[1]Cheng Jia
[1]Gaoyuan Qin
[1]Yanwen Zhang
[2]Jinchao Miao
[1] Guoqiang Ren*

YOLO-VMTC: Enhancing Steel Surface Defect Detection with a Lightweight and Context-Aware Deep Learning Approach

Abstract: - In the steel manufacturing process, the surface quality of steel not only reflects the integrity of the steel surface but also significantly influences the quality and safety of downstream production equipment. To address the shortcomings of traditional steel surface defect detection models, such as insufficient feature extraction capability, low detection accuracy, and the large computational load and model size, we propose an optimized YOLOv8-based algorithm for steel surface defect detection, termed YOLO-VMTC, which incorporates lightweight and context-aware features. The stated detection model augments the conventional YOLOv8 backbone architecture through the fusion of VanillaNet and the Multi-Scale Contextual Feature module. This integration upholds the backbone's lightweight design while incorporating multi-scale contextual features, consequentially bolstering the model's feature extraction prowess. Additionally, the neck network of YOLOv8 undergoes refinement via the introduction of the Triplet Attention Mechanism and C2fSKConv module. This adjustment minimizes computational demands and mitigates interference from irrelevant information, empowering the model to precisely identify and emphasize defect locations on steel surfaces. Furthermore, the substitution of the conventional IoU loss with WiseIoU loss within the detection framework accelerates training, conserving both computational power and time. The inclusion of Meta-ACON, a component capable of adaptively modulating its activation patterns, fortifies the model's generalization capacity, particularly when confronted with intricate and dynamic tasks like steel surface defect analysis. Experimental evaluations on the NEU-DET steel surface defect dataset reveal that this model attains an impressive mAP of 79.0% while maintaining a swift detection speed of 96.3 FPS. When compared to the baseline YOLOv8 defect detection model, it exhibits enhancements not only in detection accuracy but also in speed, offering a robust foundation for quality assurance within steel manufacturing processes.

Keywords: VanillaNet module; Context feature fusion; Triplet Attention Mechanism; Meta-ACON activation function

1. Introduction

Hot-rolled strip steel, an indispensable product in modern industrial production activities, has been extensively applied in sectors such as construction, automotive, machinery, and aerospace in recent years. The quality and efficiency of this material are directly related to the operation and development of these industries. However, during the production and manufacturing process of strip steel, factors such as the inhomogeneity of raw material composition, fluctuations in rolling process parameters, and equipment aging can lead to surface defects such as patches, rolled-in scale, and scratches on the strip steel. The presence of surface defects in hot-rolled strip steel serves as a pivotal determinant of material quality, adversely affecting aspects such as fatigue strength, corrosion resistance, and overall service lifespan [1], while also posing latent safety concerns. Consequently, the identification of these defects is of paramount importance for safeguarding the quality of steel production processes and enhancing the output of high-quality products.

As science and technology continue to progress at a rapid pace, conventional approaches to detecting surface flaws in strip steel [2], such as manual examination, ultrasonic testing, thermography, and magnetic particle in-

[1] * Jinan Campus (Swinburne College), Shandong University of Science and Technology, Jinan 250031, Shandong,China.Email:renguoqiang@sdust.edu.cn

[1] School of Materials Science and Engineering, East China University of Science and Technology, Xvhui200237, Shanghai,China.

spection, have grown progressively inadequate. These methods are hampered by their heavy dependence on human expertise, substantial costs, inefficiencies, and limited accuracy, falling short of the rigorous demands of modern industry for precise and efficient defect identification. In response, the utilization of advanced machine vision technologies and artificial intelligence algorithms for automated detection of surface defects in hot-rolled strip steel has emerged as a vibrant area of research and application [3-5]. The swift evolution of deep learning has given rise to object detection algorithms capable of autonomously pinpointing target objects within input imagery through end-to-end training, facilitating real-time, cost-effective, and efficient detection of surface defects in strip steel. The mainstream object detection algorithms can be broadly categorized into two groups: region-based methods and single-stage detection methods. Region-based techniques, notably R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], and Mask R-CNN [9], involve generating potential regions of interest through region proposal algorithms. Subsequently, each of these candidate regions undergoes feature extraction and classification, followed by regression to refine the object's precise location and bounding box. Although region-based defect detection algorithms have high detection accuracy, they are slower in computation. Single-stage detection methods directly classify and locate the entire image, achieving multi-scale detection by outputting boundary boxes of different sizes at various levels. Compared to region-based methods, these methods have slightly lower detection accuracy but offer faster detection speed and better real-time performance, making them more suitable for defect detection applications in industry.

Among the spectrum of single-stage object detection algorithms, the YOLO (You Only Look Once) algorithm stands out for its remarkable attributes, namely its swift detection speed, elevated accuracy, and proficiency in tackling multi-object detection challenges within intricate environments. In comparison to conventional manual inspection techniques and machine learning-driven detection methods, the YOLO algorithm [10-12] markedly boosts detection efficiency, curbs costs, and adeptly identifies a diverse array of defect types, thereby elevating the precision and dependability of the detection process. To further refine the performance of the YOLO algorithm in identifying surface defects in hot-rolled strip steel, researchers have embarked on a series of enhancements and optimizations.

