Steel Surface Defect Detection Based on YOLOv10

Luyu Sun, Yujun Zhang*

Abstract—To improve the accuracy of steel surface defect detection, this study proposes an improved multi-directional optimization model based on the YOLOv10n algorithm. First, we introduce innovations to the convolution (C2F) module in YOLOv10n with MSMHSA (Multi-Scale Multi-Head Self-Attention) and EMA (Enhanced Multi-Scale Attention) modules. These modules enhance learning and expression capabilities by reshaping across different scales, multiple attention heads, and channel dimensions. Second, a Multi-Dilated Channel Refinement (MDCR) module is employed. The MDCR module captures spatial features across various receptive field ranges through multiple depthwise separable convolutions, enabling more effective multi-level feature integration. Finally, a Context Aggregation module is embedded within the neck network as a generic building block for multi-head context aggregation, leveraging the inductive bias of local convolution operations to facilitate rapid convergence. Experimental results show that the improved model achieves a mean Average Precision (mAP) of 78.7% in steel surface defect detection tasks, marking a significant improvement of 4.3% over the original YOLOv10n model. In practical applications, the improved model can quickly and accurately locate and classify various steel surface defects, meeting the needs of steel surface defect detection in industrial production.

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

I. Introduction

STEEL production is an important standard for measuring a country's industrial capabilities. Steel is widely used across various industries, including but not limited to construction [1], transportation, energy, and military industries. Therefore, the quality of steel production has become a critical aspect of industrial manufacturing. With the development of artificial intelligence technology and the spread of intelligent manufacturing projects, intelligent detection systems have increasingly become a popular topic in the industrial field [2]. Steel surface defect detection can be divided into contact detection and non-contact detection. Contact detection involves receiving data by directly contacting the sample surface with the sensor elements of the detection device. In contrast, non-contact detection collects parameter information from the sample surface using electromagnetic and photoelectric processes without any surface contact [3].

Magnetic Particle Testing (MPT) and Liquid Penetrant Testing (LPT) are two methods of contact detection. MPT is based on the magnetic field changes formed in magnetic materials under an external magnetic field. By applying a magnetic field to the object under inspection, any surface or near-surface defects cause a disruption in the magnetic lines of force, attracting magnetic particles to form visible defect indications. MPT is primarily used for ferromagnetic materials such as iron, nickel, cobalt, and their alloys. LPT, on the other hand, involves using a low-viscosity penetrant liquid (usually fluorescent or dyed) that enters open surface defects in the material, such as cracks. After proper cleaning and developing, the defect location becomes visible due to a clear color contrast. LPT can be applied to almost all non-porous materials, including metals, plastics, and ceramics.

In non-contact detection methods, using image sensors for visual inspection is more advantageous, as it enables high-speed contactless inspection and simple visual analysis with only standard computers and dedicated image processors. In recent years, with the rapid development of computer technology, machine vision technology has also made significant progress. Based on this, steel surface defect technology is evolving towards automation and artificial intelligence. By utilizing high-resolution cameras, deep learning models, and image processing algorithms, steel surface defect detection can achieve automatic defect detection and intelligent classification. The application of machine vision technology eliminates human factors, greatly improving inspection speed and accuracy.

II. Related work

Many researchers have used traditional machine learning techniques to achieve steel surface defect detection. Zaghdoudi R et al. proposed an efficient steel surface defect classification system, which achieves excellent classification accuracy. The proposed system applies the binary Gabor pattern (BGP) descriptor, used for the first time in steel surface defect classification, to extract local texture features from defect images. Then, a principal component analysis (PCA)-based dimensionality reduction procedure is used to obtain a compact representation of the defect image. Finally, an SVM multi-class classifier provides the final decision [4]. Varsha A et al. extracted texture features facilitated by gray-level co-occurrence matrix (GLCM) analysis. GLCM explores spatial relationships between neighboring pixels to quantify image intensity. The extracted features are sent as input to a Random Forest Classifier to optimize the model, and SHAP plots are used to explain model output [5]. Hao Z et al. proposed a new combined damage detection method to classify various degrees of cross-sectional loss due to damage, such as steel corrosion, using a k-Nearest Neighbor (kNN) machine learning classifier.

Manuscript received October 30, 2024.revised February 21, 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).

A finite element (FE) model of an in-service railway bridge was developed and validated using vibration data from field tests, and these combined FE and field data were used for training and testing [6]. Hwang Y I et al. designed an ultrasonic non-destructive testing method to detect surface defects in 304SS steel plates. It involves using linear discriminant analysis (LDA) of pixel information from short-time Fourier transform (STFT) generated from GW data, with differences in STFT pixel counts between sound and defective samples as the main factor distinguishing the two groups [7]. Huang J et al. proposed an online acoustic emission pattern recognition technique based on multi-fractal characteristics (MF-DBSCAN). It first extracts damage features of low-carbon steel based on multifractal analysis, which is combined with the mechanical behavior characteristics of metal materials and typical scanning electron microscope (SEM) fracture images. Finally, the DBSCAN method is used for unsupervised clustering based on multifractal characteristics, training the model for online pattern recognition of unknown data to identify in-service crack patterns in power systems [8]. Kim B et al. developed a quantitative assessment method that evaluates rust formation on steel plates by using k-means clustering in the corroded areas of a given image. k-means clustering for automatic corrosion detection is based on GrabCut segmentation and Gaussian mixture models (GMM). The color of the corroded surface of the cut edge area is quantitatively analyzed based on the HSV (Hue, Saturation, Value) color space [9]. Ye X et al. used particle swarm optimization support vector machine (PSO-SVM) for accurate classification of hot-rolled strip steel surface defects. First, effective preprocessing of the defect image is performed; then, local binary patterns (LBP), histograms of oriented gradients (HOG), and gray-level co-occurrence matrices (GLCM) are extracted. Principal component analysis (PCA) is applied for feature dimensionality reduction. Finally, an SVM classification model is established, with parameters optimized using particle swarm optimization (PSO) [10]. In recent years, deep learning has become mainstream in object detection. Compared to traditional machine learning, deep learning can learn more complex patterns from large datasets. Deep learning has been applied across numerous fields, including but not limited to drone target recognition and addressing environmental issues. Due to its tremendous potential in target recognition, many researchers have applied deep learning in the field of steel surface defect detection. Litvintseva A et al. developed a method for real-time identification and classification of metal surface defects through images. This algorithm aims to improve production standards and process efficiency. Litvintseva A et al. applied deep learning (DL) and computer vision (CV) technologies to address defect detection on steel sheets, comparing convolutional neural network (CNN) architectures and identifying various steel defects. The outcome of this work is a comparative analysis of DL models, selecting an algorithm designed for real-time defect search and classification. Using a CNN model, a tool can be created that greatly facilitates the work [11].

Shi X, Zhou S et al. introduced an improved network based on Faster R-CNN. This model adopts the ConvNeXt architecture as the feature extraction structure within Faster R-CNN. Furthermore, a Convolutional Block Attention Module (CBAM) is used to enhance the model's focus on surface defects while suppressing features from complex backgrounds. Finally, the k-means clustering algorithm is utilized to generate anchor points better suited for surface defects [12].

Lin C Y et al. proposed a deep learning approach for automatic defect detection on steel surfaces. The system architecture is divided into two parts. The first part uses a modified Single Shot MultiBox Detector (SSD) model to identify potential defects. Then, a deep residual network (ResNet) is used to classify three types of defects: Rust, Scar, and Sponge [13].

Liu B et al. proposed an improved deep learning method based on existing techniques, called low-pass U-Net, to further enhance defect segmentation performance. First, a low-pass filter is implemented in the encoder before downsampling to prevent aliasing and isolate high-frequency information. The high-frequency features are transmitted to the decoder to aid segmentation. An innovative adaptive variance Gaussian low-pass layer is then used to generate different filters for each spatial position of the feature map, reducing computational resource usage. Finally, an improved Hypercolumn module is used at the end of the decoder to upsample and fuse feature maps at different resolutions, with Subpixel replacing bilinear interpolation to optimize the upsampling results [14].

