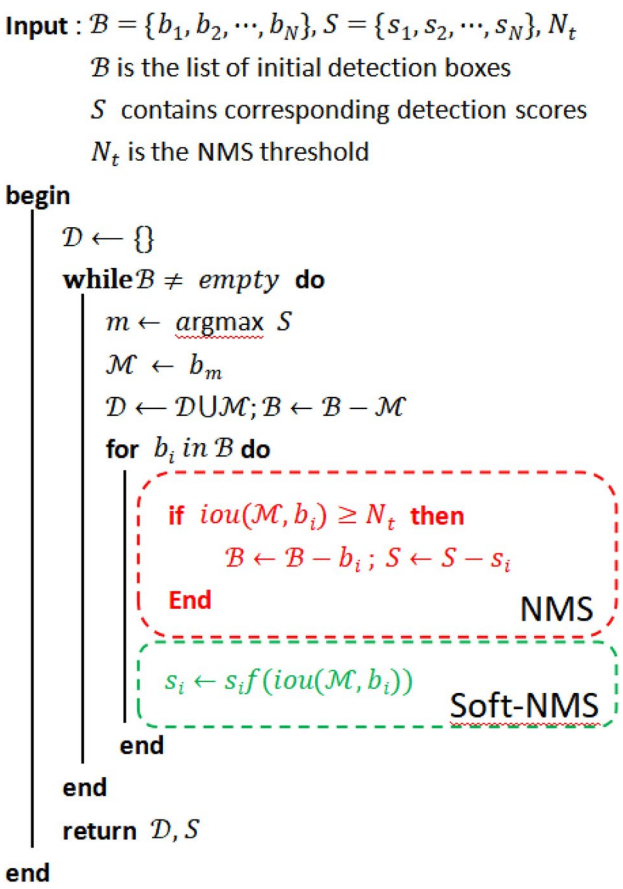

Fig. 6. SoftNMS.

The specific steps of Soft-NMS are as follows:

1. Sort candidate boxes: Sort the detected candidate boxes in descending order based on confidence (usually classification scores), prioritizing those with higher confidence.
2. Select the highest-confidence box: From the sorted list of candidate boxes, choose the one with the highest confidence as the current box and add it to the final result list.
3. Compute IoU (Intersection over Union): For the remaining candidate boxes, calculate their IoU (Intersection over Union) with the current box M.
4. Update confidence scores: For each candidate box with an IoU greater than a certain threshold with the current box M (i.e., boxes with significant overlap), do not remove them directly, but gradually reduce their confidence through the following method: Adjust the confidence score of each overlapping candidate box according to different decay functions. The commonly used decay functions in Soft-NMS include: Linear decay:

$$s_i = s_i * (1 - IOU(M, B_i)). \tag{1}$$

Gussian decay:

$$s_i = s_i * e^{\frac{-IOU(M,B_i)^2}{\sigma}}. \tag{2}$$

where s_i is the original confidence score of the candidate box, $IOU(M, B_i)$ is the IoU value between the current box M and the candidate box B_i, and σ is the parameter that control the decay rate.

5. Remove low-confidence boxes: Eliminate candidate boxes with confidence scores below the threshold from the list.
6. Repeat the above steps: Select the new highest-confidence box from the remaining candidate boxes and repeat steps 3-5 until all candidate boxes have been processed.
7. Output results: The remaining candidate boxes are output as the final detection results.

Soft-NMS improves detection performance by retaining more potential objects through the confidence decay mechanism, without affecting localization accuracy. Based on the advantages of Soft-NMS, we replaced the traditional NMS in the YOLOv9 algorithm with Soft-NMS. By smoothly decaying the confidence scores of overlapping boxes, YOLOv9 with Soft-NMS effectively avoids the information loss problem caused by traditional NMS in cases of highly overlapping bounding boxes and resolves the issue of missed detections when multiple objects are closely adjacent or overlapping.

CISBA module

CBAM (shown in Fig. 7) is an attention mechanism module that combines spatial attention and channel attention[63].

Although it can adapt the importance of each channel in the feature map through the channel attention module, highlights key local regions in whole images still room need to improve, thereby this paper proposes a new module, named CISBA module, the structure shown in Fig. 8.

From Fig. 8, the CISBA module integrates three core mechanisms: the channel attention mechanism, the Involution operation, and the spatial attention mechanism. These mechanisms optimize the input feature map at their respective levels, thereby significantly enhancing the model's feature extraction capabilities.

Firstly, the channel attention mechanism refines the channel dimension of the feature map by assigning different weights to each channel. It captures the global features of each channel using average pooling and max pooling techniques, subsequently generating channel weights via a convolutional layer. These weights are then used to adjust the contribution of each channel, enhancing the features of the most discriminative channels and enabling the network to focus on more valuable information. In the CISBA module, the channel attention mechanism is applied to the input feature map initially, ensuring that the network prioritizes the most informative channels during subsequent processing. To preserve the integrity of the original information, residual connections are incorporated, allowing the input features to be directly added to the output of the channel attention mechanism. This helps prevent information loss due to excessive processing. Next, the Involution operation enhances the representation of local features by dynamically generating convolutional kernels. Unlike traditional fixed convolutional kernels, Involution adaptively generates kernels based on the input feature map. These dynamically generated kernels are more flexible in capturing local spatial relationships, particularly when processing complex local features, thus demonstrating higher accuracy and efficiency. The Involution operation generates convolutional kernel weights through a convolutional layer, then performs unfolding, weighting, and summing operations on the input feature map to produce an optimized local feature map. In the CISBA module, the Involution operation receives the feature map weighted by the channel attention mechanism, further improving its local spatial representation. To ensure the preservation of original feature information, residual connections are again introduced, adding the original input features directly to the result after Involution. This facilitates more effective gradient propagation during network training.Finally, the spatial attention mechanism optimizes the spatial distribution of the feature map by assigning weights to spatial locations based on their importance. It begins by performing pooling operations on the input feature map, extracting feature information for each spatial location via average pooling and max pooling, respectively. A spatial attention map is then generated through a convolutional layer. This attention map represents the weights of each spatial location, which, after being processed by a Sigmoid activation function, is element-wise multiplied with the original feature map to highlight the features of critical regions while suppressing the influence of irrelevant ones. In the CISBA module, the spatial attention mechanism is applied following the Involution operation, further enhancing the feature map at the spatial level. As with the previous mechanisms, residual connections are crucial in this