Wang et al. [13] infused new vitality into the research in this field by introducing the Swin-Transformer-YOLOv5 algorithm, an innovative approach designed to achieve efficient and precise detection of surface defects in hot-rolled strip steel. This model, underpinned by GhostNet, streamlines computational demands and parameter count. Its integration of Swin-Transformer and CoordAttention modules elevates feature extraction, while BiFPN fortifies scale-invariant detection capabilities, adeptly identifying both subtle and extensive defects.Lu et al. [14] introduced the Resformer-Unet architecture, a U-shaped design that synergizes convolutional neural networks with Transformers, enhancing multi-scale feature extraction and the capture of both global and local image details. This architecture cleverly employs skip connections to mitigate information loss during the downsampling process.Wan et al. [15] have innovated by integrating SPD-Conv to preserve spatial integrity, reducing feature loss during downsampling. Their SPPFCSPC module, enriched with CBS modules and SPPF, bolsters the model's feature extraction and fusion, while the CA mechanism accentuates the significance of key feature channels.Li et al. [16] presented a model that integrates MSFE for multi-scale feature extraction with EFF for feature fusion, achieving high performance with reduced computational overhead in steel surface defect detection.Fan et al. [17] enhanced the YOLOv5 framework with ACD-YOLO, optimizing anchor boundaries with an advanced genetic algorithm and incorporating CAM to sharpen the model's focus on critical areas, thus elevating detection precision.Lu et al. [18] refined the YOLOv5s model by incorporating the ASFF and CARAFE modules for heightened sensitivity to fine-grained features, alongside lightweight modules for streamlined model architecture.Wang et al. [19] proposed an innovative hybrid model that combines the strengths of ResNet and Vision Transformer, ingeniously addressing the challenge of uneven information density distribution in the surface defect detection process of Integrated Circuits (ICs). On the other hand, Zhang et al. [20] presented an enhanced PP-YOLOE-m network that strengthens feature extraction and spatial perception capabilities through automated data augmentation techniques and carefully coordinated attention mechanisms within the CSPRes structure. Additionally, the introduction of the SIoU loss function aims to improve the accuracy of regression predictions.Wang et al. [21] and Lv et al. [22] further contributed with lightweight algorithms for real-time defect detection, integrating advanced com-

ponents to enhance feature extraction and detection speed, while navigating the delicate balance between accuracy, speed, and model complexity.Despite the proliferation of optimization algorithms in defect detection, striking an optimal balance between detection precision, speed, and model compactness remains an ongoing challenge.

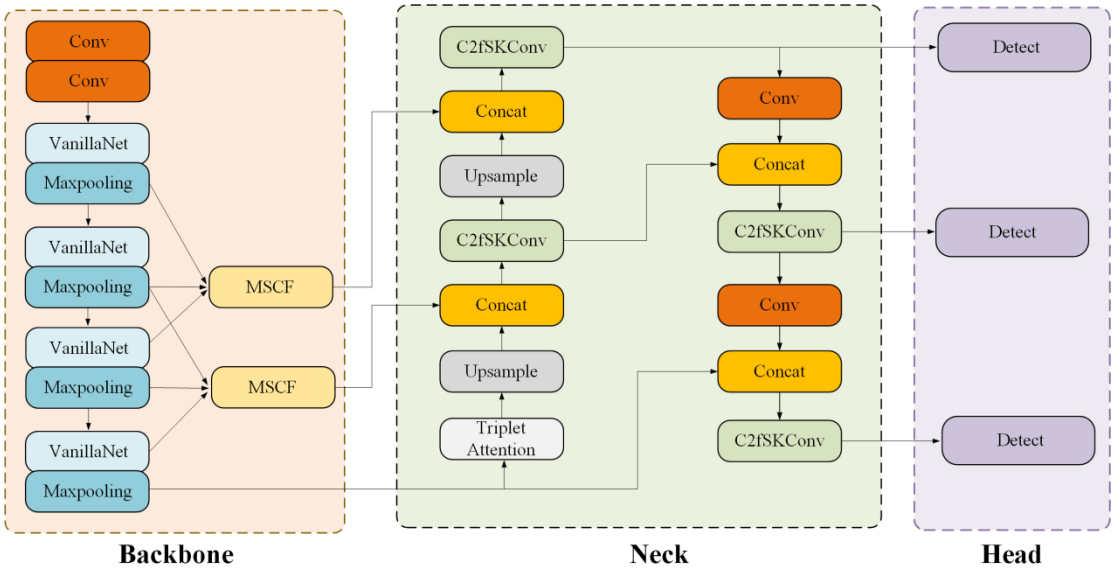


Figure 1. YOLO-VMTC network structure.

To overcome the existing constraints in the realm of steel strip surface defect detection, including inadequate feature extraction, reduced detection accuracy, along with significant computational demands and model complexity, this research presents a streamlined approach named YOLO-VMTC. This algorithm builds upon the YOLOv8 framework and undergoes optimization, offering the following key advancements:

(1) Acknowledging the intricate relationship between steel surface defects and their adjacent textures and structures, a Multi-scale Context Information Fusion Module (MSCF) has been devised within the Backbone architecture. This module aims to augment the model's ability to comprehend and interpret contextual features more comprehensively. Furthermore, the integration of a lightweight VanillaNet module [23] into the core of the target detection model ensures that high-level performance is maintained while simultaneously minimizing the consumption of computational resources and reducing inference time.

(2) A Triplet attention mechanism [24] has been integrated into the Neck structure, establishing interactions across different dimensions through rotational operations, capturing cross-dimensional dependencies, and thus enhancing defect recognition in complex backgrounds.

(3) By combining the variable kernel convolution SKConv [25] with the C2f module, a novel feature extraction structure, C2fSKConv, has been devised. This structure allows the model to maintain high precision with lower computational complexity and adapt more flexibly to the diversity and complexity of steel surface defects.

(4) Compared to traditional activation functions, Meta-ACON [26] dynamically adjusts the activation function based on the input samples, learning a more robust representation of steel surface images, which aids the model in better accommodating the various complexities that may arise in steel surface defect detection tasks.

(5) The Wise-IoU loss [27] has been implemented in lieu of the CIoU loss, enhancing the accuracy in measuring the similarity between target bounding boxes. This substitution contributes to an improvement in the model's overall defect detection performance.

The schematic illustration of the proposed YOLO-VMTC model's architecture is presented in Figure 1. Extensive experiments utilizing the publicly available NEU-DET dataset have validated the exceptional performance of the

YOLO-VMTC model in steel surface defect detection, fulfilling the stringent practical requirements of industrial steel strip defect inspection.

