# YOLOv8-FCS: A more focused YOLOv8 model for defect detection in images of steel surface

**Bingtao Hu**
School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology
**Rongsheng Lu**
rslu@hfut.edu.cn
School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology
**Dahang Wan**
School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology
**Sailei Wang**
School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology
**Jiajie Yin**
School of Instrument Science and Opto-electronics Engineering, Hefei University of Technology

**Research Article**

**Keywords:** Object detection, Steel surface defect detection, Attention mechanism, Image preprocessing, Neural network, Deep learning

**Posted Date:** May 13th, 2024

**DOI:** https://doi.org/10.21203/rs.3.rs-4368440/v1



**Additional Declarations:** Competing interest reported. The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rongsheng Lu reports financial support was provided by National Natural Science Foundation of China (NSFC) Grant No. 51875164). Rongsheng Lu reports was provided by National key Research and Development Program of China (No. 2018YFB2003801)

# Abstract

Defect detection in steel surface is crucial for engineering quality control. Traditional methods for detecting surface defects on steel materials have issues such as low detection accuracy, slow speed, low level of intelligence, and insufficient utilization of images. In response to these challenges, this paper proposes an improved YOLOv8 model for efficient and accurate detection of defects on steel surface. Firstly, we introduce a single-channel adversarial input strategy (AIS) to enhance the utilization of single-channel images and improve the network's detection effectiveness. Secondly, we utilize various attention modules to enhance the Neck and detection head of the network, thereby further improving the network's expressive power and detection performance. Finally, experiments were conducted on three open datasets, achieving a mAP (mean average precision) of 77.3% on the NEU-DET dataset, outperforming YOLOv8 at 74.1%, a mAP of 65.5% on the GC10 dataset, outperforming YOLOv8 at 64.0%, and a mAP of 73.8% on the Magnetic-tile-defect-datasets, outperforming YOLOv8 at 71.2%. Additionally, the average detection speed of this model is 93 frames per second, effectively balancing detection accuracy and efficiency.


# 1 Introduction

Steel materials are indispensable in automotive, defense, machinery manufacturing, chemical, and light industries. However, various types of defects, especially surface defects such as cracks, scabs, curls, voids, wear, and scratches, are generated during the production process of steel materials due to raw materials and process issues, which have a fatal impact on the corrosion resistance and strength of steel materials[1]. These defects not only affect the appearance of products but also impact the economic benefits of factories. Therefore, quality control of steel materials is crucial.

Early detection of surface defects in steel materials was based on traditional image processing, relying on manual design by engineers and algorithm personnel. Many researchers[2, 3]manually crafted features and utilized SVM(support vector machines) or BP networks for classification, achieving the recognition of multi-class defects. For instance, Batsuuri [2]used SIFT features for defect detection and SVM for defect classification. Martins [4]proposed an automatic system based on image analysis technology that, with the help of neural networks, classified three types of defects with clear geometric shapes. Similar work includes[5]. However, manual feature crafting requires high expertise from algorithm personnel, and with the variety and complexity of defects, many researchers have started using CNN directly for feature extraction and classification[6]. Whether using CNN or SVM for defect classification on steel material surfaces, it is done at the image level, making it difficult to quantify the size and severity of defects, which is not conducive to subsequent production management. Therefore, many researchers are now using object detection algorithms based on deep learning to detect defects on the surface of steel materials.

Object detection plays a pivotal role in computer vision, aiming to accurately identify and locate multiple objects of different classes from images or videos. Object detection algorithms can be broadly categorized into one-stage and two-stage approaches. Currently, most two-stage object detection algorithms are based on the RCNN[7–9] series, such as[10], which replaced conventional convolution kernels with deformable convolution kernels and used multi-scale feature layers to extract feature maps of defects of different scales, improving the localization accuracy of the Faster-RCNN network. Hou [11]also made improvements to the Faster-RCNN network, proposing a new two-stage network (CANet) based on context information and spatial attention to effectively perceive and utilize features of small defects. Similar two-stage methods include[12–14].