Yin T et al. improved the R-CNN algorithm by incorporating a new model. In the Faster R-CNN network, a Feature Pyramid Network (FPN) is added, allowing the network to integrate both high-level and low-level feature information. Additionally, the Region of Interest (RoI) Pooling module is replaced with RoI Alignment to reduce quantization error, helping to improve mean Average Precision (mAP). Cycle GAN is used for data augmentation, and a multi-layer RoI alignment is introduced to address extreme aspect ratio issues [15].

Liu X et al. proposed a surface defect detection method combining an attention mechanism with a multi-feature fusion network. This method uses the traditional SSD model as the basic framework, selecting the ResNet50 network for feature extraction. The fusion of low-level and high-level features is complementary, improving detection accuracy. Furthermore, a channel attention mechanism is introduced to filter and retain important information, reducing computational load and increasing detection speed [16].

Li S et al. proposed a hybrid network architecture (CNN-T) that combines a CNN with a Transformer encoder. The CNN-T network has strong inductive bias and global modeling capabilities. The CNN first extracts low-level and local features from the image, and then the Transformer encoder globally models these features to capture abstract and high-level semantic information. Finally, these are sent to a multi-layer perceptron classifier for classification [17].

Although the above methods bring numerous inno-

vations in various aspects, they also present certain limitations. Traditional machine learning often relies on manual feature extraction, which cannot eliminate human factors. When dealing with complex data, such as images or natural language, traditional algorithms may struggle to capture latent patterns. Furthermore, Convolutional Neural Networks (CNNs) require a large number of samples and substantial computational resources during training, which significantly increases costs. When training samples are insufficient, CNNs are prone to overfitting. In industrial production environments, computational resources are extremely limited, which can lead to slower detection speeds, decreased accuracy, and even impact production efficiency. Given these circumstances, this paper proposes an improved YOLOv10 algorithm designed to enhance the accuracy of steel surface defect detection. The main contributions of this paper are as follows:

1. A self-fusion C2f module is integrated into the backbone network to capture features at different levels, accommodating various details and contextual information. This improvement significantly enhances the mAP by approximately 3.2 percentage points.

2. The use of a Multi-Dilated Channel Refinement (MDCR) module within the backbone network enhances feature representation and more effectively captures multi-dimensional feature information, further improving detection accuracy. Based on the previous step, the mAP increased by approximately 0.7 percentage points.

3. Embedding the Context Aggregation module into the backbone network enhances the model's comprehension and performance by aggregating contextual information, leading to an additional mAP increase of approximately 0.5 percentage points.

III. Method Introduction

Faster R-CNN, SSD, RetinaNet, and YOLO are all classic models in object detection. However, Faster R-CNN has high computational complexity, limited ability to detect small objects, and is not suitable for real-time detection environments. SSD has lower accuracy than two-stage detectors and a complex structure, with high complexity in model training and inference. RetinaNet requires long training times and complex hyperparameter tuning. YOLOv10, on the other hand, inherits the advantages of the YOLO series. Compared with previous versions, YOLOv10 improves model architecture and adopts a more sophisticated loss function, further enhancing detection accuracy, especially in detecting small objects and dense scenes. It also optimizes model size and computational efficiency, resulting in faster inference speeds suitable for real-time applications. YOLOv10 has stronger generalization ability and uses a more advanced feature extraction network, capable of capturing richer contextual information, thus improving detection performance. The current YOLOv10 versions include: YOLOv10b, YOLOv10l, YOLOv10m, YOLOv10n, YOLOv10s, and YOLOv10x. These versions mainly differ in model size, complexity, application scenarios, and computational resource requirements. In industrial production environments, such as steel production, computational resources are extremely limited, and fast object recognition is required. YOLOv10n has been specially optimized for ultra-low-resource environments, offering extremely fast speeds, making it highly suitable for real-time tasks like steel surface defect detection. Therefore, this paper chooses to modify YOLOv10n. The architecture of YOLOv10 is shown in Figure 1.The algorithm mainly consists of three parts: the backbone, neck, and head. The YOLOv10 backbone module is built on the Efficient Layer Aggregation Network (ELAN)[18], focusing on multi-level feature extraction while maintaining rich feature representation capabilities with reduced parameters and computation. The Compact Inverted Block (CIB) combines depthwise convolution and pointwise convolution to process the mixed spatial and channel information, effectively reducing computation while retaining rich feature information. Unlike other YOLO series, the most notable feature of the YOLOv10 backbone is the separation of spatial downsampling and channel transformation. Typically, YOLO models perform both operations simultaneously using 3×3 convolutions, while YOLOv10 first adjusts the channel dimensions with pointwise convolutions and then performs spatial downsampling with depthwise convolutions. This design reduces computational complexity and parameter count, significantly enhancing efficiency, especially in larger networks. The "rank-guided" mechanism in the backbone analyzes the intrinsic rank at each stage, identifying stages with high redundancy and replacing them with more efficient modules. This strategy reduces network redundancy without impacting performance, ensuring that computational resources are allocated to the most needed areas. Finally, the backbone optimizes computational efficiency by decoupling spatial downsampling from channel dimension reduction, which not only decreases FLOPs (floating point operations) but also ensures high-quality feature retention, particularly yielding good results in detecting small objects.

The Head module is a critical component of the model responsible for object classification, bounding box regression, and confidence prediction. By integrating multi-scale feature fusion mechanisms, it processes features from different resolution levels to achieve precise detection of objects of various scales. The classification branch predicts object category probabilities, the regression branch accurately locates bounding boxes, and the confidence branch evaluates whether the predicted box contains a valid object and its reliability. To enhance detection performance, the Head module adopts a lightweight design, significantly reducing computational overhead to meet real-time requirements in embedded and edge computing environments. Additionally, some versions incorporate an adaptive feature fusion module, which dynamically adjusts weights based on the characteristics of the input image, optimizing detection performance in complex scenarios. By leveraging optimized loss functions, the Head module further balances classification accuracy and bounding box regression precision, delivering excellent robustness and resource efficiency. With multi-scale feature prediction, adaptive fusion, and