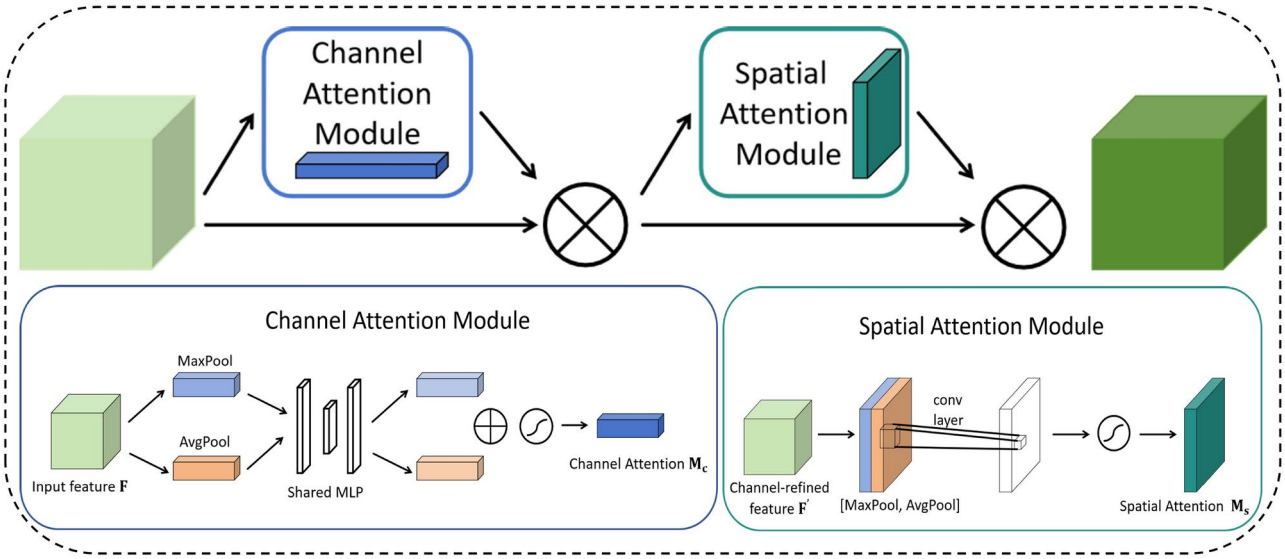

Fig. 7. The structure of CBAM.

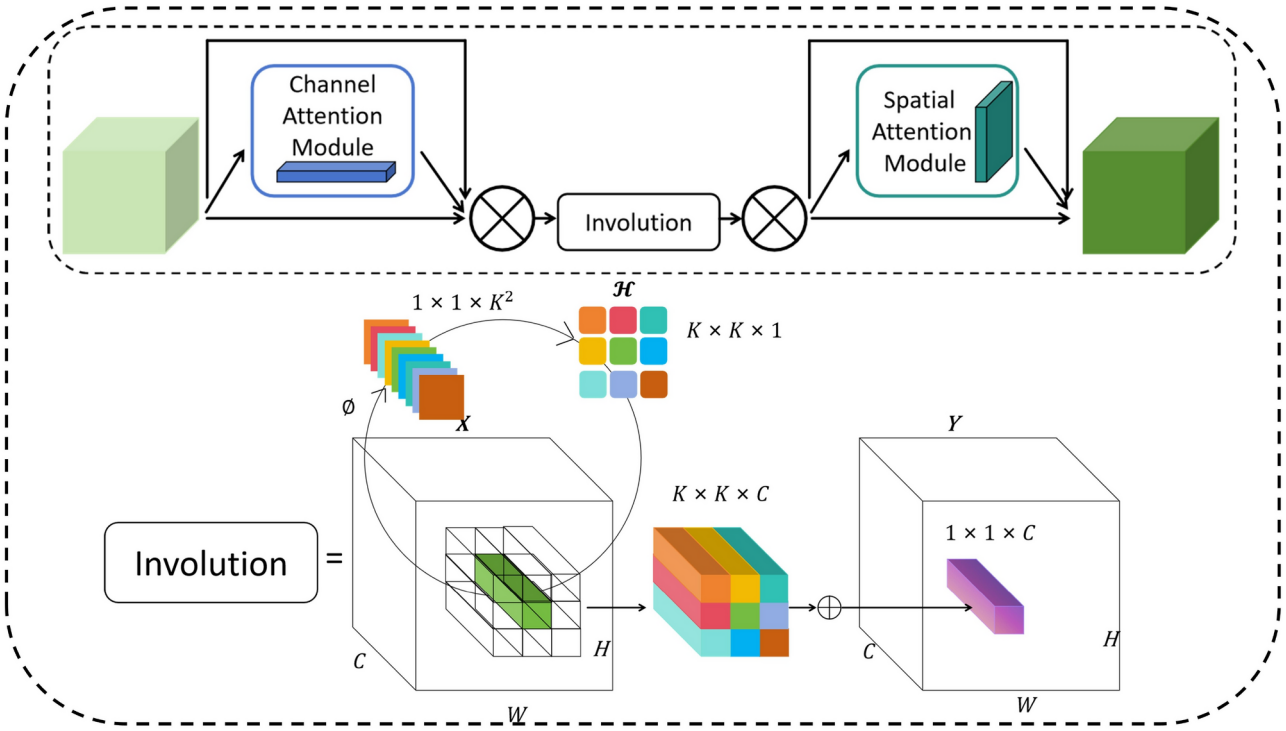

Fig. 8. The structure of CISBA Module.

stage, linking the output of the spatial attention mechanism to the feature map after Involution. This ensures that the original feature information is not discarded and supports stable model training and convergence.

By integrating these mechanisms, the CISBA module achieves optimization of the feature map across multiple levels, significantly improving the model's expressive capacity and overall performance.

Experiments and results

Experimental environment

In this paper, the experiments were conducted on a Linux system with the experimental environment consisting of Python 3.8.18, PyTorch 1.13.1, and CUDA 11.7. All models were trained using distributed training across eight NVIDIA A30 24GB GPUs. During the training process, the number of epochs was set to 300, and the batch size was 64. The SGD optimizer was used, with an initial learning rate of 0.01. The input image sizes were all resized to 640 × 640 pixels. Additionally, all networks in the experiments utilized the official pre-trained weights.

Evaluation indicator

In this paper, we selected four main metrics: (i) P (the ratio of the samples as correct by the algorithm to the total number of samples) (ii) R (the ratio of positive samples in a sample which are predicted to be right) (iii) mAP_{50}^{val} (the mean average precision at an IoU (Intersection over Union) threshold of 0.5) (iv) $mAP_{50:95}^{val}$ (the average means average precision over the IoU threshold from 0.5 to 0.95).

$$P = \frac{P_T}{P_T + P_F}. \tag{3}$$

$$R = \frac{P_T}{P_T + N_F}. \tag{4}$$

$$AP = \int_0^1 P_r(R_e)\, dR. \tag{5}$$

$$\text{mAP} = \frac{\sum_{i=1}^{C} AP}{C} \tag{6}$$

where P_T denotes the positive samples which are assigned correctly; P_F denotes the positive samples which are assigned incorrectly; N_F denotes the negative samples which are assigned incorrectly. C denotes the number of classes.

Categorie	NEU-DET	GC10-DET
0	Crazing	Waist folding
1	Inclusion	Pu
2	patches	WI
3	Pitted_surface	Cg
4	Rolled-in_scale	Water Spot
5	Scratches	Oil Spot
6	–	Silk Spot
7	–	Inclusion
8	–	Rolled Pit
9	–	Crease

Table 1. The categories of each dataset.

