electronics

MDPI

Article

Experimental Design of Steel Surface Defect Detection Based on MSFE-YOLO—An Improved YOLOV5 Algorithm with Multi-Scale Feature Extraction

Lin Li *, Ruopeng Zhang, Tunjun Xie, Yushan He, Hao Zhou and Yongzhong Zhang

School of Electronic Information and Physics, Central South University of Forestry and Technology, Changsha 410004, China; 20221200509@csuft.edu.cn (R.Z.); 20201200378@csuft.edu.cn (T.X.); 20212544@csuft.edu.cn (Y.H.); 20231100403@csuft.edu.cn (H.Z.); t20010595@csuft.edu.cn (Y.Z.)
* Correspondence: t20060540@csuft.edu.cn

Abstract: Integrating artificial intelligence (AI) technology into student training programs is strategically crucial for developing future professionals with both forward-thinking capabilities and practical skills. This paper uses steel surface defect detection as a case study to propose a simulation-based teaching method grounded in deep learning. The method encompasses the entire process from data preprocessing and model training to validation analysis and innovation optimization with the goal of deepening students' understanding of AI technology and enhancing their ability to apply it to real-world scenarios. We have designed an experimental framework that incorporates the Efficient Multi-Scale Attention (EMA) mechanism into the Backbone network. This approach helps students understand the principles of feature extraction and the core functions of attention mechanisms. Additionally, we introduced a novel architecture—Convolution 3 Dilated Convolution X (C3DX)—into the Neck network. This architecture effectively expands the network's receptive field, improves its ability to capture multi-scale information, and thus enhances defect detection accuracy. Furthermore, the implementation of the Efficient Intersection over Union (EIoU) loss function optimizes the bounding box predictions, further increasing the model's accuracy and robustness. Overall, the teaching design not only ensures that the content remains at the cutting edge of technology but also emphasizes its practicality and operability. This approach enables students to effectively apply theoretical knowledge to real-world engineering projects.

Keywords: experimental teaching; artificial intelligence; deep learning; object detection; surface defect detection

Citation: Li, L.; Zhang, R.; Xie, T.; He, Y.; Zhou, H.; Zhang, Y. Experimental Design of Steel Surface Defect Detection Based on MSFE-YOLO—An Improved YOLOV5 Algorithm with Multi-Scale Feature Extraction. *Electronics* **2024**, *13*, 3783. https://doi.org/10.3390/electronics13183783

Academic Editor: Dah-Jye Lee

Received: 1 August 2024
Revised: 12 September 2024
Accepted: 19 September 2024
Published: 23 September 2024

1. Introduction

With the rapid development of Artificial Intelligence (AI) technology, AI's role in driving industrial upgrading and social progress has become increasingly prominent. Mastering AI technology has become a core requirement for future professionals. Therefore, integrating AI education into the talent cultivation system is crucial for training highly qualified professionals who can meet industry demands. Steel, as a fundamental material widely used in various industries, often encounters defects such as cracks, scratches, and spots during production and application. These defects not only significantly reduce the strength and lifespan of steel but may also pose serious safety risks [1]. Applying cutting-edge AI algorithms to steel surface defect detection not only effectively solves real-world engineering problems but also provides students with valuable opportunities to learn and master advanced technologies. By participating in these experiments, students can gain a deep understanding of AI concepts and learn how to apply them to practical problems, thereby acquiring the practical skills needed for future careers in engineering and technology.

The development of steel defect detection has undergone three stages, reflecting the advancements in image processing technology. Initially, traditional manual inspection required substantial human resources and was prone to subjective errors. This was followed by the use of machine vision, which, while more efficient, involved manual feature extraction and struggled in complex environments. With breakthroughs in deep learning, researchers began applying these algorithms to steel defect detection, achieving superior accuracy compared to the conventional methods [2].

Existing object detection algorithms are generally categorized into two-stage and one-stage approaches [3]. The former is divided into two stages: the first stage creates region proposals, the second is responsible for classification and localization. Nevertheless, the latter skips the region proposal stage and obtains the category and location of the target immediately. So, the two-stage detection algorithms have a high degree of accuracy. Regions with a CNN (R-CNN) series, including the Faster R-CNN [4] and Cascade R-CNN, ref. [5] are considered the most representative algorithms of the two-stage detection algorithms, which are also more frequently used in defect detection [6–9]. The inclusion of the region proposal network structure limits speed despite improving the detection accuracy of the R-CNN series algorithms. Meeting real-time requirements becomes more challenging under specific circumstances.

To fulfill the requirements of industry, relevant researchers presented one-stage object detection algorithms to obtain lighter and faster network models. The exemplary one-stage algorithms include RetinaNet [10], You Look Only Once (YOLO) series [11–14], etc. The one-stage detection algorithm transforms the object detection problem into a regression problem, thereby achieving an optimal trade-off between speed and accuracy. For instance, according to the literature [1], the mean average precision (mAP) of YOLOv3 on the steel surface defect dataset NEU-DET is 1% worse than that of Faster R-CNN, but the speed is three times that of Faster R-CNN. Numerous academics applied and modified the one-stage algorithm YOLO series widely due to it performing defect detection exceptionally well.

Guo et al. [15] added a module based on the Transformer [16] to the Backbone of YOLOv5 in steel surface defect detection to enhance its global feature information extraction capability. Zhu et al. [17] proposed the LSin Transformer structure for steel surface defect detection, which significantly improved detection accuracy. However, due to the limitations inherent in the Swim Transformer [18] itself, the model struggles to meet real-time requirements effectively. Yi et al. [19] introduced the MobileViT module into the Backbone network of YOLOX [20] for steel surface defect detection to enhance the feature extraction capability of the network. Incorporating Transformers into the YOLO series algorithms improves the insufficient feature extraction capability of traditional convolutional methods. However, its computational complexity and challenging training render it unsuitable for applications in the industrial domain.

Xie et al. [21] used MobileNetV2 to replace YOLOv4's feature extraction network and added Efficient Channel Attention (ECA) to implement adaptive weight assignment and quickly infer a single image for only 18.44 ms, which achieved 86% mAP and 68% mAP on the metal surface defect datasets GCT10 and NEU-DET, respectively. Ling et al. [22] optimized the detection performance of YOLOv5 by combining multi-scale detection blocks with an attention mechanism. Their model achieved a detection rate of 72% in mean average precision (mAP) for steel surface defects. The detection speed of these algorithms has been somewhat enhanced; however, a persistent challenge remains in achieving an overall satisfactory detection accuracy. Currently, the YOLO algorithm has introduced multiple versions such as YOLOv7 [23] and YOLOv8 [24]. However, multiple experiments have shown that these versions of the network are unable to significantly improve detection accuracy despite the increase in the number of parameters. We have summarized the limitations of the above study in Table 1.

Table 1. Comparison of performance of different detection methods.

Model	Advantages	Disadvantages	Improvements
Traditional Manual Inspection	Visual identification, capable of detecting rare defects	Relies on subjective judgment, low efficiency, inaccurate, resource-intensive	Not suitable for large-scale or complex environments, both efficiency and accuracy are insufficient
Traditional Machine Vision	Automation reduces manual intervention	Manual feature extraction is complex, not suitable for complex environments, computationally intensive	Feature extraction complexity and unsuitability for complex detection environments
R-CNN Series (e.g., Faster R-CNN)	High accuracy, widely used in defect detection	Slower speed, challenging to meet real-time requirements, computation–ally complex	Region proposal network limits speed and real-time performance
YOLO Series	Real-time detection, fast, suitable for large-scale detection	Precision for small defects is insufficient, accuracy slightly lower	Precision issues, especially for small-sized defects
Transformer based YOLO improvements	Enhanced global feature extraction capability	High computational complexity, training difficulties, struggles with real-time requirements	Computational complexity and training challenges limit industrial application

To address these challenges, we propose an improved YOLOv5 model—MSFE-YOLOv5 (YOLOv5 with Multi-Scale Feature Extraction). The model integrates Efficient Multi-Scale Attention (EMA) and the novel Convolution 3 Dilated Convolution X (C3DX) structure to enhance the detection accuracy of small defects while maintaining computational efficiency. Additionally, the model serves as a valuable educational tool for students, allowing them to engage with real-world AI applications in a hands-on, simulation-based learning environment. By working with state-of-the-art models and understanding the trade-offs between accuracy and efficiency, students can develop critical skills that prepare them for future challenges in AI-driven industries.

