# Lightweight Steel Surface Defect Detection Based on YOLOv10

Luyu Sun, Yujun Zhang*

Abstract—Steel surface defect detection is of significant importance for ensuring the quality of steel production, and it requires high-precision real-time detection capabilities. Based on this, this paper proposes an improved model based on YOLOv10. First, we innovate the C2f module of YOLOv10 by introducing the Star network and EMA attention mechanism, resulting in the C3_Star_EMA module. This module aims to map the input into a high-dimensional nonlinear feature space through star operations (element-wise multiplication), thereby enhancing the model's expressive power and performance. Meanwhile, the MobileOneBlock module is incorporated into the backbone network. This module reduces the parameters and computational complexity significantly through a multi-branch convolution design and parameter reorganization. Finally, the C2fAFF module is introduced, which innovates the C2f module by using the AFF attention mechanism. Through multiple iterations, the feature fusion process is gradually optimized, assigning appropriate weights between features of different scales, thus improving the model's detection ability for multi-scale objects. The improved model achieves a mean average precision (mAP) of 79.2% on the NEU-DET dataset, which is 4.9% higher than the baseline YOLOv10n, with a 16.7% reduction in parameters and an 18.3% reduction in GFLOPs. The improved model effectively enhances the accuracy and speed of real-time steel surface defect detection.

Index Terms—Steel surface defect detection, feature fusion, YOLO, object detection

## I. Introduction

STEEL production is an important indicator of the industrial capacity of modern society. Steel is widely used in various industries and is an indispensable material in modern society. Therefore, the speed and accuracy of steel surface defect detection are of high importance. Due to manufacturing processes and equipment limitations, steel surfaces often exhibit various defects, including but not limited to cracks, inclusions, spots, pitting, and other imperfections [1]. These defects can affect the service life of steel materials and even lead to significant engineering accidents. Therefore, steel surface defect detection plays a critical role in the steel production process.

Traditional defect detection methods are influenced by human factors and suffer from issues such as low efficiency, missed detections, and false positives. The traditional methods include Acoustic Emission (AE) testing [2], Laser Scanning Technology (LST) [3], Ultrasonic Testing (UT) [4], and Computed Tomography (CT) [5].

Acoustic Emission (AE) testing technology identifies potential defects by monitoring high-frequency acoustic signals on the surface or inside steel materials (such as those generated during crack propagation or plastic deformation). These acoustic signals are received and processed by sensors to determine the location and type of the defects. This method is particularly suitable for monitoring the dynamic behavior of metallic materials and can achieve real-time monitoring, making it ideal for stress testing and structural health monitoring.

Laser Scanning Technology (LST) captures changes in surface morphology by scanning the steel surface and analyzing the variation in the reflected signal of the laser beam. This method is particularly effective for detecting tiny defects, enabling efficient and precise capture of the 3D morphology of steel surfaces, which is advantageous for detecting defects at the microscopic scale.

Ultrasonic Testing (UT) leverages the propagation characteristics of ultrasound through materials. By detecting the reflected ultrasound signals, it can identify defects. Since ultrasound can penetrate deep into the material, this method not only effectively detects surface defects but also uncovers hidden internal defects like cracks and voids.

Computed Tomography (CT) is a non-destructive testing method based on X-rays. It scans the sample from multiple angles and reconstructs a 3D image using a computer. Compared to traditional X-ray testing, CT provides more detailed and clearer internal structural images, making it more accurate and effective for detecting internal defects in steel, such as porosity, cracks, and inclusions.

Compared to traditional defect detection methods, deep learning has shown significant advantages in steel surface defect detection, particularly in areas such as automation, accuracy, robustness, big data processing, real-time feedback, and multi-modal data fusion. Deep learning methods offer higher efficiency and flexibility compared to traditional methods. With the continuous development of deep learning technology, it is poised to become an essential tool in the field of steel quality inspection, driving detection systems toward higher efficiency and intelligence.

Defect detection methods using deep learning include CNN [6], Faster R-CNN [7], Swin Transformer [8], ResNet-50[9], and GAN [10]. Convolutional Neural Networks (CNN) are deep learning models specifically designed for image data. Through multiple layers of convolution, pooling, and fully connected layers, CNNs can automatically extract features from raw images

Manuscript received March 7, 2025; revised May 23, 2025. This work was supported by the Key Laboratory of Internet of Things Application Technology on Intelligent Construction, Liaoning Province (2021JH13/10200051)

Luyu Sun is a graduate student of School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (e-mail: 1471060876@qq.com).

Yujun Zhang is a Professor of School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (e-mail: 1997zyj@163.com).

and detect defects on steel surfaces or other products. Faster R-CNN is a deep learning-based object detection model aimed at improving the efficiency and accuracy of object detection while significantly accelerating the detection process. The Swin Transformer is a novel visual model based on the Transformer architecture, specifically designed for visual tasks. By introducing local windows and shifted windows in the Transformer, it enhances computational efficiency and reduces spatial complexity. ResNet-50 is a deep convolutional neural network architecture, a variant of ResNet, where "50" indicates the number of layers in the network. This architecture effectively addresses the training challenges in deep networks, making it easier to train very deep networks. GAN (Generative Adversarial Networks) is a deep learning model composed of a generator and a discriminator. Through adversarial training, the generator learns to generate realistic data samples, while the discriminator learns to differentiate between real and generated data. The generator and discriminator continually optimize their respective capabilities through their adversarial relationship, ultimately enabling the generator to produce highly realistic samples.

## II. Related work

In recent years, many researchers have made progress in steel surface defect detection using traditional defect detection methods. Boudiaf A et al. [11] proposed an improved AlexNet model, which was obtained using transfer learning techniques and replaced the classifier part with one or more new fully connected layers. At the same time, the AlexNet model, transfer learning, and various machine learning algorithms were combined to create a hybrid model. Zheng X et al. [12] proposed a feature extraction method based on the combination of Legendre Wavelet Transform and Autoencoder Network (LWT-AE). This method first uses LWT to extract defect features and removes redundant components through statistical and texture parameters. Then, an AE network is used to reduce the dimensionality of the features. Finally, two classifiers, SVM and BPNN, are used to improve the generalization ability of the method. Zaghdoudi R et al. [13] improved the recognition rate of steel surface defect classification by introducing a new classifier combination method. This method uses LCCMSP and DCP, balancing accuracy and time consumption. These features were then fed into SVM and RF classifiers, generating four basic classifiers. Finally, Bayesian fusion rules were applied to integrate the outputs of these classifiers, ultimately producing the classification decision.

Compared to traditional defect detection methods, deep learning has several advantages in steel surface defect detection, including automation, efficiency, strong data adaptability, and robustness. It can handle complex and large-scale data and can automatically learn and extract efficient features. Traditional methods, on the other hand, require extensive manual design and adjustment, and they perform less well when facing complex, dynamic defect patterns. As a result, deep learning has become increasingly widespread in steel surface defect detection in recent years. Jiyang Q et al. [14] proposed an improved Faster R-CNN algorithm for strip steel surface defect detection, which enhanced the network's feature extraction ability using a differential convolution module and Swin Transformer. The CBAM-BiFPN module was used to improve the network's attention to defects. RoI alignment layers replaced RoI pooling layers to improve defect localization accuracy, and Soft NMS was used to optimize non-maximum suppression and remove redundant boxes. Overall, this algorithm significantly improved the accuracy and efficiency of strip steel surface defect detection. Liang C et al. [15] proposed a high-precision industrial defect detection network that uses the CEM module for multi-scale fusion to enhance semantic representation. Then, the FEM module optimizes feature information at the end of the backbone network, combined with the HAM module to extract important features. Finally, the DFPN network was proposed to significantly enhance the detection ability of the target. Ibrahim A A M S et al. [16] used a new method to improve defect detection and classification accuracy. This method used part of the pre-trained VGG16 model as a feature extractor and a new convolutional neural network (CNN) as a classifier to classify six types of defects occurring on steel surfaces. Dong X et al. [17] proposed the EAIRNet network to improve steel surface defect detection accuracy. EAIRNet consists of three core modules: a multi-scale feature extractor, SEIM module, and RAGM module. First, the network uses ResNet to extract multi-scale features from the steel surface environment, then integrates features from the encoder stage to extract edge features. The SEIM module facilitates the interaction between significant features and edge features, enhancing detection accuracy. Additionally, the RAGM module strengthens the shallow decoder layers' attention to defective regions. He Y et al. [18] proposed a sample generation method based on Generative Adversarial Networks (GAN) to improve the accuracy of steel plate surface defect detection models. To improve the quality of generated samples, this method proposed a two-stage sample generation process: the production phase generates defects on defect-free background samples, and the elimination phase learns to remove defects from defective samples. By minimizing the differences between the two-stage generated samples, the model can generate background samples close to real samples while ensuring that the defect samples are more realistic.