2. METHODS

2.1. VanillaNet Network

The VanillaNet network architecture, proposed by Huawei Noah's Ark Lab, is an ultra-simplified design that addresses efficient computation in resource-constrained environments by reducing network complexity, while maintaining or even enhancing model accuracy. Its structure is exceedingly streamlined, comprising three main components: the stem, body, and head sections. The stem section is responsible for transforming the input image's three channels into multiple channels and performing initial downsampling. The body section, the core feature extraction part of the network, adopts an extremely concise design, with each stage consisting of a single layer of network layers. The VanillaNet model, consisting of a series of identical blocks stacked sequentially, undergoes a transformation in its feature maps wherein the resolution decreases incrementally while the number of channels escalates after each stage. This architectural approach skillfully harmonizes computational efficiency with robust feature representation, enabling VanillaNet to capture intricate image features while adhering to a low computational footprint. The structural configuration of VanillaNet is visually represented in Figure 2.

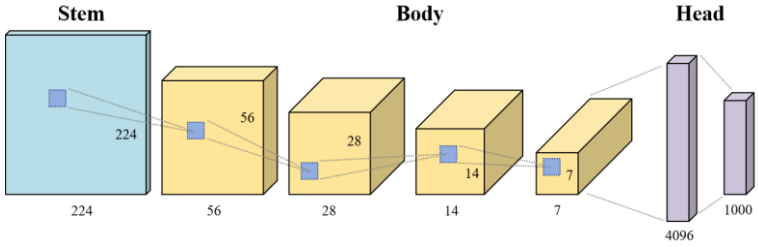


Figure 2. VanillaNet Structure.

Due to VanillaNet's composition of lightweight convolutional modules and efficient feature fusion strategies, it possesses formidable feature extraction capabilities. Consequently, the backbone network constructed based on the VanillaNet module can more effectively capture target features within images, thereby enhancing both the speed and accuracy of surface defect detection.

2.2. Meta-ACON

ACON is an innovative activation function capable of adaptively learning whether neurons should remain inactive or become activated, offering a novel paradigm for the selection of activation functions. The expressions for the Smooth max and ACON activation functions are presented as follows:

$$S_\beta(x_1,\ldots,x_n)=\frac{\sum_{i=1}^{n} x_i \cdot e^{\beta x_i}}{\sum_{i=1}^{n} e^{\beta x_i}} \tag{1}$$

$$\begin{aligned} Meta\text{-}ACON &= S_\beta(\eta_a(x),\eta_b(x)) \\ &= \eta_a(x)\frac{e^{\beta\eta_a(x)}}{e^{\beta\eta_a(x)}+e^{\beta\eta_b(x)}}+\eta_b(x)\frac{e^{\beta\eta_b(x)}}{e^{\beta\eta_a(x)}+e^{\beta\eta_b(x)}} \\ &= \eta_a(x)\frac{1}{1+e^{-\beta(\eta_a(x)-\eta_b(x))}}+\eta_b(x)\frac{1}{1+e^{-\beta(\eta_b(x)-\eta_a(x))}} \\ &= (\eta_a(x)-\eta_b(x))\cdot Sigmoid(\beta(\eta_a(x)-\eta_b(x)))+\eta_b(x) \end{aligned} \tag{2}$$

Among them, β is a hyperparameter, η_a and η_b is a monotone linear function. In the process of neural network construction and training, ReLU and Swish are the most commonly used activation functions in nonlinear transformation, and Meta-ACON unifies the expression of the two and simplifies the construction of activation functions. When $\beta = 1, \eta_a = x$, $\eta_b = 0$ then $Meta\text{-}ACON = x \cdot Sigmoid(x)$ is the Swish activation function; when $\beta = 0$, the calculation result of Meta-ACON is equivalent to a linear function. By controlling the parameter β, the activation function can be adaptively adjusted, which simplifies the selection of activation functions in the model configuration process, saves a lot of time and resources generated by testing the model and combining different activation functions, and enables the model to optimize its activation strategy according to its feedback during the training process, which not only reduces the workload of hyperparameter tuning but also makes the model design more reasonable.

2.3. MSCF module

The Multi-scale Context Feature Fusion structure (MSCF) represents a strategy aimed at bolstering feature extraction capabilities. It achieves this by incorporating features spanning diverse scales, thereby capturing intricate details alongside multi-scale contextual information embedded within feature maps. This process concurrently retrieves output feature vectors from the target layer and its immediately neighboring layers. Following alignment procedures, the dimensions of these three vectors are harmonized. The target layer's output vector undergoes a Sigmoid transformation, which is then applied element-wise to multiply with the output vectors from the adjacent layers. The resulting products are aggregated through summation, culminating in the fusion of multi-scale contextual features. This fusion process significantly augments the representational richness and expressiveness of the features.

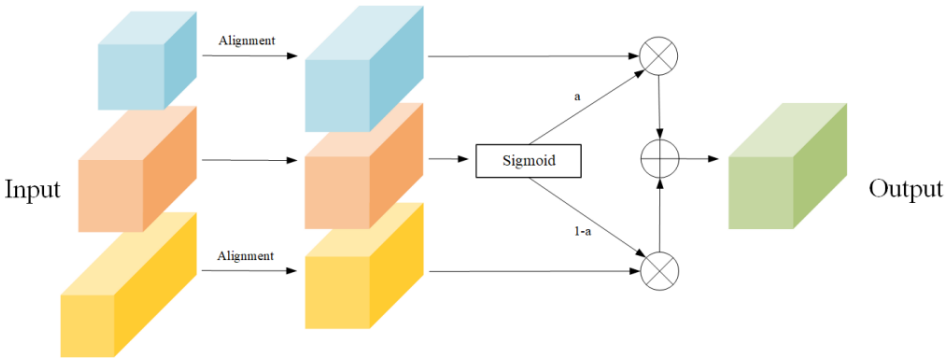


Figure 3. Schematic diagram of MSCF structure principle.