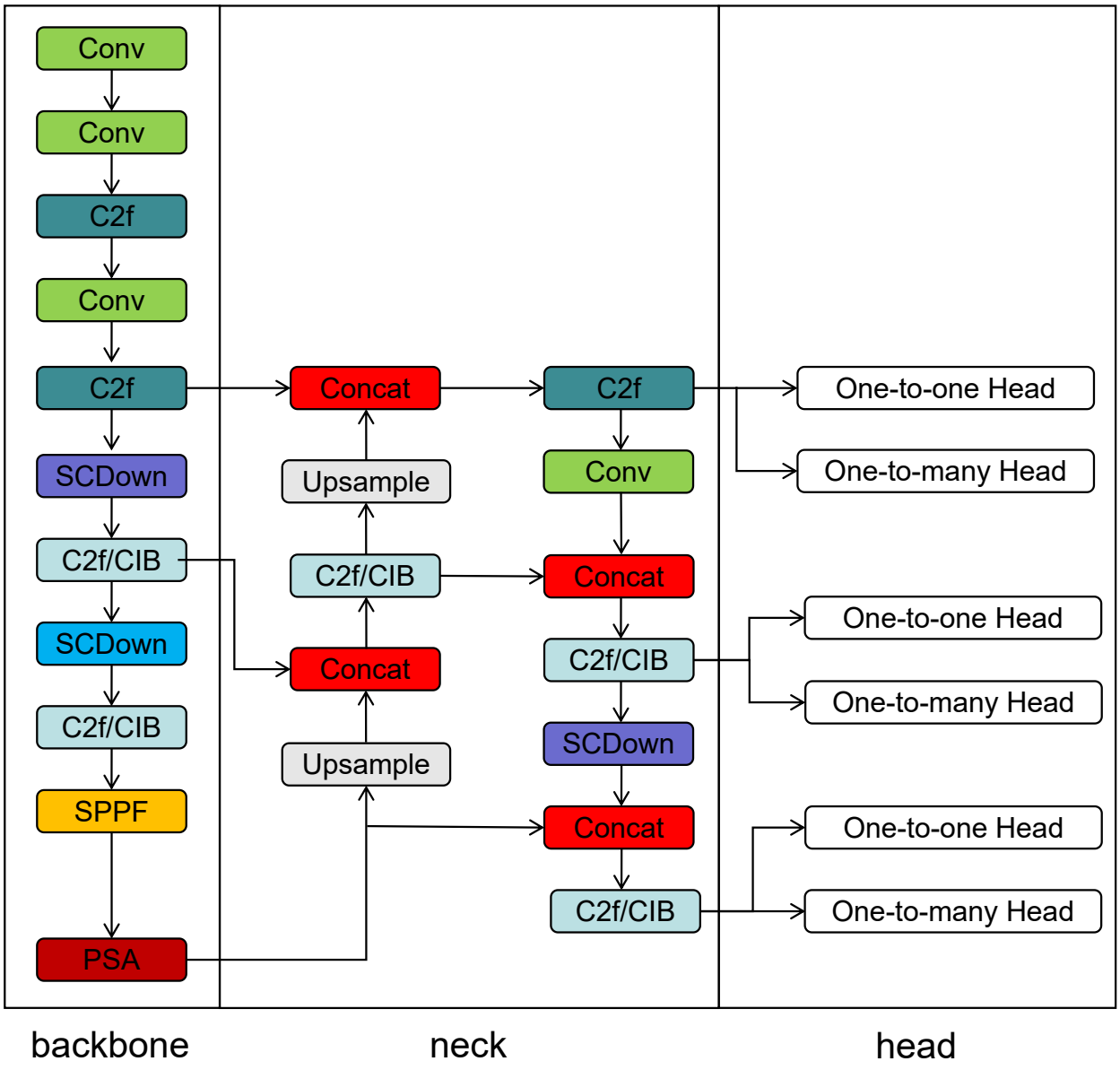

Fig. 1: YOLOv10 network architecture

efficiency, the Head module performs exceptionally well in applications like steel surface defect detection, providing a reliable solution for real-time object detection.

The neck module integrates multi-scale features from the backbone to assist the detection head in accurately classifying objects and regressing bounding boxes. This network primarily optimizes multi-scale feature integration through Feature Pyramid Networks (FPN)[19], Path Aggregation Networks (PAN)[20], and efficient feature fusion strategies. This not only enhances the model's performance in detecting objects of various sizes but also significantly reduces computational costs. FPN is a bottom-up feature fusion mechanism that combines feature layers of different resolutions extracted from the backbone. By integrating high-level and low-level features, the model can detect objects of varying scales simultaneously. In addition to FPN, the neck network also incorporates a Path Aggregation Network (PAN), which further enhances feature fusion capabilities. It employs a top-down feature transfer approach to pass features from high-resolution layers to low-resolution layers, allowing low-level features to obtain global context information. Additionally, the neck network employs efficient strategies during feature fusion to reduce computational overhead and improve performance. Especially in environments with limited computational resources (such as embedded devices or edge computing), these optimizations enable YOLOv10 to significantly reduce inference time while maintaining accuracy. Some versions of the neck network also include an adaptive feature fusion module, which can dynamically adjust the weights of feature fusion based on the different characteristics of the input image, optimizing detection performance in various scenarios. This adaptability helps improve the model's robustness and detection accuracy in complex environments.

The improved network architecture is shown in Figure 2. First, a Context Aggregation module is added to the neck network to aggregate contextual information and enhance feature representation. Then, a Multi-

Dimensional Context Relation (MDCR) module is incorporated to strengthen the model's ability to capture context and improve target detection accuracy across multiple scales. Finally, a C2f_MSMHSA_EMA module is introduced, which integrates the C2f module, a Multi-Scale Multi-Head Self-Attention (MSMHSA) mechanism, and an Efficient Multi-Scale Attention (EMA) module to enhance the model's multi-scale feature fusion capabilities. By improving YOLOv10 in these three aspects, the model's robustness and accuracy in detecting variations in defect types, sizes, and positions are significantly enhanced.

A. C2f_MSMHSA_EMA

C2f is a commonly used methodology or mechanism in computer vision and deep learning tasks. It aims to enhance model performance and efficiency through staged processing, transitioning from coarse global analysis to fine local optimization. In steel surface defect detection, C2f leverages multi-scale feature extraction and staged coarse-to-fine processing to accommodate the diversity and size variations of defects, significantly improving detection accuracy. It performs exceptionally well in detecting small targets and complex defects. Additionally, C2f enhances model robustness by integrating global and local features, effectively addressing complex backgrounds and environmental noise. Its lightweight design meets the real-time detection requirements of production lines while reducing false positives and false negatives, thereby improving detection reliability.

The Multi-Scale Multi-Head Self-Attention mechanism consists of Multi-Scale Processing and Multi-Head Self-Attention. MSMHSA first processes and extracts features at different scales through Multi-Scale to accommodate objects of varying sizes in the image, capturing both details and global structures. Multi-Scale utilizes convolutional layers with different dilation rates to extract multi-scale features from the image, with each feature representing different receptive field sizes, helping the model to model from local details to global features. Then, the Multi-Head Self-Attention mechanism is applied to the features at each scale. Multi-Head Self-Attention captures dependencies at different positions in the image through multiple parallel attention heads, allowing the model to perform global feature modeling from multiple perspectives. The formulas for the self-attention mechanism is as follow:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Here, Q, K, and V represent the matrix forms of query, key, and value, respectively.$d_k$represents the dimension of the key vector. The formulas for multi-head self-attention are as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{2}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{3}$$

Here, $W_i^Q \in R^{d_{\text{model}} \times d_k}$, $W_i^K \in R^{d_{\text{model}} \times d_k}$, $W_i^V \in R^{d_{\text{model}} \times d_v}$, $W^O \in R^{hd_v \times d_{\text{model}}}$, where d_{model} represents the sequence encoding length.

The EMA module [21] is an efficient multi-scale attention mechanism capable of simultaneously capturing channel and spatial information, effectively enhancing feature representation without adding significant parameters or computational cost. By combining channel and spatial information, it achieves information retention along the channel dimension while reducing computational burden. This combination helps capture cross-channel relationships in feature representation while avoiding the reduction of channel dimensions, thus improving model performance. Structurally, the EMA module adopts a multi-scale parallel subnetwork structure, which includes a parallel subnetwork handling a 1x1 convolution kernel and another handling a 3x3 convolution kernel. This structure effectively captures cross-dimensional interactions and establishes dependencies between different dimensions, thereby enhancing feature representation. The EMA module is an improvement and optimization based on the Coordinate Attention (CA) module. The CA module achieves fusion of cross-channel and spatial information by embedding positional information into the channel attention map. Building on this foundation, the EMA module further develops this concept, capturing cross-dimensional interactions through parallel subnetwork blocks and establishing dependencies across different dimensions. The EMA module utilizes a parallel subnetwork design, which aids in the aggregation and interaction of features, thereby enhancing the model's capability to model long-distance dependencies. This design avoids extensive sequential processing and deep layers, making the model more efficient and effective. Figure 3 shows the structure of the EMA module.