The primary contributions of this paper can be summarized as follows:

1. Experimental Design Framework Based on Deep Learning: We propose a deep learning-based experimental design framework that integrates artificial intelligence with industrial applications. This framework not only provides an innovative solution for steel surface defect detection but also serves as a teaching tool aimed at guiding students to learn and master relevant technologies, equipping them with the skills necessary for real-world industrial applications.
2. Introduction of Efficient Multi-Scale Attention (EMA) [25] Mechanism: By incorporating the EMA mechanism into the Backbone network of the YOLOv5 model, pixel-level relationships are captured through cross-dimensional interactions. Utilizing convolution kernels of varying sizes, the model efficiently fuses multi-scale contextual information, significantly enhancing the feature extraction capabilities and detection accuracy with only a slight increase in the computational cost.
3. Proposed Novel C3DX Module: In the Neck of the network, we introduce the Convolution 3 Dilated Convolution X (C3DX) module. This module uses dilated convolutions with different dilation rates to capture diverse receptive fields and integrate multi-scale contextual information, further improving defect detection precision. In addition to boosting detection performance, this module helps students understand the concept of receptive fields, fostering their innovative thinking skills.
4. Model Validation Across Multiple Datasets: The improved MSFE-YOLOv5 model has been validated on the NEU-DET, GC10-DET, Severstal Steel, and Crack500 datasets, with mean average precision (mAP) increases of 4.7%, 4.5%, 3.1% and 3.0%, respectively. These results demonstrate the model's excellent performance in detection and generalization, while the experiments help students develop practical skills and the ability to solve real-world problems.

This study aims to bridge the gap between technological innovation and educational impact, providing a robust solution for steel surface defect detection while cultivating the next generation of professionals in artificial intelligence and engineering.

A glossary of key concepts and acronyms can be found in Appendix A to facilitate reader comprehension.

2. Materials and Methods

2.1. YOLOv5

In this section, the specific structure of the YOLOv5 network is introduced to allow students to understand the specific experimental methods of object detection using neural networks. This section introduces the structure of the YOLOv5 network to help students understand the experimental methods for object detection using neural networks. The YOLOv5, one of the YOLO series algorithms, has four scales: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The s-model is more appropriate for industrial fields because it has the fewest parameters and computations. The model employed in this study is the most recent version v6.0, as shown in Figure 1, which is mainly divided into four parts: Input, Backbone, Neck and Head. Input consists of the three channels of the input image.

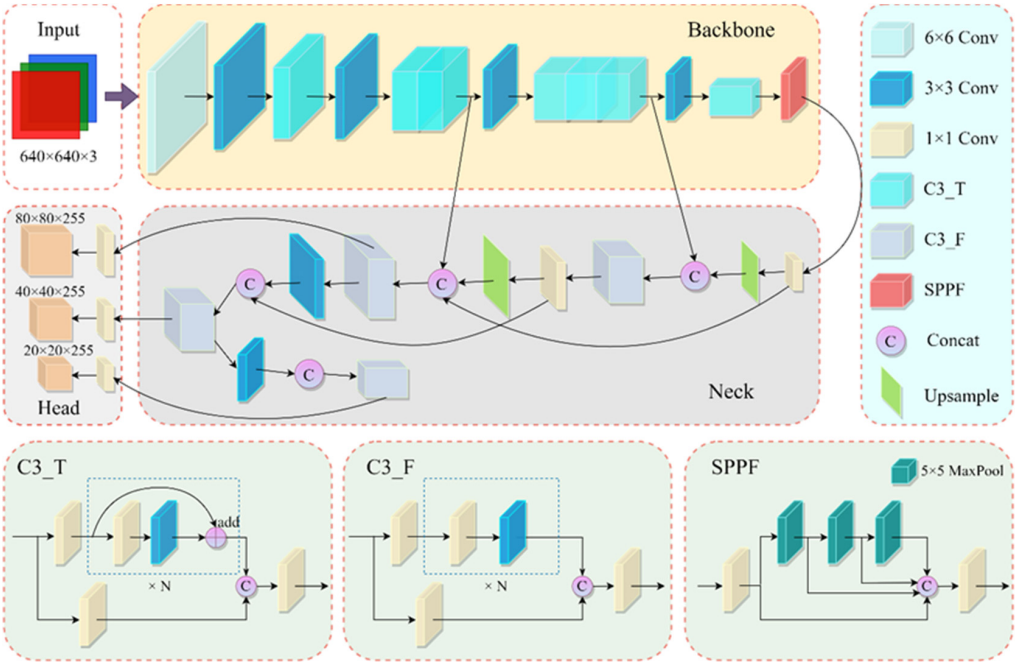


Figure 1. YOLOv5s v6.0 structure.

The Backbone network is mostly utilized for feature extraction from input images, containing the Conv, C3, and SPPF modules. Among them, Conv denotes the combined module of Conv2d, Batch Normalization (BN), and the SiLU activation function, which is the basic convolution unit for the YOLOv5 network. C3_T and C3_F are two structures of C3, which is the primary module of the YOLOv5 network, and the C3_T structure is used in the Backbone. It employs the CSPNet [26] structure, splitting it into two branches. One branch decreases the computation by 1×1 convolution, while the other branch completes the main feature learning through N residual units. Ultimately, the two branches are subjected to the concatenate operation. The SPPF module is composed of three 5×5 maximum pooling operations (MaxPools), which can solve the multi-scale problem to a certain extent.

In the Neck network, Conv and C3 are still included, but C3 in the Neck does not avail itself of the use of the residual structure. The main role of the Neck is to further process the features extracted from the Backbone by fusing the multi-layer characteristics utilizing the

Path Aggregation Network (PAN) [27] architecture, resulting in richer target features and a greater ability to express features. The PAN is an improvement over the FPN and has two paths: one is top-down for the up-sampling operation to pass down the stronger semantic features from the top layer and improve the semantic information throughout the entire pyramid; the other is bottom-up for down-sampling to pass up the stronger localization features from the bottom layer and to perform feature fusion for different detection layers from the Backbone.

For the input feature maps, such as those shown in Figure 1 with dimensions of 640 × 640, the Head module is designed to generate three distinct sizes of feature maps, specifically measuring 80 × 80, 40 × 40, and 20 × 20. These feature maps are utilized for the accurate prediction of small-, medium-, and large-target classes as well as bounding box coordinates.

2.2. Method

To enable students to thoroughly grasp the complete process of object detection using deep learning and clearly understand the core functions and roles of each stage, we have conducted an innovative optimization and improvement on the YOLOv5 network structure. In this study, we propose several enhancements to YOLOv5s for improved detection of steel surface defects. Firstly, we incorporate an efficiently designed attention mechanism called EMA into the C3_T module of the Backbone. Secondly, we enhance the receptive field by introducing dilated convolution to the structure of the C3_F in the Neck. Lastly, we increase the prediction box accuracy by introducing EIoU Loss. These methods ultimately lead to an improved accuracy in detecting steel surface defects. The structure of MSFE-YOLOv5s is illustrated in Figure 2.