2.4. Wise-IoU loss

Within the context of YOLO v8, the CIoU loss metric serves to quantify the closeness of the predicted bounding box to the target object, integrating the aspect ratio factor into the IoU loss to better align the target box in terms of overlap, centroid distance, and aspect ratio. However, the diversity in dimensions and shapes of defects found in the steel strip surface defect dataset, coupled with their varying distributions within images, can compromise the precision of the model's bounding box predictions. To mitigate this challenge, the Wise-IoU loss is employed in the defect detection algorithm as a substitute for CIoU loss. This switch minimizes the deviation between the anchor box and target box, thereby refining prediction accuracy and fortifying the bounding box loss's fitting capabilities.

The formula for Wise-IoU is as follows：

$$L_{Wise\text{-}IoU} = R_{Wise\text{-}IoU} L_{IoU} \tag{3}$$

$$R_{Wise\text{-}IoU} = \exp\left(\frac{(x-x_{gt})^2 + (y-y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) \tag{4}$$

Herein, IoU denotes the degree of overlap between the true bounding box and the predicted one; $*$ signifies the separation of W_g, H_g from the computational graph, thereby effectively eliminating factors that impede convergence.

2.5. Triplet Attention Mechanism

Triplet Attention is an innovative attention computation method that captures cross-dimensional interactions through a tripartite branch structure, establishing interdependencies between different dimensions via rotational operations followed by residual transformations. The first branch rotates the input tensor by 90 degrees along the height dimension, the second branch by 90 degrees along the width dimension, and the third branch undergoes no rotation. The tensors from the three branches are processed through a Z-pooling layer, which reduces the dimensionality of the tensor by concatenating features from max and average pooling. Post Z-pooling, the tensors pass through a convolutional layer followed by a batch normalization layer to generate intermediate output vectors. The intermediate vectors from the three branches are aggregated by attention weighting and simple averaging to yield the final output. This mechanism enhances the network's capacity for feature learning and expressiveness.

The neck network in YOLOv8 is tasked with the integration of feature maps across various scales, culminating in the generation of final feature maps used for predictions. The introduction of the Triplet attention mechanism within the neck network enables the model to aggregate feature information from different dimensions, focusing more intently on key areas within the input features, and producing more accurate and refined feature maps. This contributes to the defect detection model's enhanced capability to identify targets within images, reducing instances of false positives and missed detections.

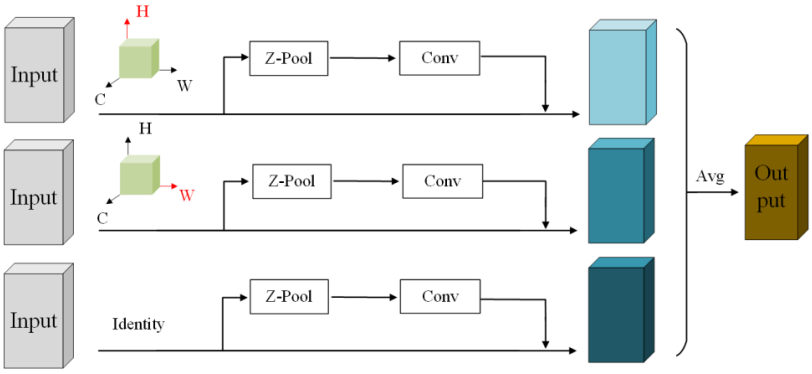


Figure 4. Triplet Attention structure diagram.

2.6. C2fSKConv structure

Traditional deep convolutional neural networks utilize fixed-size convolutional kernels to process targets of all scales and shapes, which can lead to suboptimal detection performance when the scale and shape of the targets vary significantly. To address this, the SKConv structure is introduced, which adaptively learns the shape and size of the convolutional kernels based on the varying spatial structure of the input data, aggregating information from multiple convolutional kernels. The primary process consists of two main parts: split and fuse. During the split phase, the input data undergoes various convolutional operations for feature extraction at different scales, while in the fuse phase, a gating mechanism controls the flow of information from different convolutional branches. This gating mechanism generates a weight vector that is used to compute a weighted sum with the outputs of the different convolutional branches, yielding the final output feature map.

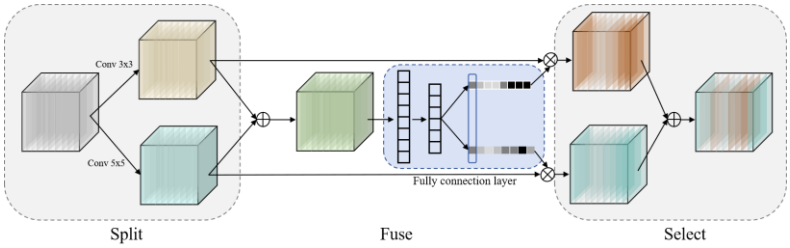


Figure 5. Selective Kernel Convolution Module.

By integrating the SKConv structure with the C2f module, a novel feature extraction module named C2fSKConv has been designed, as depicted in Figure 6. This architecture is capable of capturing a richer set of feature information, which contributes to enhancing the model's feature representation capabilities.