This paper integrates the concepts of MSMHSA and EMA into the C2f module of YOLOv10, resulting in the proposed C2f_MSMHSA_EMA layer. The neck network aims to extract multi-scale features and further fuse these features to enhance the model's object detection capability. These feature details are crucial for understanding the overall context of the entire image. Therefore, the C2f_MSMHSA_EMA module is added to the neck network's final layer. The introduction of C2f_MSMHSA_EMA effectively enhances the extraction of contextual features and multi-level features. With the incorporation of C2f_MSMHSA_EMA into YOLOv10, the model can capture the boundaries of target objects and detailed information of complex shapes more accurately, especially for detecting smaller objects.

B. ContextAggregation

To further enhance YOLOv10's extraction of contextual information features, this paper incorporates the ContextAggregation module [22] into the neck. The ContextAggregation module combines an object's local features with the surrounding background information, allowing for more accurate target localization. When detecting small objects, relying solely on local features may lead to false detections; however, by introducing

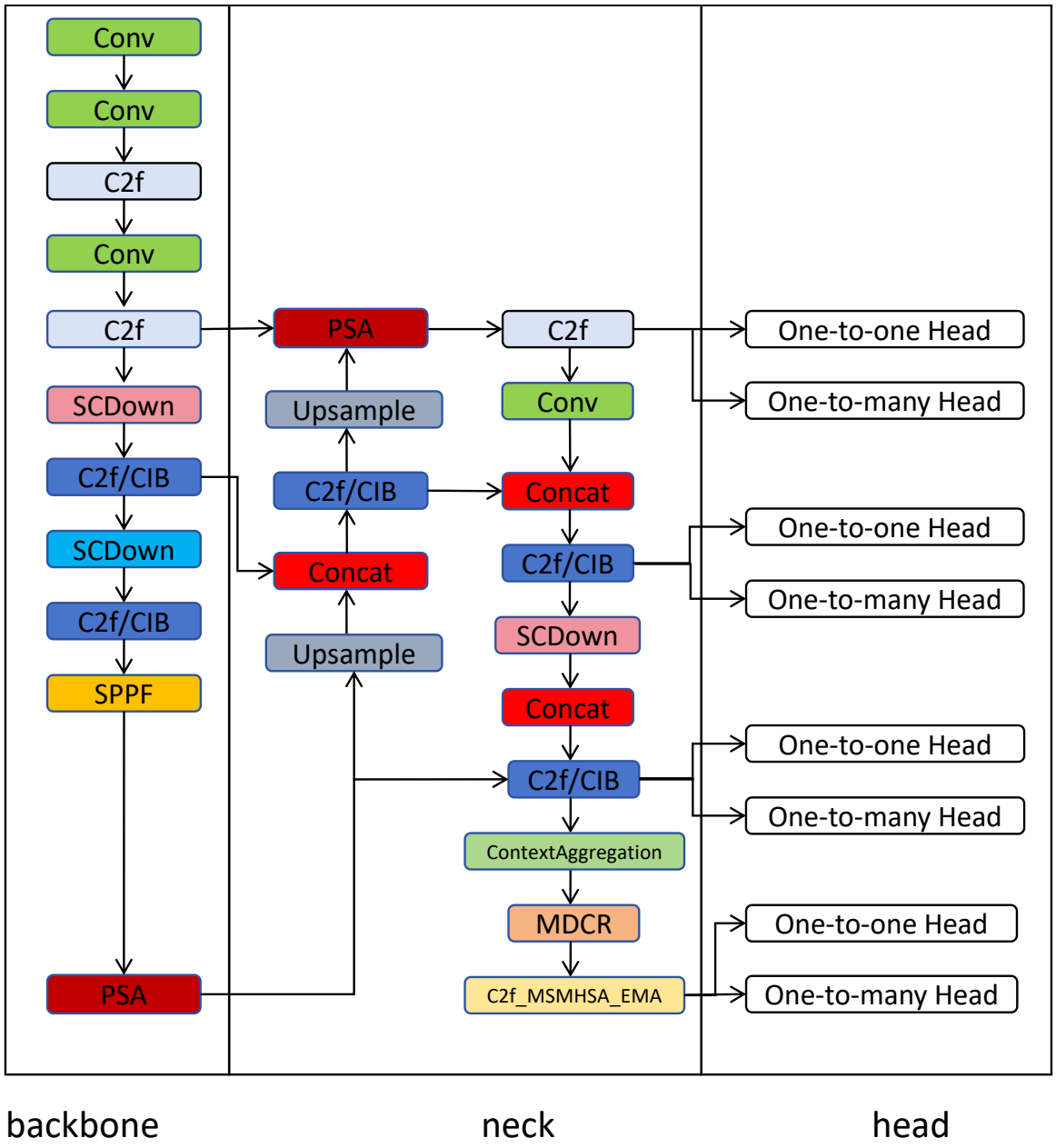

Fig. 2: Improved YOLOv10 network architecture

contextual information from the surrounding area, the model can better distinguish between objects and background, reducing detection errors. Additionally, in real-world scenarios, target objects may be partially occluded by other objects. By aggregating contextual information, the model can use the unobstructed surrounding areas to infer the complete shape and position of the target object, enhancing detection performance in occluded scenes and improving the model's practicality in real-world applications. The formula for the ContextAggregation module is as follow:

$$\mathbf{Y} = (\mathcal{A}\mathbf{V})\mathbf{W}_1 + \mathbf{X}, \tag{4}$$

Here, X and Y are the input and output vectors of the layer under consideration,$\mathcal{A} \in R^{N\times N}$ is the affinity matrix, representing the neighborhood for context aggregation.Here,$\mathbf{V} \in R^{N\times C}$ is a transformation of X obtained through a linear projection $\mathbf{V} = \mathbf{X}\mathbf{W}_2$.$\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable parameters.$\mathcal{A}_{ij}$ is the affinity value between X_i and X_j.By multiplying the affinity matrix with V, information can be propagated across features based on affinity values. The modeling capability of this context aggregation module can be enhanced by introducing multiple affinity matrices, allowing the network to acquire contextual information across x through several paths.Let $\{V^i \in R^{N\times \frac{C}{M}} \mid i = 1, \ldots, M\}$ be a slice of V, where M is the number of affinity matrices, also known as the number of heads.The multi-head version of the formula is as follow:

$$\mathbf{Y} = \text{Concat}(\mathcal{A}_1\mathbf{V}_1, \ldots, \mathcal{A}_M\mathbf{V}_M)\mathbf{W}_2 + \mathbf{X} \tag{5}$$

Here, $\mathcal{A}_m$represents the affinity matrix for each head.Compared to the single-head version, different $\mathcal{A}_m$ matrices can potentially capture different relationships in the feature space, thereby increasing the representa-