Although the above defect detection methods have achieved innovation and progress in many areas, they still have certain drawbacks and limitations. Traditional defect detection methods cannot escape the interference of human factors, are inefficient, and struggle to handle complex defects. They are also costly and not adaptable to dynamic environments. Manual detection is influenced by the operator's experience, leading to inconsistent results, and accuracy is limited in large-scale data or complex defect detection. The investment and maintenance costs of traditional sensors and detection equipment are high, and they cannot provide sufficient stability and adaptability.While Convolutional Neural Networks

(CNNs) have eliminated the need for human intervention in feature extraction, they come with several drawbacks. These include high demands for computational resources and memory, the need for large amounts of labeled data to prevent overfitting, complex model structures, poor interpretability, long training times, limited invariance to translation and rotation in input data, and the necessity for extensive hyperparameter tuning. These limitations hinder their application in complex industrial scenarios that require real-time processing with small data volumes. To address these challenges, this paper proposes an improved lightweight YOLOv10 algorithm. The goal is to reduce the total number of parameters while maintaining the original model's accuracy in detecting steel surface defects. The specific improvements include:

1. In the head part, the C3_Star_EMA module is used to replace the original C2f module, enabling the model to exchange information between the core layers and peripheral nodes of the network. The aggregation effect of the central node reduces redundant and unnecessary complexity in information flow while ensuring efficient information transfer between nodes and maintaining the model's generalization ability. This improvement reduces parameters by approximately 9.3% compared to the original model and increases accuracy by 1.4

2. In the head part, the C2fAFF module is used to replace the original C2f module. By combining local and global attention mechanisms, the model's attention to key information is enhanced, improving feature expression ability. This improvement reduces parameters by approximately 11.1% compared to the original model and increases accuracy by 2.1%.

3. In the backbone part, the MobileOneBlock module is used to replace the original C2f module. This modification reduces computational overhead through efficient convolution operations and network design while maintaining high model performance. This improvement reduces parameters by approximately 16.7% compared to the original model and increases accuracy by 4.9%.

## III. Method Introduction

SSD, Fast R-CNN, DETR, and YOLO are classic defect detection algorithms. SSD (Single Shot MultiBox Detector) can perform both object localization and classification tasks simultaneously and has high detection speed. SSD can detect objects at different feature layers, allowing it to handle objects of different scales. It completes the object detection task in one shot, unlike two-stage detection methods that first generate candidate boxes. However, SSD has lower accuracy and poor detection capability for small objects. DETR generates target bounding boxes and class labels directly through the Transformer model. Unlike traditional Convolutional Neural Network (CNN) methods, DETR uses the Transformer structure, which has stronger global context modeling capabilities, enabling it to capture object information over a wider range in the image. However, DETR requires longer training times, high computational power, and more data for effective training. It also has weak detection capabilities for small objects and is not suitable for real-time detection in complex industrial environments. YOLOv10, compared to its previous versions, introduces a consistent dual assignment strategy, using one-to-many label assignments during training to provide rich supervision signals, and one-to-one matching during inference. This improvement reduces inference latency and computational complexity while maintaining high accuracy. YOLOv10 currently has versions such as YOLOv10n, YOLOv10s, YOLOv10m, YOLOv10l, YOLOv10x, and others, each suited for detection in different environments. Depending on available resources and detection requirements, different versions can be selected. YOLOv10n has lower parameters and computational complexity while achieving high inference speed and detection accuracy, making it more suitable for steel surface defect detection in complex industrial environments.

The network architecture of YOLOv10 is shown in Figure 1.

Like traditional YOLO series, YOLOv10 consists of four main parts: Input, Backbone, Neck, and Head. The Input part is responsible for receiving the input image and preprocessing it to enable the model to better extract image features. The input part resizes the received image and normalizes it to reduce the dynamic range of the data, improving computational stability. Finally, SCDown is used to separate spatial information and channel information, enhancing feature extraction capability while reducing computational load.

The Backbone part employs a series of lightweight optimization strategies to significantly reduce computational costs while maintaining strong feature extraction capabilities, making it more suitable for real-time detection tasks. The Backbone separates spatial information and channel information using Spatial Decoupled Convolution (SDC) and Channel Decoupled Convolution (CDC), improving computational efficiency. Depthwise Separable Convolution is used to reduce parameters and computation, while large Kernel Convolution increases the model's receptive field, making it easier to capture large target features. Finally, Partial Self-Attention is used to enhance target features and improve detection performance in complex scenes, enhancing model practicality. The above parts form the Backbone's Lightweight Backbone Network, designed to enhance feature extraction ability. The Lightweight Residual Block uses a lightweight Residual Block to reduce the gradient vanishing problem and enhance feature representation. It uses 1×1 convolution for channel fusion, reducing computational overhead while maintaining effective feature representation. The Lightweight Residual Block ensures the stability of Backbone training. Lastly, the Backbone utilizes the Global Feature Extraction Layer to further extract important information from higher-level feature maps and optimize multi-scale feature representation, while reducing computational redundancy to further improve inference speed.

The Neck is the key part connecting the Backbone and Head, mainly responsible for integrating multi-layer features, enhancing multi-scale detection capability while maintaining computational efficiency. YOLOv10 does not use the traditional YOLO series' FPN and