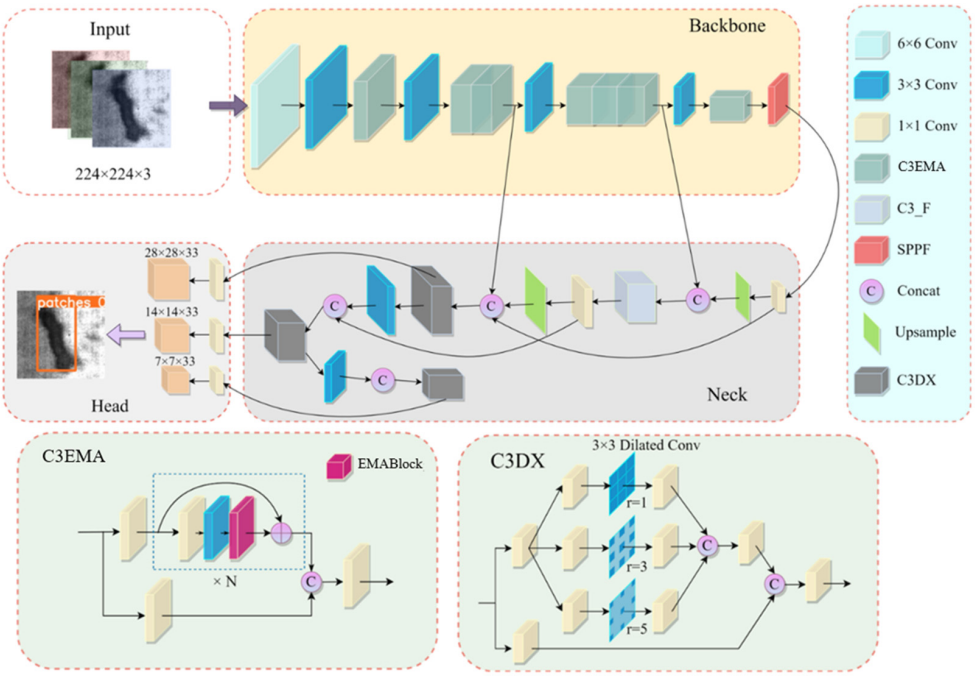


Figure 2. MSFE-YOLOv5s structure.

2.2.1. EMA

The attention mechanism is essentially similar to how humans observe things outside of themselves. In general, when people observe external things, they prefer to first observe certain important local information about the things before combining the information from different regions to form an overall impression of the observed thing [28]. It was originally used for Natural Language Processing and subsequently applied to Computer Vision tasks. The main idea behind the attention mechanism is for the model to learn to focus on the

crucial information and ignore the irrelevant information. The process essentially involves learning the weight assignment using the relevant feature maps, then adding the procured weights on top of the original feature maps to generate new weights.

Incorporating attention allows the model to better focus on key regions, thereby improving performance. In this paper, we propose embedding the EMA mechanism into the C3 module of the YOLOv5 model's Backbone, enabling the model to focus more on extracting critical features. The EMA mechanism efficiently captures spatial information and inter-channel relationships, addressing the lack of spatial information in channel attention mechanisms like Squeeze and Excitation (SE) [29], and the limitations of mixed attention in CBAM [30] for capturing long-range dependencies. By designing a multi-scale parallel sub-network, it establishes both short-range and long-range dependencies, overcoming the limitations of the CBAM in modeling long-range dependencies. Furthermore, by reshaping parts of the channel dimension into the batch dimension, it significantly reduces the number of parameters and computational complexity, thereby enhancing the detection speed of the network. Its specific structure is shown in Figure 3.

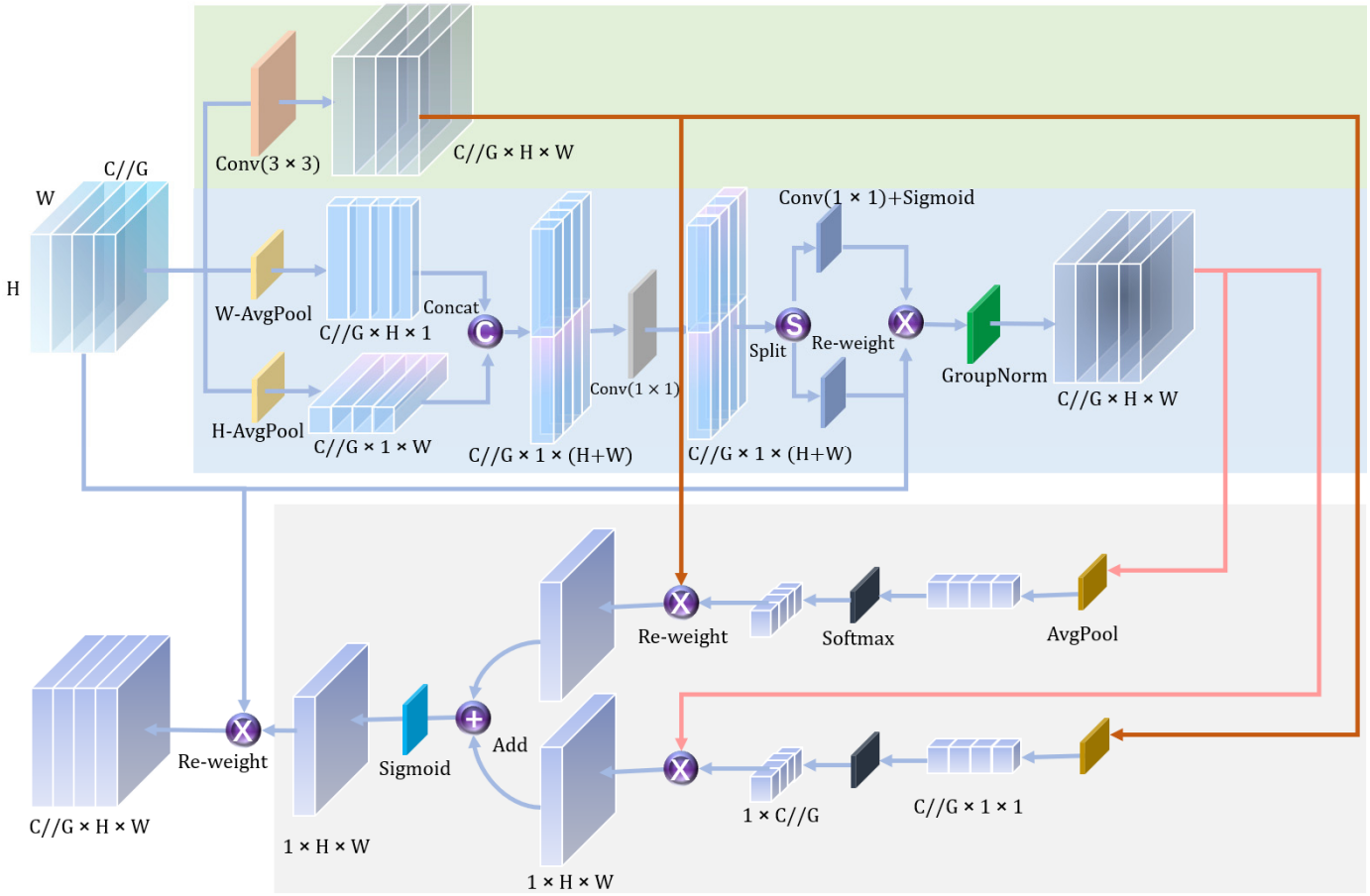


Figure 3. EMA Block structure.

The EMA Block can be viewed as a computational unit capable of taking any intermediate feature tensor X as input and outputting a transformed tensor with the enhanced characterization Y of the same sizes as the X tensor. The specific operations are as follows:

Given an input feature tensor $X \in R^{C \times H \times W}$, EMA divides it into G sub-features along the channel dimension, denoted as $X = [X_0, X_1, \ldots, X_{G-1}]$, where $X \in R^{C//G \times H \times W}$. Without loss of generality, choosing G << C, experiments have shown that when G = 8, the detection performance is optimal, indicating that the Backbone is able to adequately capture intricate features.

