WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 4, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v4n2.17
Journal homepage: https://wepub.org/index.php/IJCSIT/index

An Improved YOLOv8n Algorithm for Steel Surface Defect Detection

Dongmei Ma *, Na Zhang, Xuelong Lv

School of Physics and Electronic Engineering, Northwest Normal University, Lanzhou, China
*Corresponding Author: Dongmei Ma

ABSTRACT

Due to the manufacturing process and other aspects, steel surface defects vary in shape and size, which is a difficult point for defect detection. For this reason, this paper proposes a steel surface defect detection algorithm based on YOLOv8n. The algorithm improves the C2f module of Backbone in this network by adding Context Guided Block (CG block), which can effectively extract the local features, surrounding context and global context, and integrate this information to improve the precision and accuracy of detection. Meanwhile, we also change the detection header of YOLOv8 to Detect_ASFF by introducing the Adaptive Spatial Feature Fusion (ASFF) technique. This improvement effectively filters out the conflicting information, which can allow the model to have a better adaptability at different scales. In this paper, Shape-IoU is adopted as the bounding box loss function to make up for the traditional IoU loss that ignores the intrinsic properties of the bounding box in the calculation process. After optimisation, mAP@0.5 are improved to 67.9% and mAP@0.5~0.95 are improved to 35.1%, which is 2.8% and 1.1% respectively. The improved algorithm in this paper has improved in detection accuracy, which proves the practicality and effectiveness of the algorithm in this paper.

KEYWORDS

YOLOv8n, Context Guided Block, ASFF, Shape-IoU

1. INTRODUCTION

Steel is present in every aspect of our daily lives and is used in a wide range of applications such as building support structures, durable components in automotive manufacturing, appliance housings and internal components, pipelines for the transmission of liquids or gases, and critical parts of machinery. Maintaining the integrity and quality of steel surfaces is critical to manufacturing. However, during the manufacturing and processing of steel, lapses in manufacturing processes and production can lead to a variety of product defects. These defects are of various types, including punched holes (Pu), welded seams (Wl), oil spots (Os), silk spots (Ss), rolled pits (Rp), creases (Cr), etc. All of them impair the corrosion and wear resistance of steel, so the detection of defects on steel surfaces is particularly critical.

In recent years, target detection based on convolutional neural network has been rapidly developed in various fields, especially for the detection of material surface defects, which has become one of the research hotspots in the industrial field [1]. According to the algorithm flow, target detection algorithms are mainly divided into two categories: one is the two-stage algorithm (Two-Stage) represented by R-CNN [2], Fast R-CNN [3] and Faster R-CNN [4], this algorithm divides the task of target detection into two steps of candidate region generation and target classification and localisation, and outputs the class probability of each candidate region and its precise The other is the One-Stage

algorithm represented by SSD [5], RetinaNet [6] and YOLO (You Only Look Once) series [7-14], which directly performs the classification and bounding box adjustment without generating multiple candidate regions.The YOLO series has been improved in several versions to enhance the speed and accuracy to cope with different application scenarios. accuracy to cope with different application scenarios and challenges.

In summary, this paper proposes a steel surface defect detection algorithm based on YOLOv8n, with the following specific contributions:

(1) Since defects make up a relatively small portion of the steel surface image and may be more difficult to identify under complex lighting and background conditions. In this paper, the C2f module in the feature extraction network is improved by introducing the Context Guided Block [15] module, which is capable of fusing the captured information to generate richer feature representations without significantly increasing the computational effort, thereby improving the performance of target detection.

(2) Due to the variability of defects on steel surfaces, it is often difficult to capture the nuances of all types of defects.ASFF [16] is able to adaptively fuse features at different scales, enabling the Head part of YOLOv8 to more comprehensively utilise multi-scale feature information, thus improving the accuracy and robustness of detection.

(3) The traditional loss function fails to take into account the directional mismatch between the real frame and the predicted frame, resulting in slow and inefficient model convergence. To solve this problem, this paper adopts the Shape-IoU [17] loss function as the boundary loss function.This loss function evaluates not only the size of the overlapping region, but also considers the shape (e.g., aspect ratio and orientation) of the bounding box and its scale (e.g., size) by introducing shape and scale factors.In this way, the model can more accurately measure the similarity between the predicted frame and the real frame, so as to comprehensively learn the geometric properties of the target and improve the accuracy of bounding box regression.

