


## Article

# Research on Metallurgical Saw Blade Surface Defect Detection Algorithm Based on SC-YOLOv5

Lili Meng <sup>1,\*</sup>, Xi Cui <sup>1</sup>, Ran Liu <sup>2,\*</sup>, Zhi Zheng <sup>1</sup> , Hongli Shao <sup>3</sup>, Jinxiang Liu <sup>1</sup>, Yao Peng <sup>1</sup> and Lei Zheng <sup>1</sup>

<sup>1</sup> College of Mechanical Engineering, North China University of Science and Technology, Tangshan 063210, China

<sup>2</sup> College of Metallurgy & Energy, North China University of Science and Technology, Tangshan 063210, China

<sup>3</sup> Tangshan Metallurgical Saw Blade Co., Ltd., Tangshan 063000, China

\* Correspondence: mengll@ncst.edu.cn (L.M.); liuran@ncst.edu.cn (R.L.)

**Abstract:** Under the background of intelligent manufacturing, in order to solve the complex problems of manual detection of metallurgical saw blade defects in enterprises, such as real-time detection, false detection, and the detection model being too large to deploy, a study on a metallurgical saw blade surface defect detection algorithm based on SC-YOLOv5 is proposed. Firstly, the SC network is built by integrating coordinate attention (CA) into the Shufflenet-V2 network, and the backbone network of YOLOv5 is replaced by the SC network to improve detection accuracy. Then, the SIOU loss function is used in the YOLOv5 prediction layer to solve the angle problem between the prediction frame and the real frame. Finally, in order to ensure both accuracy and speed, lightweight convolution (GSConv) is used to replace the ordinary convolution module. The experimental results show that the mAP@0.5 of the improved YOLOv5 model is 88.5%, and the parameter is 31.1M. Compared with the original YOLOv5 model, the calculation amount is reduced by 56.36%, and the map value is increased by 0.021. In addition, the overall performance of the improved SC-YOLOv5 model is better than that of the SSD and YOLOv3 target detection models. This method not only ensures the high detection rate of the model, but also significantly reduces the complexity of the model and the amount of parameter calculation. It meets the needs of deploying mobile terminals and provides an effective reference direction for applications in enterprises.

**Keywords:** YOLOv5; metallurgical saw blade; lightweight; deep learning; defect detecting



**Citation:** Meng, L.; Cui, X.; Liu, R.; Zheng, Z.; Shao, H.; Liu, J.; Peng, Y.; Zheng, L. Research on Metallurgical Saw Blade Surface Defect Detection Algorithm Based on SC-YOLOv5. *Processes* **2023**, *11*, 2564. <https://doi.org/10.3390/pr11092564>

Academic Editors: Xin Yao and Huaqing Xie

Received: 9 August 2023

Revised: 21 August 2023

Accepted: 25 August 2023

Published: 27 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the context of intelligent manufacturing, machine vision [1,2] has become an important research direction for artificial intelligence, and same-target defect detection has become the focus of research. In recent years, with the rapid development of China's industrial level, the demand for metallurgical saw blades has increased significantly. While ensuring the output of metallurgical saw blades, the quality requirements have also become an important issue. As a multi-blade tool [3], which is essentially a consumable, the saw blade will still have defects even though metallurgical technology and the manufacturing process of saw blades have been continuously improved in recent years. These defects, such as cracks, pitting, and scarring, greatly affect the service life of a saw blade. The real-time performance of manual detection [4,5] cannot be guaranteed. Therefore, a good surface defect detection technology for metallurgical saw blades can not only ensure real-time performance, but also ensure surface quality, which is of great significance to service life.

With the development of industrial intelligence, defect detection is broadly divided into two directions [6–8]. One is the direction based on traditional image processing, and the other is the direction based on deep learning image detection. In traditional image processing, Liu et al. [9] proposed a two-branch balanced saliency model based on discriminant features for fabric defect detection. This method can be used for accurate fabric

defect detection and even surface defect detection for other industrial products. Joung et al. [10] used infrared imaging to detect defects in pipelines. Zhang et al. [11] proposed an algorithm combining local binary patterns (LBPs) and a gray-level co-occurrence matrix (GLCM). The local feature information and overall texture information of the defect image are extracted using LBPs and the GLCM, respectively. There are many ways of feature extraction. However, for specific traditional image processing methods, real-time processing and accuracy should be considered to meet the demand. In image detection methods based on deep learning, faster RCNN, YOLO and other series of networks are mainly used. Faster RCNN is a typical representative of two-stage target detection model, but the detection and training process is still relatively complex. Sun et al. [12] proposed a new face detection scheme based on deep learning and obtained advanced face detection performance by improving the faster RCNN framework. Shou et al. [13] proposed an improved region-based convolutional neural network (RCNN) fast detection method for airport detection in large-scale remote sensing images. Multi-scale training is applied to faster RCNN to enhance the robustness of the network for airport detection of different scales. The core idea of YOLO is to transform the target detection into a regression problem [14,15], using the whole map as the input of the network, just after a neural network, so that YOLO uses the whole graph as the input to the network, and just goes through a neural network to obtain the position of the bounding box and its category. Wang et al. [16] investigated the improved YOLOv4 algorithm using a shallow feature enhancement mechanism for the problems of insensitivity to small objects and low detection accuracy in traffic light detection and recognition. Xian et al. [17] used a triple loss function in YOT-Net in order to improve defect detection accuracy for copper elbows. Image similarity was used to enhance the feature extraction capability. The YOT-Net method for copper elbow surface defects was proposed. Wang et al. [18] proposed an optimized micro YOLOv3 algorithm with less computation and higher accuracy to solve the problem of insufficient accuracy of the original micro YOLOv3 algorithm in target detection in a lawn environment. Xue et al. [19] proposed an improved forest fire small-target detection model based on YOLOv5 for the problems of high unpredictability and strong destructiveness of forest fires. The model improves the backbone layer of YOLOv5 and adds an attention mechanism module to improve the identifiability of small forest fire targets.

This paper focuses on a defect detection algorithm based on deep learning, which can have high transmission efficiency while ensuring high detection accuracy, and the model is lightweight enough to meet the needs of enterprise applications. Taking the metallurgical saw blade of a metallurgical saw blade factory in Tangshan as the research object, the main network Shuffnet-V2 [20] was replaced in the YOLOv5 model, and the coordinate attention mechanism CA [21–23] was added to effectively improve the ability to extract metallurgical saw blade features. The lightweight convolution GSConv module was used to reduce the volume of the model, accelerate the speed of model reasoning, and realize the lightweight nature of the model. The regression loss function SIoU was introduced to accelerate the fitting of the data [24,25]. At the same time, the application of the improved lightweight identification model was studied.

## 2. Improved YOLOv5 Methods

### 2.1. YOLOv5s Target Detection Models

YOLOv5 is a classical algorithm for single-stage target detection. The YOLOv5s architecture comprises the input, backbone network, neck network, and prediction head [26]. The backbone is responsible for feature extraction. The neck is responsible for feature fusion. The head contains three detection heads, which are responsible for outputting detection information. The YOLOv5s network is the network with the smallest depth and the smallest width of the feature map in the YOLOv5 series. The YOLOv5 network structure is shown in Figure 1 below.