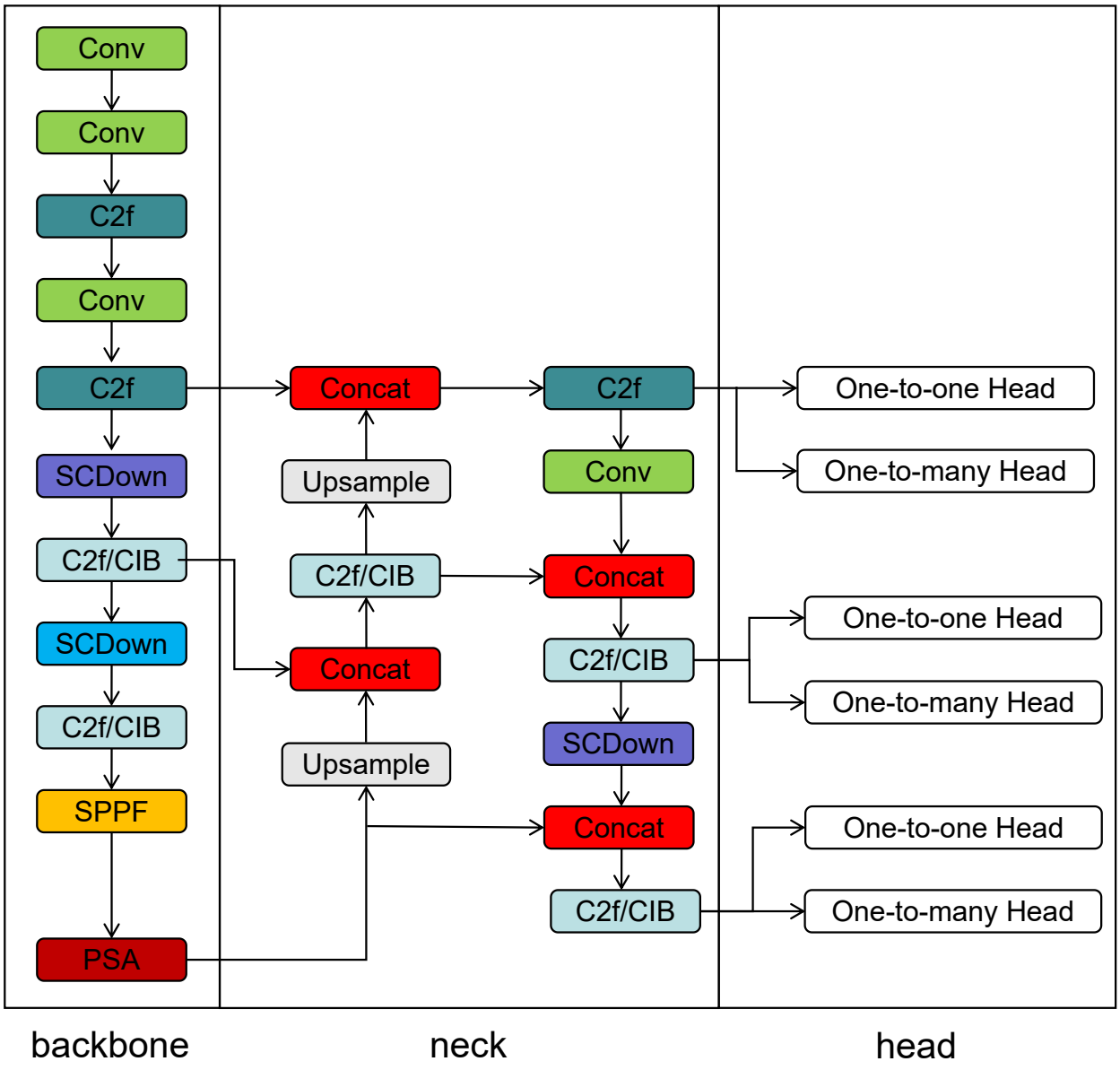


Fig. 1: YOLOv10 network architecture

PAN structures but instead introduces its self-developed LG Block (Level-Guided Block). This module is used to optimize the fusion of features at different scales, addressing the information imbalance problem in traditional FPN (Feature Pyramid Networks) and PAN (Path Aggregation Networks), making the feature flow more reasonable and improving small object detection capability.

YOLOv10's Head still uses the Anchor-based mechanism. Anchor boxes are pre-defined rectangular boxes of multiple sizes at each position of the feature map. Each position on the feature map generates multiple different-sized anchor boxes to detect objects of various sizes. The Head part matches these anchor boxes with the predicted boxes output by the network, and adjusts the predicted box parameters to fit the real targets.

The structure diagram of the improved model is shown in Figure 2. First, the MobileOneBlock is used to reduce computational redundancy, improving training stability and information flow. Next, the C3_Star_EMA is used to replace the original C2f module, enhancing the model's generalization ability and feature extraction capability, while reducing model complexity. Finally, the C2fAFF module is introduced to improve the model's accuracy and robustness, while also reducing computational load.

### A. MobileOneBlock

The structure of MobileOne is shown in Figure 3.

MobileOne is a lightweight network architecture inspired by efficient network designs such as MobileNet and EfficientNet, aiming to achieve higher computational efficiency and lower inference latency. MobileOneBlock [19] is the core module of this architecture, balancing accuracy and computational efficiency, allowing the network to operate efficiently with limited computational resources. By adopting techniques such as depthwise separable convolution, bottleneck structures, residual connections, and optimized activation functions, MobileOneBlock enables the network to perform efficiently

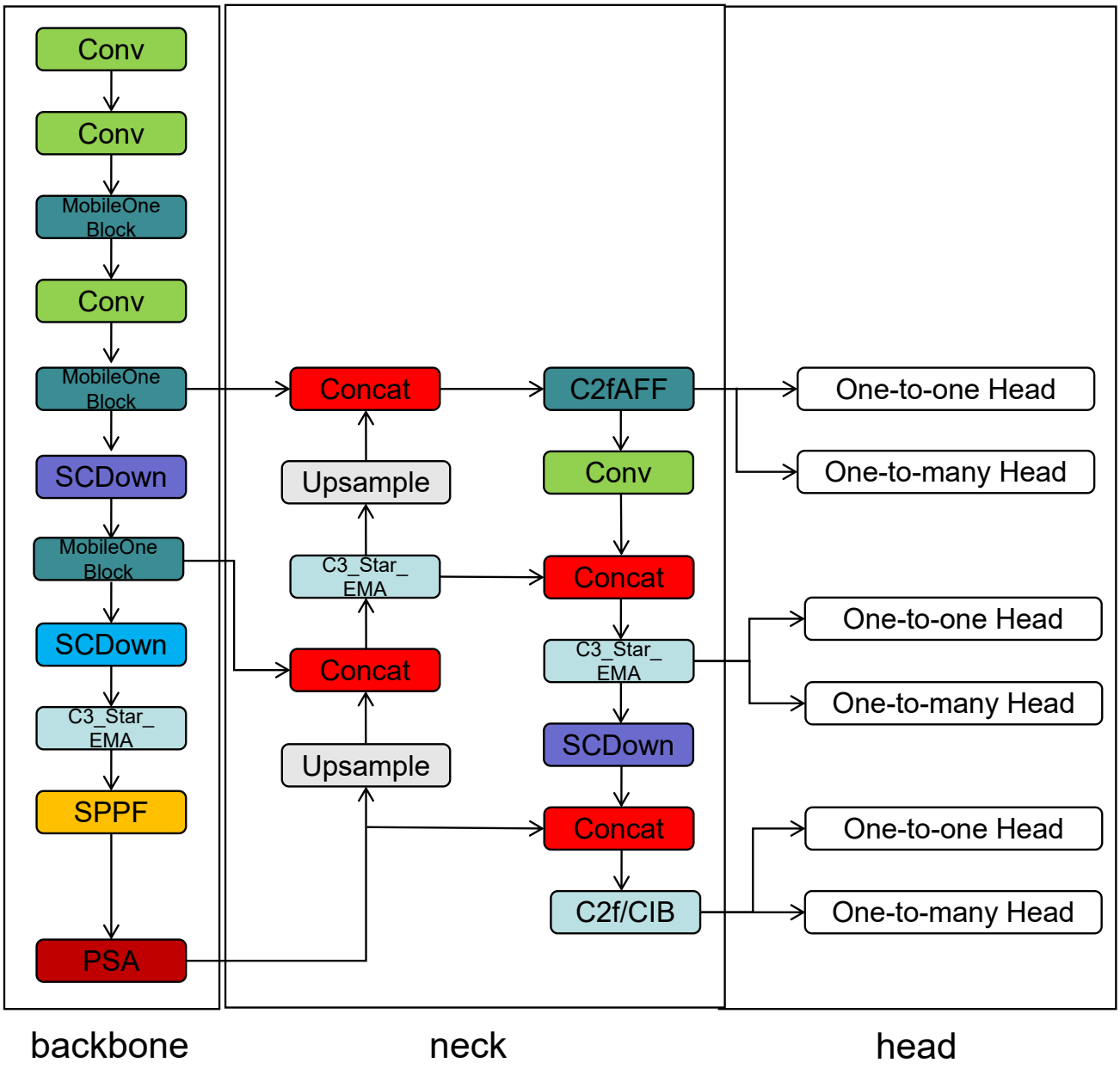


Fig. 2: Improved YOLOv10 network architecture

on low-power hardware. In particular, depthwise separable convolution, as one of the key components of MobileOneBlock, decomposes the standard convolution operation into depthwise convolution and pointwise convolution, effectively reducing computational load and thus improving overall computational efficiency. The core idea of depthwise separable convolution is to decompose the traditional convolution operation into two steps: Depthwise Convolution and Pointwise Convolution. Depthwise Convolution applies convolution only to each input channel.Assume the input feature map has $C_{\text{in}}$ channels,The size of the convolution kernel is $K \times K$.Then, for each input channel, an independent $K \times K$ convolution kernel is used. Each convolution kernel convolves with the corresponding input channel, producing a single-channel feature map.Each convolution kernel convolves with the corresponding input channel, producing a single-channel feature map. The computational cost is $C_{\text{in}} \times K \times K \times H \times W$.After depthwise convolution, pointwise convolution fuses all output channels using a 1×1 convolution. The role of this convolution kernel is to combine the outputs of the depthwise convolution across channels to generate the final output feature map. Assume the number of channels in the output feature map is $C_{\text{out}}$ ,then the computational cost of the pointwise convolution is $C_{\text{in}} \times C_{\text{out}} \times H \times W$.Compared to traditional convolution, depthwise separable convolution significantly reduces computational cost and parameter count. The computational cost of traditional convolution is $C_{\text{in}} \times C_{\text{out}} \times K \times K \times H \times W$,The computational cost of depthwise separable convolution is the sum of the computational costs of depthwise convolution and pointwise convolution, that is $C_{\text{in}} \times K \times K \times H \times W + C_{\text{in}} \times C_{\text{out}} \times H \times W$,through this decomposition, the computational cost of depthwise separable convolution is greatly reduced compared to traditional convolution.Especially when both