While the two-stage object detection algorithms have advantages in accuracy and detection effectiveness, they often have complex models and slower speeds. Therefore, more researchers are focusing on one-stage algorithms that balance speed and accuracy more effectively, such as YOLO[15–19]and SSD[20, 21], and improving them to enhance the detection accuracy of steel surface defects. For example, in literature[22], Res2Net blocks were employed to replace the backbone components of YOLOv5, expanding the receptive field, extracting features of different scales, and further improving the detection accuracy using a decoupled head. The number of parameters is smaller than that of the two-stage network, but the number of parameters in the network is still huge. To reduce the number of parameters, Qian[23]used ShuffleNetv2 as the feature extraction network and proposed a lightweight feature pyramid network (LFPN) to improve the efficiency of multi-scale feature fusion. Subsequently, Liu[24] proposed a method based on ghost convolution. Although the two methods have reduced the number of parameters and computational complexity of the network, they have not shown a significant improvement in detection efficiency. Similarly, Yang et al[25–27]also made lightweight improvements to the YOLO series of algorithms and introduced

attention mechanisms to enhance the network's expressive power while making up for the deficiency of low detection efficiency of the above target detection methods. However, the interpretability of these articles for the expression effect is not strong. Therefore, Chen[28]not only improves the defect detection accuracy by incorporating attention mechanisms based on YOLOX but also provides interpretability analysis of the added attention through Grad-CAM[29], allowing for an intuitive understanding of the advantages of the attention module.

In addition, the object detection methods mentioned above[22–28] have addressed or alleviated the issues of diverse defect scales, diverse defect types, and real-time detection in steel surface inspection. However, when using grayscale images as network inputs, researchers mostly convert the grayscale images into pseudo-RGB images with three identical channels, resulting in redundant features and underutilization of the original image. While the methods mentioned above are significant for defect detection in steel surface images, the following issues persist:

1. Inefficient detection or relying solely on classification models for image-level detection. Some researchers have utilized traditional algorithms for defect detection in steel surface images, which exhibit poor robustness and generalization.
2. Some object detection models fail to strike a good balance concerning model parameters, computational complexity, inference speed, and detection performance. For example, some researchers employ two-stage models for object detection, which have high model complexity, large amount of parameter calculation, slow speed, and are not conducive to deployment. Although some researchers opt for one-stage networks for defect detection in steel surface images, which are easier to deploy, they encounter challenges such as subpar detection performance.
3. Most defect detection algorithms for steel surface images directly convert grayscale images into pseudo-RGB images with three identical channels, resulting in feature redundancy and inadequate utilization of input images.
4. Some researchers have demonstrated that incorporating hybrid attention mechanisms can improve the detection performance of object detection models in steel surface images. However, currently, there are limited studies on hybrid attention modules.

To attain an equilibrium among the crucial aspects of detection accuracy, speed, and model parameters while maximizing the utilization of available features, we propose an efficient object detection network for defect detection in steel surface images based on the YOLOv8 framework. The contributions of this paper can be summarized as follows:

1. Propose a one-stage detection network, YOLOv8-FCS, for steel surface defect detection.
2. Introduce an image preprocessing method called AIS that can fully utilize the prior features of grayscale images.
3. Apply a hybrid attention module to the model.
4. Validate the effectiveness and feasibility of the model on three open-source detection datasets.

The paper is structured as follows: Section 2 provides a brief overview of related work, including datasets, attention mechanisms, and image preprocessing techniques. Section 3 introduces the architecture of the YOLOv8-FCS network and the proposed adversarial input strategy. Section 4 tests the proposed algorithm on three open datasets and presents the experimental results. Section 5 concludes the article.

## 2 Related Work

# 2.1 Dataset

Table 1 summarizes some of the classic steel surface images. In the table, "Dataset" represents the name of the dataset, "Categories" indicates the number of defect categories in the dataset, "Images" denotes the total number of images, "Year" represents the year of publication, and "Website" provides the source for accessing the images. In this paper, we conducted experiments with three datasets, NEU-DET, GC10-DET, and Magnetic-tile-defect-datasets, to demonstrate the generalizability of the proposed model.