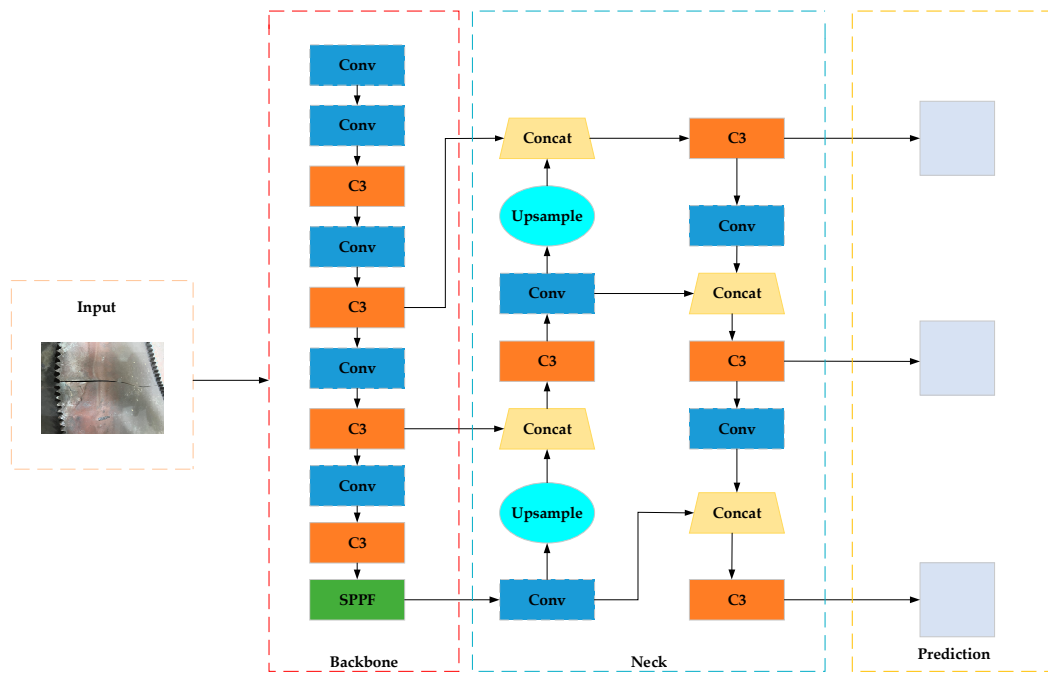


Figure 1. YOLOv5 network architecture.

2.2. SC-YOLOv5 Improved Model

To meet the needs of low recognition accuracy and a large number of parameters, this study improves the YOLOv5 model. While improving the recognition accuracy, the parameters of the model are reduced and the model inference speed is accelerated. In this study, the coordinate attention mechanism was introduced into the Shuffenet-V2 network (a lightweight network) to construct the SC network, and the SC network structure was used as the lightweight backbone network of the YOLOv5 model. The lightweight convolution module (GSConv) was used to replace the convolution module, and SIoU was introduced to redefine the loss function so that the lightweight degree of the backbone network was effectively improved. The improved SC-YOLOv5 model network structure based on YOLOv5 is shown in Figure 2.

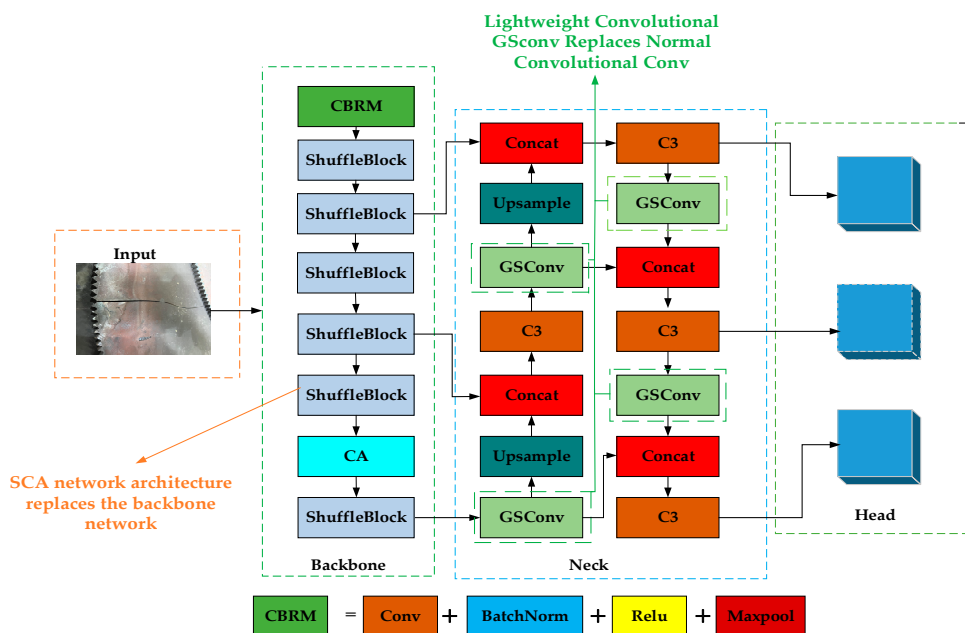


Figure 2. SC-YOLOv5 network structure.

In Figure 2, the SC-YOLOv5 network structure is divided into four parts, in which Input inputs and processes the dataset. Backbone extracts the features of the input metallurgical saw blade images and uses the SC network module as a new backbone network to improve the feature extraction ability of the model for key information. The acquired feature map is fused by Neck, and the lightweight convolution module is used instead of the ordinary convolution module. Head performs regression prediction, introduces the SIOU function to calculate the regression loss, and improves the convergence ability of the model.

### 2.3. Shuffenet-V2 Architecture

The most important part of the Shufflenet V2 network structure is the basic residual unit (block), which has two branching structures. As shown in Figure 3, the first structure performs a channel division operation at the input and divides the input feature map into two branches [27]; the primary branch contains three convolution operations the secondary branch does not perform any operation, and the input and output channels of each branch remain the same. The second structure splits the feature map into two branches, with three convolution operations in the primary branch and one depth-divisible convolution and one pointwise convolution in the secondary branch. The residual unit merges the output feature maps of the two branches by splicing at the output and further performs channel blending on the merged feature maps. Different subgroups are randomly extracted for rearrangement into new feature maps so that the next group convolution can fuse the input feature information from different groups, improving the information flow between channel groups and ensuring that the input and output channels are correlated.

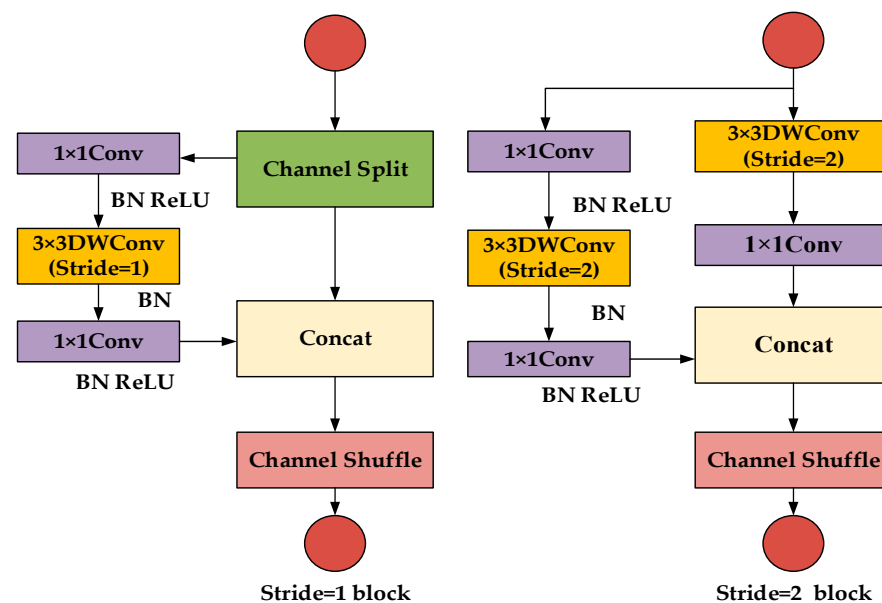


Figure 3. Basic residual unit network structure.