For the grouped features, the EMA mechanism employs a parallel sub-architecture, leveraging both 1 × 1 convolution kernels and 3 × 3 convolution kernels for feature extraction. Specifically, within the 1 × 1 convolution branch, inspired by the handling in CA (Coordinate Attention) [31], average pooling is introduced to encode features along both the height and width dimensions, resulting in direction-aware feature maps in these two orientations. Subsequently, these feature maps undergo a concatenation operation followed by a convolution step, aiming to achieve cross-directional attention–perception interactions.

After applying a sigmoid function for nonlinear transformation, global average pooling is employed once more to extract global spatial information.

$$Z_C = \frac{1}{H \times W} \sum_j^H \sum_i^W x_c(i, j) \tag{1}$$

This global information, after undergoing nonlinear fitting through a Softmax function, is then subject to a matrix dot product operation with the local information obtained from the parallel 3 × 3 convolution branch, resulting in the first globally and locally mixed encoding matrix.

On the 3 × 3 convolution branch, after performing the corresponding convolutions and global average pooling to obtain the global information, this information undergoes nonlinear processing via Softmax. Subsequently, a matrix dot product is conducted between this processed global information and the features from the 1 × 1 branch after undergoing the GroupNorm, yielding the second spatial attention matrix.

By simply adding the weight information from both matrices, and then passing the result through a sigmoid function, we obtain the attention weight matrix. Multiplying the input features by this matrix results in the feature map processed by the EMA mechanism.

Since EMA aggregates the output features from the two parallel branches through cross-dimensional interactions, it captures pixel-wise/pair-wise relationships, thereby enhancing the feature extraction capabilities of the Backbone portion of the YOLO model.

The EMA Block is a light-weight and plug-and-play attention block with long-range dependencies, which can be added straight to C3_T. The merged module is identified as C3EMA. It can enhance the extraction ability of feature information while only increasing a modicum of the computation, and improve the performance of the whole network. This study compares multiple attention mechanisms and also explores the effects of the two insertion methods shown in Figure 4a,b on the network. In Section 3.4.1, there is a detailed analysis. The (a) method with the highest mean average precision is selected for this paper.

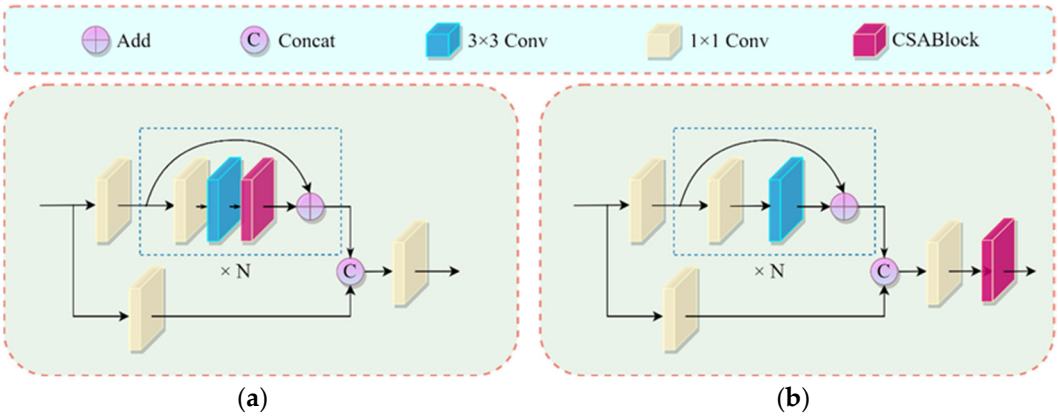


Figure 4. Two insertion methods: (**a**) residual layer insertion; (**b**) final layer insertion.

2.2.2. C3DX

The attention mechanism is simply the process of assigning weights to feature information, and it is intended to improve the feature representation ability of key regions

and suppress the unimportant parts. Given that the Backbone's primary function is feature extraction and that various features have varying degrees of importance, a feature weight assignment method will function well in the Backbone and can therefore significantly improve the network performance. Nevertheless, it is not essential to add the attention mechanism to the Neck network because the feature weights are already assigned in the Backbone from the beginning, signifying the crucial feature information is already extracted.

More rich feature information is required considering the primary role of the Neck is feature integration. For this reason, we improve the C3_F module in the Neck by introducing dilated convolution to design a new structure, C3DX, as shown in Figure 2. Dilated convolution can enlarge the receptive field of each point on the feature map without extra parameters, learning more multi-scale contextual information [32].

C3DX remains to execute the CSP structure; it is divided into two branches and the computation is reduced through 1×1 convolution and one of the branches will go through a multi-branch dilated convolution block. To reduce the computing burden, each branch of this multi-branch block first undergoes a 1×1 convolution, it cuts the number of input channels in half, and then undergoes dilated convolutions with different dilated rates. In this paper, the convolution kernel size of the dilated convolution is set to 3×3, the dilation rate is set to 1, 3, and 5 from top to bottom, and the padding operation is used for size adjustment. This can capture the contextual information of various scale features without increasing extra parameters. Following a 1×1 convolution to resize the number of channels, the three feature maps integrating the contextual information of different scales are concatenated, and then the number of channels is restored to the original input channel number by a 1×1 convolution. Finally, after passing through the multi-branch dilated convolution block, feature integration is performed with another branch of the CSP.

The C3DX module can capture information in different regions of the image with different dilated rates, that is, different receptive fields, and obtain multi-scale information, which is more suitable for the front layer of the detection layer. Therefore, the last three C3_F modules are swapped out for C3DX modules in this paper. When compared to the original C3_F module, C3DX can obtain more rich feature information. And the average precision of surface defect detection can be improved with a slightly increased parameter and computation.

2.2.3. EIoU Loss

The YOLOv5 applies CIoU Loss [33] as a localization loss function, its formula is shown in Equations (2)–(4):

$$L_{CIoU} = 1 - IoU + \frac{\rho^2\left(b,\ b^{gt}\right)}{c^2} + \alpha v \tag{2}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{\omega^{gt}}{h^{gt}} - arctan\frac{\omega}{h}\right)^2 \tag{3}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{4}$$

where IoU is the intersection over the union between the prediction box and the ground truth (GT) box, L_{CIoU} is the CIoU loss function, $\rho^2\left(b,\ b^{gt}\right)$ is the Euclidean distance between the center point of the prediction box b and the center point of the GT box b^{gt}, c is the diagonal distance of the closed rectangle containing both the prediction box and the GT box, α is the balance scale weight factor, v is the similarity factor measuring the aspect ratio of the prediction box and the GT box, ω^{gt} and h^{gt} are the width and height of the GT box, and ω and h are the width and height of the prediction box, respectively.

Although CIoU Loss takes into account the overlapping area, center point distance, and aspect ratio of the bounding box regression, it has a problem in that it uses the aspect ratio as an influence factor. For example, it is possible that the centers of two bounding

boxes are the same, and the aspect ratios are also identical, but their specific widths and heights differ, which may lead to inconsistencies with the regression target. Due to this, this paper introduces EIoU Loss [34] to replace CIoU Loss. EIoU Loss is based on CIoU by splitting the aspect ratio, it is transformed to a width and height regression, and its formula is as follows in Equation (5):

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b,\ b^{gt})}{c^2} + \frac{\rho^2(\omega,\ \omega^{gt})}{C_\omega^2} + \frac{\rho^2(h,\ h^{gt})}{C_h^2} \tag{5}$$

where C_ω and C_h represent the width and height of the minimum outer rectangle of the prediction and ground truth box, respectively. EIoU can accelerate network fitting, improve regression correctness, and more precisely reflect the width and height difference between the predicted box and the ground truth box.