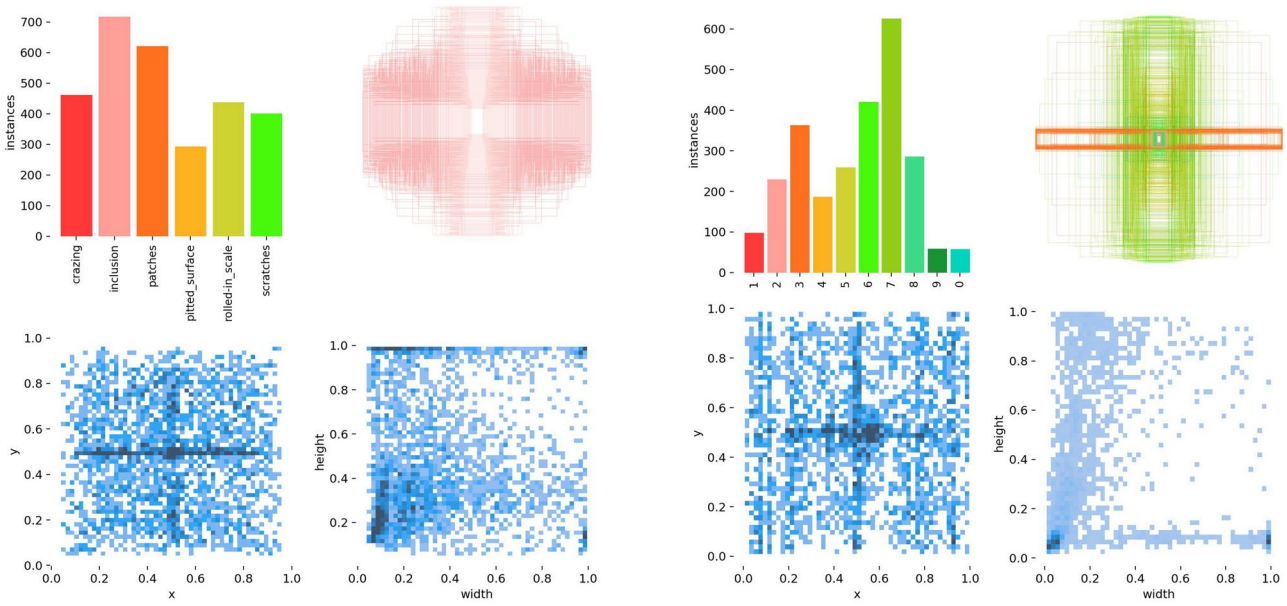

Fig. 9. Details of the training sets for the two datasets.

Experimental datasets

To validate the effectiveness of the proposed improved model, we conducted experiments on two public datasets: NEU-DET and GC10-DET.

1. NEU-DET

NEU-DET (Northeastern University Surface Defect Database for Detection) is a surface defect detection dataset developed by Northeastern University in China. It includes six typical types of defects found on steel surfaces, with 300 images for each type. Each image has a resolution of 200×200 pixels[64].

2. GC10-DET

The GC10-DET (GongChang Surface Defect Database for Detection) dataset consists of 14,300 images depicting steel surface defects from real industrial scenarios. It covers 10 common types of metal surface defects, with approximately 1,430 images for each type. Each image has a resolution of 256×256 pixels[65].

The datasets were split into training, validation, test sets in a 7:2:1 ratio and converted labels to numerical values, see Table 1 for details.

In order to further shown the details of the training sets,such as the sizes and categories et.al. We visualized it shown in Fig. 9.

Dataset	NEU-DET			
Metric	*P*	*R*	mAP_{50}^{val}	$mAP_{50:95}^{val}$
YOLOv9c	75.9	71.2	75.6	44.9
+EMA	75.3	73.2	75.6	45.2
+PyConv	75.4	74.1	78.7	46.5
+SoftNMS	73.8	74.4	77.9	49.5
+CISBA	**74.5**	**74.3↑3.1%**	**77.6↑2%**	**50↑5.1%**

Table 2. Ablation experiments on the NEU-DET dataset.

Dataset	GC10-DET			
Metric	*P*	*R*	mAP_{50}^{val}	$mAP_{50:95}^{val}$
YOLOv9c	71.5	69.9	70.6	37.5
+EMA	72.4	71.1	73.8	39.8
+PyConv	71	67.3	69.2	36.2
+SoftNMS	70.2	72.6	74.2	39
+CISBA	**74.5↑3%**	**70.1↑0.2%**	**73↑2.4%**	**39.9↑2.4%**

Table 3. Ablation experiments on the GC10-DET dataset.

Figure 9 shows the number of instances for each category in the training datasets of the two datasets (top left), the size and number of bounding boxes (top right), the center coordinates of the bounding boxes (bottom left), and the height and width of the bounding boxes (bottom right).

Ablation experiments

In this section, we conduced five sets of ablation experiments under the same environmental to validate the effectiveness of each improved operation, including introduce EMA, design PyConv3, PyConv2 and CISBA modules, and use the SoftNMS instead of NMS in the YOLOv9c algorithm on the NEU-DET dataset and GC10-DET dataset, the results shown in Tables 2 and 3, bold indicates the results of our improved model.

From Table 2, first, used the EMA module in backbone network after the second Conv module, the metric *P* of the network was 75.3, representing a slight decrease of 0.6% compared with the original model. However, the metric *R* and were increased by 2% and 0.3%, respectively. Second, the original Conv module in backbone network and head were replaced by PyConv2 (pyramidal convolution with two layers) and PyConv3 (pyramidal convolution with three layers). Based on the incorporation of the EMA module, we introduced pyramid convolution, and the results showed that all four evaluation metrics of the model improved. Notably, the mAP_{50}^{val} increased by 3.1%, significantly enhancing the model's detection capabilities while also validating the effectiveness of the pyramid convolution. Third, we used SoftNMS instead of NMS. The results in Table 2 indicate that the $mAP_{50:95}^{val}$ has a significant improvement. This suggests that the model's ability to detect small targets has been further enhanced, which also validates the effectiveness of this operation. Finally, we used new module, CISBA. The *P*, mAP_{50}^{val} and $mAP_{50:95}^{val}$ improved, indicating that the model's ability to detect small targets has been enhanced, which also demonstrates the effectiveness of this module.

According to the Table 3, our improvements are effective, each improvement has increased the model's detection capability to some extent. Comparisons with baseline algorithm, YOLOv9c, the four metrics *P*, *R*, mAP_{50}^{val} and $mAP_{50:95}^{val}$ of our algorithm were increased by 3%, 0.2%, 2.4% and 2.4%, respectively.

To further validate the effectiveness of the proposed algorithm, we visualized representative detection result images, as shown in Figs. 10 and 11, Figure 10 shows representative detection result images from two datasets. The first row is for the NEU-DET dataset, and the second row is for the GC10-DET dataset. From left to right, the images are: the original image (A(1), B(1)), the detection result of the original model (A(2), B(2)), and the detection result of the improved algorithm (A(3), B(3)). Figure 11 visually compares the detection results on the NEU-DET and GC10-DET datasets, divided into two display groups (left and right). In each group, the original images (Fig. 11(A1),(B1)), detection results using the YOLOv9c (Fig. 11(A2),(B2)), and detection results using the proposed EPSC-YOLO algorithm (Fig. 11(A3),(B3)) are shown in sequence.

According to the Fig. 10, in the NEU-DET dataset, the improved EPSC-YOLO algorithm detect the inclusion missed by YOLOv9c (see Fig. 10(A3)), the confidence score of EPSC-YOLO is higher than YOLOv9c. In the GC10-DET dataset, althongh the confidence score of EPSC-YOLO is not higher than YOLOv9c, but it close to YOLOv9c and the improved EPSC-YOLO algorithm detects Oil Spot missed by YOLOv9 (see Fig. 10(B3)).

According to the Fig. 11, in the NEU-DET dataset, the improved EPSC-YOLO algorithm detect the scratch missed by YOLOv9c (see purple box in Fig. 11(A3)), in the GC10-DET dataset, the improved EPSC-YOLO algorithm detects Silk Spot missed by YOLOv9 (see red box in Fig. 11(B3)). Besides, the confidence score of EPSC-YOLO is higher than YOLOv9c in two datasets.