The ShuffleNetV2 network structure is mainly composed of the Conv1 layer, MaxPool layer, Stage 2 layer, Stage 3 layer, Stage 4 layer, Conv5 layer, and FC layer. The Stage 2 layer, Stage 3 layer, and Stage 4 layer are composed of the superposition of basic residual units. Among them, the Stage 2 layer and the Stage 4 layer are superimposed with a total of 4 basic residual units, and the Stage 3 layer is superimposed with a total of 8 basic residual units. The step size of the first basic residual unit in each Stage is 2, the main purpose is for downsampling, and the step size of other basic residual units is 1. A network with different complexity can be designed by changing the number of output channels in each structural layer of the network structure. Based on the ShuffleNet V2  $1 \times$ , the number of output channels in the Conv1 layer, MaxPool layer, Stage 2 layer, Stage 3 layer, Stage

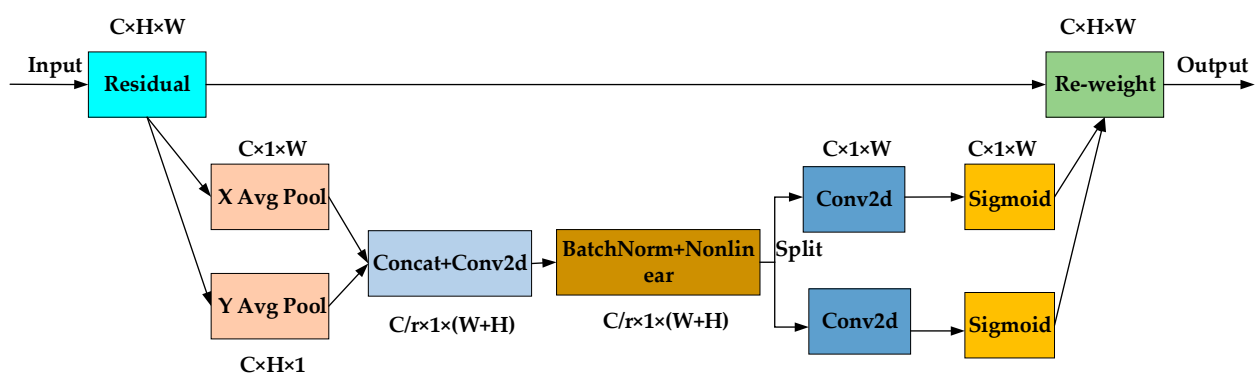
4 layer, Conv5 layer, and FC layer is 24, 24, 116, 232, 464, 1024, and 1000, respectively, as shown in Table 1.

**Table 1.** ShuffleNetV2 network architecture.

Layer	Output Size	KSize	Stride	Repeat	Output Channels
Image	$224 \times 224$				3
Conv1	$112 \times 112$	$3 \times 3$	2	1	24
MaxPool	$56 \times 56$	$3 \times 3$	2		24
Stage 2	$28 \times 28$		2	1	116
	$28 \times 28$		1	3	
Stage 3	$14 \times 14$		2	1	232
	$14 \times 14$		1	7	
Stage 4	$7 \times 7$		2	1	464
	$7 \times 7$		1	3	
Conv 5	$7 \times 7$	$1 \times 1$		1	1024
FC					1000

#### 2.4. Coordinate Attention Mechanisms

Most previous attention mechanisms for lightweight networks used SE modules [28,29], which only consider interchannel information but ignore positional information. Although later CBAM modules [30] tried to extract positional attention information by convolution after reducing the number of channels, convolution can only extract local relations and lacks the ability to extract long-range relations. Therefore, a new efficient attention mechanism, coordinate attention (CA), is proposed, which is able to encode horizontal and vertical location information into channel attention, allowing mobile networks to focus on a wide range of location information without incurring too much computational cost. It is shown in Figure 4, where  $C$  is the number of channels,  $W$  is the width,  $H$  is the height, and  $r$  is the dimension.



**Figure 4.** Coordinate attention network structure.

The advantages of the location attention module are mainly as follows:

1. It captures not only interchannel information, but also direction-dependent positional information, which helps the model to better locate and identify the target.
2. It is flexible and lightweight enough to be easily inserted into the core structure of a mobile network.
3. It can be used as a pre-trained model for a variety of tasks such as detection and segmentation, both of which show good performance improvement.

#### 2.5. Redefined SIoU Loss Functions

YOLOv5s uses the CIoU loss function [31,32], although the length–width ratio of the predicted bounding box to the real bounding box is introduced on the basis of the DIoU loss, which makes the loss function pay more attention to the shape of the bounding box.

However, the calculation of CIoU loss is relatively complex, which may lead to a large computational overhead in the training process. In order to solve the above problems, the loss function SIOU is introduced into the improved network model, which not only considers the overlapping area, distance, length, and width, but also considers the angle between the prediction box and the real box. The SIOU loss function consists of four cost functions: angle cost, distance cost, shape cost, and IoU cost.

The SIOU loss function is as follows:

$$L_{\text{box}} = 1 - IOU + \frac{\Delta + \Omega}{2} \quad (1)$$

The overall loss function is as follows:

$$L = W_{\text{box}}L_{\text{box}} + W_{\text{cls}}L_{\text{cls}} \quad (2)$$

In the formulas above,  $IOU$  denotes conventional regression loss,  $\Delta$  denotes distance loss,  $\Omega$  denotes shape loss,  $W_{\text{box}}$  denotes frame loss model volume,  $L_{\text{box}}$  denotes regression loss,  $W_{\text{cls}}$  denotes categorical loss model volume, and  $L_{\text{cls}}$  denotes focal loss.

### 2.6. Lightweight Convolutional GSConv

In the lightweight model design, the deep neural network has only used deep separated convolution (DSC) from start to finish. Although the number of parameters is reduced and the separation of channels and regions is achieved, DSC does not effectively use the feature information of different channels at the same spatial location. In order to make the output of the DSC as close to the SC as possible, a new method, GSConv, was introduced, which, as shown in Figure 5, introduces the SC-generated information (dense convolution operation) into each part of the DSC-generated information using shuffle. This method allows the information from SC to be completely blended into the output of DSC. GSConv first downsamples the inputs for normal convolution and then performs deep convolution using DSC; the results from SC and DSC are stitched together, and finally, a shuffle operation is performed.

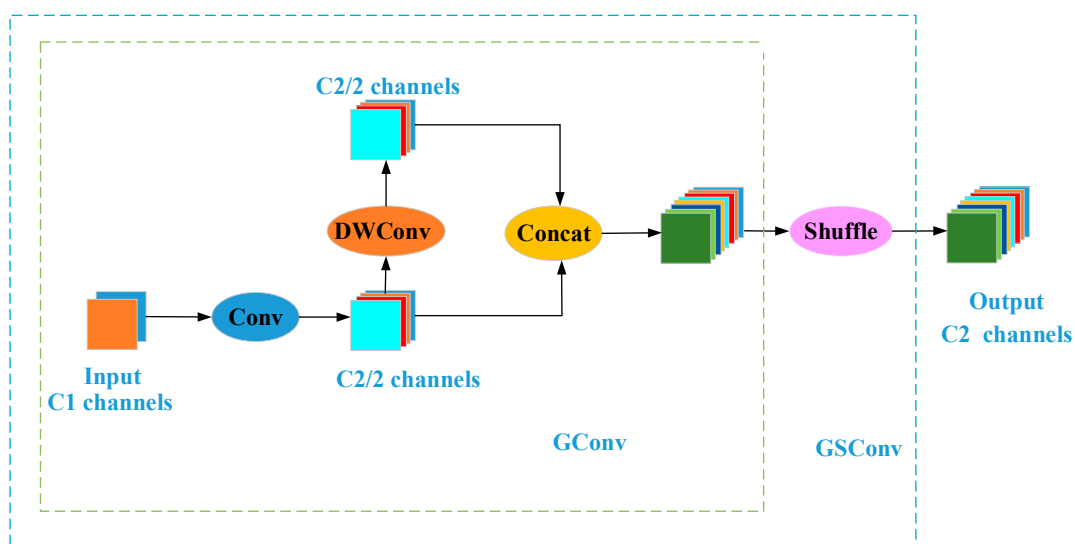


Figure 5. GSConv network structure.