2. YOLOV8N ALGORITHM FRAMEWORK

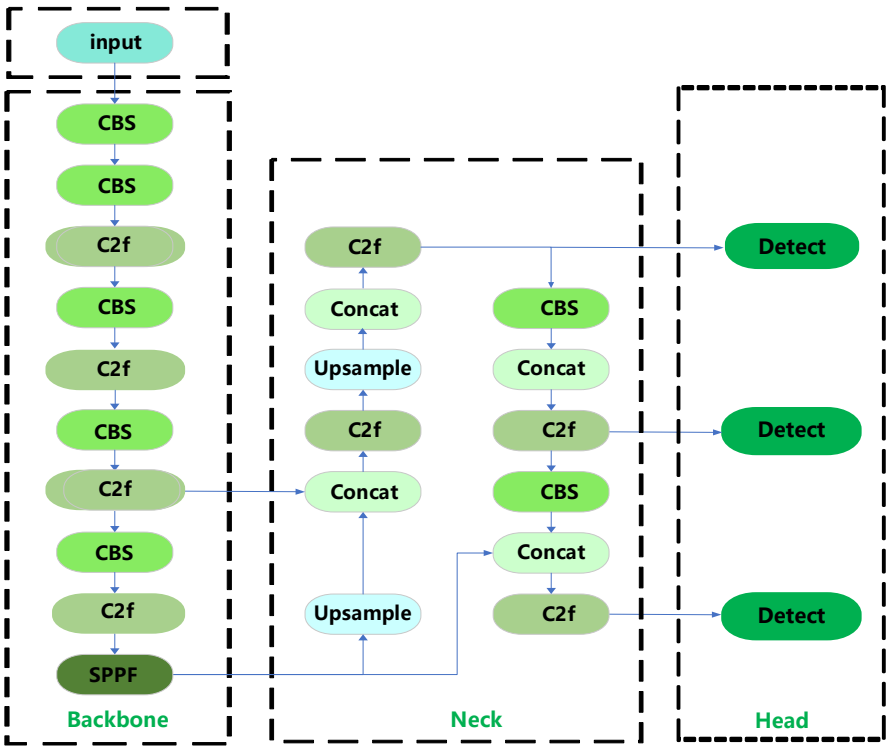


Figure 1. YOLOv8n network structure

YOLOv8 is optimised for model performance and flexibility based on the predecessor models in the YOLO series, providing higher accuracy and faster inference in target detection tasks, and is able to handle complex scenes effectively. Its network structure consists of three main components: Backbone is responsible for feature extraction, using C2f module and other improved techniques to increase efficiency and convergence speed; Neck focuses on multi-scale feature fusion, retaining the concept of path aggregation network and introducing SPPF to speed up the processing; and Head uses an uncoupled header design, transforming into an Anchor-Free approach to reduce the the conflict between classification and regression, thus improving the model performance. Overall, YOLOv8 has been optimised in several aspects to provide the advantages of high performance, flexibility and ease of use. The overall architecture of the network is shown in Fig.1.

3. ALGORITHM OF THIS PAPER

3.1. Model Structure

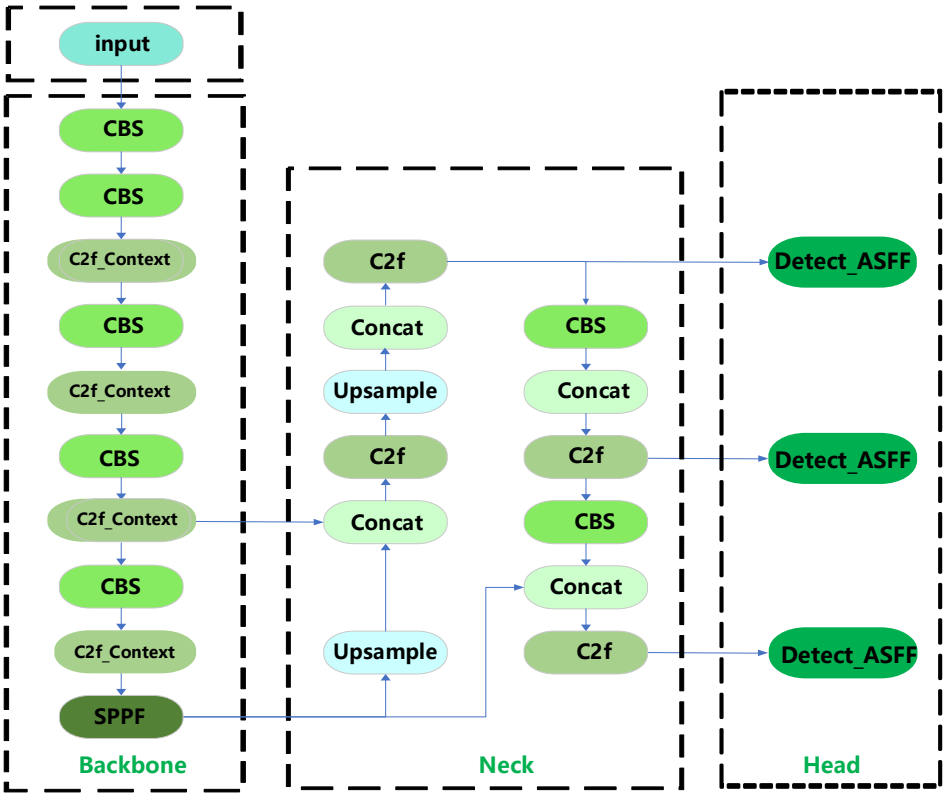


Figure 2. Structure of model in this paper

In this paper, the improved structure is shown in Fig. 2. The C2f module in Backbone is modified based on YOLOv8n to introduce the Context Guided Block module, which is able to understand the overall situation of the environment in which the target is located when analysing or processing information in order to identify and classify the target more accurately. This information includes not only the characteristics of the target itself (e.g. shape, colour and texture), but also surrounding objects, background, lighting conditions and other potential distractions. It is more effective for dealing with targets of different sizes and shapes, and can significantly improve detection accuracy. Moreover, CG block enhances the ability of feature representation. This enables the model to identify targets in the image more accurately and reduces false and missed detections. In addition, the detection header of YOLOv8 is changed to Detect_ASFF using ASFF, which mainly introduces an adaptive spatial feature fusion that effectively filters out conflicting information, thus enhancing the scale invariance. Shape-IoU is used as the bounding box loss function of the model, which is a new loss method for border regression, which makes up for the shortcomings of the traditional loss function that ignores

the intrinsic attributes of the border during the calculation process by focusing on the shape and scale of the target border, and enhances the scale invariance of the features by fusing the features with different scales in an adaptive way and filtering out the conflicting information effectively.

3.2. Context Guided Block (CG block)

Context Guided Block (CG block) mimics the contextual understanding ability of the human visual system, which can learn the joint features of local features, surrounding environment context, and then introduce global contextual features in order to dramatically improve the performance of the computer vision model in complex scenes, making it more accurate and effective in tasks such as target recognition, object detection and scene understanding. This module contains the following parts:

(1) Local feature extractor (floc(*)): Local features are learnt using standard convolutional layers. Features are extracted from local regions of the image using standard 3×3 convolutional layers. These extracted local features are then combined with surrounding context features to improve the model's sensitivity to local features and enhance the network's overall understanding of each region.