Table 1
Steel Surface Defect Detection Datasets

| DataSet | Categories | Images | Year | Website |
|---|---|---|---|---|
| NEU-DET | 6 | 1800 | 2013 | http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm |
| Severstal-Steel-Defect-Detection | 4 | 18074 | 2019 | https://www.severstal.com/ |
| Guangdong-Aluminum-Defect-Detection | 27 | 3686 | 2016 | https://tianchi.aliyun.com/dataset/dataDetail?dataId=140666 |
| BSData-dataset | 1 | 1104 | 2021 | https://github.com/2Obe/BSData?tab=readme-ov-file |
| GC10-DET | 10 | 3570 | 2020 | https://github.com/lvxiaoming2019/GC10-DET-Steellic-Surface-Defect-Datasets?tab=readme-ov-file |
| Magnetic-tile-defect-datasets | 5 | 1344 | 2018 | https://github.com/abin24/Magnetic-tile-defect-datasets. |

## 2.2 Attention Mechanism

The attention mechanism is a method that mimics the human visual and cognitive systems[30]. It can select the most critical information for the current task from much information. It is applied in machine learning and deep learning tasks to enhance model performance. Depending on the application and task, attention mechanisms can be categorized into three main types: hard attention mechanism, soft attention mechanism, and self-attention mechanism. In the hard attention mechanism, the model focuses only on a specific input part, directly discarding irrelevant parts. Unlike the hard attention mechanism, the soft attention mechanism allows the model to assign weights to different input parts rather than focusing only on a specific part. The self-attention mechanism allocates attention to different positions by capturing dependencies between different locations. The soft attention mechanism is the most commonly used in steel surface defect detection. Representative articles include SE[30], CBAM[31], ECA[32], among others. In this paper, the attention module EMABTK is also composed of soft attention.

## 2.3 Image Preprocessing

Image preprocessing techniques[33] in object detection involve processing operations on input images to extract and enhance the object information, providing better input for subsequent detection. Familiar image preprocessing techniques include image grayscale conversion, image denoising, and image enhancement. Grayscale conversion refers to converting a color image to a grayscale image. In numerous tasks, considering image brightness information alone suffices, rendering the inclusion of color information unnecessary. Therefore, converting a color image to a grayscale image simplifies the processing and reduces computational complexity. During the image acquisition and transmission process, various interferences and noises, such as Gaussian and salt-and-pepper noise, often affect the image. Filters can mitigate the impact of noise on image quality. Image enhancement[34] improves images' quality and visual effects by altering attributes such as contrast, brightness, and color.

In this paper, we employ the AIS image preprocessing method. We apply two different pixel transformations to enhance the input single-channel grayscale image. Then, the resulting image is concatenated with the original input image to obtain a three-channel image fed into the detection network.

# 3 Method

## 3.1 YOLOv8-FCS algorithm

The paper proposes improvements to the YOLOv8 network by introducing various modules and strategies, presenting a network suitable for steel surface defect detection. As shown in Fig. 1, the YOLOv8-FCS network in this paper comprises three main

parts: Backbone, Neck, and Head. Initially, before entering the backbone, the AIS method transforms the single-channel grayscale image into a three-channel image. Then, the preprocessed images are inputted into the backbone. The structure of the backbone is similar to YOLOv5, but it replaces the C3 module with the more gradient-rich C2f module. Features are extracted through multiple convolution modules in the backbone and enhanced through the SPPF module before entering the PANet connection in the Neck for bidirectional feature fusion. Finally, the features are fed into the Head layer for defect prediction and localization.

## 3.2 Single-channel image adversarial input strategy

In this study, we propose a novel single-channel image input strategy to improve the model's efficiency and performance in processing grayscale images. Traditional methods often use the gray2rgb conversion, duplicating the single-channel grayscale image into three color channels to adapt to convolutional neural networks designed for processing color images. However, this approach needs to improve on significant data redundancy issues since the information in the three channels is identical. To overcome this limitation, we explore an innovative grayscale image input strategy that enhances the model's performance by introducing different information in the three channels.

Figure 2 illustrates the comparison between our approach and the conventional method. Figure 2(a) depicts the traditional grayscale image input method, which directly applies the gray2rgb conversion, resulting in data redundancy as the content in the three channels is the same. In contrast, our strategy employs two opposing filtering methods to process the original grayscale image, generating three channels with inconsistent information, as shown in Fig. 2(b). We apply a mean filter in the first channel to smooth the original grayscale image. Mean filtering eliminates noise and minor image disturbances while preserving the overall contours and structures. In the second channel, different detail enhancement filters with varying parameters are used to highlight the image's details and texture information. Detail enhancement filtering can increase the local contrast of the image, making subtle features more pronounced and providing rich texture information for the model. The third channel uses the original image as input, retaining the raw image information. Through this design, our grayscale image input strategy effectively utilizes the three channels to represent different levels of image information, from global to detail, avoiding data redundancy and aiming to provide more valuable features for deep learning model learning.