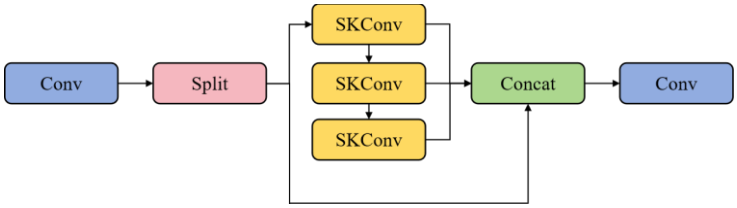
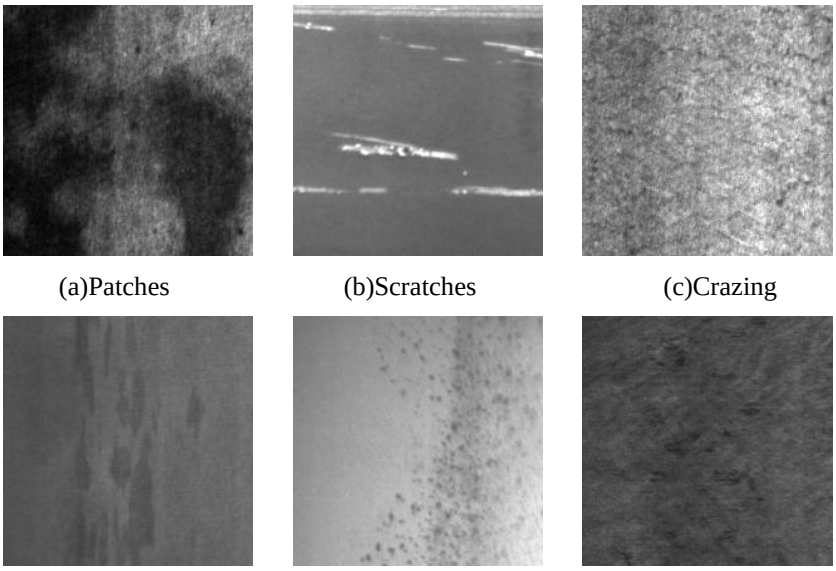


Figure 6. C2fSKConv Module.

3. EXPERIMENT AND RESULT ANALYSIS

3.1. Experimental environment and data set

The experimental configuration for this study comprised an Intel(R) Xeon(R) Gold 5218 CPU and an NVIDIA GeForce RTX 3090 GPU with 24GB of memory. The development environment leveraged PyCharm as the programming tool, while PyTorch 1.10 served as the deep learning framework. The dataset utilized originated from the NEU-DET[28], an open-access repository maintained by Northeastern University in China, featuring six distinct categories of steel strip surface defects: Crazing (Cz), Inclusion (In), Patches (Pa), Pitted Surface (Ps), Rolled-in Scale (Rs), and Scratches (Sc). Each category encompassed a set of 300 images, amounting to a total of 1800 images depicting various steel strip surface defects. This dataset was randomly partitioned into training and testing subsets at an 80:20 ratio, resulting in 1440 images for training and 360 images for testing. Figure 7 visually showcases the six defect categories encompassed within the dataset.



(a)Patches (b)Scratches (c)Crazing

(d)Inclusion (e)Pitted Surface (f)Rolled-in Scale

Figure 7. Six defect categories in the NET-DET dataset.

For the experimental phase, Python served as the programming language of choice, while CUDA 11.3 was employed to expedite the training process of the devised defect detection model. The input images were resized to 640x640 pixels, and the training was carried out over a total of 1000 iterations. The learning rate was set at 0.01, accompanied by a momentum of 0.937, and a batch size of 128 images per iteration. The specifics of the experimental environment and the configured parameters are outlined in Table 1 for clarity.

Table 1. Experimental Environment and Parameter Configuration.

Name	Parameter
CPU	Intel(R) Xeon(R) Gold 5218
GPU	NVIDIA GeForce RTX 3090 24G
Python version	3.8
Operating system	Windows 10
Deep learning framework	Pytorch 1.10
CUDA	11.3
Input_Size	640x640
Epoch	1000
Learning_rate	0.01
Momentum	0.937

3.2 Evaluation Indicators

To evaluate the effectiveness of the steel strip surface defect detection model, a set of performance metrics were utilized, namely Recall (R), Precision (P), Average Precision (AP), and Mean Average Precision (mAP). The methodologies for calculating these metrics were also detailed, allowing for a comprehensive assessment of the model's performance.

$$R = \frac{TP}{TP + FN} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{3}$$

$$mAP = \frac{\sum_{i=0}^{n} AP(i)}{n} \tag{4}$$

Herein, TP denotes the number of true positive samples correctly identified by the model, FN represents the number of true positive samples that the model failed to recognize, and FP indicates the quantity of false positive samples where the model erroneously classifies negative samples as positive. Additionally, we have taken into account the model's parameter count (Params), computational load (GFLOPs), detection speed (FPS), and model size as comparative metrics in our experiments.

3.3. Ablation experiment

To validate the efficacy of YOLO-VMTC, YOLO v8 was employed as the baseline network. Ablation studies were conducted on four distinct structural components: the VanillaNet module, the HWD module, the CA attention mechanism, and the C2fSKConv structure. The presence of √ in the table indicates the utilization of a particular module. The results of the ablation experiments are presented in Table 2.

Table 2. Ablation experiment.

Method	VanillaNet	Triplet Attention	MSCF	C2fSKConv	mAP/%	GFLOPs	Params
YOLOv8					76.2	8.2	3.01×106
Method1	√				76.1	6.6	2.06×106
Method2		√			76.5	8.3	3.01×106
Method3			√		77.1	8.2	3.01×106
Method4				√	76.4	7.4	2.66×106
Method5	√	√			77.3	6.6	2.07×106
Method6	√		√		78.2	6.6	2.06×106
Method7	√			√	76.9	6.6	2.39×106
Method8		√	√		78.4	8.3	3.01×106
Method9		√		√	78.6	8.2	2.67×106
Method10			√	√	78.5	7.6	2.66×106
Method11	√	√	√		78.8	6.7	2.07×106
Method12	√	√		√	78.7	5.7	1.73×106
Method13		√	√	√	78.5	7.5	2.67×106
Method14	√	√	√	√	79.0	5.8	1.61×106

The results of the ablation study indicate that Method1, which employs the VanillaNet module and an improved backbone network of the baseline model YOLO v8, experienced a slight decrease of 0.1% in mAP compared to the baseline model, while computational load was reduced by 19.5% and the parameter count was decreased by 31.5%. This suggests that the improved backbone network significantly reduced the parameter count and computational load with only a marginal loss in detection accuracy.The incorporation of Triplet Attention in Method2, MSCF in Method3, and C2fSKConv in Method4 led to an increase in mAP, with Method4 maintaining a lower computational complexity and parameter count. Overall, the combined improvements in Method6, Method7, Method8, and Method9 demonstrated higher mAP values, with Method6 achieving 78.2% and Method9 reaching 78.6%, indicating that these optimized modules synergistically enhanced performance. Notably, Method12, although slightly lower in mAP than Method9, exhibited reduced parameter count and computational complexity, which may confer faster speeds in practical detection tasks.Method13, while showing a minor decrease in mAP compared to Method11 and Method12, further reduced the parameter count, effectively controlling model complexity with a negligible trade-off in performance. Method14 introduced an improved backbone network with VanillaNet and MSCF, and optimized the neck network with the Triplet attention mechanism and C2fSKConv structure. Compared to the baseline model, Method14 increased mAP by 2.8%, reduced computational load by 29.2%, and decreased the model's parameter count by 46.5%. When compared with other methods, Method14 demonstrated the best performance, enhancing efficiency while maintaining model performance.