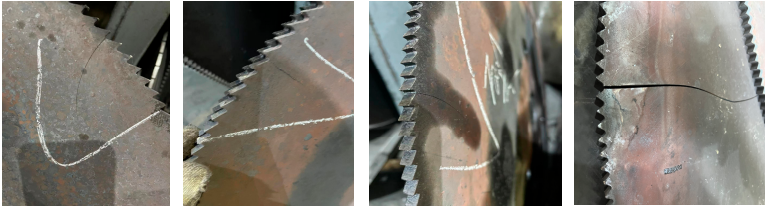




## 3. Metallurgical Saw Blade Dataset

### 3.1. Collection of Datasets

The dataset used industrial cameras mainly collected from a metallurgical saw blade factory in Tangshan. In order to increase the diversity of metallurgical saw blades and improve the generalization ability of the recognition model, natural image data from differ-

ent angles and different environments were selected when collecting datasets. Through random noise, Gaussian blur, random cropping, random rotation, random translation, and other data amplification methods of the original image, three kinds of metallurgical saw blades with cracks, pitting corrosion, and scarring were collected. The metallurgical saw blade defect part of the picture is shown in Table 2.

**Table 2.** Metallurgical saw blade partial data sheet.

Photos	Description
	Collected images of some metallurgical saw blade defects, numbered 000001–000004.
	Collected images of some metallurgical saw blade defects, numbered 000005–000008.
	Collected images of some metallurgical saw blade defects, numbered 000009–000012.
	Collected images of some metallurgical saw blade defects, numbered 000013–000016.
	Collected images of some metallurgical saw blade defects, numbered 000017–000020.

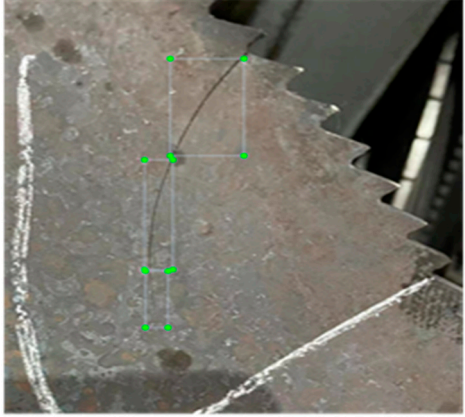

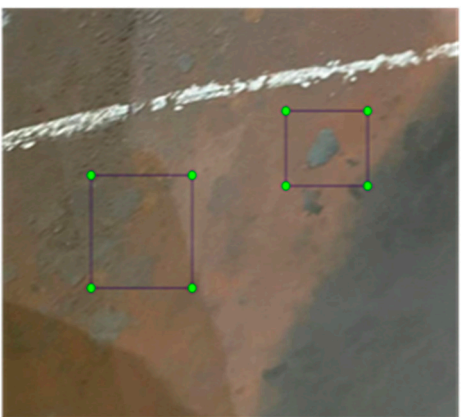
### 3.2. Dataset Processing

The collected dataset needed to be pre-processed, including data labeling, label conversion, and data storage.

1. Use labelling software ([https://gitcode.net/mirrors/tzutalin/labelimg?utm\\_source=csdn\\_github\\_accelerator](https://gitcode.net/mirrors/tzutalin/labelimg?utm_source=csdn_github_accelerator), accessed on 8 August 2023) to label the dataset information,

use the labeling software to frame the position of defects in the image, and save it as an xml format file, as shown in Table 3.

**Table 3.** Multiple defect labeling data.

Photos	Description
	<p>The picture on the left shows the crack defects of metallurgical saw blades, which are easily caused by the large temperature difference during the quenching process.</p>
	<p>The picture on the left shows a metallurgical saw blade scarring defect, which is easily caused by poor surface cleaning when cutting steel.</p>
	<p>The figure on the left represents the pitting corrosion defects of metallurgical saw blades. When there are localized damages on the surface of metallurgical saw blades, such as small holes, microcracks, and scratches, corrosive media and oxides are easily gathered and thus form pitting corrosion.</p>

2. The annotation information required by the Yolo algorithm is the coordinates of the center of the target frame in the sample image and the width and height of the target frame. Therefore, the xml annotation file needs to be converted. The specific steps are as follows:
  - Calculate the YOLO type annotation data: set the coordinates of the center point of the target frame on the sample image as  $(x, y)$  and the width and height of the target frame



as  $w$  and  $h$ , respectively. According to the known data of  $\langle x_{min} \rangle$ ,  $\langle x_{max} \rangle$ ,  $\langle y_{min} \rangle$ , and  $\langle y_{max} \rangle$ , they can be represented as follows (3):

$$\begin{aligned} w &= (x_{max}) - (x_{min}) \\ h &= (y_{max}) - (y_{min}) \\ x &= (x_{min}) + \frac{w}{2} \\ y &= (y_{min}) + \frac{h}{2} \end{aligned} \quad (3)$$

- Normalization: Let the width normalization coefficient be  $dw$  and the height normalization coefficient be  $dh$ , as shown in Equation (4):

$$dw = \frac{1}{width} dh = \frac{1}{height} \quad (4)$$

- Data normalization requires multiplying the  $x$  coordinate of the center point of the target frame and the width  $w$  of the target frame by the width normalization coefficient  $dw$ , and multiplying the coordinate  $y$  of the center point of the target frame and the height  $h$  of the target frame by the height normalization coefficient  $dh$ , as shown in Equation (5):

$$\begin{aligned} \hat{x} &= x \bullet dw \\ \hat{y} &= y \bullet dh \\ \hat{w} &= w \bullet dw \\ \hat{h} &= h \bullet dh \end{aligned} \quad (5)$$

3. Data storage: The YOLO-type txt annotation file contains a row of data, followed by category information, the  $x$  coordinate of the center point of the target box, the  $y$  coordinate of the center point of the target box, the width  $w$  of the target box, and the height  $h$  of the target box.

### 3.3. Experimental Environment

This experiment was carried out using a Win11 operating system and NVIDIA GeForce GTX 105 graphics card. The model was built, trained, and validated using the Pytorch framework. The parameters of the YOLO model were initialized, and during the training process, the applied network model was designed with adaptive anchor frames, with initial anchor frame sizes set to [10, 13, 16, 23, 30, 33], [30, 45, 59, 61, 62, 119], and [90, 116, 156, 198, 326, 373]. To perform network training, the learning rate was set to 0.01, and the number of training epochs was set to 200. The test platform is shown in Table 4.

**Table 4.** Experimental platform.

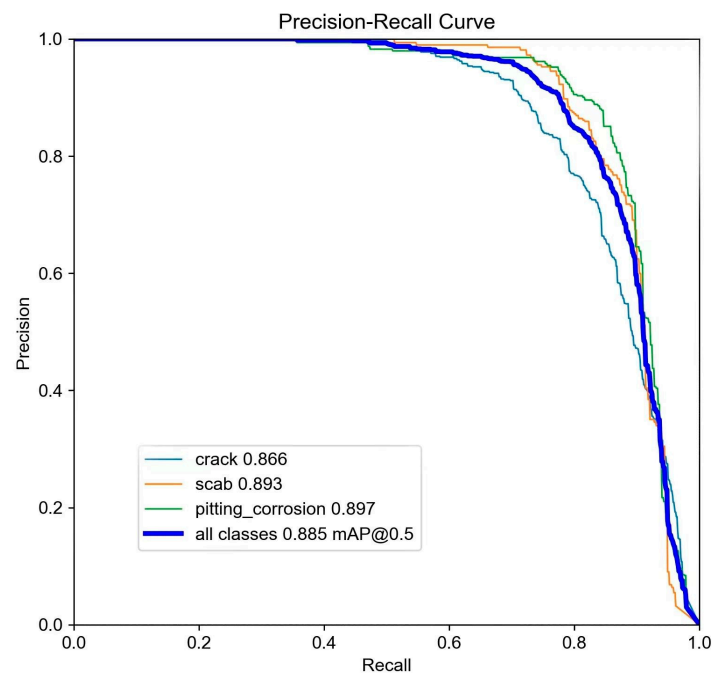
Designation	Configure
CPU	Intel Core i11400H 2.7 GHZ
Graphics Board	NVIDIA GeForce GTX 105
CUDA	CUDA 11.7
Operating System	W11 Operating System
Experimental Platform	Python 3.8, Pytorch 3.8