## 3.3 Modules in the YOLOv8-FCS Network

Figure 3 illustrates the composition of the YOLOv8-FCS network modules. The Conv module consists of a 2D convolutional layer, a batch normalization (BN) layer, and the SiLU activation function. The Conv module performs 2D convolution operations to extract spatial and channel information from the input feature map. The output is then normalized using the BN layer to accelerate the training process and improve model stability. Finally, the SiLU activation function applies a non-linear transformation to introduce non-linear features and enhance the model's expressive power. The C2f module is used for feature fusion and consists of two Conv layers and multiple Bottleneck blocks. The SPPF module is an accelerated version of the SPP module used for multiscale feature fusion. It is composed of Conv modules and max pooling layers. The EMABTK module incorporates the EMA attention mechanism into the Bottleneck block. Traditional lightweight attention mechanisms focus on simplifying the model structure, resulting in the loss of feature information. However, the EMA attention mechanism[35] can learn effective channel descriptions without reducing the dimension. It reconstructs a portion of the channel dimensions into the batch dimensions, reducing computational overhead while preserving information from each channel. The EMA attention mechanism also groups the channel dimension into multiple sub-features, ensuring spatial semantic features are evenly distributed within each feature group. In steel surface defect images, due to variations in defect size, position, and shape, the introduction of the EMA attention mechanism promotes information transfer between different-dimensional features. This mechanism aids the model in understanding and capturing the feature representation of defects, thereby improving the model's detection capability.

## 3.4 Detection Head

The complexity of combining localization and classification in object detection has led to the development of various detection heads. There are two common types of detection heads: coupled heads and decoupled heads. A coupled head combines the classification and localization tasks into a single neural network, where a single network simultaneously performs classification

and localization. The YOLOv3-YOLOv5 algorithms employ coupled heads, where the classification branch and localization branch share parameters, reducing the amount of computation and parameters. However, due to the different focus of the classification and regression tasks, coupled heads often suffer from lower detection accuracy.

To address the issue above, YOLOX adopts a decoupled head that extracts object position and class information separately through different network branches. This method successfully mitigates conflicts in feature information required for various tasks, thereby significantly improving the model's convergence speed and detection accuracy. In YOLOv6, researchers employ a mixed channel strategy to construct a more efficient detection head. Similarly, YOLOv8 also adopts a decoupled head structure. Two parallel branches extract class and position features, respectively. Unlike YOLOv6 and YOLOX, YOLOv8's head does not have a confidence branch and only consists of classification and regression branches. The regression branch employs the integral form representation mentioned in DFL. Additionally, the channel numbers of the class and regression branch are different, enabling a better representation of the two distinct features.

In steel surface defect detection, an image may contain defects of multiple scales, types, and locations. Therefore, an effective detection head should possess scale-awareness, task-awareness, and spatial-awareness capabilities. In this research, we enhance the YOLOv8 detection head by integrating the DyHeadBlock[36] module, as depicted in Fig. 4. Figure 5 visually represents the modified structure. The DyHeadBlock module incorporates three types of attention: scale, spatial, and task awareness, effectively improving the model's detection accuracy.

## 3.5 Loss Function

In steel surface defect detection, a loss function plays a pivotal role in precisely predicting the category and location of defects in the image. The loss function measures the discrepancy between the predicted bounding boxes and class labels and the ground truth bounding boxes and class labels. By minimizing the loss function, the model's predictions can be brought closer to the ground truth, thereby enhancing the model's accuracy.

This paper's loss function comprises three components: the classification loss $L_{cls}$,the bounding box loss $L_{box}$, and the DFL loss $L_{dfl}$. The formula for the loss function is as follows:

$$L_{all} = \lambda_{cls} L_{cls} + \lambda_{box} L_{box} + \lambda_{dfl} L_{dfl}$$

1

$L_{all}$ encompasses three components, where $\lambda$ is a hyperparameter representing the weights assigned to each component. These weights can be adjusted based on the specific requirements before training. In this paper, the weights for the three components are 0.5, 7.5, and 1.5, respectively.