3.4. Activation function comparison experiment

Activation functions introduce nonlinear transformations into neural networks, enabling the learning of more complex features within the input data. Different activation functions may affect the model's convergence rate and generalization capability. Generally, activation functions with larger derivatives, such as ReLU, may accelerate convergence but could also lead to model instability. The Sigmoid function has an output range between 0 and 1,

which can be interpreted as a probability, but may cause saturation in the output layer, affecting the model's generalization performance. The YOLO-VMTC model utilizes the Meta-ACON adaptive activation function. To compare the impact of different activation functions on the YOLO-VMTC model, ReLU, SiLU, LeakyReLU, and Mish were selected for comparative experiments. The results of these experiments are presented in Table 3.

Table 3. Comparison experiment of activation function.

Activation Function	mAP/%	Cr/%	In/%	Pa/%	Ps/%	Rs/%	Sc/%
ReLU[29]	77.4	45.3	86.5	94.0	91.2	55.2	86.9
SiLU[30]	78.1	46.8	87.7	96.1	90.1	53.4	88.6
LeakyReLU[31]	78.7	48.5	86.4	95.2	90.2	57.2	89.1
Mish[32]	78.8	49.3	86.2	94.3	89.3	56.1	88.9
Meta-ACON	79.0	49.7	86.8	95.8	90.4	57.6	91.0

The experimental results reveal discernible differences in the impact of various activation functions on object detection performance. The conventional ReLU, while simple and efficient, achieved an mAP of 77.4%, outperformed by other activation functions. SiLU, by integrating the properties of the Sigmoid and linear functions, realized a modest performance enhancement, achieving the highest detection accuracy of 87.7% in the 'In' category. LeakyReLU, which allows a non-zero gradient for negative inputs, further improved model performance, reaching an mAP of 78.7%. Mish, an emerging activation function, showed results comparable to LeakyReLU. The Meta-ACON adaptive activation function demonstrated the best performance in terms of mAP, highlighting its advantage in adaptively adjusting and enhancing the model's detection capabilities.

3.5 Comparative experiment of different IoU loss functions

To further validate the impact of different IoU loss functions on the YOLO-VMTC steel strip surface defect detection model, CIoU, SIoU, GIoU, and WiseIoU loss functions were selected for comparative analysis. The experimental results are presented in Table 4.

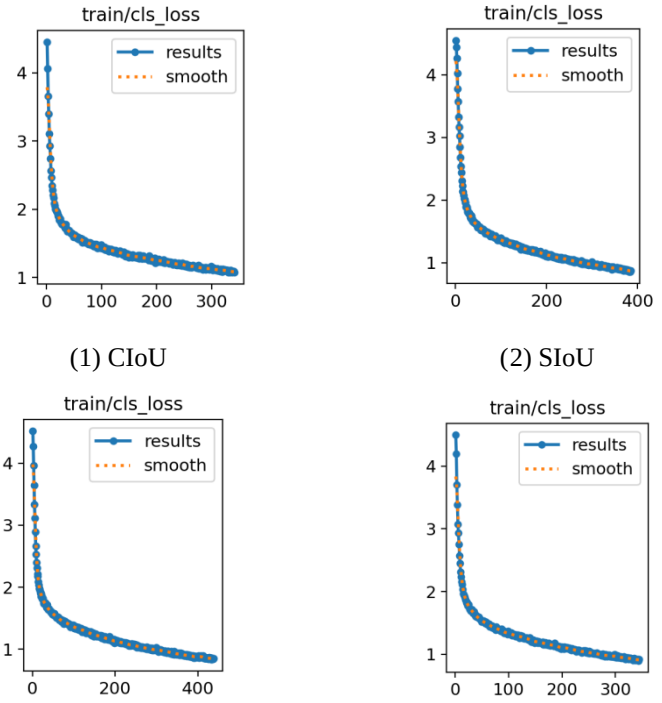
Table 4. Comparison experiment of different IoU losses.

Loss	mAP/%	FPS	Volume/MB
CIoU[33]	76.7	106	5.2
SIoU[34]	75.3	93	5.2
GIoU[35]	78.4	100	5.2
WiseIoU	79.0	96.3	5.2

The experimental results indicate that among the four IoU loss functions evaluated, there was no change in the model size. However, there were differences in performance as measured by the mAP metric. The YOLO-VMTC models trained with the SIoU and CIoU loss functions exhibited the lowest mAP values, at 75.3% and 76.7% respectively. In contrast, the model trained with the WiseIoU loss function demonstrated the best mAP performance, reaching 79.0%. Regarding detection speed, the models trained with the CIoU and GIoU loss functions were the fastest, with speeds of 106 frames per second and 100 frames per second, respectively. The model trained

with the WiseIoU loss function followed closely with a speed of 96.3 frames per second. Through the analysis and comparison of the aforementioned experimental comparative data, the YOLO-VMTC detection model showed the best mAP performance and relatively fast detection speed when employing the Wise-IoU loss function.