3. Results

To effectively teach students how to apply neural networks for steel surface defect detection, we have adopted a comprehensive and structured learning approach, as illustrated in Figure 5. First, in the introduction and background knowledge phase, we introduce the significance of steel surface defect detection and its industrial applications. This phase helps students understand the real-world relevance of the technology and provides them with a solid theoretical foundation. Next, students proceed to the data preparation phase, where they learn how to select and process publicly available datasets and apply image enhancement techniques. This process not only enhances students' data processing skills but also improves the model's adaptability to unknown or complex scenarios. In the model development phase, students gain a thorough understanding of the YOLO series models, with a particular focus on the customized MSFE-YOLO model. This phase emphasizes how the Multi-Scale Feature Fusion (MSFE) mechanism improves detection accuracy, helping students understand the architecture and functionality of the model, appreciate its performance in practical applications, and develop their innovative capabilities. Subsequently, students participate in hands-on activities related to model training and tuning. By analyzing the detection results to optimize model performance and ensure accuracy in complex backgrounds, this phase develops students' practical skills and problem-solving abilities. In the model deployment and application phase, students learn how to deploy the trained model in real-world scenarios, detect surface defects in steel materials in real-time, and analyze detection accuracy and robustness. This phase enhances students' practical operation skills and prepares them for applications in industrial environments.

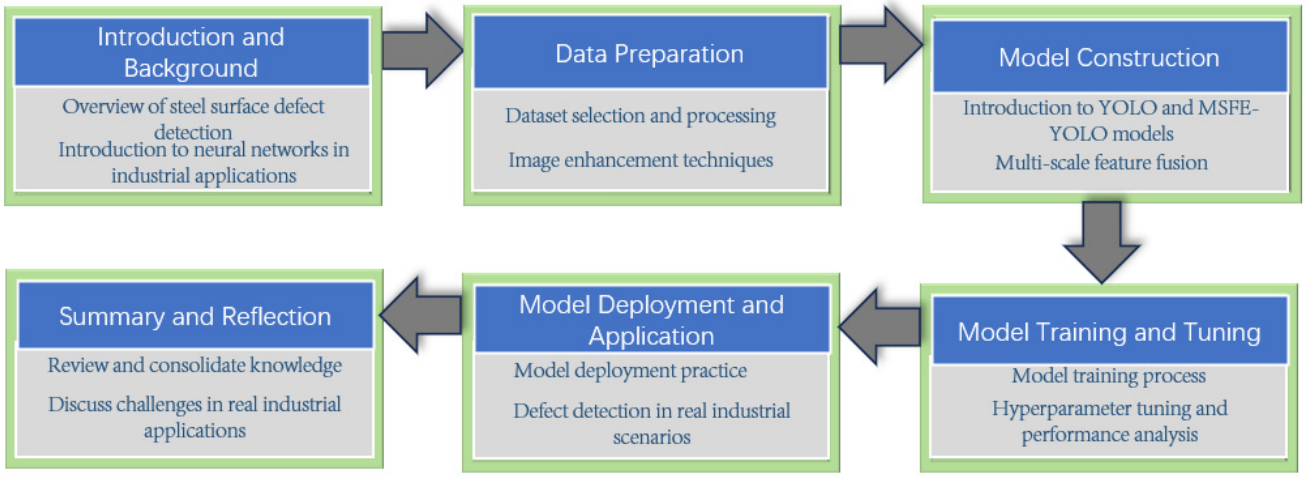


Figure 5. The complete experimental teaching process.

The overall framework for the experiment, which utilizes neural networks to detect surface defects in steel materials, is illustrated in Figure 6. In the data preparation phase, we leveraged the publicly available dataset of steel surface defects to their fullest extent. Subsequently, a series of image enhancement techniques, including but not limited to random rotation, non-uniform scaling and cropping, horizontal/vertical flipping, and the innovative mosaic tiling method, were implemented. These techniques effectively mitigated the issue of data scarcity while significantly enhancing the model's adaptability to unknown or complex scenarios. Following this, we introduced the deeply customized and optimized MSFE-YOLO model, which, while inheriting the efficient detection characteristics of the YOLO series, incorporates a Multi-Scale Feature Extraction (MSFE) mechanism. This integration enables the model to capture nuanced texture variations and defect features on steel surfaces with greater precision. Through model training and parameter tuning, we constructed a defect detection model that is both swift and accurate. In the final stage of the experiment, the trained model was utilized to detect various types of defects on steel surfaces. The experimental results demonstrate that this model is not only capable of accurately identifying common defects such as cracks, rust, and scratches but also maintains stable detection performance against complex backgrounds. The generated detection outcomes closely align with actual conditions.

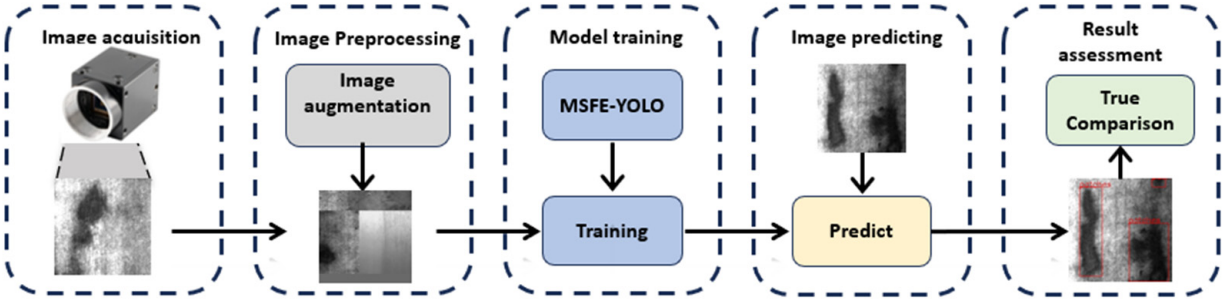
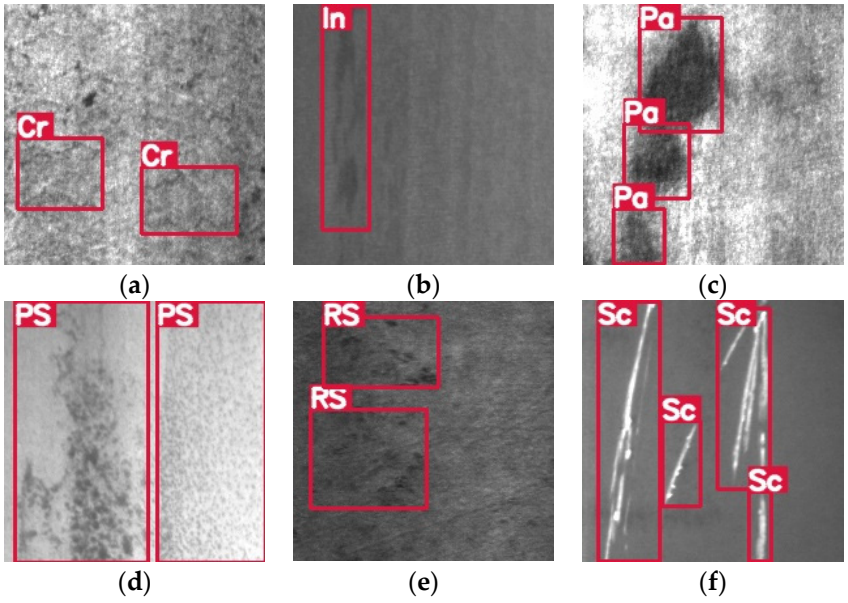


Figure 6. The process diagram of experimental simulation for steel surface defect detection based on MSFE-YOLO network.

This comprehensive process, spanning from data preparation to model deployment, incorporates multiple innovative strategies to elevate detection accuracy and efficiency. It provides students with a hands-on understanding of the entire workflow for industrial defect detection using neural networks.