The classification loss, denoted as $L_{cls}$, utilizes the BCE(binary cross-entropy) loss and can be mathematically expressed as follows:

$$L_{cls} = -[C_i log C_i' + (1 - C_i) \log \left(1 - C_i'\right)]$$

2

$C_i$ and $C_i'$ represent the class's actual and predicted values.

The bounding box loss, $L_{box}$, utilizes the CIoU Loss, which considers multiple factors such as position, shape, and orientation. This careful consideration allows the model to learn the characteristics of the target bounding boxes more effectively, thus enhancing its performance in steel surface defect detection. The expression for the CIoU Loss is as follows:

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

3

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

4

$$\alpha = \frac{v}{1 - IoU + v}$$

5

Here, $\rho$ represents the Euclidean distance, $b$ and $b^{gt}$ denote the center coordinates of the predicted and ground truth bounding boxes, $w$ and $h$ represent the width and height of the predicted bounding box, $w^{gt}$ and $h^{gt}$ represent the width and height of the ground truth bounding box, $c$ denotes the diagonal length of the smallest enclosing box covering the two boxes, $v$ evaluates the consistency of aspect ratio and $\alpha$ is a trade-off parameter.

We compute the DFL by utilizing the distances between the positive sample labels and the center points of the predicted bounding boxes concerning each side. Before calculating the DFL loss, converting the positive sample labels($x_{min}$,$y_{min}$, $x_{max}$, $y_{max}$)into distances from the center to the left, top, right, and bottom edges is necessary. The expression for the DFL loss is as follows:

$$DFL\left(S_i\square S_{i+1}\right) = -[\left(y_{i+1} - y\right)\log\left(S_i\right) + \left(y - y_i\right)\log\left(S_{i+1}\right)]$$

6

# 4 Experiments

# 4.1 Experimental Platform and Relevant Metrics

The evaluation of network performance primarily relies on mAP during the training process and the performance of the trained network on the validation set. Precision(P), recall(R), and mAP are adopted as performance evaluation metrics to assess the detection results quantitatively. The expressions for P and R are as follows:

$$P = TP/(TP + FP)$$

7

$$R = TP/\left(TP + FN\right)$$

8

True positives (TP): The number of samples that are positive and correctly classified as positive by the classifier; True negatives (TN): The number of samples that are negative and correctly classified as negative by the classifier; False positives (FP): The number of samples that are negative but incorrectly classified as positive by the classifier; False negatives (FN): The number of samples that are positive but incorrectly classified as negative by the classifier.

Average Precision (AP) is the area under the P-R curve. Generally, a higher AP value indicates a better classifier performance. mAP takes the AP values for each class separately, calculates the average of all class APs, and represents a comprehensive measure of the average precision for the detected objects. Table 2 presents the experimental platform utilized in this study.

Table 2
Experimental Platform

| platform | specifications |
|---|---|
| CPU | 13th Gen Intel(R) Core(TM) i5-13600KF |
| GPU | NVIDIA GeForce RTX 4070Ti |
| Operating System | Windows 11 |
| Framework | Pytorch 1.21 |

# 4.2 Experimental Dataset

To demonstrate the versatility of our model, we conducted experiments on three open-source datasets: NEU-DET[37], GC10[38], and Magnetic tile defect(MTD)[39].

The NEU-DET dataset is an open dataset specifically designed for hot-rolled steel strip defect detection. It consists of 1800 images, with 1440 randomly selected for training and 360 images for validation. The images have a size of 200×200, which we resized to 224×224 before inputting them into the network. The dataset includes six defect classes: crazing, inclusion, patches, pitted_surface, rolled-in_scale, and scratches. Figure 6(a) shows the distribution of each class.

The GC10 dataset is collected from real industrial settings, specifically designed for steel plate surface defect detection. It consists of 10 different defect classes and comprises 3570 grayscale images. Out of these, there are 2294 labeled images available for use. We randomly selected 1836 images as the training set and 458 as the validation set. Figure 6(b) illustrates the distribution of each class.

The MTD dataset is a publicly available dataset for magnetic tile defect detection. It consists of 1344 images, with 392 images containing defects. We randomly selected 314 images for training and 78 images for validation. The dataset encompasses five defect classes: MT_Uneven, MT_Blowhole, MT_Break, MT_Crack, and MT_Fray. Figure 6(c) depicts the distribution of each class.