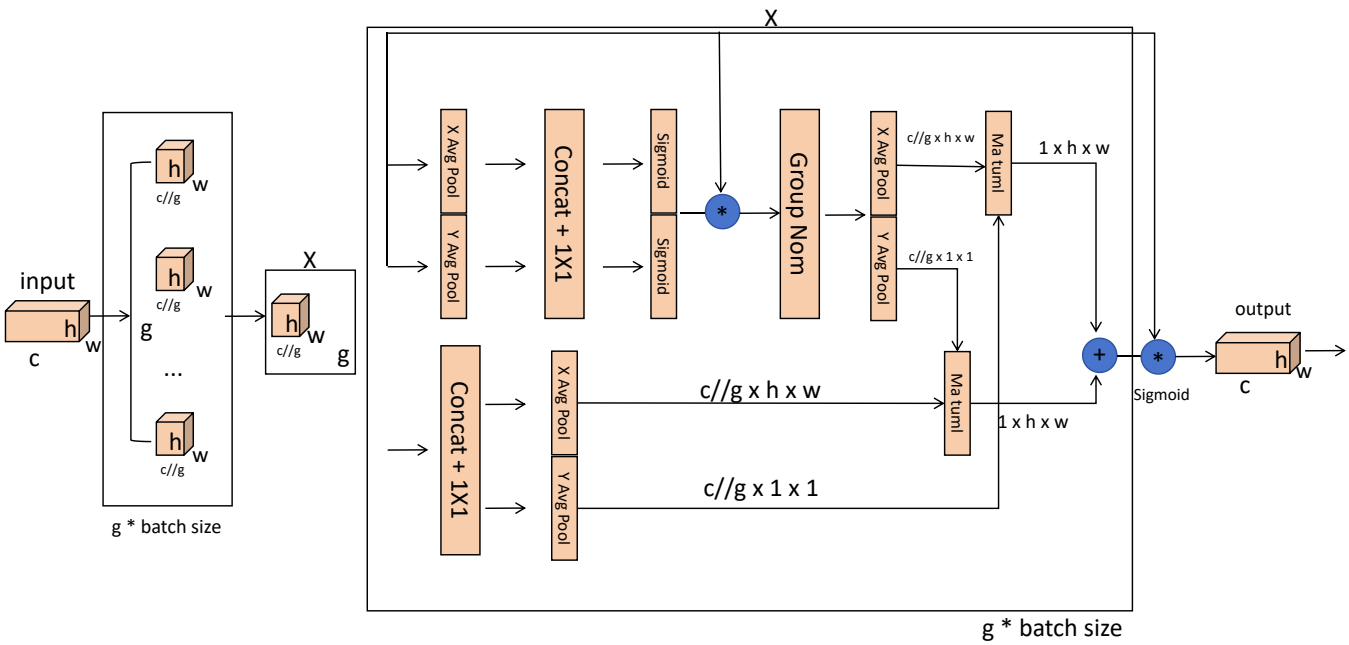

Fig. 3: EMA network architecture."g" means the divided groups, "X Avg Pool" represents the 1D horizontal global pooling and "Y Avg Pool" indicates the 1D vertical global pooling, respectively.

tion capacity of context aggregation. Note that during the context aggregation process using affinity matrices, only spatial information is propagated; no cross-channel information exchange occurs in the affinity matrix multiplication, and there are no nonlinear activation functions involved.

Context Aggregation can operate in a manner similar to the attention mechanism. First, the model generates three different vectors for each pixel or region from the feature map: Query, Key, and Value. These vectors respectively represent the information needs, semantic features, and contributed contextual information of that region. The model then calculates the correlation scores between the query of each pixel (or region) and the key vectors of other pixels, similar toContext Aggregation operates in a manner similar to the attention mechanism. Specifically, the model first generates three distinct vectors for each pixel or region in the feature map: Query, Key, and Value. The Query vector represents the information needs of the pixel or region, the Key vector encodes the semantic features of other regions, and the Value vector represents the contextual information contributed by each region. In this way, the model can extract the necessary key information from local regions while integrating global context to improve the understanding of different parts of the image.

Next, the model calculates the correlation scores between the Query vector of each pixel (or region) and the Key vectors of other pixels (or regions), similar to the dot product calculation in traditional attention mechanisms. This calculation quantifies the relationship between different regions, determining how important each pixel or region is in the global context. The scores are then normalized using a softmax function, transforming them into attention weights that reflect the relative contribution of each region to the others.

Finally, using the normalized attention scores, the model applies weighted summation to the Value vectors of each pixel or region, aggregating relevant contextual information from different regions. This process allows the model to combine local region information with global context, ensuring that it not only focuses on local details but also leverages global information for a more comprehensive understanding and reasoning. This step enables the model to better capture the relationships between global and local contexts, enhancing its ability to understand and recognize targets. the dot product calculation in the attention mechanism. These scores are then normalized using a softmax function to obtain the attention weights of each pixel relative to other regions. Finally, by using the normalized attention scores, the value vectors of each pixel or region are weighted and summed, thereby aggregating relevant contextual information. This step allows the integration of local area information with global context, enabling the model to leverage global information to enhance its understanding of individual targets.

C. MDCR

MDCR [23] enhances multi-scale feature extraction and channel information representation, capturing features across different receptive field ranges. It more accurately models the differences between objects and backgrounds, improving its ability to locate small objects. The organic combination of these modules enhances detection performance and robustness. MDCR uses multiple depthwise separable convolution layers with different dilation rates to capture spatial features across various receptive field sizes, enabling a more detailed extraction of differences between objects and backgrounds, which improves detection performance on small targets. Figure 4 shows the structure of the MDCR

module.

The MDCR (Multi-Dilation Contextual Representation) module is typically represented through its core operations. The process can be divided into two major parts: dividing the input feature map along the channel dimension and applying depthwise separable dilated convolution. These two parts are realized through four steps: input feature division, depthwise separable dilated convolution, channel splitting and recombination, and pointwise convolution with aggregation, to achieve the functionality of MDCR.

1. Input Feature Division: The formula structure is as follows:

$$\mathbf{F}_a \in R^{H\times W\times C} \Rightarrow (\mathbf{a}_i)_{i=1}^{4} \in R^{H\times W\times \frac{C}{4}} \tag{6}$$

The MDCR module first divides the input feature map $\mathbf{F}_a$ along the channel dimension into four different "heads." Each head focuses on a different subset of features to achieve multi-scale feature extraction.

2. Depthwise Separable Dilated Convolution: The formula structure is as follows:

$$\mathbf{a}_i' = DDWConv(\mathbf{a}_i), \quad i \in \{1,2,3,4\} \tag{7}$$

where

$$\mathbf{a}_i')_{i=1}^{4} \in R^{H\times W\times \frac{C}{4}} \tag{8}$$

Each head undergoes depthwise separable dilated convolution. Here, the dilated convolution uses different dilation rates d_1, d_2, d_3, d_4 to control the receptive field size for each head. By using different dilation rates, MDCR can introduce spacing in the feature map through dilated convolution to capture a larger range of contextual information. This enables it to capture spatial features at various scales, from small to large, while reducing computational complexity and improving efficiency.

3. Channel Splitting and Recombination: The formula structure is as follows:

$$\mathbf{a}_j^i)_{j=1}^{\frac{C}{4}} \in R^{H\times W\times 1} \Rightarrow (\mathbf{h}_j)_{j=1}^{\frac{C}{4}} \in R^{H\times W\times 4} \tag{9}$$

The features of each head $\mathbf{a}_i'$ after dilated convolution are further split into single channels, each with dimensions $H \times W \times 1$. These single-channel features are interleaved across different heads to form new feature representations $(\mathbf{h}_j)_{j=1}^{\frac{C}{4}} \in R^{H\times W\times 4}$. Through this operation, the MDCR module can establish connections between features at different scales and enhance the diversity of multi-scale features. This interleaving ensures that information from different dilation rates is fused, enabling better representation of complex objects and backgrounds in scenes.

4. Pointwise Convolution and Aggregation: The formula structure is as follows:

$$\mathbf{F}_o = \delta(\mathcal{B}(W_{\text{outer}}([\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_j]))) \tag{10}$$

Finally, the MDCR module applies pointwise convolution to fuse information within and across groups. Pointwise convolution is a 1×1 convolution that transforms the channel dimension without changing spatial resolution. Additionally, MDCR uses batch normalization (denoted as $\mathcal{B}$) and an activation function (such as ReLU, denoted as δ) to enhance the stability and non-linearity of feature representation. The output of this step, $\mathbf{F}_o \in R^{H\times W\times C}$, is the final result of the MDCR module. It achieves lightweight and efficient feature representation by aggregating multi-scale and multi-head information.