(2) Surrounding Context Extractor (fsur(*)): the surrounding context extractor uses extended convolution (e.g., null convolution) to increase the size of the receptive field to capture a wider range of contextual information. When recognising an object, in addition to the features of the object itself, its surroundings provide important clues. fsur(*), on the other hand, looks at features over a larger area, not just local details, and can enhance the model's overall understanding of the object.

(3) Joint feature extractor (fjoi(*)): enables the effective fusion of information from different feature extractors (e.g., local and contextual feature extractors) to form richer feature representations through connectivity layers and batch normalisation (BN) and parametric ReLU (PReLU) operations.

(4) Global Context Extractor (fglo(*): The Global Context Extractor aggregates the information extracted from the joint features through Global Average Pooling (GAP) to obtain global information of the whole feature graph. This information is then further processed using two Fully Connected Layers , which helps the network to understand the global information of the entire image.

The structure of Context Guided Block is shown in Fig. 3, firstly the feature map is input to floc(*) and fsur(*) extractors respectively by over 1x1 convolution; floc(*) extracts local features using 3x3 ordinary convolution and fsur(*) extracts surrounding context features using 3x3 dilation convolution; fjoi(*) converts floc(*) and fsur(*) outputs into a richer feature representation by performing a Concat operation followed by Batch Normalization (BN) and Parametric ReLU (PReLU); fglo (*) extracts global contextual features, and performs Global Average Pooling (GAP) and Multi-Layer Perceptron on the inputs, and multiplies the resulting weights and inputs by element by element multiplication. These components work together to improve the network's ability to understand features at various scales in complex scenes.

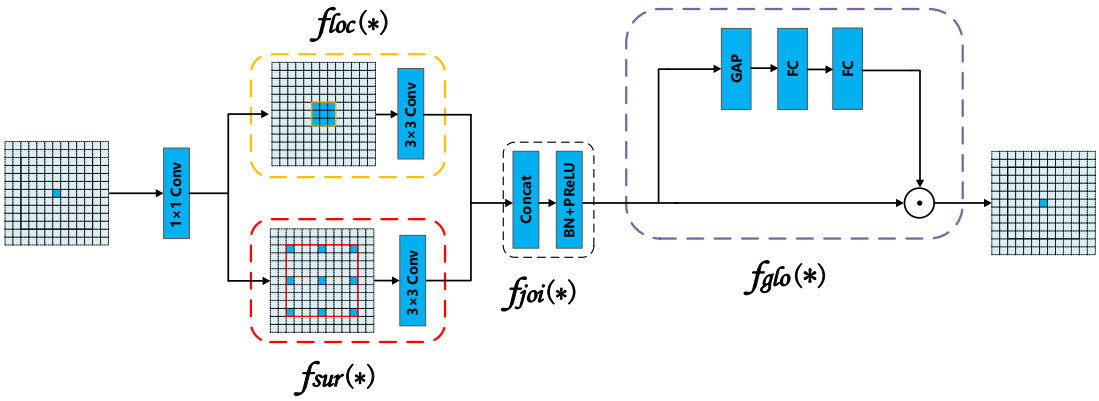


Figure 3. Context Guided Block Structure

Two types of Residual Connection are used in CG Block. It helps the model to learn highly complex features and improves gradient backpropagation during training. One is connecting the CG Block input and the output of fjoi(*) is called Local Residual Learning (LRL), the structure is shown in Fig. 4; the other is connecting the CG Block input and the output of fglo(*) is called Global Residual Learning (GRL), the structure is shown in Fig. 5.

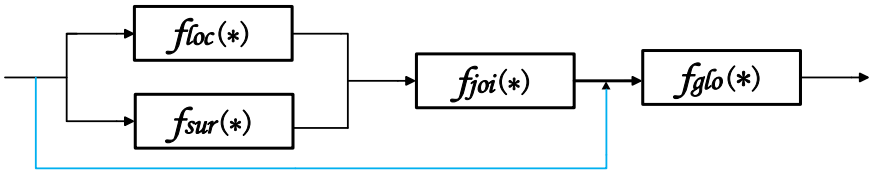


Figure 4. Local Residual Learning (LRL)

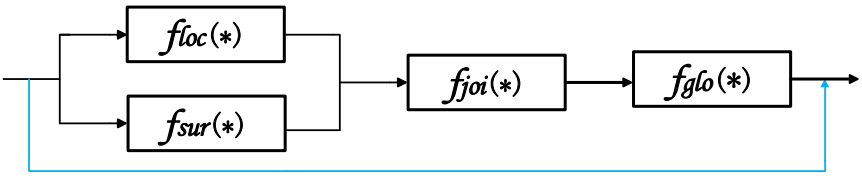


Figure 5. Global Residual Learning (LRL)

3.3. ASFF

ASFF is a feature fusion method that helps models capture defective targets at different scales, giving them better detection accuracy. The ASFF method solves the problem of inconsistency in the feature pyramid by learning the importance of features at different levels at each spatial location, and adaptively filtering out features carrying contradictory information. This innovative scheme has achieved significant results in single detectors, allowing the network to learn directly how to spatially filter at other levels, thus retaining only useful information for combination.

The Adaptive Spatial Feature Fusion (ASFF) mechanism is shown in Fig. 6, which is used for single object detection. In this structure, the features of each level (represented by different colours) are first down-sampled or up-sampled by their respective steps to ensure that all the features have the same spatial dimensions. This process ensures that features at different levels are fused at the same scale, thus enhancing the inter-matching and coordination between features. Level-1, Level-2 and Level-3 refer to features at different levels of the feature pyramid, each with a different spatial resolution.ASFF-1, ASFF-2 and ASFF-3 denote the application of the ASFF mechanism to the feature fusion at different levels. In the amplified part of ASFF-3, the features of the other layers are resized to the same size as the third layer, and then they are weighted and fused by the learned weight maps to generate the fused features that are finally used for prediction. In this way, ASFF is able to adaptively select the most useful features at each spatial location to improve detection accuracy. This approach allows the model to flexibly decide which feature layers are most important for the final prediction based on the context of each particular location and scale.