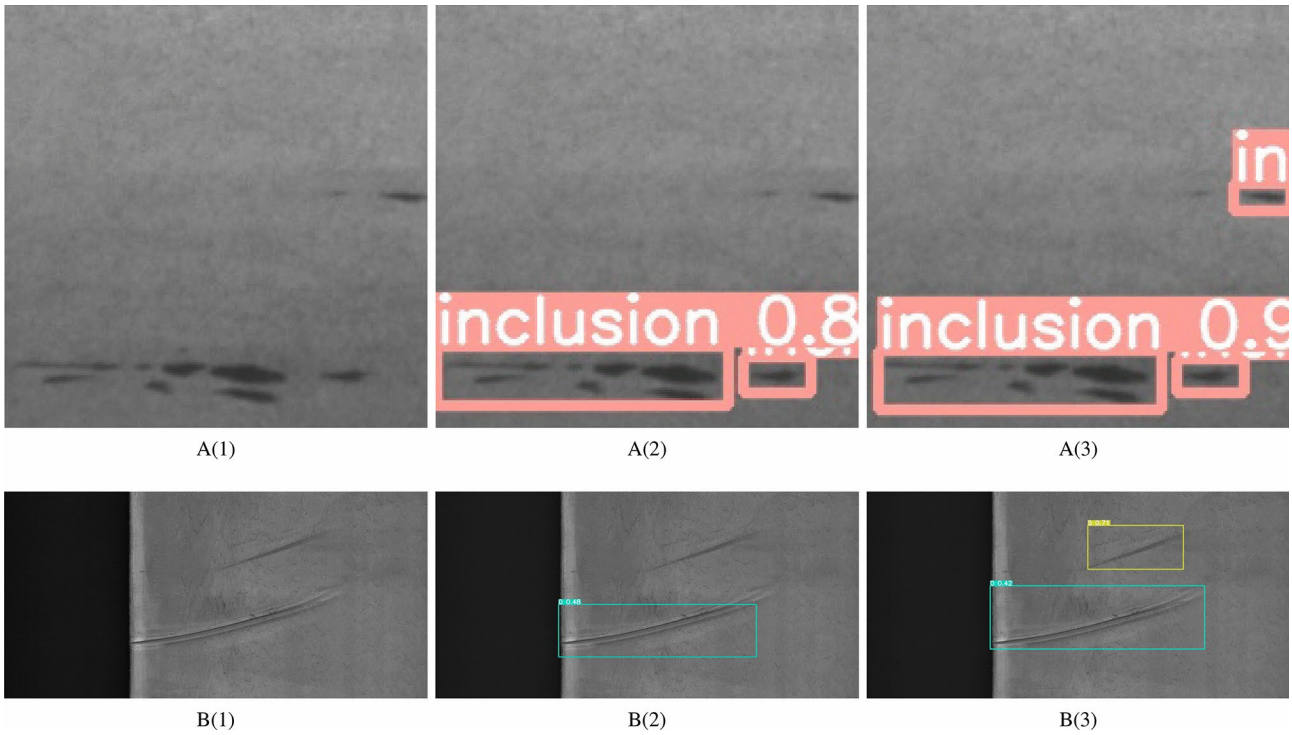

Fig. 10. Original image and visualization results of YOLOv9c and EPSC-YOLO. A(1-3): NEU-DET, B(1-3): GC10-DET.

Based on the analysis of Figs. 10 and 11, the improved algorithm not only demonstrates significant advantages in detecting small defects but also provides broader and more comprehensive coverage in detection and the performance of our proposed algorithm is better than original algorithm

Comparative experiments and results

To further validate the detection performance of the improved algorithm proposed in this paper, a comparative analysis was conducted. The proposed object detection algorithm was compared with various classic and improved algorithms, including seven classic methods (such as YOLOv5, Gold-YOLO, etc.) and five improved methods (such as the modified YOLOv5 algorithm proposed by Li et al[66]). During the model validation process, we compared the and performance of different methods over 300 training epochs. Comparisons with classic detection algorithms, the results are shown in Tables 4 and 5 and visualization results in Fig. 12. Comparisons with other improved detection algorithms, the results are shown in Tables 6 and 7.

Tables 4 and 5 respectively show the performance of different YOLO models on the NEU-DET and GC10-DET datasets.

According to the data in Table 4, in terms of *P*, EPSC-YOLO's *P* is 74.50, which performs moderately compared to other YOLO models. In terms of precision, YOLOv8 leads with a *P* of 81.00, demonstrating the strongest false positive suppression ability, ensuring the accuracy of the detection results. Following closely is YOLOv3 with a *P* of 78.00, which also performs strongly and has good accuracy. YOLOv9 and YOLOv10 have *P* values of 75.90 and 76.10, respectively, and still perform excellently in tasks requiring high precision. In comparison, YOLOv5 has a *P* of 73.58, which is slightly lower, but it still maintains good precision, suitable for various application scenarios. YOLOv6 and Gold-YOLO have precision values of 41.37 and 42.57, respectively. The lower precision suggests that they may generate more false positives, limiting their use in high-precision detection tasks.

In terms of recall (*R*), EPSC-YOLO stands out with a value of 74.30, demonstrating its exceptional ability to capture defects. YOLOv5 follows closely with an *R* of 73.70, also performing excellently and able to capture most defects. YOLOv9 and YOLOv10 have *R* values of 71.20 and 70.60, respectively, which are slightly lower but still perform well and are suitable for various defect detection scenarios. YOLOv8 has an *R* of 70.10, slightly lower than the other models but still possesses strong defect-capturing ability. Gold-YOLO and YOLOv6 have *R* values of 66.43 and 62.47, respectively, showing poorer performance, possibly failing to capture all defects, which limits their application in complex tasks.

In terms of mAP_{50}^{val}, EPSC-YOLO leads with a mAP_{50}^{val} of 77.60, demonstrating its excellent performance at IoU = 0.5. YOLOv10 follows with a mAP_{50}^{val} of 76.30, ranking second with stable performance and high precision. YOLOv9 and YOLOv8 have mAP_{50}^{val} values of 75.60 and 75.30, respectively, providing stable precision at IoU = 0.5, making them suitable for surface defect detection tasks. Gold-YOLO's mAP_{50}^{val} is 75.27, which is stable but slightly lower than YOLOv8 and YOLOv9. YOLOv5 has a mAP_{50}^{val} of 74.23, which is slightly lower than other high-precision models but still offers good detection capability. YOLOv6 has a mAP_{50}^{val} of