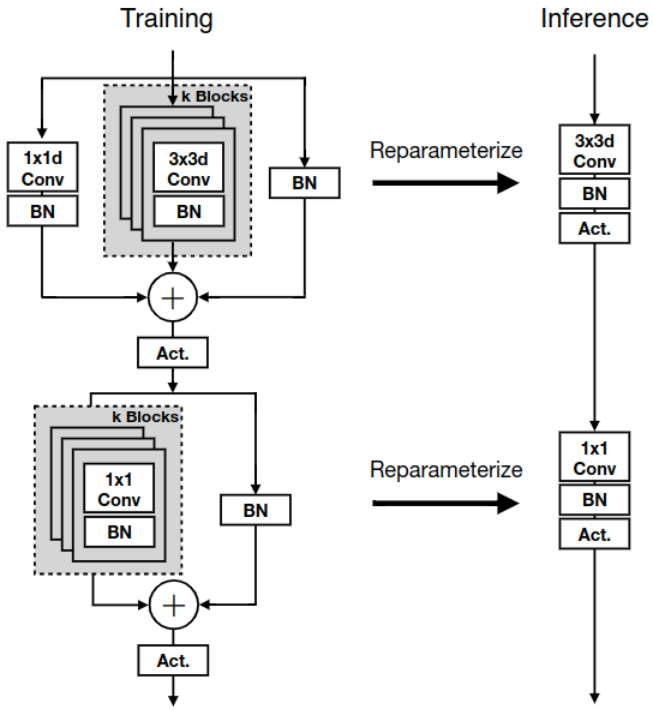


Fig. 3: MobileOneBlock Structure Diagram.

the number of input channels $C_{\text{in}}$ and output channels $C_{\text{out}}$ are large, depthwise separable convolution can greatly reduce the computational cost.MobileOneBlock uses the ReLU function as the activation function. ReLU is a simple and efficient activation function, and due to its superior computational efficiency and gradient propagation performance, it has become a commonly used activation function in deep neural networks. Its formula is as follows:

$$\text{ReLU}(x) = \max(0, x) \tag{1}$$

In traditional activation functions (such as Sigmoid and Tanh), especially in deep networks, the vanishing gradient problem is common, making the training process difficult. However, ReLU has a gradient of 1 in the positive range, which prevents the vanishing gradient issue, thus accelerating training.

### B. C3_Star_EMA

StarNet [20] is a neural network architecture designed based on a 'star-shaped' information transmission mechanism. By introducing efficient interaction between the central node and peripheral nodes, it combines global information capture with local detail representation, significantly reducing the computational complexity of traditional fully connected networks, making it more efficient when processing long sequences or high-resolution data. The StarNet structure is shown in Figure 4.

The C3 module is a key component in YOLO for efficient feature extraction and fusion, designed to enhance the performance of the model in object detection tasks. It combines multiple Bottleneck layers to strengthen feature representation while maintaining a lightweight computational load. Comprising three convolutional layers and several Bottleneck modules, the first convolutional layer has a stride of 2, which halves the feature map size to increase the receptive field and reduce computational costs. Additionally, the C3 module optimizes feature propagation and information flow through residual connections and multi-scale information fusion. This structural design not only improves the efficiency of feature extraction but also reduces the number of parameters through lightweight design, thereby accelerating the inference process.

StarNet adopts a four-stage hierarchical structure, performs downsampling through convolutional layers, and uses an improved demonstration block for feature extraction. To enhance computational efficiency, StarNet replaces the original layer normalization with batch normalization and places it after the depthwise convolution. Drawing on the design of MobileNeXt [21], StarNet adds depthwise convolution at the end of each block, with a fixed channel expansion factor of 4, doubling the network width at each stage. In addition, StarNet replaces the GELU activation function with the ReLU6 function and flexibly adjusts the network scale by modifying the number of blocks and input embedding channels.

The workflow of StarNet is divided into two stages: the information aggregation stage and the information distribution stage. The information aggregation stage involves passing the local features of each peripheral node to the central node. The central node uses weighted summation, attention mechanisms, or other aggregation methods to integrate the local features into a global representation. Mathematically, this can be expressed as:

$$h = \sum_{i=1}^{n} \alpha_i x_i \tag{2}$$

Where,

$$\alpha_i = \frac{\exp(\varphi(x_i))}{\sum_{j=1}^{n} \exp(\varphi(x_j))} \tag{3}$$

Here, $x_i$ represents the features of the i-th peripheral node.Function $\varphi(\cdot)$ for calculating similarity or importance scores.The information distribution stage refers to the process where, after obtaining the global representation h, the central node feeds it back to each peripheral node. The peripheral nodes use this global information to update their own features, allowing the local features to incorporate global context. The update formula can be a simple summation or involve a nonlinear transformation:

$$x_i' = f(x_i, h) \tag{4}$$

Here, f is typically implemented as a concatenation followed by a multilayer perceptron (MLP) or convolutional layers, etc.

EMA [22] is an efficient multi-scale attention module that reshapes part of the channels to serve as the batch dimension and divides the channel dimension into multiple sub-feature groups, thereby optimizing the distribution of spatial semantic features within each feature group. This strategy not only retains key channel information but also significantly reduces computational overhead, enhancing the model's efficiency. EMA uses a 1×1 convolution shared component of the Coordinate Attention (CA) module, which is placed in parallel with