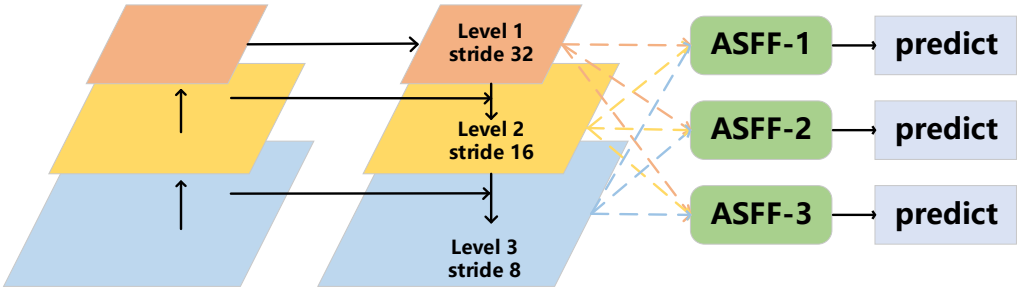


Figure 6. ASFF Algorithm Model

3.4. Shape-IoU

The loss function is used to calculate the error between the predicted value and the target value, and the smaller the result value, the better the model is. The loss function provides data to the neural network in the reverse direction so that the model can be optimised in time to make the predicted values as close as possible to the actual values. In the task of target detection, the bounding box is a common method used to represent the location of the target. Bounding box regression loss is an important part of the localisation branch of the target detection detector, and most of the computational methods focus on the geometric relationship between the Ground Truth Box (GT) box and the prediction box, and calculate the loss by using the relative positions and shapes of the bounding boxes, without taking into account the effect of the inherent properties of the bounding boxes such as their own shapes and scales on the regression of the bounding boxes. The design principle of the Shape IoU loss function is to calculate the loss by focusing on the shape and scale of the bounding box. The Shape_IoU method can calculate the loss by focusing on the shape and scale of the bounding box, which makes the regression of the bounding box more accurate [18]. In target detection, the bounding boxes of different targets may have different shapes and scales. The traditional IoU loss function only considers the degree of overlap between the bounding boxes when calculating the loss, but does not take into account the differences in the shape and scale of the bounding boxes. The Shape IoU loss function, on the other hand, introduces the concepts of shape and scale, which makes it possible to more comprehensively take into account the characteristics of the bounding boxes when calculating the loss.

As shown in Fig. 7 for the Shape-IoU loss calculation, this loss function goes through the area of overlap between the real and predicted boxes (Area of Overlap) and the total area of the merged real and predicted boxes (Area of Union), respectively, when calculating the loss. Next, the ratio of the two (i.e., the IoU value) is used to obtain a value ranging between 0 and 1, which can indicate the degree of overlap between the two bounding boxes. Finally, the IoU value is converted into a loss value by some means, which is used to guide the training and optimisation of the model.

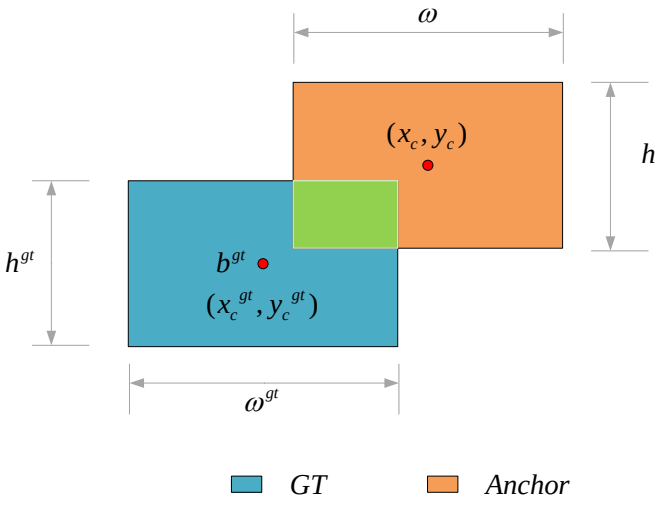


Figure 7. Shape-IoU loss calculation

The formula is as follows:

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}} \tag{1}$$

$$\omega\omega = \frac{2 \times \left(\omega^{gt}\right)^{scale}}{\left(\omega^{gt}\right)^{scale} + \left(h^{gt}\right)^{scale}} \tag{2}$$

$$hh = \frac{2\times(h^{gt})^{scale}}{(\omega^{gt})^{scale}+(h^{gt})^{scale}} \tag{3}$$

$$distance^{shape} = hh \times (x_c - {x_c}^{gt})^2/c^2 + \omega\omega \times (y_c - {y_c}^{gt})^2/c^2 \tag{4}$$

$$\Omega^{shape} = \sum_{t=\omega,h}(1 - e^{-\omega t})^{\theta}.\theta = 4 \tag{5}$$

$$\begin{cases} \omega_{\omega} = hh \times \frac{|\omega-\omega^{gt}|}{|\max(\omega,\omega^{gt})|} \\ \omega_h = \omega\omega \times \frac{|h-h^{gt}|}{|\max(h,h^{gt})|} \end{cases} \tag{6}$$

Where scale is the scaling factor, which is related to the size of the target in the dataset, and ww and hh denote the weight coefficients in the horizontal and vertical directions, respectively, and their values are related to the shape of the GT box. The corresponding bounding box regression loss is as follows:

$$L_{Shape-IoU} = 1 - IoU + distance^{shape} + 0.5 \times \Omega^{shape} \tag{7}$$

In this paper, we refer to ShapeIoU, a loss function for measuring the overlap of bounding boxes in target detection, which aims to improve the traditional IoU (Intersection over Union) metrics, especially when the box geometries and scales vary widely. The core idea of ShapeIoU is to combine the information about the overlapping regions of target boxes and the distance between centroids to provide a more comprehensive matching metric. The core idea of ShapeIoU is to take into account the overlapping areas of the target boxes, shape information, and the distance between the centroids to provide a more comprehensive match metric.