IV. Experimental Design and Implementation

A. Dataset Introduction

The NEU-DET dataset (Northeastern University Detection Dataset) is a standard dataset for surface defect detection, primarily used in the research and application of metal surface defect detection. Released by a research team at Northeastern University, it contains images of six common types of metal surface defects. The six different types of surface defects in the NEU-DET dataset include:

Crazing: Discontinuous cracks on the surface of rolled pieces, spreading out in a lightning shape from a central point. Inclusion: Thin layer folds on the surface of sheet steel, often grayish-white in appearance, with varying sizes, shapes, and irregular distributions across the steel surface. Scratches: Mechanical damage on the surface of rolled pieces, varying in length, width, and depth, often appearing along or perpendicular to the rolling direction. Rolled-in Scale: Small spots, fish-scale shapes, streaks, or irregular blocks of oxidized material distributed on either or both surfaces of sheet steel, often accompanied by a rough, dimpled texture. Pitted Surface: Localized or continuous rough areas on the surface of sheet steel, which may resemble orange peel in severe cases. It can appear on both surfaces, with uneven density along the steel strip's length. Patches: Spots or large areas of discoloration on the surface of the sheet steel, sometimes showing radiating patterns at certain angles.

Each defect type is well-represented in the dataset, displaying varied morphological characteristics that facilitate the study of model performance across different defect types. The images in the dataset are of 200x200-pixel resolution and are all grayscale. The uniformity in image specifications makes preprocessing and model input convenient. Each defect type in the dataset contains 300 images, totaling 1800 images. The NEU-DET dataset is divided in an 8:1:1 ratio into training, testing, and validation sets. The training set includes 1440 images, the test set contains 180 images, and the validation set has 180 images. This split ratio ensures the diversity and representativeness of the dataset while providing a sufficient sample size for algorithm training, testing, and validation.

B. Evaluation Metrics

In this paper, precision (P), recall (R), and mAP@0.5 are used as evaluation metrics. Precision (P) measures the accuracy of the model's prediction of "positive samples" during detection. In object detection tasks, precision is defined as the proportion of all "positive" boxes (i.e., boxes detected as target objects) that actually contain target objects. Recall (R) measures the model's ability to detect target objects during the detection process.

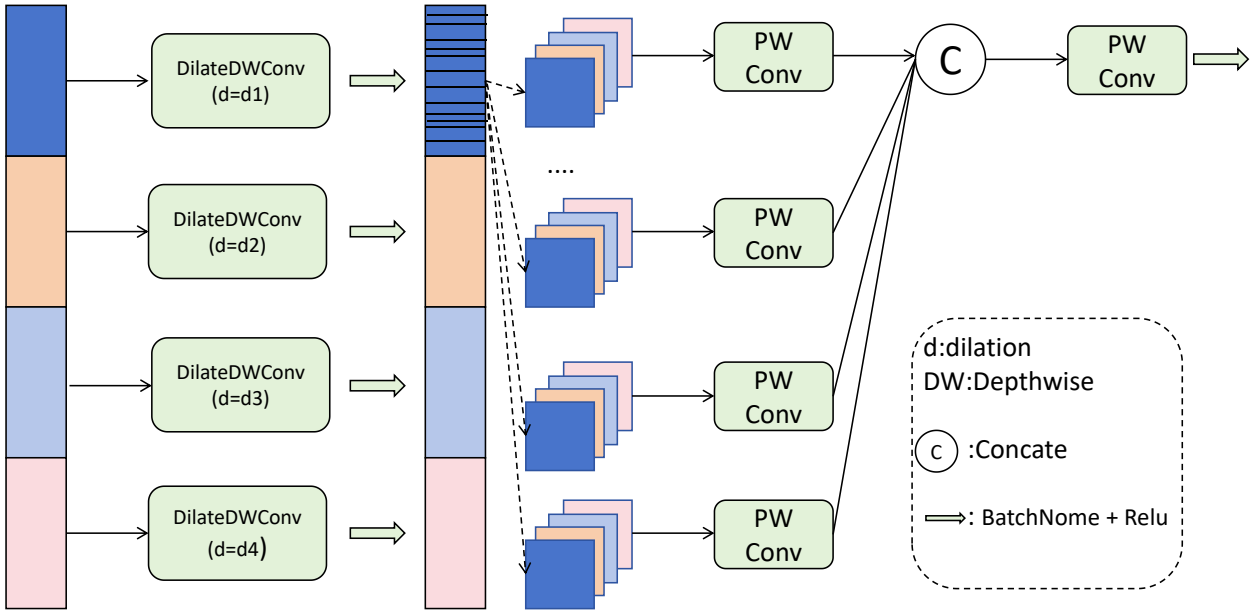

Fig. 4: MDCR network architecture

It reflects the proportion of successfully detected target objects among all existing target objects. Additionally, mAP (mean Average Precision) represents the overall performance of the model in object detection tasks. mAP@0.5 evaluates the model's overall performance by combining precision and recall to assess the model's accuracy in identifying and localizing target objects. In mAP@0.5, "@0.5" indicates that the Intersection over Union (IoU) threshold is set to 0.5. IoU is the ratio of the overlapping area between the predicted box and the ground truth box to their union area. An IoU threshold of 0.5 means that the overlap between the predicted box and the ground truth box must be greater than 50% for it to be considered a true positive (TP). Predictions with an IoU below 0.5 are considered false positives (FP). The formulas for the three metrics are as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$\text{mAP} = \frac{1}{c} \sum_{i=1}^{c} AP_i \tag{13}$$

TP represents True Positives, FP represents False Positives, and FN represents False Negatives.

C. Comparative Experiment

Figures 5 and 6 illustrate the PR curves of the YOLOv10 model before and after the improvements. These curves reflect the performance of the original YOLOv10 and the improved YOLOv10 under the same experimental conditions. The figures display the mAP@0.5 values for each category, as well as the overall mAP@0.5 value, providing a clear quantitative measure of the model's overall detection performance. Figure 5 presents the PR curve of the original YOLOv10 model, showing the performance across multiple categories, while Figure 6 shows the PR curve of the improved YOLOv10 model. It is evident from the curves that the improved algorithm achieves performance gains across multiple categories, with the overall mAP value increasing from 74.3% to 78.7%, a gain of 4.4 percentage points. This improvement not only enhances the overall accuracy of the algorithm but also significantly boosts detection performance in specific categories. For example, in the original YOLOv10 model, the mAP@50 value for the "patch" category was only 0.900, indicating relatively weak detection performance for this category. However, after the improvements, this value increased significantly to 0.937, demonstrating a significant enhancement in detection capabilities for this category. This indicates that the improved model has made breakthroughs in feature extraction and small object detection. These results show that the improvements to YOLOv10 lead to enhancements in multiple aspects, making it more accurate and better suited to real-world object detection tasks.

Figures 7 and 8 show the prediction results of the YOLOv10 model before and after the improvement. The left side represents the prediction results before the improvement, while the right side shows the results after the improvement. It is evident from the figures that the improved algorithm performs better in several aspects. The improved algorithm achieves more accurate object localization, with a significant increase in the matching degree between the bounding box and the target object, meaning the algorithm's ability to recognize and localize objects in the image has been enhanced. Additionally, the improved model demonstrates higher confidence, indicating that the algorithm's certainty in its predictions has been increased. This not only helps improve the reliability of the predictions but also effectively reduces the likelihood of misidentifications. With these improvements, the YOLOv10 model has achieved

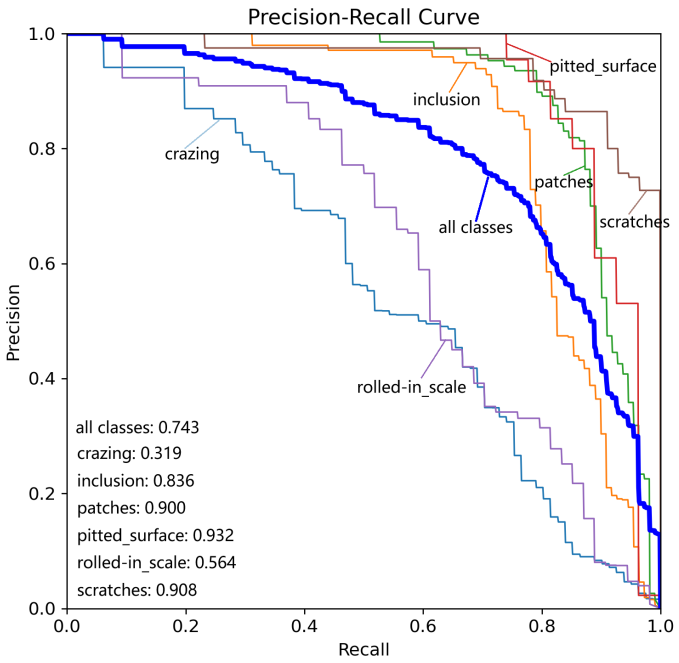

Fig. 5: FP-R curve of YOLOv10 Algorithm.