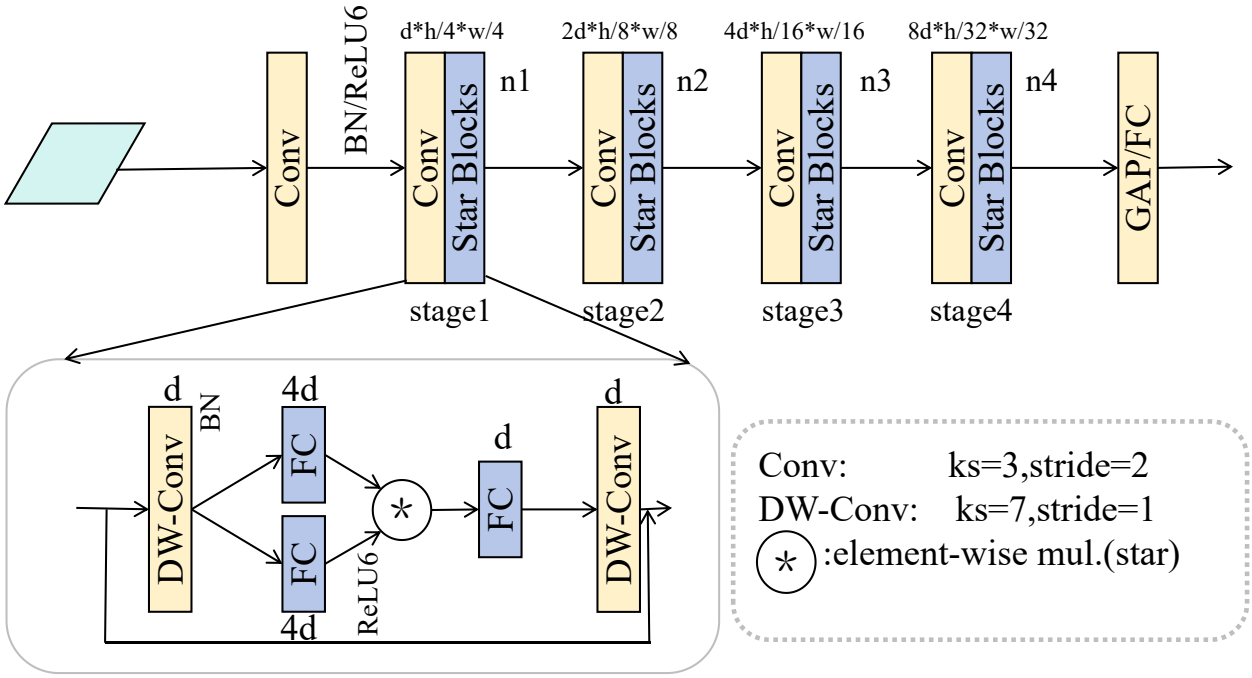


Fig. 4: star network architecture

another 3×3 convolution to improve the model's response speed. This parallel substructure effectively aggregates multi-scale spatial information, allowing the model to accelerate inference speed while more accurately extracting and expressing features. Compared to traditional serial computation methods, the parallel design of EMA effectively avoids lengthy sequential processing and excessively deep network structures, thereby reducing computational burdens while maintaining excellent feature expression ability. In terms of feature fusion, EMA further introduces a cross-dimensional interaction mechanism to ensure that channel information and spatial information can fully complement each other, thereby enhancing the completeness of feature representation. This innovative design not only strengthens the model's ability to capture local details but also improves its ability to model global structures, making EMA perform better in computer vision tasks such as classification, object detection, and semantic segmentation. The EMA structure is shown in Figure 5.

This paper integrates the STAR and EMA modules into the C3 structure, innovatively proposing the C3_Star_EMA module. This module combines the advantages of the above components, not only enhancing the diversity of feature flow but also strengthening gradient propagation through a more efficient skip connection mechanism, reducing information loss. At the same time, the introduction of EMA further strengthens the interaction between channels and spatial information, making the model more accurate in small object detection and boundary area recognition.

Furthermore, the parallel substructure design of EMA ensures that C3_Star_EMA significantly reduces computational costs while enhancing feature extraction capabilities. This allows the module to perform better in object detection and semantic segmentation tasks, enabling it to more effectively distinguish between foreground and background, thus improving detection accuracy. Additionally, thanks to its efficient computational characteristics, C3_Star_EMA is also highly suitable for lightweight model deployment, demonstrating excellent performance in efficient object detection tasks.

### C. C2FAFF

In computer vision tasks (such as object detection, semantic segmentation, and image classification), multi-level feature fusion is crucial for improving model performance. Deep neural networks typically extract features at different levels, where shallow features contain rich local detail information (such as textures and edges), while deep features carry high-level semantic information, but often lack spatial details. However, traditional feature fusion methods have certain limitations: fixed-weight fusion is difficult to adapt to changes in input data and can lead to information loss. While concatenation fusion can integrate multi-level information, it increases computational costs and may introduce feature redundancy. Traditional attention mechanisms focus only on channel or spatial information, making it difficult to flexibly adjust the weights of cross-scale features, limiting the effectiveness of fusion.

To address the above issues, this paper introduces the AFF [23] (Adaptive Feature Fusion) module, the structure of which is shown in Figure 6.

AFF dynamically computes the importance of features at different levels through an attention mechanism and implements adaptive fusion, ensuring the effectiveness of information interaction.

Overall, AFF uses the Channel Attention (CA) mechanism to calculate the fusion ratio of different features, which is equivalent to "extracting global semantic features" and retaining important information in the

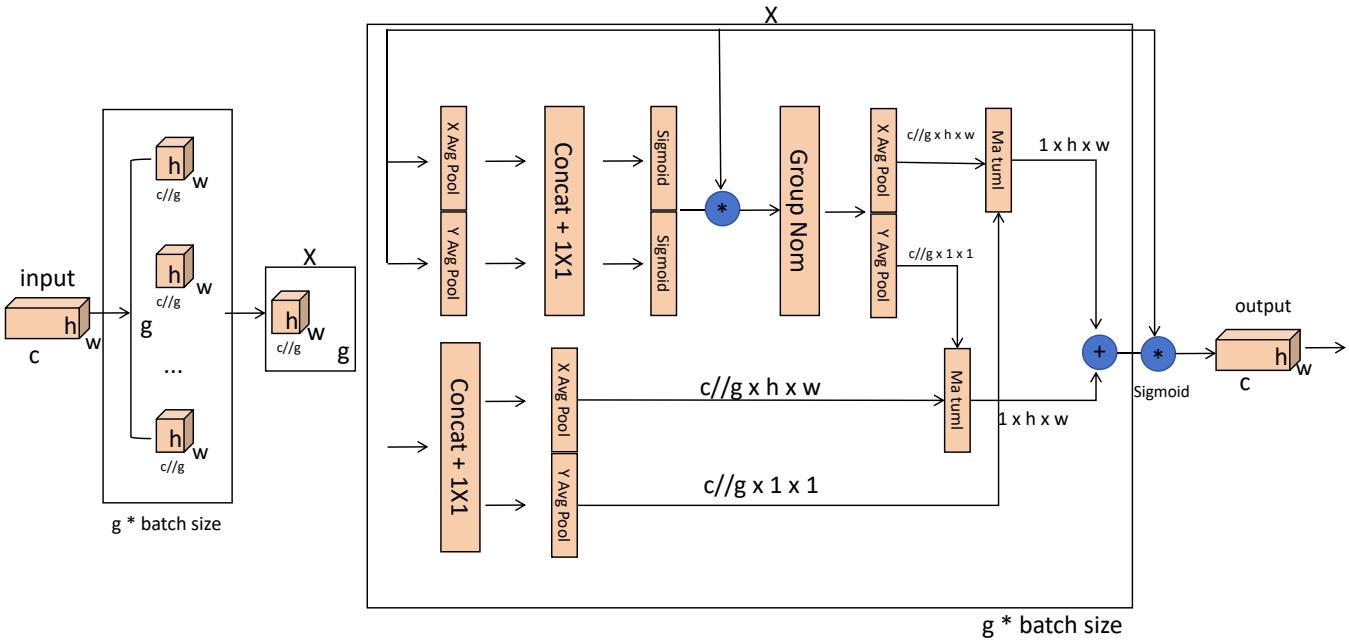


Fig. 5: EMA network architecture

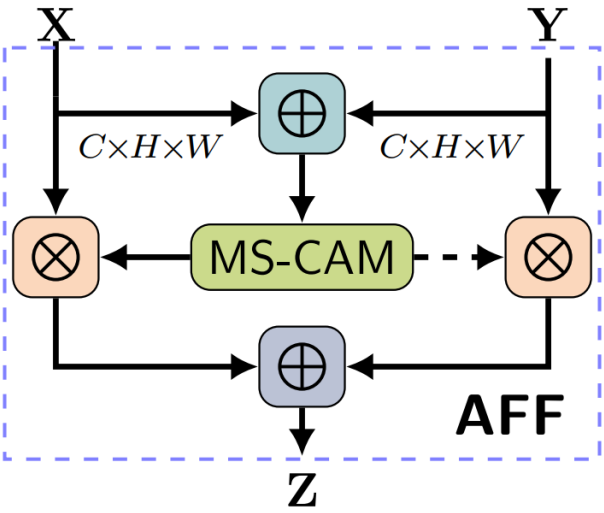


Fig. 6: AFF Structure Diagram.

channel dimension. By normalizing the weight values using a Sigmoid activation function, AFF can precisely adjust feature contributions, making fusion smarter and more efficient, thereby significantly improving the model's performance in various computer vision tasks. The formula for AFF is as follows:

$$Z = M(X \oplus Y) \odot X + (1 - M(X \oplus Y)) \odot Y, \quad (5)$$

⊕epresents pixel-wise summation as the initial feature. Z represents the fused feature. X and Y are the input features.