3.1. Dataset Preparation and Preprocessing

We used the public dataset NEU-DET [35] for training and testing, which is a dataset specifically designed for the object detection of hot-rolled steel strip surface defects, provided by Northeastern University. As shown in Figure 7, it contains 1800 images consisting of six classes of defects with 300 images of 200 × 200 pixels for each class, including Crazing (Cr), Inclusion (In), Patches (Pas), Pitted Surface (PS), Rolled-in Scale (RS), and Scratches (Sc), where the GT box for each defect category is indicated by a red box. The training set and test set are divided according to the ratio of 8:2, with 1440 images and 360 images, respectively. To improve the generalization ability of the detection model, we performed image enhancement processing on the images before inputting them into the neural network, including contrast enhancement, image rotation and flipping, zooming and cropping, and mosaic enhancement.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 7. Defect classes: (**a**) Crazing; (**b**) Inclusion; (**c**) Patches; (**d**) Pitted Surface (**e**) Rolled-in Scale; (**f**) Scratches.

3.2. Experimental Setup

Table 2 lists the hardware and major software configurations utilized in the experiments of this paper. In the process of model training, the Stochastic Gradient Descent (SGD) optimizer was used with a weight decay of 0.0005 and a momentum of 0.937; a method of warmup and cosine annealing was used to adjust the learning rate, and the initial learning rate was set to 0.01; the input image size was 224 × 224, and the Mosaic method was used at the same time; and the batch size was set to 64 and epochs to 300 times.

Table 2. Parameters of device and environment.

Name	Parameter
CPU	Intel Core i9-7960X
GPU	RTX 3080Ti 12G
Operating System	Windows10
Software environment	Python 3.7 + Pytorch 1.8.1 + CUDA 11.3

3.3. Evaluation Metrics

It is not sufficient to evaluate the actual defect detection only by the precision metric; the detection speed also needs to be taken into account. Therefore, this paper evaluated the model from four aspects: model size, inference time, average precision (AP), and mean average precision (mAP) @0.5. The model size indicates the size of the model weights generated after the training was completed; inference time refers to the overall time of the single image forward inference, it reflects the actual speed of the model.

The *AP* denotes the average value of precision under different recalls, that is, the area enclosed by the P–R curve and the coordinate axis, reflecting the comprehensive performance of precision and recall. The specific calculation of precision P, recall R, and AP are as follows:

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

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

$$AP = \int_0^1 P(R)dR \tag{8}$$

where *TP* is true positive samples, *FP* is false positive samples, and *FN* is false negative samples.

The mAP refers to the average precision of all classes, which is one of the most essential and extensively employed evaluation metrics for the current object detection model, and can be used to evaluate the overall performance of the model. The mAP@0.5 refers to the average precision of all classes when the IoU is set to 0.5. The mAP is obtained from the following Equation (9):

$$mAP = \frac{\sum AP}{N} \tag{9}$$

where N is the total number of classes detected.

3.4. Performance Evaluation of Each Module

3.4.1. Attention Effectiveness Experiment

In this study, we compared the more widely employed attention modules, such as SE, CBAM, ECA, [36], etc., and the effect on the networks when using the two insertion methods is indicated in Figure 4. The results are shown in Table 3. From this, it can be seen that the mAP@0.5 was generally higher than the (b) method when we chose the (a) insertion method. When the attention mechanism was added to the residual block, since the residual block itself has a shortcut connection, the original input information could be transferred directly to the later layers. This can be tremendously helpful in protecting information integrity, so that the weights learned by the attention mechanism can act on the original input feature map, which is more conductive to the expression of essential feature information. When the attention is added in the last layer, the weights act on the feature information that has been extracted by the C3 module, but a loss of information is inevitable during the feature extraction process. Therefore, method (a) is more effective in this respect. As a result of the additional time required for module computations in the residual structure, (b) method requires less inference time than (a) method.

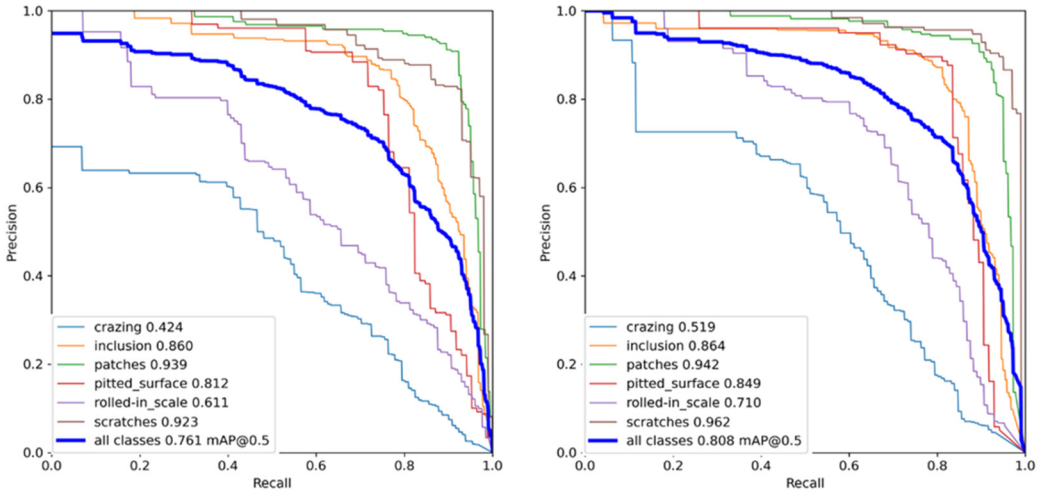
Table 3. Multiple attention compared the results in two insertion methods.

Method	(a)			(b)		
	Size (M)	mAP@0.5	Time (ms)	Size (M)	mAP@0.5	Time (ms)
None [14]	13.6	76.1	13.4	13.6	76.1	13.4
SE	13.7	78.7	14.6	13.7	78.1	13.9
CBAM	13.7	78.9	18.1	13.7	78.5	16.4
ECA	13.6	78.7	14.4	13.6	78.3	13.9
Transformer	17.9	79.2	85.1	17.9	79.1	73.8
CA	13.7	79.0	18.0	13.7	78.7	16.1
EMA (G = 4)	13.7	79.2	15.7	13.7	78.7	14.8
EMA (G = 8)	13.7	79.4	16.0	13.7	78.9	15.0

The novel attention module EMA, with G = 8, has the best effect on mAP@0.5 from Table 3 and (a) method and (b) method have about the same inference time. So, we choose the (a) method with the highest mAP.

3.4.2. Comparison Experiments

Figure 8 shows the experimental results of training using the identical configuration before and after improving the YOLOv5 model. As can be seen from the figure, each category of defect has improved, with the Crazing and Rolled-in Scale being the most noticeably improved by roughly 9% and 10%, and the mAP@0.5 has increased by 4.7% compared to 76.1% of the YOLOv5.



(**a**) (**b**)

Figure 8. Comparison of P–R curve before and after improvement. (**a**) Original YOLOv5s P–R curve; (**b**) MSFE-YOLOv5s P–R curve.

In this paper, the defect images of the test set were detected by the above two models to compare the actual detection effect before and after improvement more intuitively. Some comparative results of defect detection are shown in Figure 9. It can be seen that the MSFE-YOLOv5s network model has a better and more sufficient effect on defect detection.