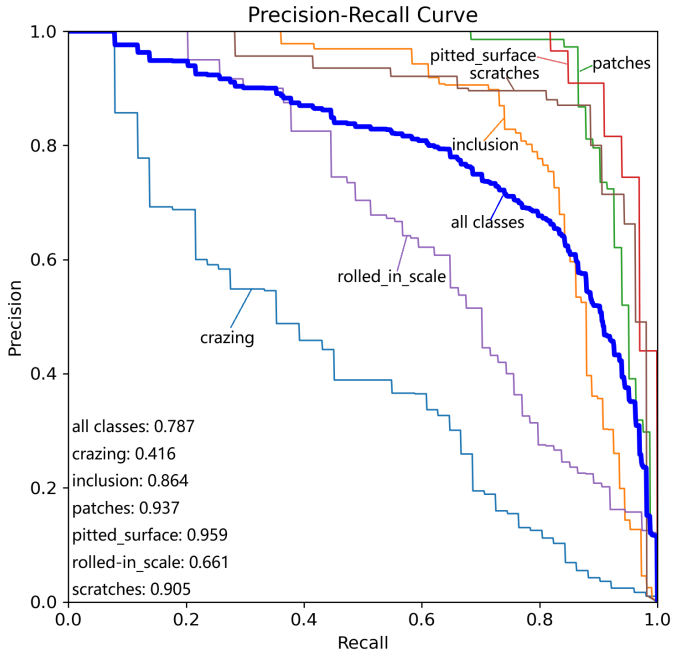

Fig. 6: FP-R Curve of the Improved YOLOv10 Algorithm.

significant gains in accuracy and robustness, making it better suited for complex industrial scenarios and various object detection tasks.

D. Ablation Study

To evaluate and validate the effectiveness of the proposed improvements, four ablation experiments were conducted. Under the same environment and training parameters, the experimental group and control group were trained or tested, and the corresponding results were recorded. The results of the ablation experiments are shown in Table 1.To comprehensively evaluate and validate the effectiveness of the proposed improvements,

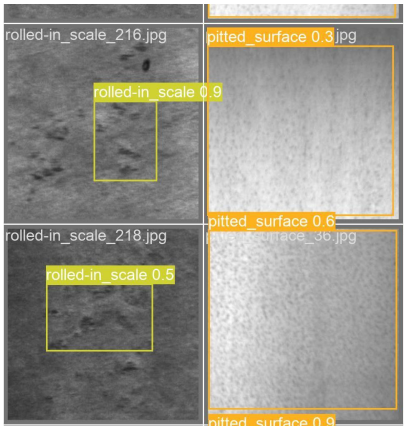

Fig. 7: Detection effect of YOLOv10 model.

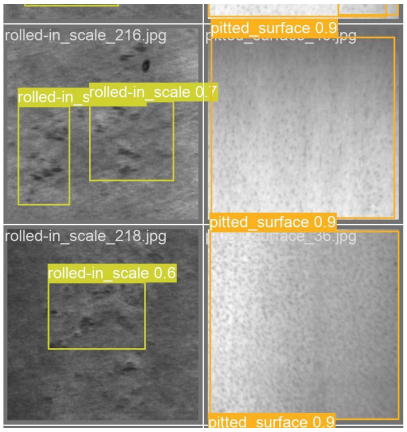

Fig. 8: The detection effect of the improved YOLOv10 model.

four ablation experiments were designed and conducted. Under the same experimental environment and training parameters, the performance of the experimental group and the control group was compared by training and testing them separately. This systematic analysis aimed to assess the impact of different model components on overall performance. During the experiments, key components were progressively removed or replaced to observe their contributions to the final detection performance, ensuring the reliability and reproducibility of the results. Finally, all experimental data were meticulously recorded and analyzed, with the specific results of the ablation experiments presented in Table 1.

The first group demonstrates the original, unmodified YOLOv10 algorithm applied to the steel surface defect detection task, achieving an mAP value of 0.743. In the second group, the C2f_MSMHSA_EMA module is introduced. This change allows the model to capture more detailed information in complex object detection tasks through multi-scale feature extraction and enhanced attention mechanisms, making it more robust for detecting small and variously scaled targets. Experimental results show that the mAP@50 value increased to 0.775. Building on this, the third group further introduces the MDCR structure for feature fusion. This modification enhances the model's object detection capability by enabling multi-domain (or multi-

TABLE I: Ablation experiments

	C2f_MSMHSA_EMA	MDCR	ContextAggregation	BOX(P)	AP50	AP(50-95)
YOLOv10n	-	-	-	71.3	74.3	40.6
YOLOv10n	√	-	-	73.1	77.5	45.5
YOLOv10n	√	√	-	73.7	78.2	47.1
YOLOv10n	√	√	√	76.4	78.7	47.7

scale, multi-channel) feature fusion. Experimental results indicate that the mAP@50 value further increased to 0.782. Subsequently, the fourth group incorporates the ContextAggregation module, helping the model capture richer feature representations, particularly in complex scenes where it better identifies object structure, edges, and background relationships. Experimental results show that this improvement raised the mAP@50 value to 0.787.

To verify the effectiveness of the algorithm, this paper compares it with mainstream object detection models on the NEU-DET dataset, including SSD, Fast RCNN, DETR, YOLOv5s, YOLOv7, and YOLOv10n. The experimental results are compared in Table 2.

Table 2 presents the mAP@0.5 values for detecting various defects across different models. As shown, this model outperforms the following models in accuracy, achieving an mAP@0.5 improvement of 0.067 over SSD, 0.020 over Fast RCNN, 0.016 over Libra Fast RCNN, 0.121 over DETR, 0.152 over YOLOv5s, 0.167 over YOLOv7, 0.023 over YOLOv8n, 0.037 over YOLOX-M, and 0.044 over YOLOv10n. Additionally, for specific defect types, particularly challenging ones like patches and pitted_surface, this model demonstrates superior accuracy compared to the aforementioned models, highlighting significant advancements in precision.

V. Conclusion

In the task of steel surface defect detection, this paper proposes an improved method based on YOLOv10. The algorithm incorporates an MDCR module in the neck network, effectively enhancing feature extraction and improving the model's ability to detect small targets and targets in complex environments. The use of a Context Aggregation module enables the model to capture richer feature representations, facilitating better recognition of object structures, edges, and background relationships. The self-fusion C2f_MSMHSA_EMA module enhances the model's capacity to capture more detailed information in complex object detection tasks through multi-scale feature extraction and enhanced attention mechanisms. The improved model achieves an mAP of 78.7% on the NEU-DET dataset. Through ablation and comparative experiments, the effectiveness of the improved model is validated.In order to address issues such as low accuracy and unstable detection performance in the practical application of steel surface defect detection, this paper proposes an improved method based on YOLOv10, aimed at enhancing the precision and robustness of defect detection. Traditional defect detection algorithms often suffer from low accuracy, missed detections, and false positives, especially when dealing with small targets, low-contrast defects, or environments with heavy background interference. To overcome these challenges, this paper introduces several innovative improvements based on YOLOv10.