## 4. Improved Model Evaluation and Comparative Ablation Experiments

### 4.1. Model Performance Evaluation

To accurately evaluate the improved SC-YOLOv5 network model, the model evaluation indicators of this experiment included the average accuracy mean mAP@0.5, confusion matrix, PR curve, and loss fitting diagram to evaluate the improved model. The PR curve reflects the relationship between precision and recall. The relationship between precision

and recall is reciprocal. If the classifier only predicts the samples with high probability as positive samples, many positive samples with relatively low probability but still satisfied will be missed, resulting in a decrease in recall. The PR curve of the improved model reflects the average accuracy  $mAP@0.5$  of the three defects in the model evaluation standard, and the  $mAP@0.5$  values of cracks, pitting corrosion, and scarring are 0.866, 0.897, and 0.893. The average  $mAP@0.5$  value of the three defects reached 0.885, as shown in Figure 6; the model has relatively high accuracy and good performance.



**Figure 6.** Model PR curve.

The confusion matrix is a summary of the prediction results of the defect types. The count value is used to summarize the number of correct and incorrect predictions by classification and is broken down by defect type, showing which part of the classification model is confused when making predictions. From Figure 7 it can be seen that the actual value is close to the predicted value, and the positive sample ratios for cracks, pitting, and scars are 0.89, 0.91, and 0.92, respectively.

The  $mAP$  is one of the metrics used to evaluate the detection performance of the improved model, combining the performance of the precision and recall metrics, taking into account the performance at different confidence levels. Specifically,  $mAP$  is obtained by averaging the average precision (AP) over all defect categories, and AP is the size of the area under the precision–recall curve. The precision and recall are shown below, where  $TP$  denotes data predicted to be defective and actually defective,  $FP$  denotes data predicted to be defective but actually not defective,  $TN$  denotes data predicted not to be defective and actually not defective, and  $FN$  denotes data predicted not to be defective but actually defective.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

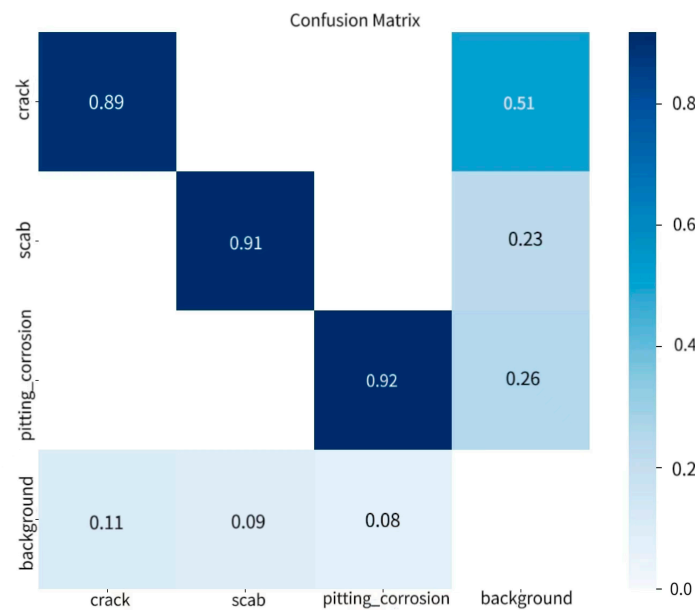


Figure 7. Model confusion matrix.

Precision indicates the accuracy of the model in predicting correctly; the higher the value, the less the model misclassifies as a positive case. Recall reflects the ability of the model to identify correct samples; the higher it is, the lower the risk of the model under-reporting (predicting a positive case as a false case). As shown in Table 5, the smaller the mean value of the loss fit plot, the more accurate the detection. Six metrics can be ideally fitted after 200 epochs of training, as expected.

Table 5. Multiple indicator fit charts.

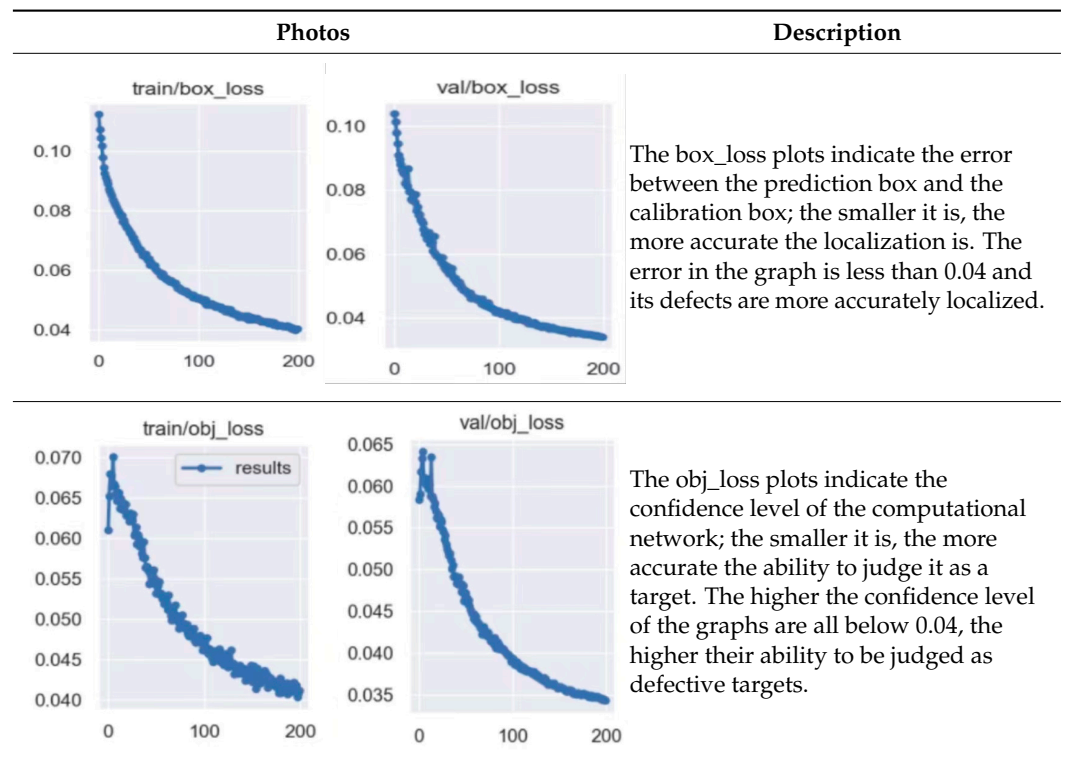
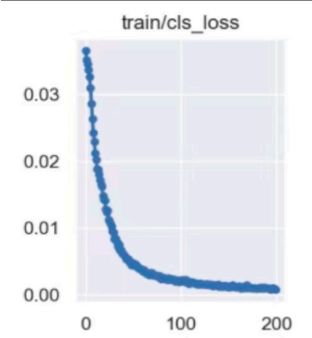
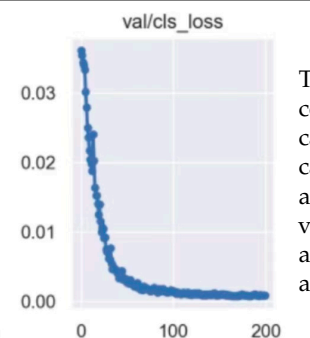
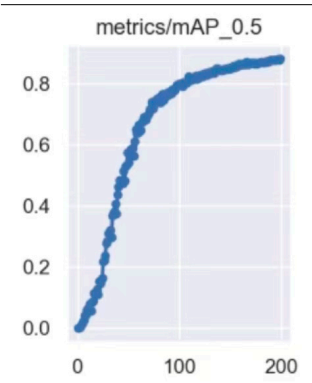
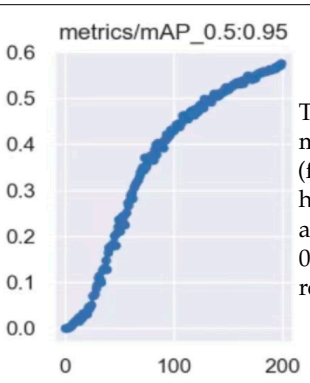
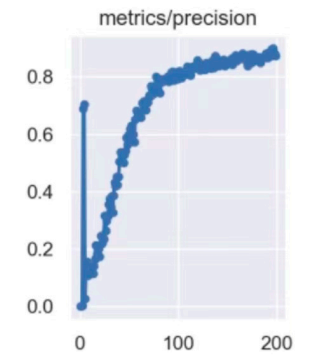
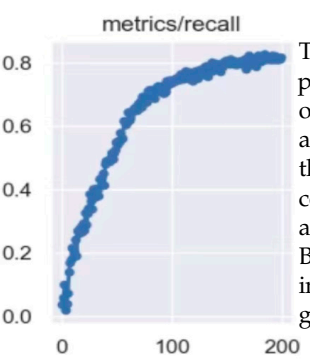