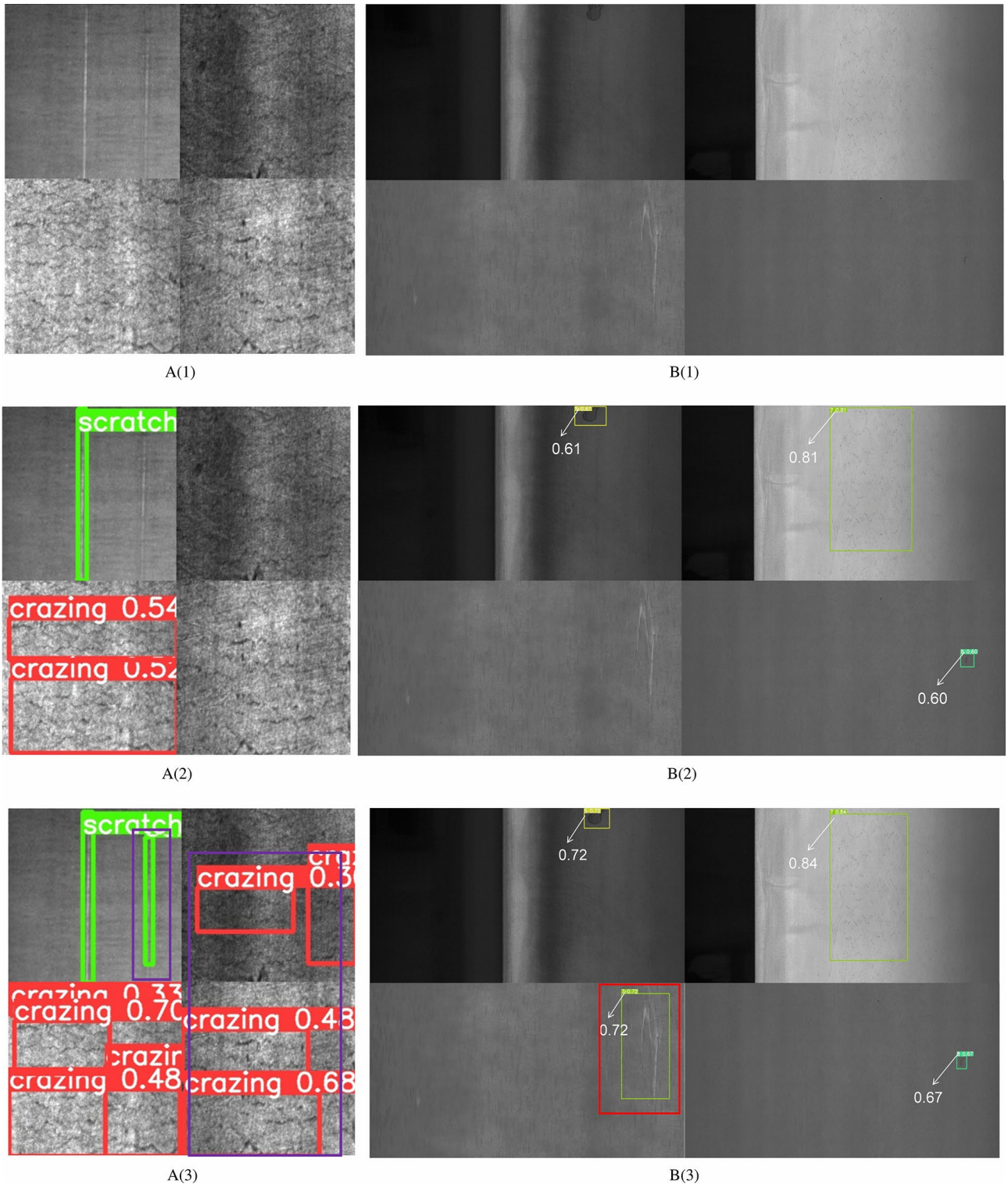

Fig. 11. Original image and visualization results of YOLOv9c and EPSC-YOLO. A(1-3): NEU-DET, B(1-3): GC10-DET.

73.37, which, while maintaining some precision, is noticeably lower than other advanced models, leading to suboptimal performance at IoU = 0.5. YOLOv3 has the lowest mAP_{50}^{val} at 72.50, indicating that its precision at mid-low IoU conditions is relatively low, potentially missing some key defects.

For the $mAP_{50:95}^{val}$ metric, EPSC-YOLO again stands out with a value of 50, far surpassing other models, indicating its ability to maintain high detection precision across multiple IoU thresholds. YOLOv8 and YOLOv9 both have a $mAP_{50:95}^{val}$ of 44.90, showing good performance across multiple IoU thresholds and providing

Models	P	R	mAP_{50}^{val}	$mAP_{50:95}^{val}$
YOLOv3	**78.00**	68.50	72.50	40.60
YOLOv5	73.58	73.70	74.23	41.48
YOLOv6	41.37	62.47	73.37	41.37
YOLOv8	81.00	70.10	75.30	44.90
Gold-YOLO	42.57	66.43	75.27	42.57
YOLOv9	75.90	71.20	75.60	44.90
YOLOv10	76.10	70.60	76.30	44.80
EPSC-YOLO (ours)	74.50	**74.30**	**77.60**	**50**

Table 4. Comparisons with other classic detection models on NEU-DET dataset.

Models	P	R	mAP_{50}^{val}	$mAP_{50:95}^{val}$
YOLOv3	68.50	69.50	69.90	35.10
YOLOv5	**75.63**	61.97	68.73	34.80
YOLOv6	31.73	53.03	65.00	31.73
YOLOv8	74.30	68.80	70.50	36.50
Gold-YOLO	33.07	55.63	66.03	33.07
YOLOv9	71.50	69.90	70.60	37.50
YOLOv10	68.80	65.60	68.50	34.50
EPSC-YOLO (ours)	74.50	**70.10**	**73**	**39.9**

Table 5. Comparisons with other classic detection models on GC10-DET dataset.

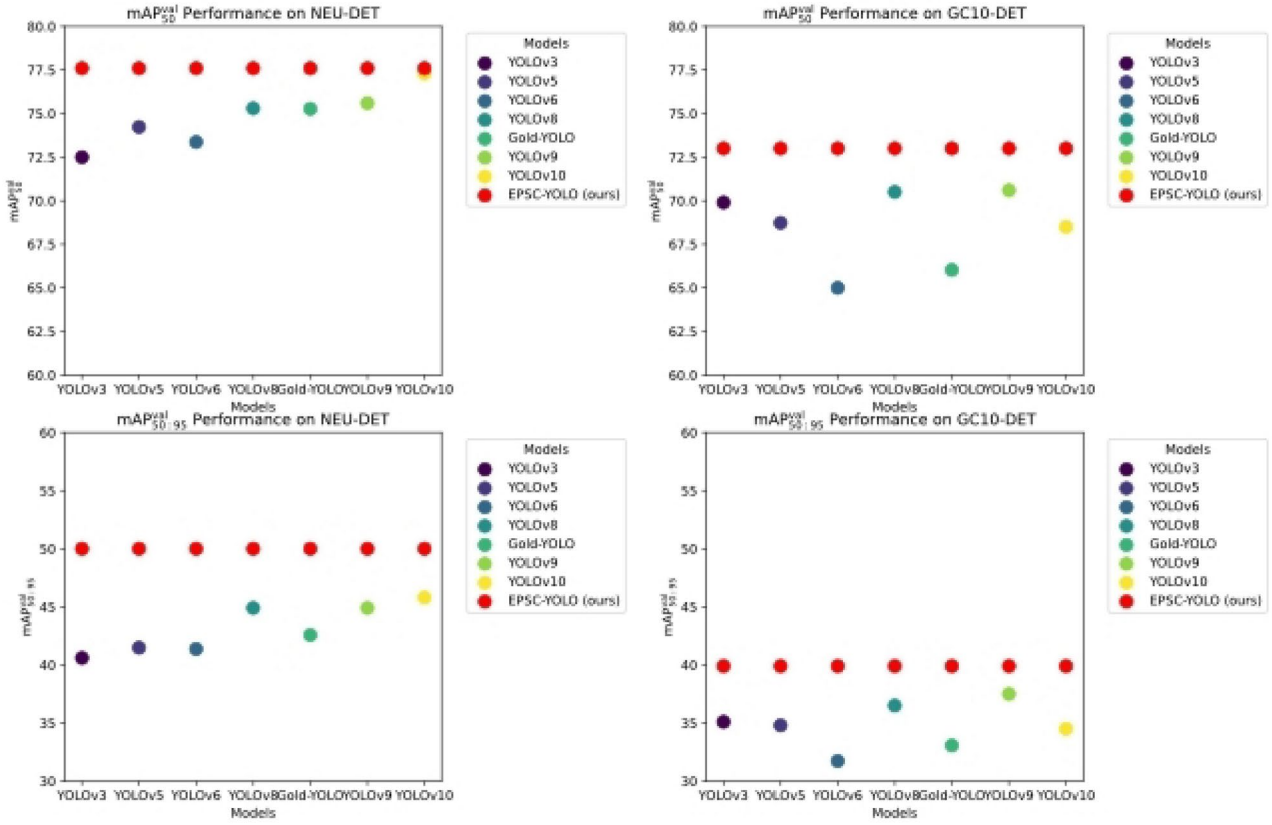

Fig. 12. Comparison of EPSC-YOLO with Other Models.