## IV. Experimental Design and Implementation

### A. Dataset Introduction

NEU-DET is a standard dataset for metal surface defect detection created by Northeastern University, widely used in deep learning, computer vision, and industrial automation fields, particularly for object detection tasks, helping to improve the accuracy and efficiency of quality inspection in smart manufacturing. The dataset contains 1,800 high-quality images (200 × 200 pixel resolution) and is divided into training (1,440 images), testing (180 images), and validation (180 images) sets in an 8:1:1 ratio to ensure the model's generalization capability.

The NEU-DET dataset covers six typical metal surface defects that are extremely common in metal processing and production, and they have a significant impact on product quality and safety. These defects include rolled-in scale, which is caused by the oxidation layer or impurities on the material surface being pressed into the metal surface during processing, resulting in scaly or uneven textures that can affect the aesthetic appearance of the metal and lead to cracks or spalling in subsequent steps; scratches, which are linear marks caused by mechanical friction from equipment wear, workpiece collisions, or transportation friction, reducing product appearance quality and potentially becoming stress concentration points that affect mechanical properties; patches, which are uneven areas caused by oxidation or material adhesion, often appearing in localized positions on the metal surface due to improper heat treatment, chemical reactions, or material contamination, and can affect corrosion resistance and coating adhesion; inclusions, which are material overlaps caused by impurities in the metal material or welding defects, such as oxides, sulfides, or other non-metallic substances, reducing the strength and toughness of the metal and potentially leading to crack initiation and propagation; cracks, which are serious defects in the metal surface or interior caused by uneven cooling, material fatigue, or stress concentration, severely affecting the integrity and safety of the metal structure and leading to early failure; and pitted surface, characterized by depressed areas caused by corrosion, oxidation, or bubble ruptures, appearing as irregular pits

or cavities that reduce the metal's corrosion resistance and fatigue strength.

## B. Evaluation Metrics

To demonstrate the effectiveness of the proposed improved model, the following metrics will be used for evaluation:

Precision (P): Precision is an important metric for measuring the accuracy of a classification model's predictions, especially for binary classification problems. It describes the proportion of true positive samples among those predicted as positive by the model. Specifically, precision reflects the accuracy of the model when predicting positive cases, and its formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{6}$$

TP (True Positive): The number of samples correctly predicted as positive by the model. FP (False Positive): The number of samples incorrectly predicted as positive, but actually negative.

Recall (R): Recall is used to measure the performance of a classification model when handling imbalanced datasets, particularly its ability to identify actual positive samples. Unlike precision, which focuses on the accuracy of the model when predicting positive cases, recall focuses on how many of the actual positive samples are correctly predicted as positive. Its formula is as follows:

$$R = \frac{TP}{TP + FN} \tag{7}$$

TP (True Positive): The number of samples correctly predicted as positive by the model. FN (False Negative): The number of samples incorrectly predicted as negative, but actually positive.

Params (Parameters): In machine learning or deep learning models, parameters refer to the adjustable variables within the model that define its structure and behavior. The types and number of parameters may vary across different types of models. By adjusting these parameters, the model can learn and optimize during the training process, thereby improving its predictive performance.

GFLOPS (Giga Floating Point Operations Per Second): GFLOPS is a unit for measuring the speed at which a computer or computing hardware (such as CPU, GPU, etc.) processes floating-point operations, representing the number of floating-point operations the hardware can execute per second. It is commonly used to assess the computational capability of processors, especially in applications that require a large amount of floating-point calculations, such as deep learning, scientific computing, etc.

Through the comprehensive evaluation of the above metrics, this paper aims to demonstrate the advantages of the proposed improved model in various aspects of performance, further validating its effectiveness.

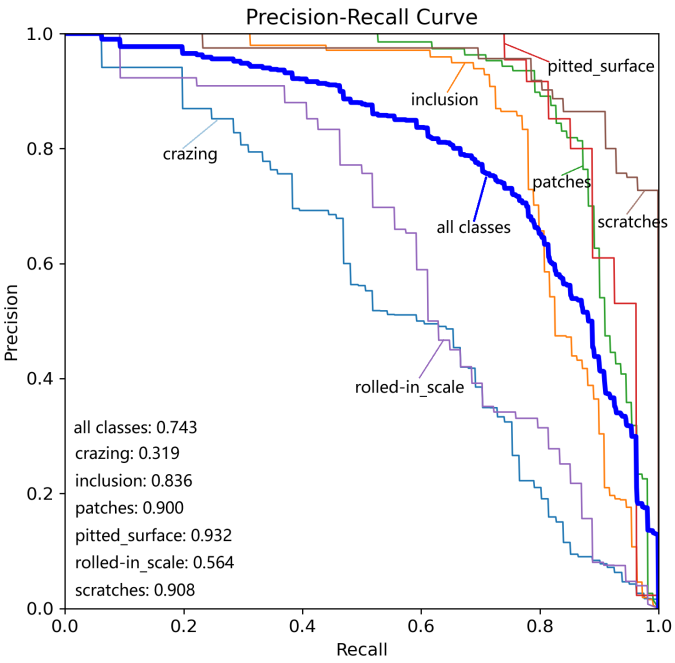


Fig. 7: FP-R curve of YOLOv10 Algorithm.

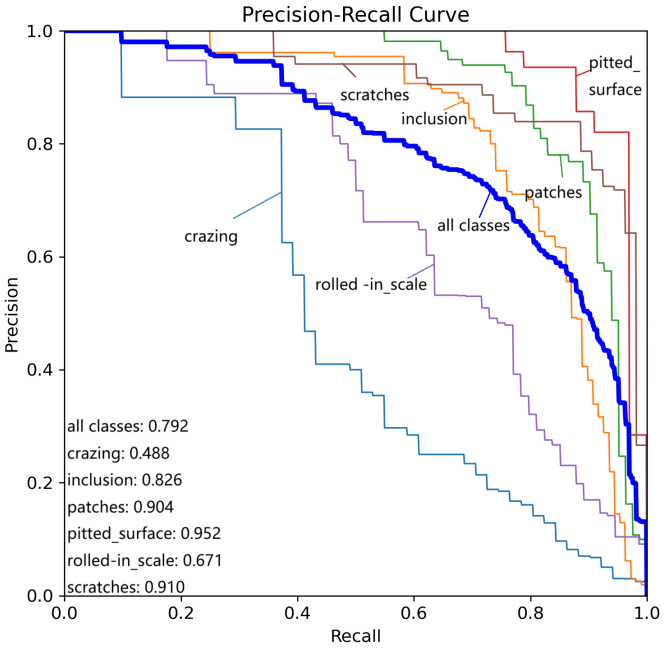


Fig. 8: FP-R Curve of the Improved YOLOv10 Algorithm.

## C. Comparative Experiment

Figures 7 and 8 show the PR curves of the model before and after improvement, reflecting the performance of the original YOLOv10 and the improved YOLOv10 under the same experimental conditions. The curves display the detection accuracy for each type of defect as well as the average detection accuracy for all defects. As shown in Figures 6 and 7, despite the lightweight modification to the original model, the improved model still maintains a high detection accuracy across various defects, with even significant improvements. Specifically, the overall detection accuracy of the improved model increased from 74.3% to 79.2%, an improvement of 4.9

percentage points. Additionally, the improved model showed significant improvements in detecting specific defects, exhibiting stronger precision. For example, the detection accuracy for crazing increased from 31.9% to 48.8%, a 16.9% improvement; for patches, it increased from 90.0% to 90.4%, a 0.4% improvement; for pitted_surface, it increased from 93.2% to 95.2%, a 2% improvement; and for rolled-in_scale, it increased from 56.4% to 67.1%, a 10.7% improvement. These results indicate that, after lightweight processing, the improved model not only retains strong feature extraction capabilities but also enhances detection accuracy for small targets, especially demonstrating stronger adaptability in complex environments, making it particularly suitable for industrial applications such as real-time steel surface defect detection.

Figures 9 and 10 present a detailed comparison of the prediction results of the model before and after the improvement process. Upon examining these images, several key enhancements in the improved model become immediately apparent. The improved model demonstrates a notable increase in detection accuracy, defect type recognition accuracy, and the ability to distinguish between different types of defects. This enhanced performance is evident in the more precise and reliable identification of defects, which is crucial for accurate quality control in industrial settings.