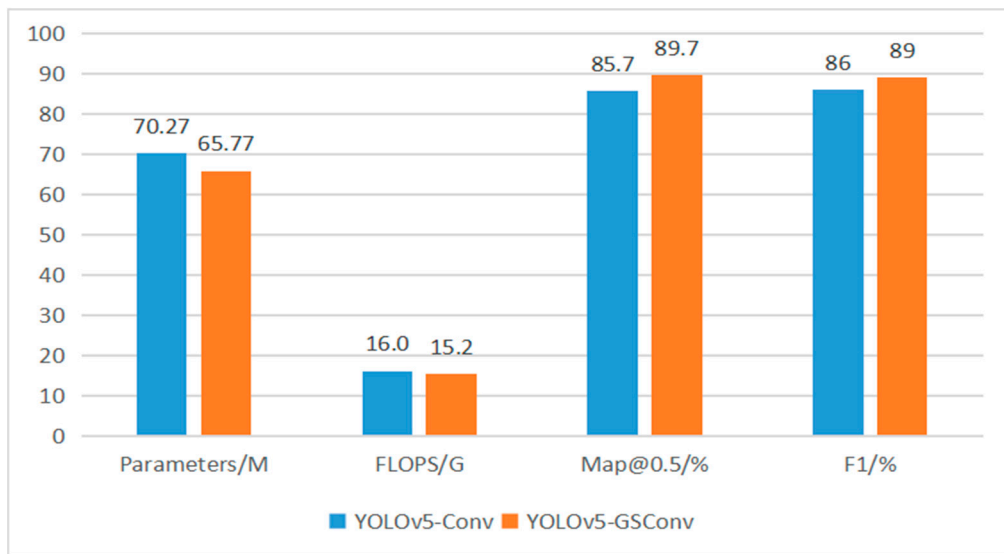
Table 5. Cont.

Photos		Description
 <p>train/cls_loss</p>	 <p>val/cls_loss</p>	<p>The cls_loss plots indicate whether the computational anchor box is correctly categorized with the corresponding calibration; the smaller it is, the more accurate the categorization is. The values in the graph are all below 0.01 and their defects are categorized more accurately.</p>
 <p>metrics/mAP_0.5</p>	 <p>metrics/mAP_0.5:0.95</p>	
 <p>metrics/precision</p>	 <p>metrics/recall</p>	<p>The mAP plots represent the average mAP over different IoU thresholds (from 0.5 to 0.95 in steps of 0.05), with higher mAP values being more accurate. In the graph mAP@0.5 Above 0.8, mAP@0.9 Above 0.6, its accuracy is relatively accurate.</p> <p>The precision plot represents the proportion of truly defective samples out of all samples that were predicted as defective. The recall plot represents the proportion of samples that are correctly predicted as defective out of all samples that are truly defective. Both the predicted value and the recall in the figure are higher than 0.8, which gives better precision results.</p>

## 4.2. Ablation Contrast Experiment

### 4.2.1. GSConv Improves the Performance of the Model

To prove that the lightweight convolution module GSConv has a better effect on the YOLOv5 network structure and to obtain a better performance of the feature network SC-YOLOv5, comparative experiments were carried out for this paper. First, the original YOLOv5 network structure was experimentally analyzed using the lightweight convolutional module GSConv and the ordinary convolutional module Conv. The experiments show that by using the lightweight convolution module GSConv in the original YOLOv5 network structure, the model parameters are reduced by 6.4%, the F1 value is increased by 0.03, and the map value is increased by 0.04, so the network lightweighting is improved and the accuracy is effectively improved, as shown in Figure 8.



**Figure 8.** GSconv lightweight module ablation experiment.

#### 4.2.2. Network Model Performance Comparison Experiment

In order to objectively evaluate the performance of the model proposed in this paper, the self-made metallurgical saw blade defect dataset was selected to test SC-YOLOv5 and compare it with SSD, YOLO v3-tiny, and YOLOv5. The comparison results are shown in Table 6.

**Table 6.** Comparison of detection performance of models.

Network Models	Model Size/MB	mAP@0.5/%	Parameters/MB
SSD	91.68	0.629	138.12
YOLOv3-tiny	16.6	0.883	86.74
YOLOv5	27.8	0.864	71.27
SC-YOLOv5	5.78	0.885	31.10

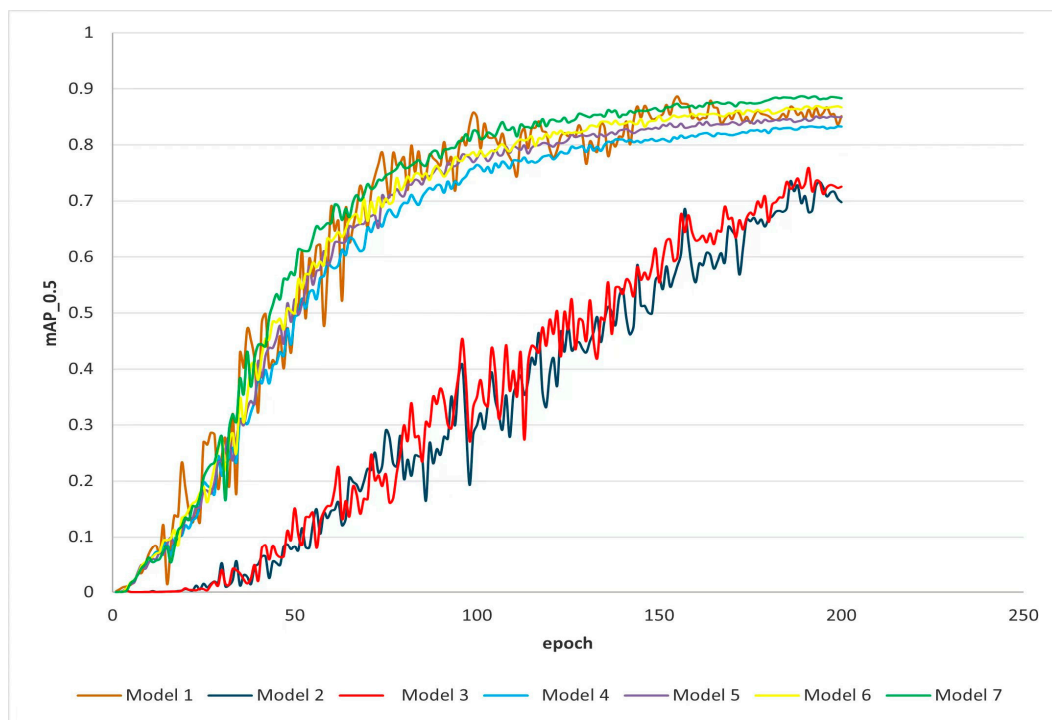
From Table 3, it can be seen that this study further compares SC-YOLOv5 with SSD, YOLOv3-tiny, and YOLOv5 models on the metallurgical saw blade dataset. It can be seen from Table 3 that the mAP@0.5 of the SC-YOLOv5 model is 88.5%, and the memory size of the model is 5.78 MB. Compared with the SSD, YOLOv3-tiny, and YOLOv5 models, the mAP@0.5 of the SC-YOLOv5 model is increased by 25.6, 0.2, and 2.1 percentage points, respectively. The memory size of the model is reduced by 93.7%, 65.1%, and 79.2%, respectively. It can be seen that compared with the SSD, YOLOv3-tiny, and YOLOv5 models, the SC-YOLOv5 model has the best overall performance on metallurgical saw blade defects, and the model complexity is the lowest, which is conducive to the deployment of an improved metallurgical saw blade defect detection model in low-power equipment.