The experiment also compared the loss function variation curves of these four loss functions, as depicted in Figure 8. The CIoU loss function converges rapidly, requiring fewer training epochs for stabilization. In contrast, the convergence rates of the SIoU and GIoU loss functions are slower, both exceeding 400 iterations. The YOLO-VDCW detection model trained with the Wise-IoU loss function demonstrated a notably shorter training duration, showing relatively stable performance after approximately 350 epochs. Although the CIoU loss converges quickly, the final detection accuracy is lower than that achieved with WiseIoU..



(1) CIoU (2) SIoU

(3) GIoU (4) WiseIoU

Figure 8. Convergence curves for different IoU losses.

3.6 Model comparison experiment

To substantiate the performance of the steel surface defect detection model YOLO-VMTC, comparative experiments with different defect detection models were conducted using the same NEU-DET dataset. The comparative experimental results are documented in Table 5.

Table 5. Comparative experiments of different models

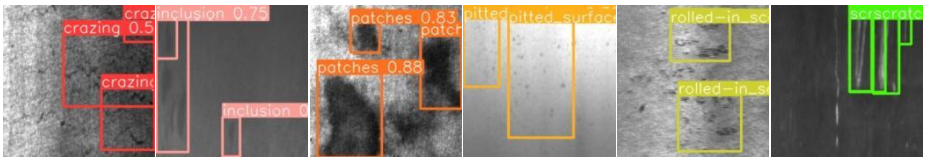
Algorithm	mAP/%	FPS
SSD	71.4	44.6
Faster RCNN	68.5	36.3
Mask RCNN	74.2	47.5
YOLOv5	74.5	75.3
YOLOv6	71.2	78.3

YOLOv7	72.9	85.2
YOLOX	72.4	83.6
YOLOv8n	76.2	88
YOLO-VMTC	79.0	96.3

As indicated in Table 5, the detection algorithms for steel surface defects implemented by Faster R-CNN, Mask R-CNN, and SSD exhibit relatively low detection accuracy and slow speeds, making them unsuitable for real-time steel surface defect detection scenarios. The detection algorithms of YOLOv5, YOLOv6, YOLOv7, and YOLOX all achieve an FPS (frames per second) value exceeding 70, which meets the requirements for real-time detection scenarios, yet there is room for improvement in accuracy. In the YOLO-VMTC algorithm, the mAP (mean Average Precision) and detection speed have reached 79.1% and 96.3 FPS, respectively, demonstrating that this algorithm has achieved a higher level of both accuracy and speed compared to other algorithms.

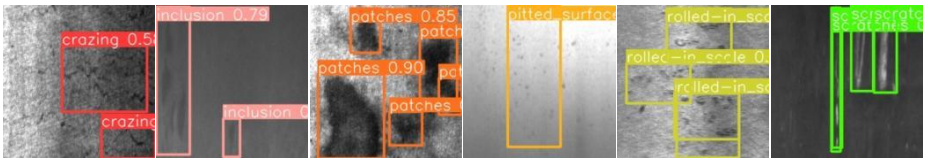
3.7 Experimental Results Analysis

Validation tests were conducted on the traditional YOLOv8 and the YOLO-VMTC algorithms using a steel surface defect detection test set, with selected detection results extracted from the experimental outcomes and compared as shown in Figure 9. Among the detection results for the six categories of steel strip surface defects in the NEU-DET test set, the conventional YOLOv8 defect detection model exhibited instances of missed detections. In contrast, the YOLO-VMTC algorithm demonstrated superior performance to the traditional YOLOv8, with higher detection accuracy and more comprehensive defect detection results. The comparative results substantiate the effectiveness of the algorithm.



Cr In Pa PS RS Sc

(1)YOLO v8 defect detection model



Cr In Pa PS RS Sc

(2)YOLO-VMTC defect detection model

Figure 9. Diagram of defect detection effect.

4. CONCLUSIONS

Addressing the shortcomings of traditional steel surface defect detection models, such as insufficient feature extraction capability, low detection accuracy, and large model computation and size, we propose an optimized lightweight steel surface defect detection algorithm based on YOLOv8, termed YOLO-VMTC. This detection algorithm enhances the precision and efficiency of steel surface defect detection through a series of innovative improvements. Specifically, the YOLO-VMTC algorithm, with optimizations to the VanillaNet and MSCF modules, strengthens the feature extraction capability of the backbone network and refines the neck network through the Triplet attention mechanism and the C2fSKConv module, effectively reducing computational complexity and

mitigating interference from irrelevant information. Furthermore, the introduction of the Meta-ACON activation function and the WiseIoU loss function has further improved the model's generalization ability and training speed.

Experimental results demonstrate that the YOLO-VMTC algorithm achieved an mAP of 79.0% on the NEU-DET dataset, with a detection speed of up to 96.3 FPS, showing a significant enhancement in both detection accuracy and speed compared to the traditional YOLOv8 detection model. Ablation studies further confirm the effectiveness of the proposed modules, and comparative experiments with different activation functions and IoU loss functions also show that Meta-ACON and WiseIoU achieved optimal detection performance in their respective categories. By comparing with other advanced object detection algorithms, YOLO-VMTC has shown excellent performance in the task of steel surface defect detection. YOLO-VMTC not only has a distinct advantage in detection accuracy but also meets the requirements for real-time processing in terms of detection speed. Compared to other versions in the YOLO series, YOLO-VMTC further improves detection speed while maintaining high precision, proving its practicality and efficiency in industrial steel strip defect detection.

The YOLO-VMTC algorithm holds significant application value in the field of steel surface defect detection. Its lightweight design and optimized network structure make it suitable for deployment in industrial sites with limited computational resources for defect detection. However, the detection accuracy when dealing with complex noisy environments and multiple categories of defects in the same image is still not satisfactory. Therefore, further improving the model's detection accuracy should remain a focus of future research.