Models	mAP_{50}^{val}	$mAP_{50:95}^{val}$
Improved-YOLOv5[66]	73.08	37.57
Regularized YOLO[67]	80.77	47.62
MSFT-YOLO[68]	75.2	—
Improved Multi-Scale YOLO-v5[69]	72	37.2
Kou's[70]	72.2	—
EDNN[65]	72.4	—
YOLO-LFPD[71]	81.2	—
WFRE-YOLOv8s[72]	79.4	—
EPSC-YOLO (ours)	**77.6**	**50**

Table 6. Comparisons with other classic detection models on NEU-DET dataset.

Models	mAP_{50}^{val}	$mAP_{50:95}^{val}$
Kou's[70]	71.3	—
EDNN[65]	65.1	—
YOLO-LFPD[71]	72.8	—
WFRE-YOLOv8s[72]	69.4	—
Improved-YOLOX[73]	70.5	—
LSD-YOLOv5[74]	67.9	—
EPSC-YOLO (ours)	**73**	**39.9**

Table 7. Comparisons with other classic detection models on GC10-DET dataset.

stable detection capabilities. YOLOv10's $mAP_{50:95}^{val}$ is 44.80, slightly lower than YOLOv8 and YOLOv9, but still maintains strong detection abilities. YOLOv5's $mAP_{50:95}^{val}$ is 41.48, showing a decrease in performance at higher IoU thresholds. YOLOv6 and Gold-YOLO have $mAP_{50:95}^{val}$ values of 41.37 and 42.57, respectively, and fail to maintain high precision at higher IoU thresholds, resulting in subpar performance in complex detection tasks.

According to the data shown in Table 5, in terms of precision (*P*), EPSC-YOLO has a precision of 74.50, which is moderate and places it in the upper range among all models. YOLOv5 has a *P* of 75.63, slightly higher than EPSC-YOLO, showing its advantage in reducing false positives. YOLOv9 has a *P* of 71.50, ranking third. Although slightly lower than YOLOv5, it still has good false positive suppression capabilities. YOLOv8 and YOLOv10 have *P* of 74.30 and 68.80, respectively, indicating that their performance in reducing false positives is relatively similar, especially YOLOv8, which performs relatively well. YOLOv3 and Gold-YOLO have precisions of 68.50 and 33.07, respectively, showing poorer performance, especially Gold-YOLO, which has a precision far lower than other models, indicating significant issues in false positive suppression. YOLOv6 has the lowest *P* of 31.73 among all models, which could result in higher false positives.

In terms of recall (*R*), EPSC-YOLO leads all models with a *R* of 70.10, demonstrating its excellent defect detection capability. YOLOv9 and YOLOv8 have recall rates of 69.90 and 68.80, respectively, close to EPSC-YOLO, and also perform well in capturing most defects. YOLOv5 has a *R* of 61.97, slightly lower than the top three, but still maintains strong defect detection capabilities. YOLOv3 has a *R* of 69.50, performing well. YOLOv6 and Gold-YOLO have *R* of 53.03 and 55.63, respectively, which are lower compared to other models, potentially leading to missed detections and affecting their application in complex scenarios.

In the mAP_{50}^{val} metric, EPSC-YOLO stands out with a mAP_{50}^{val} value of 73, ranking first and showing excellent performance at an IoU of 0.5. YOLOv9 has a mAP_{50}^{val} of 70.60, ranking second, performing excellently and suitable for high-precision detection tasks. YOLOv8 has a mAP_{50}^{val} of 70.50, closely following YOLOv9, and performs similarly well. YOLOv5 has a mAP_{50}^{val} of 68.73, slightly lower, but still provides stable detection performance. YOLOv3 and YOLOv10 have mAP_{50}^{val}s of 69.90 and 68.50, respectively, which are lower than other stronger models. Gold-YOLO has a mAP_{50}^{val} of 66.03, relatively low, indicating its precision at IoU = 0.5 is not as good as other models. YOLOv6 has a mAP_{50}^{val} mAP_{50}^{val} of 65.00, and its lower mAP_{50}^{val} value indicates that its detection precision at IoU = 0.5 is poor, affecting its application in surface defect detection.

In the $mAP_{50:95}^{val}$ metric, EPSC-YOLO again performs the best with a $mAP_{50:95}^{val}$ of 39.9, the highest among all models, indicating its ability to maintain high precision across multiple IoU thresholds. YOLOv9 and YOLOv8 have $mAP_{50:95}^{val}$s of 37.50 and 36.50, respectively, following closely and showing strong multi-IoU threshold detection capability. YOLOv5 has a $mAP_{50:95}^{val}$ of 34.80, lower than the three above, but still performs relatively stably, suitable for various detection tasks. YOLOv3 and YOLOv10 have $mAP_{50:95}^{val}$s of 35.10 and 34.50, respectively. Although their performance is average, they can still be applied in tasks with lower precision requirements. Gold-YOLO has a $mAP_{50:95}^{val}$ of 33.07, the lowest among all models, indicating its poor precision across multiple IoU thresholds, which affects its performance in complex tasks. YOLOv6 has a $mAP_{50:95}^{val}$ of 31.73, the lowest among all models, showing its weak detection ability under multiple IoU conditions.

According to Fig. 12, the mAP_{50}^{val} and $mAP_{50:95}^{val}$ of our method are higher than seven classic detection algorithms, which demonstrate that the proposed improved algorithm performs excellently across different datasets, significantly enhancing detection accuracy.

Tables 6 and 7 present a performance comparison between the EPSC-YOLO model and other classical object detection models on the NEU-DET and GC10-DET datasets.

According to the mAP_{50}^{val} performance on the NEU-DET dataset in Table 6, the results of most models are relatively close. Among them, Regularized YOLO achieves the best performance with a mAP_{50}^{val} of 80.77, slightly ahead of YOLO-LFPD, which has a mAP@50 of 81.2, with both models performing similarly. In contrast, EPSC-YOLO (ours) has a mAP_{50}^{val} of 77.6, which, although slightly lower than the first two, still demonstrates strong object detection capabilities, especially with its outstanding performance in $mAP_{50:95}^{val}$. Other models, such as Improved-YOLOv5 and Improved Multi-Scale YOLO-v5, show weaker performance on this metric, with scores of 73.08 and 72, respectively. This suggests that these models may suffer from inaccuracies in bounding box localization or misdetection, which may have impacted their mAP_{50}^{val} scores.

Under the more stringent $mAP_{50:95}^{val}$ evaluation standard, EPSC-YOLO (ours) stands out with a score of 50, significantly ahead of other models. This indicates that EPSC-YOLO has a clear advantage in precise localization and high-quality object detection, especially under high IoU thresholds, where it maintains a high level of detection accuracy. Regularized YOLO follows closely with a score of 47.62, also performing well but slightly lagging behind EPSC-YOLO under strict IoU standards. Improved-YOLOv5 and Improved Multi-Scale YOLO-v5 have relatively low $mAP_{50:95}^{val}$ scores, 37.57 and 37.2, respectively, which suggests that these models have significant deficiencies in high-precision object localization. This may be due to their inability to effectively adjust or optimize the bounding boxes, resulting in their failure to meet the higher precision detection requirements at higher IoU thresholds.