#### 4.2.3. Improvement of SC Network Structure and GSConv on Model Performance

The SC network structure is used to replace the backbone network part in the original YOLOv5, SIoU is redefined as the loss function, and the light convolution module GSConv is replaced to improve the detection effect and further lighten the model parameters. For this paper, the SIoU loss function is first replaced by CIoU, and the SE, CBAM attention mechanism, and CA attention mechanism are added at the same position of the end layer of the YOLOv5 backbone network for ablation experiments. At the same time, the lightweight convolution module is replaced to complete the final improvement of the model, as shown in Table 7. In order to effectively evaluate the performance of the improved model SC-YOLOv5, other models were selected for comparison with SC-YOLOv5. The mAP training curves of the improved model and other models are shown in Figure 9.

**Table 7.** YOLOv5 model ablation contrast experiment.

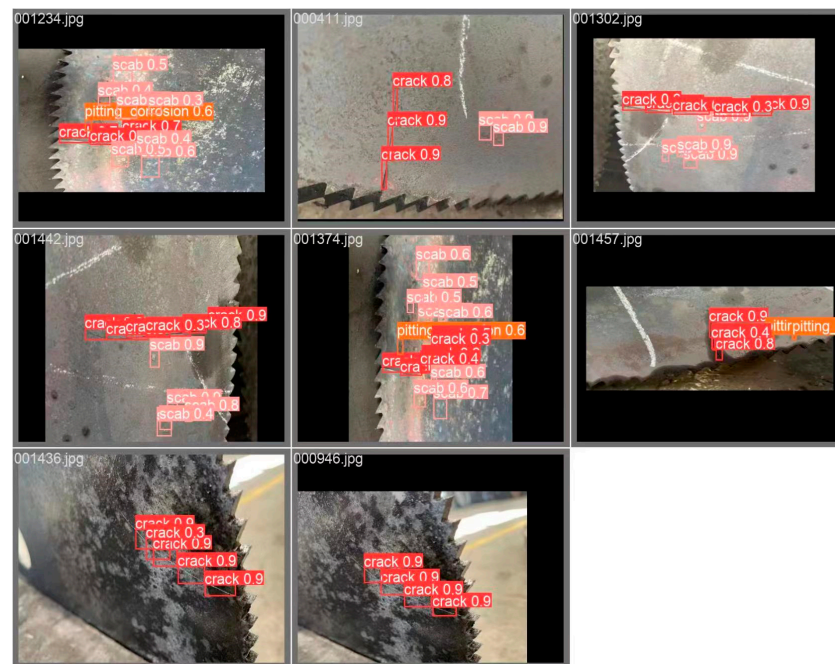
Model	ShuffleNetv2	SIOU	SE	CBAM	CA	GSCnv	Parameters/MB	Map@0.5/%	FLOPS/G
1	×	×	×	×	×	×	71.27	0.864	16.0
2	√	×	×	×	×	×	31.12	0.738	6.7
3	√	√	×	×	×	×	31.12	0.728	6.7
4	√	√	√	×	×	×	32.91	0.833	7.0
5	√	√	×	√	×	×	32.95	0.848	7.0
6	√	√	×	×	√	×	32.89	0.867	7.9
7	√	√	√	√	√	√	31.10	0.885	6.7

**Figure 9.** Variation curves of mAP for different models.

Experiments show that after the ShuffleNetV2 network structure replaces the original YOLOv5 backbone network structure, its model parameters are greatly reduced, and the lightweight degree of the network is greatly improved, as shown in Table 6. Compared with Model 6, the model parameters of SC-YOLOv5 (Model 7) in this paper are reduced by 2.5%, the map value is increased by 0.018, and the overall lightweight degree is improved. Compared with Model 5, the model parameters are reduced by 5.9%, and the map value is increased by 0.037. Compared with Model 4, the model parameters are reduced by 5.8%, and the mAP value is increased by 0.052. Compared with Model 1, the model parameters are reduced by 56.36%, and the lightweight degree of the network is greatly improved, which significantly improves the long-term transmission efficiency in industrial production, and the map value is slightly improved.

In Figure 9, compared with other models, after 200 rounds of full training, it is obvious that the improved Model 7 has a better convergence effect and better average accuracy.

In order to verify the effectiveness of the model improvement, a random mixed test diagram of multiple defects in the test set was selected, and SC-YOLOv5 was used for testing. The test results are shown in Figure 10. It can be seen from the figure that for metallurgical saw blades with various target defects, the improved model can effectively identify the types of defects and locate the target.



**Figure 10.** Randomized mixed test plot of multiple defects.

## 5. Conclusions

A metallurgical saw blade is a kind of consumable tool, and saw blade defect detection is particularly important. The surface defects of metallurgical saw blades are numerous and complex, and the small size leads to low accuracy and poor real-time performance. An improved YOLOv5 algorithm is proposed to overcome these problems. Verification experiments were carried out on the collected metallurgical saw blade dataset, and the conclusions of this investigation are as follows:

1. The ShuffleNetV2 structure and coordinate attention module were used to form a new network structure SC, which improves the network detection of small targets and multi-target detection ability, greatly reduces the number of model parameters, and improves the degree of network lightweighting.
2. The lightweight convolution module GSConv and loss function SIOU were used to improve YOLOv5, which ensures the accuracy and speed of SC-YOLOv5.
3. The improved YOLOv5 algorithm has low model complexity while ensuring accuracy. It can quickly and accurately detect small-size defect images on the surface of metallurgical saw blades, and it meets the requirements of lightweight model deployment for mobile devices.
4. This paper focuses on the detection of metallurgical saw blade surface defects. In the future, we will continue to optimize the YOLO algorithm, further improve the detection accuracy, and reduce the complexity of the model. In addition, a variety of defect categories for metallurgical saw blades will be added to meet the needs of the industry.

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

**Funding:** This study was supported by the Natural Science Foundation of Hebei Province (E2019209314) and the Natural Science Foundation of Hebei Province (E2022209086).

**Data Availability Statement:** The data presented in this study are available on demand from the first author (at mengll@ncst.edu.cn).

**Acknowledgments:** The authors would like to express their sincere gratitude to their schools and colleges and for the funding of the project. We sincerely appreciate all the support and help. In addition, we sincerely thank the editors and reviewers of this paper for their work.