4. EXPERIMENTAL RESULTS AND ANALYSES

4.1. Experimental Environment and Parameter Settings

The experimental parameter settings are basically adopted from the official recommendations of YOLOv8, the network is trained using the SGD optimiser, the initial learning rate is 0.01, the learning rate momentum is set to 0.937, the weight decay is set to 0.0005, the batch is 64, the input image size is 640 pixel*640 pixel, and the epoch is 300.The experimental hardware configurations of this paper are shown in Table 1.

Table 1. The experiment hardware configuration of this paper

Project	Content
CPU	AMD Ryzen 5 7500F 6-Core Processor
GPU	NVIDIA RTX 4070 SUPER
Video Memory	12GB
Running Memory	32GB
Operating System	Windows
Deep Learning Framework	Torch2.4.1+cu12.1
Data Processing	Python3.8

4.2. Datasets

The dataset used in this paper is GC10-DET, which is a dataset of surface defects collected in a real industry. It contains ten types of surface defects, i.e., Punch (Pu), Weld (Wl), Crescent Gap (Cg), Water Spot. Oil Spot (Os), Silk Spot (Ss), Inclusions (In), Rolled Pits (Rp), Creases (Cr), Waist Folds

(Wf). The collected defects are on the surface of the steel plate. The dataset consists of 3570 grey scale images. However, due to real industrial photography, there are more discarded images, after sorting and screening, the number of images in the dataset used in this paper is 2294.The defects in each category in the dataset are different in type, size, number and distribution. In the defect detection task of this paper, 80% of the images are randomly selected for model training and the rest are used for testing. This data distribution method aims to ensure that the model can be trained on a wide range of samples, while leaving enough test images to verify the model's generalisation ability and effectiveness.

4.3. Evaluation Metrics

The experiment uses five evaluation metrics: mean average precision (mAP), Precision, recall, Parameters and GFLOPs, which are introduced as follows:

mean Average Precision (mAP): It indicates the comprehensive weighted average of the average precision of different categories of targets during detection, and is a commonly used performance evaluation index in target detection, calculated as (8):

$$mAP = \frac{1}{c}\sum_{i=1}^{c} A\,P_i \tag{8}$$

Where AP_i is the average precision of the ith category, Average Precision (AP) represents the average of the detector's precision in each case of recall, which corresponds to the area under the Precision-Recall Curve (AUC). mAP averages the APs from the dimensions of the categories, and for each category, first calculate the the area under the Precision-Recall Curve (AUC) and then averaged over all categories. where c is the number of categories.

Precision: indicates the proportion of true positive category samples among all samples predicted as categories by the model, and is an important metric used to measure the accuracy of model prediction. It is calculated as (9).

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Where TP is the number of positive samples that are correctly detected, and FP is the number of negative samples that are detected as positive samples (positive samples refer to targets to be detected, and negative samples refer to targets that do not need to be detected).

Recall: Indicates the proportion of true positive category samples that have been successfully predicted as positive by the model, and is an indicator of the degree of coverage of positive examples by the model. The formula is (10):

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

Where FN is the number of false negative examples (negative samples are detected as positive samples).

Parameters. indicates the number of parameters in the model, including weights and biases. More model parameters may result in a more complex model, but it is also easier to overfit.

GFLOPs: Used to measure the computational speed of the model, it indicates the number of billions of floating-point operations performed by the model in a second, GFLOPs are related to the computational complexity and computational efficiency of the model.

4.4. Training Results

In the training results of the model in this paper, a series of result plots, such as confusion matrix, P-R curve, etc., are generated to initially judge the performance of the model. The visualisation image of the confusion matrix of the detection model in this paper is shown in Fig. 8, which shows the relationship between the prediction results of the model and the actual labels in the form of a matrix, which shows which part of the prediction will be confused when making a prediction.