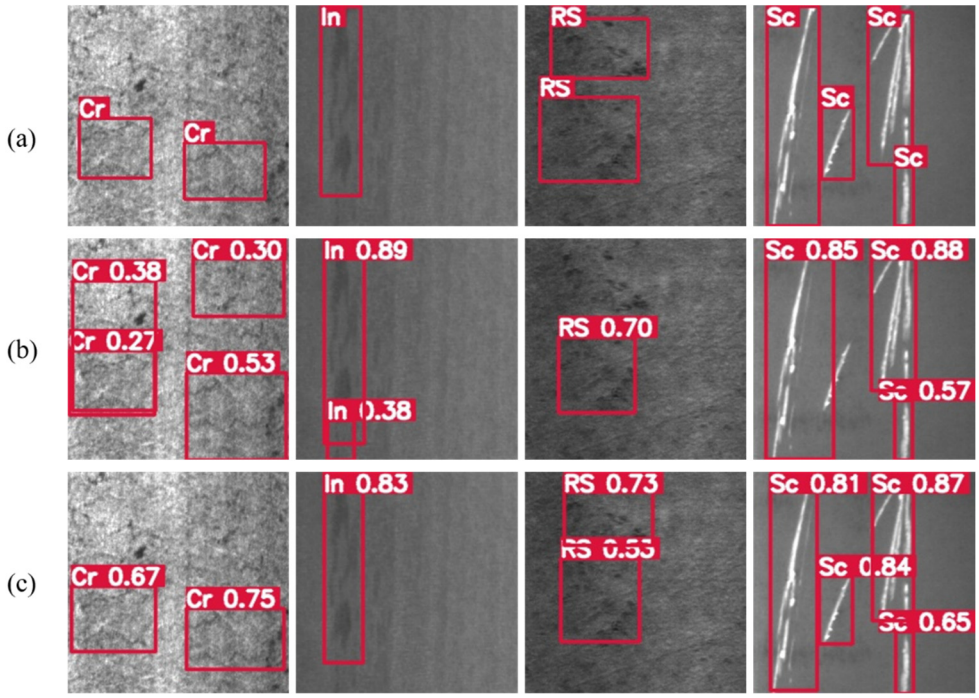


Figure 9. Comparison of model detection effect before and after improvement. (**a**) Original picture with GT; (**b**) original YOLOv5s testing result; (**c**) MSFE-YOLOv5s testing result.

Table 4 presents a comparison of the Faster R-CNN, Cascade R-CNN, RetinaNet, YOLOX, and YOLOv7 using the same dataset division standard to evaluate the effectiveness of the improved methods proposed in this paper. The results clearly show that the improved model achieves the highest mAP@0.5. Although the AP value for the Pitted Surface does not reach the maximum level, the AP values for the other defect categories are the highest. This indicates that overall, the algorithm outperforms the other models in detecting various

defects on steel surfaces. Compared to the two-stage networks like Cascade R-CNN and Faster R-CNN, this model significantly improves detection speed while ensuring real-time detection capability. Compared to the other YOLO models, this model further enhances detection accuracy while maintaining inference times at the millisecond level and keeping computational complexity low, achieving an excellent balance between accuracy and efficiency.

Table 4. Comparative experimental results.

Model	Size (M)	AP (%)						mAP@0.5 (%)	Time (ms)
		Cr	In	Pa	PS	RS	Sc		
Faster R-CNN	159.5	45.5	84.9	91.5	86.1	68.4	94.0	78.4	100.4
Cascade R-CNN	264.9	49.3	84.6	93.2	85.7	69.2	95.8	79.6	204.2
RetinaNet	145.1	49.0	82.8	94.0	87.9	66.3	91.0	78.5	54.8
YOLOv3	236.5	48.0	79.4	89.3	79.7	59.6	90.2	74.4	48.4
YOLOv5s	13.6	42.4	86.0	93.9	81.2	61.1	92.3	76.1	13.4
YOLOX	68.5	40.8	85.9	91.8	87.8	61.9	84.2	75.4	18.6
YOLOv7-tiny	11.6	48.7	82.5	93.5	83.5	53.5	88.9	75.1	11.1
YOLOv7	71.3	50.7	87.0	92.2	84.7	67.5	94.4	79.4	31.9
YOLOv8s	22.5	43.0	81.4	92.6	82.5	64.3	94.6	76.4	16.3
Our model	14.2	51.9	86.4	94.2	84.9	71.0	96.2	80.8	18.2

In order to verify the generalization of the improved model, this paper conducts experiments on the public datasets GC10-DET [37], Severstal Steel [38], and Crack500 [39], respectively. The experimental results are shown in Table 5, and the evaluation standard uses mAP@0.5. Among them, GC10-DET is a steel surface defect dataset, with a total of 2306 images of 2048 × 1000 resolution, including 10 types of defects. Severstal Steel is a steel plate surface defect dataset, which is a Kaggle competition dataset. Its image resolution is 1600 × 256 and it contains four types of defects. This paper removes the non-defective images and uses 6666 images for training and testing. The Crack500 is a dataset about road cracks, with a total of 3000 images with 500 × 500 resolution, including only the crack defect.

Table 5. Generalization verification of experimental results.

Model	GC10-DET	Severstal Steel	Crack500
YOLOv5s-pre	69.3%	57.5%	77.6
MSFE-YOLOv5s-pre	72.0% (↑2.7%)	59.7% (↑2.2%)	79.8% (↑2.2%)
YOLOv5s	61.9%	55.1%	78.1%
MSFE-YOLOv5s	66.4% (↑4.5%)	58.2% (↑3.1%)	81.1% (↑3.0%)

The experimental training environment remains unchanged, and the division ratio of the training set and the test set is still 8:2. In order to adapt to large-size images, the size of 640 × 640 is used for training. At the same time, the two cases of using and not using the pre-training model are compared. The pre-training model refers to the weight parameters obtained after the original YOLOv5 network is trained on the COCO dataset. The "pre" suffix in Table 5 indicates that the pre-training model is used. As can be seen from Table 5, after using the pre-training model on the three datasets, the mAP of the improved model can be increased by 2~3% compared with the original model, and it can be increased by 3~4% without using the pre-trained model. This shows that the improved model has good generalization.

3.4.3. Ablation Study

In order to explore the actual improvement effect of the three methods proposed in this paper on the YOLOv5 network model, an ablation experiment was conducted on the NEU-DET dataset with the same environment and parameter configuration. The study results are shown in Table 6.

Table 6. Results of ablation study.

Method	Size (M)	AP (%)						mAP@0.5 (%)	Time (ms)
		Cr	In	Pa	PS	RS	Sc		
YOLOv5s(baseline)	13.6	42.4	86.0	93.9	81.2	61.1	92.3	76.1	13.4
+C3EMA	13.7	48.1	85.3	93.7	86.5	67.8	94.9	79.4	16.0
+C3DX	14.2	50.8	85.8	94.5	85.5	69.3	95.7	80.3	18.2
+EIoU	14.2	51.9	86.4	94.2	84.9	71.0	96.2	80.8	18.2

Table 6 shows that using the EMA mechanism in the Backbone results in the most significant performance improvement for the Crazing and Rolled-in Scale, both increasing by approximately 6%. Overall, the mAP@0.5 improves by 3.3%, reaching 79.4%, with only a 2.6 ms increase in inference time, demonstrating the effectiveness of the EMA mechanism. Further, replacing C3_F with C3DX results in an additional 0.9% increase in the mAP@0.5. Replacing the loss function with EIoU further improves the mAP@0.5 by 0.5% without additional inference time. Although there is a decrease in AP for certain defect categories, these improvements are still effective overall. The ablation experiments reveal that the proposed improvements significantly enhance steel surface defect detection. With these methods, the mAP@0.5 of YOLOv5s on the NEU-DET dataset has increased from 76.1% to 80.8%.

4. Discussion

In the wave of promoting social progress and industrial upgrading, Artificial Intelligence (AI) has emerged as a key driving force. To deepen students' understanding of cutting-edge AI technologies, particularly the application of neural networks to solving complex industrial problems, we have proposed the MSFE-YOLO network and developed a corresponding steel surface defect detection experimental simulation system. This system aims to comprehensively cover both the theoretical and practical aspects of AI technology, ensuring that students can accurately grasp every step from model design to deployment.