Firstly, the MDCR (Multi-Dimensional Contextual Representation) module is introduced into the neck portion of the network. The inclusion of this module significantly strengthens the feature extraction capability, particularly in detecting small targets and the edges of defects in complex backgrounds. The MDCR module enriches the model's ability to capture multi-dimensional contextual information, enabling the model to not only capture local features but also understand the relationship between the target and its background, thus improving detection performance in complex environments. This enhancement helps the model perform better in detecting small and subtle defects by capturing features across various scales.

Secondly, the paper incorporates the Context Aggregation module, which improves the model's ability to capture richer feature representations by enhancing its perception of global information. This further boosts the model's ability to recognize object structures, edges, and background relationships. In steel surface defect detection, where the background is often complex and defects are subtle, the model needs to better understand these relationships to effectively detect defects. The Context Aggregation module optimizes the aggregation of global context, thereby improving the model's ability to recognize fine details and structures.

Additionally, the self-fusion C2f_MSMHSA_EMA module is introduced. This module combines multi-scale feature extraction with enhanced attention mechanisms, allowing the model to adaptively select and fuse features from different scales. As a result, the model's ability to capture detailed information in complex object detection tasks is enhanced. This improvement allows the model to detect defects of varying sizes and shapes more accurately and to operate reliably in diverse environments. In steel surface defect detection, where defects vary greatly in size, the self-fusion C2f_MSMHSA_EMA module effectively addresses this challenge, enabling the model to detect a wide range of defect types.

Experimental results show that the improved model achieves an mAP50 of 78.7% on the NEU-DET dataset, which is 4.4% higher than the original algorithm. This improvement demonstrates that the proposed method significantly enhances detection accuracy, particularly in

TABLE II: Comparison of Detection Performance of Different Algorithms.

Types	SSD	Fast RCNN	Libra Faster RCNN	DETR	YOLOv5s	YOLOv7	YOLOv8n	YOLOvX-m	YOLOv10n	OURS
crazing	0.411	0.421	0.415	0.261	0.201	0.185	0.494	0.555	0.319	0.416
inclusion	0.773	0.773	0.791	0.655	0.697	0.762	0.852	0.812	0.836	0.846
patches	0.922	0.919	0.908	0.898	0.931	0.908	0.839	0.930	0.900	0.937
pitted_surface	0.792	0.866	0.884	0.706	0.706	0.534	0.850	0.814	0.932	0.959
rolled-in_scale	0.695	0.654	0.707	0.565	0.397	0.539	0.624	0.537	0.564	0.661
scratches	0.729	0.969	0.920	0.910	0.875	0.778	0.926	0.863	0.908	0.905
mAP	0.720	0.767	0.771	0.666	0.635	0.618	0.764	0.750	0.743	0.787

small target detection and complex background environments. Compared to traditional detection methods, the improved model not only achieves higher precision but also maintains stable performance in quality inspection tasks in complex environments, meeting the current requirements for steel surface defect detection.

In conclusion, the YOLOv10-based improved method proposed in this paper, through the integration of the MDCR module, Context Aggregation module, and self-fusion C2f_MSMHSA_EMA module, effectively enhances the accuracy and robustness of steel surface defect detection. The improved method can handle various types of defects, adapt to complex backgrounds, and address the challenges of multi-scale targets, providing an efficient and reliable solution for automated quality inspection in the steel industry.

References

[1] X. Wang, Z. Wang, C. Guo, et al., "Application and prospect of new steel corrugated plate technology in infrastructure fields," in IOP Conference Series: Materials Science and Engineering, vol. 741, p. 012099, IOP Publishing, 2020.

[2] Z. Guo, C. Wang, G. Yang, et al., "Msft-yolo: Improved yolov5 based on transformer for detecting defects of steel surface," Sensors, vol. 22, no. 9, p. 3467, 2022.

[3] R. Mordia and A. K. Verma, "Visual techniques for defects detection in steel products: A comparative study," Engineering Failure Analysis, vol. 134, p. 106047, 2022.

[4] R. Zaghdoudi, H. Seridi, and S. Ziani, "Binary gabor pattern (bgp) descriptor and principal component analysis (pca) for steel surface defects classification," in 2020 International conference on advanced aspects of software engineering (ICAASE), pp. 1–7, IEEE, 2020.

[5] A. Varsha, K. Mundra, A. Singh, et al., "From pixels to insight: Enhancing metallic component defect detection with glcm features and ai explainability," in International Conference on Data Management, Analytics & Innovation, pp. 289–301, Springer Nature Singapore, 2024.

[6] Z. Hao, Z. Li, F. Ren, et al., "Strip steel surface defects classification based on generative adversarial network and attention mechanism," Metals, vol. 12, no. 2, p. 311, 2022.

[7] Y. I. Hwang, M. K. Seo, H. G. Oh, et al., "Detection and classification of artificial defects on stainless steel plate for a liquefied hydrogen storage vessel using short-time fourier transform of ultrasonic guided waves and linear discriminant analysis," Applied Sciences, vol. 12, no. 13, p. 6502, 2022.

[8] J. Huang, Z. Zhang, B. Zheng, et al., "Tensile damage characterization of low-carbon steel for high-end power system based on acoustic emission and mf-dbscan." Available at SSRN 4257873.

[9] B. Kim, J. Kwon, S. Choi, et al., "Corrosion image monitoring of steel plate by using k-means clustering," Journal of the Korean Institute of Surface Engineering, vol. 54, no. 5, pp. 278–284, 2021.

[10] X. Ye and S. Xu, "Study on surface defect classification of hot-rolled strip based on pso-svm," in Proceedings of the Eighth Asia International Symposium on Mechatronics, pp. 1846–1855, Springer Nature Singapore, 2022.

[11] 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.

[12] X. Shi, S. Zhou, Y. Tai, 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.

[13] C. Y. Lin, C. H. Chen, C. Y. Yang, et al., "Cascading convolutional neural network for steel surface defect detection," in Advances in Artificial Intelligence, Software and Systems Engineering: Proceedings of the AHFE 2019 International Conference on Human Factors in Artificial Intelligence and Social Computing, pp. 202–212, Springer International Publishing, 2020.

[14] B. Liu, B. Yang, Y. Zhao, et al., "Low-pass u-net: a segmentation method to improve strip steel defect detection," Measurement Science and Technology, vol. 34, no. 3, p. 035405, 2022.

[15] T. Yin and J. Yang, "Detection of steel surface defect based on faster r-cnn and fpn," in Proceedings of the 2021 7th International Conference on Computing and Artificial Intelligence, pp. 15–20, 2021.

[16] X. Liu and J. Gao, "Surface defect detection method of hot rolling strip based on improved ssd model," in Database Systems for Advanced Applications. DASFAA 2021 International Workshops, pp. 209–222, Springer International Publishing, 2021.

[17] S. Li, C. Wu, and N. Xiong, "Hybrid architecture based on cnn and transformer for strip steel surface defect classification," Electronics, vol. 11, no. 8, p. 1200, 2022.

[18] C. Y. Wang, H. Y. M. Liao, and I. H. Yeh, "Designing network design strategies through gradient path analysis," 2020.

[19] T. Y. Lin, P. Dollár, R. Girshick, et al., "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125, 2017.

[20] S. Liu, L. Qi, H. Qin, et al., "Path aggregation network for instance segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759–8768, 2018.

[21] D. Ouyang, S. He, G. Zhang, 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.

[22] P. Gao, J. Lu, H. Li, et al., "Container: context aggregation network," in Proceedings of the 35th International Conference on Neural Information Processing Systems, pp. 19160–19171, 2021.

[23] S. Xu, S. C. Zheng, W. Xu, et al., "Hcf-net: Hierarchical context fusion network for infrared small object detection," CoRR, 2024.