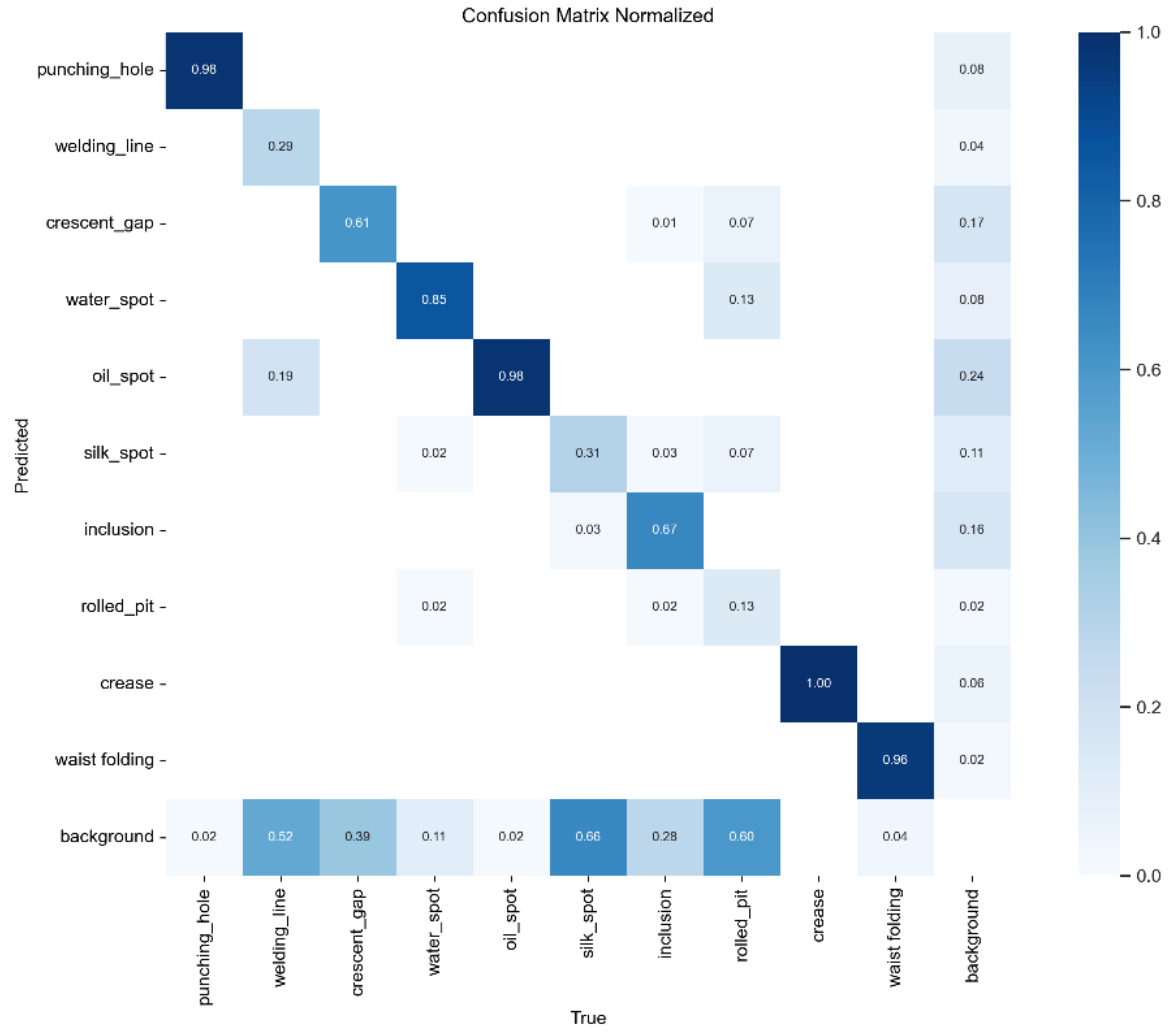
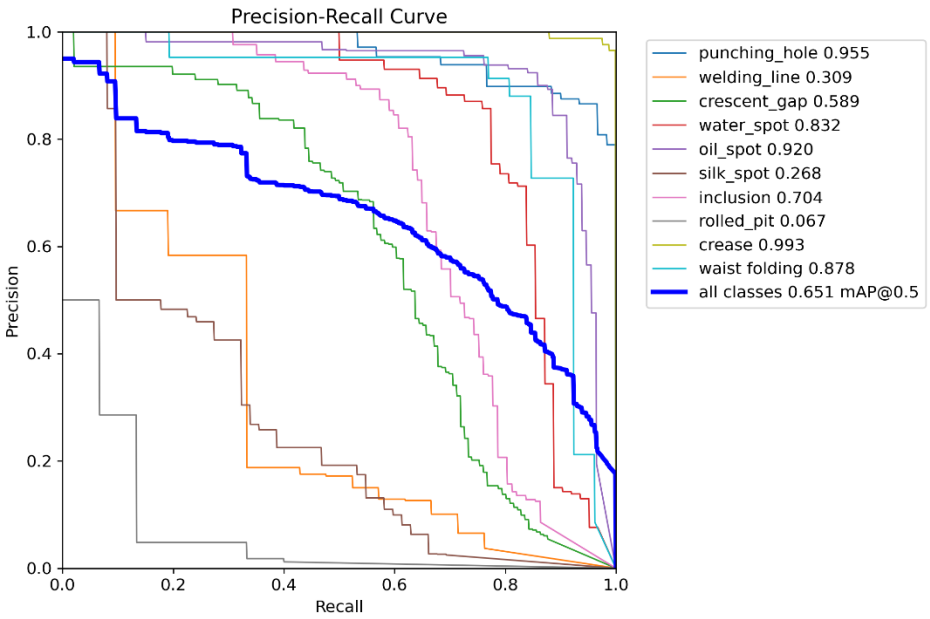


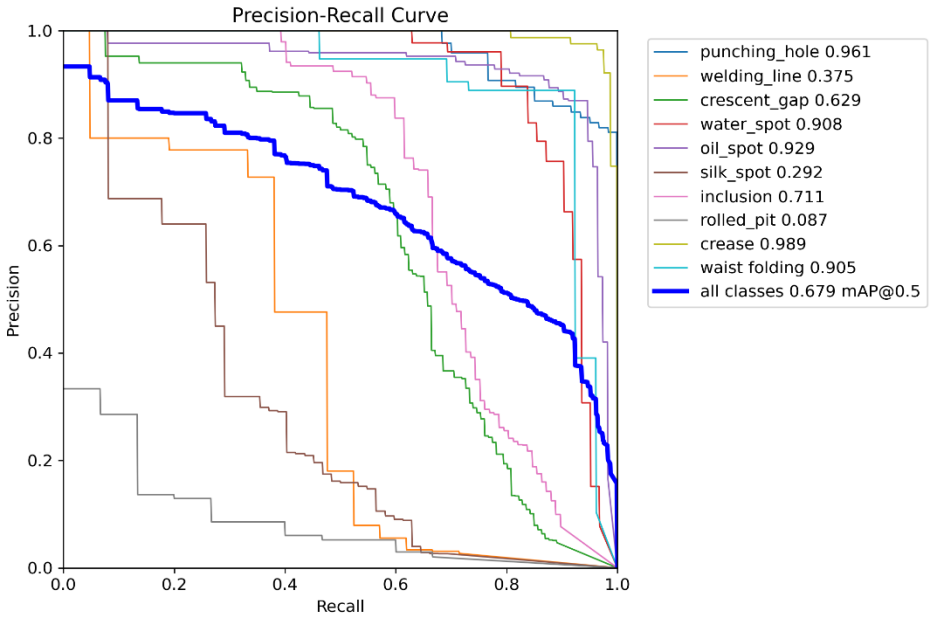
Figure 8. Confusion matrix visualization image

The P-R curve (precision-recall curve) represents the relationship curve between precision and recall under different thresholds. Where the vertical axis represents precision and the horizontal axis represents recall, the area enclosed under the P-R curve is the AP, and the average value of AP across all categories is mAP.