According to the mAP_{50}^{val} performance on the GC10-DET dataset in Table 7, it can be seen that EPSC-YOLO (ours) performs the best with a score of 73, leading other models. This indicates that EPSC-YOLO demonstrates strong object detection capabilities, especially in reliably detecting objects within this dataset. In contrast, the mAP_{50}^{val} of Kou's model is 71.3, which is slightly inferior to EPSC-YOLO but still performs well. Other models, such as EDNN (65.1), YOLO-LFPD (72.8), WFRE-YOLOv8s (69.4), and Improved-YOLOX (70.5), show weaker performance in mAP_{50}^{val}, suggesting that they may have some limitations in object detection capability on the GC10-DET dataset, potentially due to dataset characteristics or inherent model constraints.Under the stricter $mAP_{50:95}^{val}$ evaluation standard, EPSC-YOLO (ours) again stands out with a score of 39.9.

In summary, EPSC-YOLO performs exceptionally well on both the NEU-DET and GC10-DET datasets, especially under high IoU standards, leading other models and demonstrating its strong object detection capability.

Conclusion

In this paper, we propose an improved algorithm for industrial surface defect detection, EPSC-YOLOv9, aimed at addressing the challenges associated with detecting small defects and those with complex shapes. First, we analyze the impact of the convolutional structure in YOLOv9 on the detection of defects with varying shapes and sizes. Based on this analysis, we design two types of pyramid convolution to facilitate the parallel processing of multi-scale targets. Second, to tackle the difficulty of detecting small defects against complex backgrounds, we incorporate the EMA attention mechanism and propose the CISBA module to enhance its effectiveness. Finally, we replace the original NMS with SoftNMS to further improve the model's detection capability.

Experimental results demonstrate that EPSC-YOLO achieves mAP_{50}^{val} values of 77.6% and 73% on the NEU-DET and GC10-DET datasets, respectively, representing improvements of 2% and 2.4% compared to YOLOv9c. The $mAP_{50:95}^{val}$ values are 50% and 39.9%, with an increase of 5.1% and 2.4% on two datasets. Compared to classical object detection models, our model performs exceptionally well across all metrics, particularly in $mAP_{50:95}^{val}$. Additionally, our model shows significant advantages over other improved object detection models. The results indicate that EPSC-YOLO effectively meets the requirements for defect detection in real industrial applications.

In our subsequent work, we will focus on medical image segmentation. We have selected the YOLOv10-N algorithm as our baseline model because it has the fewest parameters. We hope to combine YOLOv10-N with generative adversarial networks to mitigate issues such as poor model generalization caused by an insufficient dataset size.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 6 December 2024; Accepted: 24 February 2025
Published online: 06 May 2025

References

1. Wang, C., Wei, X. & Jiang, X. An automated defect detection method for optimizing industrial quality inspection. *Eng. Appl. Artif. Intell.* **127**, 107387 (2024).
2. Birlutiu, A., Burlacu, A., Kadar, M. & Onita, D. Defect detection in porcelain industry based on deep learning techniques. In *2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 263–270, https://doi.org/10.1109/SYNASC.2017.00049 (2017).
3. Wang, T., Chen, Y., Qiao, M. & Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. *Int. J. Adv. Manuf. Technol.* (2017).

4. Feng, H. et al. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Trans. Instrum. Meas.* **63**, 877–888. https://doi.org/10.1109/TIM.2013.2283741 (2014).
5. Zeng, N. et al. A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Trans. Instrum. Meas.* **71**, 1–14. https://doi.org/10.1109/TIM.2022.3153997 (2022).
6. Yang, M., Yu, K., Zhang, C., Li, Z. & Yang, K. Denseaspp for semantic segmentation in street scenes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3684–3692, https://doi.org/10.1109/CVPR.2018.00388 (2018).
7. Pang, J. *et al.* Libra r-cnn: Towards balanced learning for object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 821–830 (2019).
8. Wang, X., Girshick, R. B., Gupta, A. K. & He, K. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 7794–7803 (2017).
9. Rudolph, M., Wehrbein, T., Rosenhahn, B. & Wandt, B. Fully convolutional cross-scale-flows for image-based defect detection. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1829–1838, https://doi.org/10.1109/WACV51458.2022.00189 (2022).
10. Lv, X., jie Duan, F., jia Jiang, J., Fu, X. & Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors (Basel, Switzerland)***20** (2020).
11. Cheng, X. & Yu, J. Retinanet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Trans. Instrum. Meas.* **70**, 1–11. https://doi.org/10.1109/TIM.2020.3040485 (2021).
12. Rudolph, M., Wandt, B. & Rosenhahn, B. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1906–1915, https://doi.org/10.1109/WACV48630.2021.00195 (2021).
13. Xu, Y., Li, D., Xie, Q., Wu, Q. & Wang, J. Automatic defect detection and segmentation of tunnel surface using modified mask r-cnn. *Measurement* **178**, 109316 (2021).
14. Chen, J., Wen, Y., Nanehkaran, Y. A., Zhang, D. & Zeb, A. Multiscale attention networks for pavement defect detection. *IEEE Trans. Instrum. Meas.* **72**, 1–12. https://doi.org/10.1109/TIM.2023.3298391 (2023).
15. Fan, S. et al. Real-time defects detection for apple sorting using NIR cameras with pruning-based yolov4 network. *Comput. Electron. Agric.* **193**, 106715 (2022).
16. Hu, W. *et al.* A method of citrus epidermis defects detection based on an improved yolov5. *Biosyst. Eng.* (2023).
17. Xiong, C., Hu, S. & Fang, Z. Application of improved yolov5 in plate defect detection. *The Int. J. Adv. Manuf. Technol.* (2022).
18. Xu, X. & Li, X. Research on surface defect detection algorithm of pipeline weld based on yolov7. *Sci. Rep.***14** (2024).
19. Zhang, C. *et al.* Automated detection and segmentation of tunnel defects and objects using yolov8-cm. *Tunnel. Undergr. Space Technol.* (2024).
20. Lu, M., Sheng, W., Zou, Y., Chen, Y. & Chen Z. Wss-yolo: An improved industrial defect detection network for steel surface defects. *Measurement* (2024).
21. Viola, P. A. & Jones, M. J. Robust real-time face detection. *Int. J. Comput. Vision* **57**, 137–154 (2001).
22. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 886–893. https://doi.org/10.1109/CVPR.2005.177 (2005).
23. Felzenszwalb, P. F., Girshick, R. B., McAllester, D. & Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 1627–1645. https://doi.org/10.1109/TPAMI.2009.167 (2010).
24. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587, https://doi.org/10.1109/CVPR.2014.81 (2014).
25. He, K., Zhang, X., Ren, S. & Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824 (2015).
26. Girshick, R. Fast r-cnn. *Comput. Sci.* (2015).
27. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031 (2017).
28. Redmon, J., Divvala, S. K., Girshick, R. B. & Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 779–788 (2015).
29. Liu, W. *et al.* Ssd: Single shot multibox detector. In *European Conference on Computer Vision* (2015).
30. Redmon, J. & Farhadi, A. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525, https://doi.org/10.1109/CVPR.2017.690 (2017).
31. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007, https://doi.org/10.1109/ICCV.2017.324 (2017).
32. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement. arXiv:abs/1804.02767 (2018).
33. Tan, M., Pang, R. & Le, Q. V. Efficientdet: Scalable and efficient object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10778–10787, https://doi.org/10.1109/CVPR42600.2020.01079 (2020).
34. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934 (2020).
35. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. Scaled-yolov4: Scaling cross stage partial network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13024–13033, https://doi.org/10.1109/CVPR46437.2021.01283 (2021).
36. Carion, N. *et al.* End-to-end object detection with transformers. arXiv:2005.12872 (2020).
37. Ge, Z., Liu, S., Wang, F., Li, Z. & Sun, J. Yolox: Exceeding yolo series in 2021. arXiv:2107.08430 (2021).
38. Wang, C.-Y., Yeh, I.-H. & Liao, H. You only learn one representation: Unified network for multiple tasks. *J. Inf. Sci. Eng.* **39**, 691–709 (2021).
39. Chen, Q. *et al.* You only look one-level feature. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13034–13043, https://doi.org/10.1109/CVPR46437.2021.01284 (2021).
40. Jocher, G. YOLOv5 by Ultralytics, https://doi.org/10.5281/zenodo.3908559 (2020).
41. Xu, S. *et al.* Pp-yoloe: An evolved version of yolo. arXiv: 2203.16250 (2022).
42. Li, C. *et al.* Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022).
43. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475 (2023).
44. Jocher, G., Chaurasia, A. & Qiu, J. Ultralytics YOLO (2023).
45. Wang, C. *et al.* Gold-yolo: Efficient object detector via gather-and-distribute mechanism. arXiv:2309.11331 (2023).
46. Wang, C.-Y., Yeh, I.-H. & Liao, H.-Y. M. Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616 (2024).
47. Wang, A. *et al.* Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458 (2024).
48. Xuan, W. et al. A lightweight modified Yolox network using coordinate attention mechanism for PCB surface defect detection. *IEEE Sens. J.* **22**, 20910–20920. https://doi.org/10.1109/JSEN.2022.3208580 (2022).
49. Peng, C., Li, X. & Wang, Y. Td-yoloa: An efficient yolo network with attention mechanism for tire defect detection. *IEEE Trans. Instrum. Meas.* **72**, 1–11. https://doi.org/10.1109/TIM.2023.3312753 (2023).
50. Wang, R., Liang, F., Wang, B. & Mou, X. ODCA-YOLO: An omni-dynamic convolution coordinate attention-based yolo for wood defect detection. *Forests* **14**, 1885 (2023).
51. Tang, J. et al. A lightweight surface defect detection framework combined with dual-domain attention mechanism. *Expert Syst. Appl.* **238**, 121726 (2023).