Author Contributions: Conceptualization, Cheng Jia, Yanwen Zhang, Xilin Zhang and Guoqiang Ren; Data curation, Gaoyuan Qin; Formal analysis, Gaoyuan Qin; Investigation, Xilin Zhang; Methodology, Cheng Jia, Yanwen Zhang and Guoqiang Ren; Resources, Yanwen Zhang; Software, Cheng Jia and Gaoyuan Qin; Supervision, Guoqiang Ren; Validation, Gaoyuan Qin; Writing – original draft, Yanwen Zhang; Writing – review & editing, Cheng Jia, Xilin Zhang and Guoqiang Ren. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset address used in this paper http://faculty.neu.edu.cn/songkechen/zh_CN/zhym/263269/list/index.htm

Acknowledgments: Support by colleagues and the university is acknowledged.

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

REFERENCES

[1] Luo Q, Fang X, Liu L, et al. Automated visual defect detection for flat steel surface: A survey[J]. IEEE Transactions on Instrumentation and Measurement, 2020, 69(3): 626-644.
[2] Hu C, Ma R, Du X, et al. Boundary-aware residual network for defect detection in strip steel products[J]. Evolving Systems, 2024: 1-15.
[3] Liu G, Ma Q. Strip steel surface defect detecting method combined with a multi-layer attention mechanism network[J]. Measurement Science and Technology, 2023, 34(5): 055403.
[4] Liu R, Huang M, Gao Z, et al. MSC-DNet: An efficient detector with multi-scale context for defect detection on strip steel surface[J].
[5] Mi Z, Gao Y, Xu X, et al. Steel strip surface defect detection based on multiscale feature sensing and adaptive feature fusion[J]. AIP Advances, 2024, 14(4).
[6] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
[7] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
[8] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
[9] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
[10] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
[11] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[12] Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems, 2016, 29.
[13] Wang Q, Dong H, Huang H. Swin-Transformer-YOLOv5 for lightweight hot-rolled steel strips surface defect detection algorithm[J]. Plos one, 2024, 19(1): e0292082.
[14] Lu K, Wang W, Pan X, et al. Resformer-Unet: A U-shaped Framework Combining ResNet and Transformer for Segmentation of Strip Steel Surface Defects[J]. ISIJ International, 2024, 64(1): 67-75.
[15] Wan W, Wang L, Wang B, et al. Space to depth convolution bundled with coordinate attention for detecting surface defects[J]. Signal, Image and Video Processing, 2024, 18(5): 4861-4874.
[16] Li Z, Wei X, Hassaballah M, et al. A deep learning model for steel surface defect detection[J]. Complex & Intelligent Systems, 2024, 10(1): 885-897.
[17] Fan J, Wang M, Li B, et al. ACD-YOLO: Improved YOLOv5-based method for steel surface defects detection[J]. IET Image Processing, 2024, 18(3): 761-771.
[18] Lu J, Zhu M, Ma X, et al. Steel Strip Surface Defect Detection Method Based on Improved YOLOv5s[J]. Biomimetics, 2024, 9(1): 28.
[19] Wang X, Gao S, Guo J, et al. Deep Learning-Based Integrated Circuit Surface Defect Detection: Addressing Information Density Imbalance for Industrial Application[J]. International Journal of Computational Intelligence Systems, 2024, 17(1): 1-18.
[20] Zhang Y, Liu X, Guo J, et al. Surface defect detection of strip-steel based on an improved PP-YOLOE-m detection network[J]. Electronics, 2022, 11(16): 2603.
[21] Wang Q, Dong H, Huang H. Swin-Transformer-YOLOv5 for lightweight hot-rolled steel strips surface defect detection algorithm[J]. Plos one, 2024, 19(1): e0292082.
[22] Lv, Baozhan, et al. "Research on Surface Defect Detection of Strip Steel Based on Improved YOLOv7." Sensors 24.9 (2024): 2667.
[23] Chen H, Wang Y, Guo J, et al. Vanillanet: the power of minimalism in deep learning[J]. Advances in Neural Information Processing Systems, 2024, 36.
[24] Misra D, Nalamada T, Arasanipalai A U, et al. Rotate to attend: Convolutional triplet attention module[C]//Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021: 3139-3148.
[25] Li X, Wang W, Hu X, et al. Selective kernel networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 510-519.
[26] Ma N, Zhang X, Liu M, et al. Activate or not: Learning customized activation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8032-8042.
[27] Tong Z, Chen Y, Xu Z, et al. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism[J]. arXiv preprint arXiv:2301.10051, 2023.
[28] He Y, Song K, Meng Q, et al. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features[J]. IEEE transactions on instrumentation and measurement, 2019, 69(4): 1493-1504.
[29] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]//Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011: 315-323.
[30] Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning[J]. Neural networks, 2018, 107: 3-11.
[31] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc. icml. 2013, 30(1): 3.
[32] Misra D. Mish: A self regularized non-monotonic activation function[J]. arXiv preprint arXiv:1908.08681, 2019.
[33] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.
[34] Gevorgyan Z. SIoU loss: More powerful learning for bounding box regression[J]. arXiv preprint arXiv:2205.12740, 2022.
[35] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 658-666.

ABOUT THE AUTHOR

Cheng Jia

Jinan Campus (Swinburne College), Shandong University of Science and Technology, Jinan 250031, Shandong,China.

E-mail: cs_jiacheng@126.com

Gaoyuan Qin

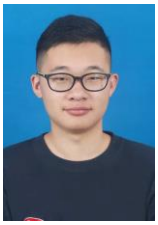
Jinan Campus (Swinburne College), Shandong University of Science and Technology, Jinan 250031, Shandong,China.

E-mail: 202213040216@sdust.edu.cn

Yanwen Zhang

Jinan Campus (Swinburne College), Shandong University of Science and Technology, Jinan 250031, Shandong,China.

E-mail:202113020631@sdust.edu.cn



Xilin Zhang

School of Materials Science and Engineering, East China University of Science and Technology, Xvhui200237, Shanghai,China

E-mail: 1617434798@qq.com

Guoqiang Ren

Jinan Campus (Swinburne College), Shandong University of Science and Technology, Jinan 250031, Shandong,China

E-mail: renguoqiang@sdust.edu.cn