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

## References

1. Feng, L.; Tang, D.; Chen, M. Development of a measuring system for the diameter of a smooth plug gauge based on machine vision technology. *Meas. Control* **2023**, *56*, 966–974. [[CrossRef](#)]
2. Sarker, A.; Grift, T. Monitoring Postharvest Color Changes and Damage Progression of Cucumbers Using Machine Vision. *J. Food Res.* **2023**, *12*, 37–50. [[CrossRef](#)]
3. Pavlopoulos, G.A.; Hooper, S.D.; Sifrim, A.; Schneider, R.; Aerts, J. Medusa: A tool for exploring and clustering biological networks. *BMC Res. Notes* **2011**, *4*, 384. [[CrossRef](#)]
4. Benfenati, A.; Causin, P.; Oberti, R.; Stefanello, G. Unsupervised deep learning techniques for automatic detection of plant diseases: Reducing the need of manual labelling of plant images. *J. Math. Ind.* **2023**, *13*, 5. [[CrossRef](#)]
5. Li, J.; Li, S.; Li, X.; Miao, S.; Dong, C.; Gao, C.; Liu, X.; Hao, D.; Xu, W.; Huang, M. Primary bone tumor detection and classification in full-field bone radiographs via YOLO deep learning model. *Eur. Radiol.* **2023**, *33*, 4237–4248. [[CrossRef](#)]
6. Wang, L. Retraction Note: Application of deep learning to detect defects on the surface of steel balls in an IoT environment. *J. Supercomput.* **2023**, *79*, 13907. [[CrossRef](#)]
7. Hou, X.; Liu, M.; Zhang, S.; Wei, P.; Chen, B. CANet: Contextual Information and Spatial Attention Based Network for Detecting Small Defects in Manufacturing Industry. *Pattern Recognit.* **2023**, *140*, 109558. [[CrossRef](#)]
8. Abagiu, M.M.; Cojocaru, D.; Manta, F.; Mariniuc, A. Detecting Machining Defects inside Engine Piston Chamber with Computer Vision and Machine Learning. *Sensors* **2023**, *23*, 785. [[CrossRef](#)]
9. Liu, Z.; Wang, M.; Li, C.; Ding, S.; Li, B. A dual-branch balance saliency model based on discriminative feature for fabric defect detection. *Int. J. Cloth. Sci. Technol.* **2022**, *34*, 451–466. [[CrossRef](#)]
10. Joung, O.J.; Kim, Y.H. Application of an IR thermographic device for the detection of a simulated defect in a pipe. *Sensors* **2006**, *6*, 1199–1208. [[CrossRef](#)]
11. Zhang, L.; Jing, J.; Zhang, H. Fabric defect classification based on LBP and GLCM. *J. Fiber Bioeng. Inform.* **2015**, *8*, 81–89. [[CrossRef](#)]
12. Sun, X.; Wu, P. Face detection using deep learning: An improved faster RCNN approach. *Neuro Comput.* **2018**, *299*, 42–50. [[CrossRef](#)]
13. Yin, S.; Li, H.; Teng, L. Airport detection based on improved faster RCNN in large scale remote sensing images. *Sens. Imaging* **2020**, *21*, 49. [[CrossRef](#)]
14. Yin, H.; Xu, J.; Wang, Y.; Hu, D.; Yi, W. A Novel Method of Situ Measurement Algorithm for Oudemansiella raphaniplies Caps Based on YOLO v4 and Distance Filtering. *Agronomy* **2022**, *13*, 134. [[CrossRef](#)]
15. Li, Z.; Zhang, J.; Wen, Y.; Dong, Y.; Xu, W. A multiscale fusion YOLOv3-based model for human abnormal behavior detection in special scenarios. *J. Transp. Eng.* **2023**, *149*, 04022150. [[CrossRef](#)]
16. Wang, Q.; Zhang, Q.; Liang, X.; Wang, Y.; Zhou, C. Traffic lights detection and recognition method based on the improved YOLOv4 algorithm. *Sensors* **2021**, *22*, 200. [[CrossRef](#)] [[PubMed](#)]
17. Xian, Y.; Liu, G.; Fan, J.; Yu, Y.; Wang, Z. YOT-Net: YOLOv3 combined triplet loss network for copper elbow surface defect detection. *Sensors* **2021**, *21*, 7260. [[CrossRef](#)] [[PubMed](#)]
18. Wang, X.; Lv, F.; Li, L.; Yi, Z.; Jiang, Q. A novel optimized tiny YOLOv3 algorithm for the identification of objects in the lawn environment. *Sci. Rep.* **2022**, *12*, 15124. [[CrossRef](#)]
19. Xue, Z.; Lin, H.; Wang, F. A small target forest fire detection model based on YOLOv5 improvement. *Forests* **2022**, *13*, 1332. [[CrossRef](#)]
20. Li, H.; Zeng, X.; Li, Y.; Zhou, S.; Wang, J. Convolutional neural networks based indoor Wi-Fi localization with a novel kind of CSI images. *China Commun.* **2019**, *16*, 250–260. [[CrossRef](#)]
21. Zha, M.; Qian, W.; Yi, W.; Hua, J. A lightweight YOLOv4-Based forestry pest detection method using coordinate attention and feature fusion. *Entropy* **2021**, *23*, 1587. [[CrossRef](#)]
22. Zhang, Y.; Zhou, A.; Zhao, F.; Wu, H. A lightweight vehicle-pedestrian detection algorithm based on attention mechanism in traffic scenarios. *Sensors* **2022**, *22*, 8480. [[CrossRef](#)]
23. Li, Y.; Han, B.; Zeng, S.; Xu, S.; Yuan, C. End-to-End Point Cloud Completion Network with Attention Mechanism. *Sensors* **2022**, *22*, 6439. [[CrossRef](#)] [[PubMed](#)]
24. Du, S.; Zhang, B.; Zhang, P. Scale-sensitive IOU loss: An improved regression loss function in remote sensing object detection. *IEEE Access* **2021**, *9*, 141258–141272. [[CrossRef](#)]
25. Han, G.; Li, T.; Li, Q.; Zhao, F.; Zhang, M.; Wang, R.; Yuan, Q.; Liu, K.; Qin, L. Improved algorithm for insulator and its defect detection based on YOLOX. *Sensors* **2022**, *22*, 6186. [[CrossRef](#)]



26. Wang, L.; Zheng, H.; Yin, C.; Wang, Y.; Bai, Z.; Fu, W. Dense Papaya Target Detection in Natural Environment Based on Improved YOLOv5s. *Agronomy* **2023**, *13*, 2019. [[CrossRef](#)]
27. Kirik, M.; Hamamreh, J.M. A novel interference signal superposition algorithm for providing secrecy to subcarrier number modulation-based orthogonal frequency division multiplexing systems. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, 4678. [[CrossRef](#)]
28. Bai, Y.; Yan, Z.; Li, C. An attention mechanism-based deep regression approach with a sequence decomposition-granularity reconstruction-integration model for urban daily water supply forecasting. *J. Hydrol.* **2023**, *617*, 129032. [[CrossRef](#)]
29. Xu, B.; Cui, X.; Ji, W.; Yuan, H.; Wang, J. Apple grading method design and implementation for automatic grader based on improved YOLOv5. *Agriculture* **2023**, *13*, 124. [[CrossRef](#)]
30. Li, L.; Fang, B.; Zhu, J. Performance Analysis of the YOLOv4 Algorithm for Pavement Damage Image Detection with Different Embedding Positions of CBAM Modules. *Appl. Sci.* **2022**, *12*, 10180. [[CrossRef](#)]
31. He, X.; Cheng, R.; Zheng, Z.; Wang, Z. Small object detection in traffic scenes based on YOLO-MXANet. *Sensors* **2021**, *21*, 7422. [[CrossRef](#)] [[PubMed](#)]
32. Luo, Q.; Jiang, W.; Su, J.; Ai, J.; Yang, C. Smoothing Complete Feature Pyramid Networks for Roll Mark Detection of Steel Strips. *Sensors* **2021**, *21*, 7264. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.