52. Ma, Z. et al. A lightweight detector based on attention mechanism for aluminum strip surface defect detection. *Comput. Ind.* **136**, 103585 (2022).
53. Chen, H., Du, Y., Fu, Y., Zhu, J. & Zeng, H. Dcam-net: A rapid detection network for strip steel surface defects based on deformable convolution and attention mechanism. *IEEE Trans. Instrum. Meas.* **72**, 1–12. https://doi.org/10.1109/TIM.2023.3238698 (2023).
54. Xiao, G., Hou, S. & Zhou, H. Pcb defect detection algorithm based on cdi-yolo. *Sci. Rep.* **14** (2024).
55. Zheng, H., Chen, X., Cheng, H., Du, Y. & Jiang, Z. MD-YOLO: Surface defect detector for industrial complex environments. *Opt. Lasers Eng.* (2024).
56. Ouyang, D. *et al.* Efficient multi-scale attention module with cross-spatial learning. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1–5 (2023).
57. Su, Y. & Yan, P. A defect detection method of gear end-face based on modified yolo-v3. In *2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 283–288, https://doi.org/10.1109/CYBER50695.2020.9279161 (2020).
58. Wang, G.-Q. et al. A high-accuracy and lightweight detector based on a graph convolution network for strip surface defect detection. *Adv. Eng. Inform.* **59**, 102280 (2024).
59. Li, Y. *et al.* Efc-yolo: An efficient surface-defect-detection algorithm for steel strips. *Sensors (Basel, Switzerland)***23** (2023).
60. Duta, I. C., Liu, L., Zhu, F. & Shao, L. Pyramidal convolution: Rethinking convolutional neural networks for visual recognition. arXiv:2006.11538 (2020).
61. Neubeck, A. & Van Gool, L. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, 850–855, https://doi.org/10.1109/ICPR.2006.479 (2006).
62. Bodla, N., Singh, B., Chellappa, R. & Davis, L. S. Soft-nms - improving object detection with one line of code. In *2017 IEEE International Conference on Computer Vision (ICCV)* 5562–5570 (2017).
63. Woo, S., Park, J., Lee, J.-Y. & Kweon, I.-S. Cbam: Convolutional block attention module. arXiv:1807.06521 (2018).
64. He, Y., Song, K., Meng, Q. & Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **69**, 1493–1504. https://doi.org/10.1109/TIM.2019.2915404 (2020).
65. Lv, X., jie Duan, F., jia Jiang, J., Fu, X. & Gan, L. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors (Basel, Switzerland)***20** (2020).
66. Li, Z., Wei, X., Hassaballah, M., Li, Y. & Jiang, X. A deep learning model for steel surface defect detection. *Complex Intell. Syst.*[SPACE]https://doi.org/10.1007/s40747-023-01180-7 (2023).
67. Zou, Y. & Fan, Y. An infrared image defect detection method for steel based on regularized yolo. *Sensors* **24**, 1674. https://doi.org/10.3390/s24051674 (2024).
68. Guo, Z., Wang, C., Yang, G., Huang, Z. & dong Li, G. MSFT-YOLO: Improved yolov5 based on transformer for detecting defects of steel surface. *Sensors (Basel, Switzerland)***22** (2022).
69. Wang, L., Liu, X., Ma, J., Su, W. & Li, H. Real-time steel surface defect detection with improved multi-scale yolo-v5. *Processes* **11**, 1357. https://doi.org/10.3390/pr11051357 (2023).
70. Kou, X., Liu, S., Cheng, K.I.-C. & Qian, Y. Development of a yolo-v3-based model for detecting defects on steel strip surface. *Measurement* **182**, 109454 (2021).
71. Lu, J., Zhu, M., Qin, K. & Ma, X. Yolo-lfpd: A lightweight method for strip surface defect detection. *Biomimetics* **9**, 607 (2024).
72. Huang, Y., Tan, W., Li, L. & Wu, L. Wfre-yolov8s: A new type of defect detector for steel surfaces. *Coatings* **13**, 2011 (2023).
73. Wang, X. & Zhuang, K. An improved yolox method for surface defect detection of steel strips. In *2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA)*, 152–157 (IEEE, 2023).
74. Zhao, H. et al. Lsd-yolov5: A steel strip surface defect detection algorithm based on lightweight network and enhanced feature fusion mode. *Sensors* **23**, 6558 (2023).

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 11972266, the National Natural Science Foundation of China under Grant No. 12402344, the National Natural Science Foundation of China Youth Project under Grant No. 12401669, and the Research Project of the Hubei Provincial Department of Education under Grant No. B2023064.

Author contributions

R.S., Z.C., W.Q. conceived the experiments, R.S., Z.C. and L.B. conducted the experiment(s), Z.C., R.S. and S.K. analysed the results. All authors reviewed the manuscript.

Additional information

Correspondence and requests for materials should be addressed to K.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.