Fig .9(a) shows the P-R curve under the training of YOLOv8n model, and Fig. 9(b) shows the P-R curve under the training of this model. Each point corresponds to a threshold value, representing the precision and recall of the classifier under the threshold value. In general, when the P-R curve is closer to the upper right corner, it means that the model can guarantee high precision and high recall, i.e., the prediction results are more accurate. In this paper, compared with the original model, the curve is closer to the upper right corner, which indicates that the precision and recall are both higher. Partial visualisation results of the model are shown in Fig. 10.

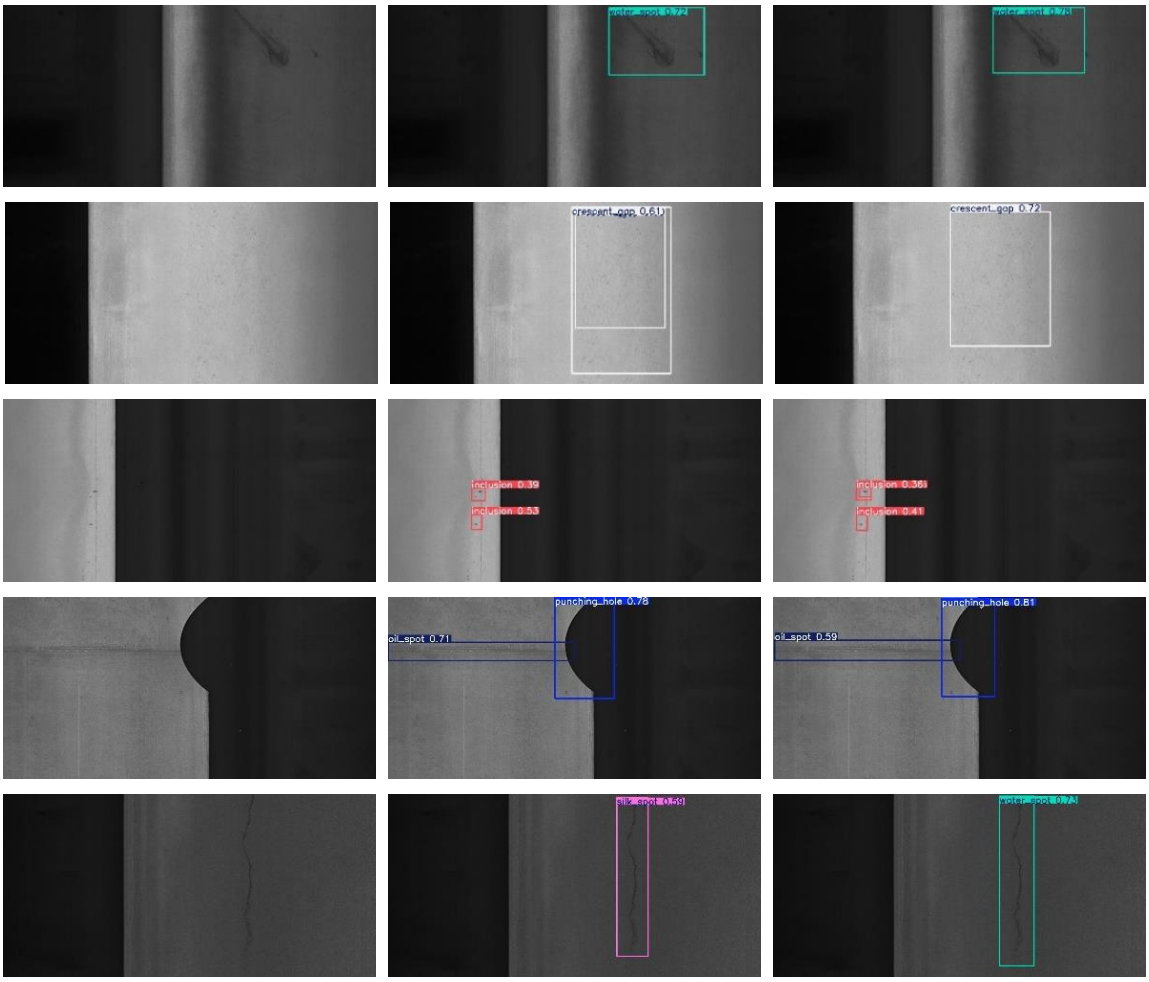


(a) YOLOv8n



(b) Ours

Figure 9. Precision-recall curve



Original Image YOLOv8n Ours

Figure 10. Partial visualisation results of the model

4.5. Ablation Experiments

In order to deeply analyse the impact of individual improvements on the baseline model, a series of ablation experiments were designed in this study to assess the specific impact of each improvement point on the model performance, the results of which are shown in Table 2.

Table 2. Ablation experiments on GC10-DET dataset

Model	mAP@0.5/%	mAP@0.5~0.95/%	Parameters/M	GFLOPs/G
YOLOV8n	65.1	34.0	3.0	8.1
YOLOV8n+C2f_Context	67.2	34.2	2.5	6.8
YOLOV8n+ASFF	66.8	34.9	4.4	10.3
YOLOV8n+ ShapeIoU	67.3	34.7	3.9	8.9
YOLOV8n+ASFF+C2f_ Context	67.5	34.3	3.9	8.9
YOLOV8n+ASFF+C2f_ Context +ShapeIoU	67.9	35.1	3.9	8.9