Compared to the original model, the improved model has made significant strides in prediction accuracy. It exhibits higher confidence levels in its predictions, which is a testament to its enhanced robustness. This robustness is particularly important in dynamic industrial environments where conditions can vary widely. The improved model effectively reduces both missed detections and false positives, which are critical factors in maintaining high detection reliability. By minimizing these errors, the model shows higher stability and consistency in its performance, which is essential for real-time applications.

Moreover, the improved model's ability to handle complex and dynamic industrial environments is a significant advantage. Its enhanced practicality is particularly evident in real-time defect detection tasks, where speed and accuracy are paramount. The model's comprehensive improvements in accuracy, efficiency, and robustness make it a more reliable tool for high-precision and real-time steel surface defect detection.

In summary, the improved model not only outperforms the original model in terms of detection accuracy and robustness but also demonstrates a higher level of practicality and reliability in complex industrial applications. These enhancements collectively make the improved model a superior choice for addressing the challenges of steel surface defect detection, ensuring higher production quality and efficiency in the steel manufacturing industry.

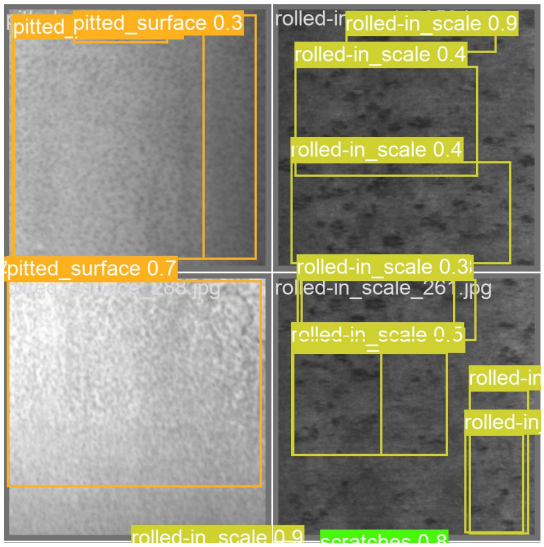


Fig. 9: Detection effect of YOLOv10 model.

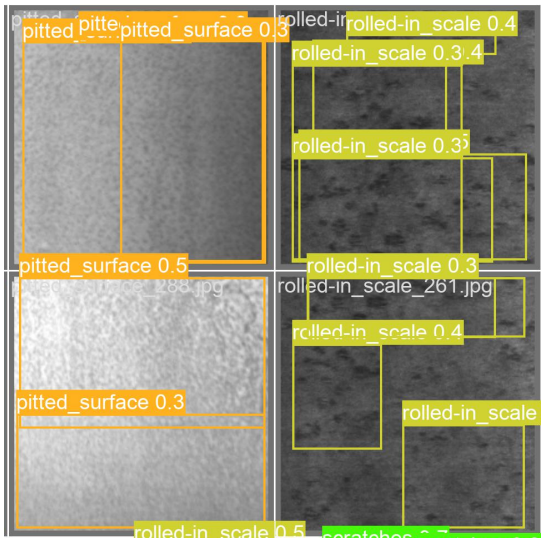


Fig. 10: The detection effect of the improved YOLOv10 model.

## D. Ablation Study

To validate the effectiveness of each improvement module in terms of object detection accuracy and model lightweighting, this paper designs four ablation experiments. All experiments were conducted under the same environment and training parameters to ensure the comparability of the results. During the experiment, both the experimental group and the control group were trained and tested, and the performance metrics for each group were recorded in detail. The experimental results are shown in Table 1.

In the first experiment, we used the original YOLOv10 model. This model achieved an mAP (mean Average Precision) value of 0.743 on the standard test set, with a parameter count of 2.7M. As a baseline, the first experiment provides a foundation for comparison in subsequent experiments.

In the second experiment, we added our custom C3_Star_EMA module to the original YOLOv10 model. This module combines the Star network and the EMA (Exponential Moving Average) module, aiming to enhance the diversity of feature flow, reduce information loss, and strengthen the interactions across channels and spatial dimensions. By doing so, the C3_Star_EMA module effectively improved the model's feature repre-

TABLE I: Ablation experiments

| | C3_Star_EMA | C2FAFF | MobileOneBlock | AP50 | parameters(M) | GFLOAPs |
|---|---|---|---|---|---|---|
| YOLOv10n | - | - | - | 74.3 | 2.7 | 8.2 |
| YOLOv10n | √ | - | - | 75.7 | 2.45 | 8.0 |
| YOLOv10n | √ | √ | - | 76.4 | 2.4 | 7.9 |
| YOLOv10n | √ | √ | √ | 79.2 | 2.25 | 6.7 |

sentation capability, raising the mAP value to 0.757 while reducing the parameter count to 2.45M. This shows that the module enhances model accuracy while effectively reducing computational complexity.

In the third experiment, we introduced the C2fAFF module. This module dynamically calculates and integrates the importance of features from different layers, enhancing the interaction of effective information and thereby improving the model's detection accuracy. With the introduction of this module, the mAP value further increased to 0.764, and the parameter count decreased to 2.4M, further demonstrating the model's balance between efficiency and accuracy.

In the fourth experiment, we incorporated the MobileOneBlock module, which decomposes standard convolution operations into depthwise convolutions and pointwise convolutions. This significantly reduces the computational load and improves the model's computational efficiency. The introduction of this module resulted in a significant increase in the mAP value to 0.792, while the parameter count decreased to 2.25M, indicating that it can notably improve detection accuracy while reducing the model's size.

To validate the overall effectiveness of this algorithm, this paper compares the improved YOLOv10 model with several classic object detection algorithms, including Faster R-CNN, YOLOv3, YOLOv5s, YOLOv7s, YOLOv8s, YOLOvX_s, and YOLOv10n. The experimental results are shown in Table 2.

The experimental data shows that the improved model outperforms other classic models in terms of both accuracy and parameter count. Specifically, compared to Faster R-CNN, the mAP increased by 0.018, while the parameter count decreased by 134.85M; compared to YOLOv3, the mAP increased by 0.018, and the parameter count decreased by 7.35M; compared to YOLOv5s, the mAP increased by 0.046, and the parameter count decreased by 5.05M; compared to YOLOv7s, the mAP increased by 0.071, and the parameter count decreased by 7.05M; compared to YOLOv8s, the mAP increased by 0.005, and the parameter count decreased by 8.85M; compared to YOLOvX_s, the mAP increased by 0.104, and the parameter count decreased by 6.65M; compared to YOLOv10n, the mAP increased by 0.049, and the parameter count decreased by 0.45M.

In summary, the improved approach proposed in this study has made significant breakthroughs in several aspects. Not only has it improved object detection accuracy, but it has also significantly reduced the model's parameter count while ensuring efficient computation. Through the collaborative optimization of various modules, such as the self-designed C3_Star_EMA module, C2fAFF module, and MobileOneBlock module, this model has further enhanced its adaptability to small targets and complex scenarios while maintaining high efficiency. Especially in tasks like steel surface defect detection, the model has demonstrated exceptional performance, proving its immense potential in practical applications.

This model has shown excellent performance across various classical object detection algorithms, especially in balancing accuracy and computational complexity. For example, despite extensive optimizations for model lightweighting, the improved YOLOv10 model still achieves higher mAP values on several commonly used benchmark datasets. Through the synergistic effect of the various modules, the model's detection capability, particularly for small targets, has been significantly enhanced, even in the presence of noise and complex backgrounds.

Additionally, the reduction in parameter count not only alleviates the computational burden but also greatly improves the model's adaptability in resource-constrained environments. This makes the model well-suited for real-time applications in industrial production scenarios, particularly for automated defect detection and quality monitoring on high-speed production lines.

Overall, the improved approach proposed in this study not only overcomes the limitations of traditional object detection methods in terms of accuracy and computational efficiency but also provides an efficient and reliable solution for industrial automation, intelligent manufacturing, and other fields. With further optimization and the expansion of application scenarios, this model is expected to deliver even greater value in practical applications and contribute to the advancement and innovation of object detection technology.