Specifically, we have implemented multidimensional optimization strategies for the MSFE-YOLO network. First, by introducing the EMA mechanism at the Backbone level, we significantly enhanced the network's focusing capability during feature extraction, allowing the model to more effectively concentrate on key defect areas on steel surfaces, thereby improving the richness and accuracy of feature representation. Second, we meticulously designed the C3DX module in the Neck part, which effectively addresses the limitations of the single-scale features in complex defect recognition by fusing multi-scale features, significantly enhancing the model's ability to recognize defects of varying sizes and shapes. Finally, we made targeted improvements to the loss function to reduce the interference of complex backgrounds with detection results, further improving the network's detection accuracy and robustness in complex industrial environments. Although these improvements have shown significant enhancements in feature extraction and defect recognition, there are still aspects worth noting. For instance, while the C3DX module excels at multi-scale feature fusion, it may have limitations when handling extreme sizes or special shapes of defects; furthermore, lightweight modifications to the C3 network could further improve detection speed.

Our experiments validate the four main contributions mentioned in the introduction, particularly the significant performance improvements on the NEU-DET dataset. These results indicate that the proposed improvements are effective in practical applications.

Future work should explore the potential applications of these methods in other industrial fields, such as more complex industrial environments or different types of defect detection. Additionally, further optimization of the existing model is recommended to enhance its applicability and efficiency in real-world scenarios.

This experiment integrates knowledge from specialized courses like image processing, pattern recognition, and computer vision. It covers the entire chain of deep learning vision tasks, including data preprocessing, model construction, parameter tuning, training, and result evaluation, providing a platform that closely integrates theory and practice.

Author Contributions: Conceptualization, L.L. and R.Z.; methodology, L.L. and R.Z.; software, R.Z. and L.L.; writing—original draft preparation, T.X. and R.Z.; writing—review and editing, L.L., Y.H. and Y.Z.; validation, L.L. and H.Z.; visualization, R.Z.; supervision, L.L.; formal analysis, T.X. and R.Z.; investigation, L.L., R.Z. and H.Z.; resources, L.L. and T.X.; data curation, L.L.; project administration, R.Z.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation in China (Grant No. 61902436), the Hunan Provincial Research Project on Teaching Reform in Colleges and Universities (Grant No. HNJG-20230458), and the Hunan Provincial Research Project on Teaching Reform for Degree and Graduate Students (Grant No. 2023JGYB57).

Data Availability Statement: All data are contained within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

To facilitate reader comprehension, we provide a glossary in Table A1, which includes key concepts and acronyms.

Table A1. Glossary of Key Concepts and Acronyms.

Name	Meanings
YOLO	YOLO (You Only Look Once) is a deep learning model widely used for object detection tasks. Its core idea is to transform the object detection problem into a regression problem, predicting multiple classes and a bounding box
EMA	Efficient Multi-Scale Attention.
C3DX	Convolution 3 Dilated Convolution X
C3	The C3 module is a feature extraction structure in the YOLOv5 that enhances feature extraction and fusion capabilities by incorporating a Cross-Stage Partial (CSP) network. This design further optimizes the model's ability to capture and integrate features effectively.
mAP	mAP (Mean Average Precision) is a comprehensive metric used to evaluate a model's detection accuracy and localization precision across all categories, with higher values indicating better performance.

References

1. Vilček, I.; Řehoř, J.; Carou, D.; Zeman, P. Residual stresses evaluation in precision milling of hardened steel based on the deflection-electrochemical etching technique. *Robot. Comput.-Integr. Manuf.* **2017**, *47*, 112–116. [CrossRef]
2. Abbes, W.; Elleuch, J.F.; Sellami, D. Defect-Net: A new CNN model for steel surface defect classification. In Proceedings of the 2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC 2024), Marrakech, Morocco, 21–23 May 2024. [CrossRef]
3. Nguyen, H.-V.; Bae, J.-H.; Lee, Y.-E.; Lee, H.-S.; Kwon, K.-R. Comparison of Pre-Trained YOLO Models on Steel Surface Defects Detector Based on Transfer Learning with GPU-Based Embedded Devices. *Sensors* **2022**, *22*, 9926. [CrossRef] [PubMed]
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
5. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1483–1498. [CrossRef] [PubMed]
6. He, Y.; Jin, Z.; Zhang, J.; Teng, S.; Chen, G.; Sun, X.; Cui, F. Pavement surface defect detection using mask Region-Based convolutional neural networks and transfer learning. *Appl. Sci.* **2022**, *12*, 7364. [CrossRef]

7. Si, B.; Yasengjiang, M.; Wu, H. Deep learning-based defect detection for hot-rolled strip steel. *J. Phys. Conf. Ser.* **2022**, *2246*, 012073. [CrossRef]
8. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W. A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 592878. [CrossRef]
9. Shi, X.; Zhou, S.; Tai, Y.; Wang, J.; Wu, S.; Liu, J.; Xu, K.; Peng, T.; Zhang, Z. An improved faster R-CNN for steel surface defect detection. In Proceedings of the 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP 2022), Shanghai, China, 26–28 September 2022. [CrossRef]
10. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef]
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
12. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Jocher, G.R.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Hogan, A.; Hajek, J.; Diaconu, L.; Kwon, Y.; Defretin, Y.; et al. *Ultralytics/Yolov5: V5.0-YOLOv5-P6 1280 Models, AWS, Supervise.Ly and YouTube Integrations*; Zenodo: Geneva, Switzerland, 2021.
15. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOV5 based on transformer for detecting defects of steel surface. *Sensors* **2022**, *22*, 3467. [CrossRef]
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 6000–6010.
17. Zhu, W.; Zhang, H.; Zhang, C.; Zhu, X.; Guan, Z.; Jia, J. Surface defect detection and classification of steel using an efficient Swin Transformer. *Adv. Eng. Inform.* **2023**, *57*, 102061. [CrossRef]
18. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021. [CrossRef]
19. Yi, C.; Xu, B.; Chen, J.; Chen, Q.; Zhang, L. An improved YOLOX model for detecting strip surface defects. *Steel Res. Int.* **2022**, *93*, 2200505. [CrossRef]
20. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
21. Xikun, X.; Changjiang, L.; Meng, X. Application of attention YOLOV 4 algorithm in metal defect detection. In Proceedings of the 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT 2021), Chongqing, China, 22–24 November 2021. [CrossRef]
22. Wang, L.; Liu, X.; Ma, J.; Su, W.; Li, H. Real-Time Steel Surface Defect Detection with Improved Multi-Scale YOLO-v5. *Processes* **2023**, *11*, 1357. [CrossRef]
23. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023. [CrossRef]
24. Jocher, G.; Chaurasia, A.; Qiu, J. *Ultralytics YOLO, Version 8.0.0*; Ultralytics Inc.: Los Angeles, CA, USA, 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 10 June 2023).
25. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
26. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *IEEE Conf. Proc.* **2020**, *2020*, 1571–1580.
27. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
28. Brauwers, G.; Frasincar, F. A General Survey on attention Mechanisms in Deep Learning. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 3279–3298. [CrossRef]
29. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
30. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Computer Vision—ECCV 2018*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; pp. 3–19. [CrossRef]
31. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Montreal, BC, Canada, 10–17 October 2021. [CrossRef]
32. Yao, C.; Tang, Y.; Sun, J.; Gao, Y.; Zhu, C. Multiscale residual fusion network for image denoising. *IET Image Process.* **2021**, *16*, 878–887. [CrossRef]
33. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef]

34. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
35. He, Y.; Song, K.; Meng, Q.; Yan, Y. An End-to-End steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1493–1504. [CrossRef]
36. Qilong, W.; Banggu, W.; Pengfei, Z.; Peihua, L.; Wangmeng, Z.; Qinghua, H. ECA-Net: Efficient channel attention for deep convolutional neural networks. *IEEE Conf. Proc.* **2020**, *2020*, 11531–11539.
37. Lv, X.; Duan, F.; Jiang, J.-j.; Fu, X.; Gan, L. Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network. *Sensors* **2020**, *20*, 1562. [CrossRef] [PubMed]
38. Severstal: Steel Defect Detection. Available online: https://www.kaggle.com/c/severstal-steel-defect-detection (accessed on 21 May 2021).
39. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1525–1535. [CrossRef]