Comparing the data in Table 2 for analysis, YOLOv8n is the benchmark model, and the Context Guided Block is introduced into the c2f module in its Backbone, 2.1 per cent improvement over the baseline model on mAP@0.5/% and mAP@0.5~0.95/% improved by 0.2%; this shows that the C2f_Context module can more effectively achieve feature extraction and improve detection accuracy;

using ASFF to improve the detection head of YOLOv8n to form a new detection head Detect_ASFF, which mainly introduces an adaptive spatial feature fusion method, comparing with the baseline model, mAP@0.5/% improved by 1.7% and mAP@0.5~0.95/% improved by 0.9%, which demonstrates that this improvement is able to enhance the feature fusion effect, thus significantly improving the detection performance of the model. Using Shape-IoU as the bounding box loss function of the model in the benchmark model, mAP@0.5/% improved by 2.2% and mAP@0.5~0.95/% improves by 0.7%, which shows that this improvement can enhance the progress of the bounding box regression in target detection, thus improving the performance. Introducing C2f_Context and Detect_ASFF simultaneously. Compared to the original model, mAP@0.5/% improved by 2.4%, mAP@0.5~0.95/% improved by 0.3%. Introducing the improvement at the same time, a more significant performance improvement was obtained, mAP@0.5 is 67.9%, mAP@0.5~0.95 is 35.1%, Increased by 2.8% and 1.1% respectively. Analysed by ablation experiments, from the basic algorithm YOLOv8n to different module combination experiments, the improved algorithm in this paper has improved in detection accuracy, proving the practicality and effectiveness of the algorithm in this paper.

4.6. Comparison Experiments

In order to evaluate the effectiveness of the proposed algorithm, this paper compares the performance of several one-stage detection methods in terms of predicted mAP, detection time per image, and model volume when using the same GC10-DET dataset. The detection results of different models are shown in Table 3.

Table 3. Inspection results on GC10-DET

Model	mAP@0.5/%	mAP@0.5~0.95/%	Parametes/M	GFLOPs/G
YOLOv5n	65.7	33.5	2.5	7.1
YOLOv5s	65.7	35.2	9.1	23.8
YOLOv5l	67.3	35.3	53.1	134.7
YOLOv8s	67.6	34.6	11.1	28.5
YOLOV8n+ASFF+C2f _Context +Shape-IoU	67.9	35.1	3.9	8.9

Experiments show that the algorithm in this paper increases the mAP@0.5/% by 2.2%, 2.2%, 0.6%,and 0.3% compared to Yolov5n, Yolov5s, Yolov5l, and Yolov8s, respectively; while comparing Yolov5s, Yolov5l, and Yolov8s, the GFLOPs decreased by 14.9G, 125.8G, and 19.6G, respectively, further verifying the generality and effectiveness of the improved algorithm in this paper. We believe this is due to the fact that the improved algorithm enhances the feature representation capability by capturing and fusing contextual information, which improves the detection accuracy and robustness of the model, and the model is able to identify the target in the image more accurately.

5. CONCLUSION

The model in this paper is proposed to address the current problem of defect detection performed on steel surfaces. The introduced Context Guided Block module enables the model to not only focus on local features, but also capture surrounding context and global context information, and fuse the captured local features, surrounding context and global context information, thus improving the accuracy and robustness of the model for target detection. This ability of information fusion is especially important for dealing with target detection in complex scenes. In addition, ASFF is used to improve the detection head of YOLOv8 to form a new detection head Detect_ASFF, which is able to adaptively fuse features of different scales, enabling the Head part of YOLOv8 to more comprehensively utilise multi-scale feature information, thus improving the accuracy and robustness

of detection. The loss function due to Shape-IoU is selected as the bounding box loss function of the model in this paper, which enables the model to learn the geometric characteristics of the target more comprehensively during the training process, reduces the occurrence of false and missed detection, and improves the accuracy of the bounding box regression. The test results show that the method has a good improvement in the accuracy of steel defect detection, which can provide a better reference for the positioning and detection of surface defects on metal parts. However, there is still room for improvement in the current optimisation algorithm architecture, and subsequent studies can try to continue the optimisation of the backbone network and optimise the feature fusion method of the neck structure, in order to enhance the model's ability to detect multi-scale defects.

REFERENCES

[1] Su Hu, Zhang Jiabin, Zhang Bohao et al. A review of surface defect detection based on visual perception [J]. Computer Integrated Manufacturing Systems, 2023, 29(01): 169-191. SU H, ZHANG J B, ZHANG B H, et al. Review of surface defect inspection based on visual perception [J]. Computer Integrated Manufacturing Systems, 2023, 29(01): 169-191.

[2] 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.

[3] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[4] 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.

[5] Liu, W. et al. SSD: Single Shot MultiBox Detector [C]. ECCV. 2016, 28: 2137.

[6]] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Pro ceedings of the IEEE international conference on computer vision. 2017: 2980-298 8.

[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition, June 26-July 1, 2016, Las Vegas, NV, USA. New York: IEEE, 2016:779-788.

[8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[9] Redmon J, Farhadi A. Yolov3: An incremental improvement [J]. arXiv preprint arXiv:1804.02767, 2018.

[10] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection [J]. arXiv preprint arXiv:2004.10934, 2020.

[11] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications [J]. arXiv preprint arXiv:2209.02976, 2022.

[12] Li M J, Wang H, Wan Z B. Surface defect detection of steel strips based on improved YOLOv4 [J]. Computers and Electrical Engineering, 2022, 102: 108208.

[13] Yi C C, Xu B, Chen J, et al. An improved yolox model for detecting strip surface defects [J]. steel research international, 2022, 93(11): 2200505.

[14] Wang Z, Liu W S. Surface defect detection algorithm for strip steel based on improved YOLOv7 model [J]. IAENG International Journal of Computer Science, 2024, 51(3):308-316. yolo

[15] Wu T, Tang S, Zhang R, et al. Cgnet: A light-weight context guided network for semantic segmentation [J]. IEEE Transactions on Image Processing, 2020, 30: 1169-1179. (CGB)

[16] Shen, Chunhua, et al. "Adaptively spatial feature fusion f or object detection." Pattern Recognition Letters 137 (20 19): 27-37. ASFF

[17] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection [C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.shapeiou

[18] He Yun arty. Research on small target detection algorithm based on deep learning [D]. Jilin University, 2024. DOI:10.27162/d.cnki.gjlin.2024.004707.