## V. Conclusion

This paper proposes an improved model based on YOLOv10n for steel surface defect detection tasks. While maintaining model lightweighting, this study significantly enhances detection accuracy through an innovative improvement strategy, achieving a 4.9% increase in mAP. Furthermore, the model's parameter count is reduced by 16.7% compared to the original version, effectively alleviating the computational burden and improving the model's adaptability in resource-constrained environments. Notably, the model's real-

TABLE II: Comparison of Detection Performance of Different Algorithms.

| Types | SSD | Fast R-CNN | YOLOv3 | YOLOv5s | YOLOv7s | YOLOX_s | YOLOv8s | YOLOv10n | OURS |
|---|---|---|---|---|---|---|---|---|---|
| crazing | 45.2 | 47.4 | 50.8 | 42.3 | 36.8 | 39.0 | 50.1 | 31.9 | 48.8 |
| inclusion | 81.8 | 78.6 | 87.0 | 86.5 | 86.5 | 80.3 | 89.1 | 83.6 | 82.6 |
| patches | 88.0 | 95.2 | 76.2 | 78.5 | 76.6 | 77.0 | 93.1 | 90.0 | 90.4 |
| pitted_surface | 84.2 | 86.9 | 92.9 | 93.0 | 96.1 | 78.6 | 78.1 | 93.2 | 95.2 |
| rolled-in_scale | 60.2 | 60.5 | 86.9 | 86.9 | 83.1 | 86.1 | 72.5 | 56.4 | 67.1 |
| scratches | 72.2 | 95.7 | 70.7 | 59.6 | 53.3 | 62.0 | 89.2 | 90.8 | 91.0 |
| mAP | 71.9 | 77.4 | 77.4 | 74.6 | 72.1 | 68.8 | 78.7 | 74.3 | 79.2 |
| parameters(M) | 21.2 | 137.1 | 9.6 | 7.3 | 9.3 | 8.9 | 11.1 | 2.7 | 2.25 |
| GFLOAPs | 62.7 | 20.2 | 23.6 | 17.0 | 26.7 | 18.2 | 28.4 | 8.2 | 6.9 |

time performance in industrial scenarios has been significantly enhanced.

Steel surface defect detection has high practical value, especially in industrial production. Rapid and accurate detection of small defects on steel surfaces is crucial for production efficiency and product quality. Traditional defect detection methods rely mostly on manual inspection or conventional image processing techniques. These methods are not only time-consuming and labor-intensive, but also struggle to maintain detection accuracy and robustness when dealing with complex backgrounds and small targets.

This research, based on an improved version of YOLOv10n, demonstrates strong performance in steel surface defect detection, particularly in detecting small targets. Steel surface defects are often tiny and susceptible to noise interference, making them difficult for traditional methods to capture efficiently. The proposed improved model addresses this challenge by employing various optimization techniques, such as the self-designed C3_Star_EMA module and the C2fAFF module, significantly enhancing sensitivity to small targets and improving detection accuracy.

Experimental validation shows that the improved model not only achieves a significant increase in detection accuracy but also exhibits considerable improvements in operational efficiency. Compared to traditional methods, the improved model enables real-time detection of steel surface defects and maintains high accuracy even in complex environments. This makes the model highly practical and operational in real-world industrial applications, especially for automated defect detection on high-speed production lines.

Overall, the proposed improved model based on YOLOv10n successfully balances detection accuracy and model lightweighting. In the task of steel surface defect detection, it demonstrates clear advantages in small target detection and real-time detection capabilities. The model not only operates stably in complex industrial environments but also significantly improves detection efficiency, offering a more efficient and reliable solution for quality control in industrial production.

## References


[1] Z. Li, X. Wei, M. Hassaballah, and et al., "A deep learning model for steel surface defect detection," Complex & Intelligent Systems, vol. 10, no. 1, pp. 885–897, 2024.

[2] M. G. Droubi, N. H. Faisal, F. Orr, and et al., "Acoustic emission method for defect detection and identification in carbon steel welded joints," Journal of constructional steel research, vol. 134, pp. 28–37, 2017.

[3] T. Dai, X. Jia, J. Zhang, and et al., "Laser ultrasonic testing for near-surface defects inspection of 316l stainless steel fabricated by laser powder bed fusion," China Foundry, vol. 18, pp. 360–368, 2021.

[4] X. Jian, I. Baillie, and S. Dixon, "Steel billet inspection using laser-emat system," Journal of Physics D: Applied Physics, vol. 40, no. 5, p. 1501.e, 2007.

[5] A. R. Dakak, Automatic defect detection in industrial CT volumes of casting. PhD thesis, Université de Lyon, 2022.

[6] A. Litvintseva, O. Evstafev, and S. Shavetov, "Real-time steel surface defect recognition based on cnn," in 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), pp. 1118–1123, IEEE, 2021.

[7] X. Shi, S. Zhou, Y. Tai, and et al., "An improved faster r-cnn for steel surface defect detection," in 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP), pp. 1–5, IEEE, 2022.

[8] B. Tang, Z. K. Song, W. Sun, and et al., "An end-to-end steel surface defect detection approach via swin transformer," IET Image Processing, vol. 17, no. 5, pp. 1334–1345, 2023.

[9] M. Jeong, M. Yang, and J. Jeong, "Hybrid-dc: A hybrid framework using resnet-50 and vision transformer for steel surface defect classification in the rolling process," Electronics, vol. 13, no. 22, p. 4467, 2024.

[10] K. Liu, A. Li, X. Wen, and et al., "Steel surface defect detection using gan and one-class classifier," in 2019 25th International Conference on Automation and Computing (ICAC), pp. 1–6, IEEE, 2019.

[11] A. Boudiaf, S. Benlahmidi, A. Dahane, and et al., "Development of hybrid models based on alexnet and machine learning approaches for strip steel surface defect classification," Journal of Failure Analysis and Prevention, vol. 24, no. 3, pp. 1376–1394, 2024.

[12] X. Zheng, W. Liu, and Y. Huang, "A novel feature extraction method based on legendre multi-wavelet transform and auto-encoder for steel surface defect classification," IEEE Access, vol. 12, pp. 5092–5102, 2024.

[13] R. Zaghdoudi, A. Bouguettaya, and A. Boudiaf, "Steel surface defect recognition using classifier combination," The International Journal of Advanced Manufacturing Technology, vol. 132, no. 7, pp. 3489–3505, 2024.

[14] Q. Jiyang and W. Yufan, "Strip steel surface defect detection algorithm based on improved faster r-cnn," CHINA WELDING, vol. 33, no. 2, pp. 11–22, 2024.

[15] C. Liang, Z. Z. Wang, X. L. Liu, and et al., "Sdd-net: A steel surface defect detection method based on contextual enhancement and multiscale feature fusion," IEEE Access,

2024.
[16] A. A. M. S. Ibrahim and J. R. Tapamo, "Transfer learning-based approach using new convolutional neural network classifier for steel surface defects classification," Scientific African, vol. 23, p. e02066, 2024.
[17] X. Dong, Y. Li, L. Fu, and et al., "Edge-aware interactive refinement network for strip steel surface defects detection," Measurement Science and Technology, vol. 36, no. 1, p. 016222, 2024.
[18] Y. He, S. Li, X. Wen, and et al., "A high-quality sample generation method for improving steel surface defect inspection," Sensors, vol. 24, no. 8, p. 2642, 2024.
[19] P. K. A. Vasu, J. Gabriel, J. Zhu, and et al., "Mobileone: An improved one millisecond mobile backbone," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7907–7917, IEEE, 2023.
[20] X. Ma, X. Dai, Y. Bai, and et al., "Rewrite the stars," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5694–5703, IEEE, 2024.
[21] D. Zhou, Q. Hou, Y. Chen, and et al., "Rethinking bottleneck structure for efficient mobile network design," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pp. 680–697, Springer International Publishing, 2020.
[22] D. Ouyang, S. He, G. Zhang, and 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), pp. 1–5, IEEE, 2023.
[23] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, "Attentional feature fusion," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3560–3569, IEEE, 2021.