# 4.3 Ablation Experiments

Results on the NEU-DET dataset are presented in Table 3, indicating a significant improvement in the performance of the YOLOv8-FCS model by introducing three technical improvements: AIS, DyHeadBlock, and EMABTK. The complete model, incorporating all three improvements, achieved mAP50 and mAP50-95 scores of 77.3% and 44.5%, respectively, outperforming the baseline YOLOv8n model with 74.1% and 41.8%. Moreover, despite the increased parameters and computational complexity observed in the complete model, its recall improved, demonstrating enhanced detection capabilities.

Similar effectiveness of these technical improvements was observed on the GC10 dataset, as shown in Table 4. The complete model achieved mAP50 and mAP50-95 scores of 65.5% and 33.6%, respectively, exhibiting stable improvement compared to the baseline model. Despite the comparatively modest progress on this dataset, considering the potentially higher complexity or diversity of the GC10 dataset, such improvement still validates the robustness and adaptability of the YOLOv8-FCS model. Changes in precision and recall also reflect the model's adaptability to different scenarios.

On the MTD dataset, the complete model also demonstrated excellent performance, as shown in Table 5, achieving mAP50 and mAP50-95 scores of 73.8% and 50.3%, respectively. Notably, the complete model achieved the highest precision of 84.8%, showcasing its significant capability in reducing false positives. The improvement in recall also indicates the model's ability to provide more comprehensive coverage of actual targets, which is crucial for applications such as defect detection that require high precision and recall.

Figure 7 depicts the mAP curves during the training process on the NEU-DET, GC10, and MTD. It is evident from the figure that the mAP values of the YOLOv8-FCS model surpass those of the YOLOv8 model on all three datasets. Furthermore, through a comprehensive analysis of the results from ablation experiments on the three datasets, it is evident that the three technical improvements (AIS, DyHeadBlock, and EMABTK) are critical for enhancing the performance of the YOLOv8-FCS model. These

improvements enhance detection accuracy (mAP) and optimize precision and recall, enabling the model to deliver better performance and adaptability across different application scenarios. While these improvements inevitably increase parameters and computational complexity, the trade-off is reasonable considering the significant performance improvement. Overall, the YOLOv8-FCS model, empowered by these innovative improvements, demonstrates strong competitiveness and broad application potential in object detection.

Table 3
The results of ablation experiments on the NEU-DET dataset

| Method | Baseline(YOLOv8n) | | | | Our Models | | | |
|---|---|---|---|---|---|---|---|---|
| AIS | | √ | | | √ | | √ | √ |
| DyHeadBlock | | | √ | | √ | √ | | √ |
| EMABTK | | | | √ | | √ | √ | √ |
| mAP50(%) | 74.1 | 75.6 | 75.4 | 74.3 | 76.0 | 73.7 | 75.8 | 77.3 |
| mAP50-95 (%) | 41.8 | 43.4 | 43.8 | 42.0 | 44.2 | 41.1 | 43.3 | 44.5 |
| Params(M) | 2.87 | 2.87 | 3.32 | 3.62 | 3.32 | 4.08 | 3.62 | 4.08 |
| Flops(G) | 8.1 | 8.1 | 9.6 | 9.7 | 9.6 | 11.2 | 9.7 | 11.2 |
| FPS | 157.0 | 156.3 | 108.2 | 125.4 | 107.5 | 93.6 | 124.7 | 93.0 |
| P (%) | 74.2 | 74.0 | 71.1 | 67.3 | 73.1 | 66.5 | 66.3 | 71.7 |
| R(%) | 67.0 | 68.5 | 69.6 | 69.7 | 70.0 | 70.0 | 71.8 | 72.7 |

Table 4
The results of ablation experiments on the GC10 dataset

| Method | Baseline(YOLOv8n) | | | | Our Models | | | |
|---|---|---|---|---|---|---|---|---|
| AIS | | √ | | | √ | | √ | √ |
| DyHeadBlock | | | √ | | √ | √ | | √ |
| EMABTK | | | | √ | | √ | √ | √ |
| mAP50(%) | 64.0 | 67.6 | 64.3 | 65.1 | 66.0 | 64.9 | 66.6 | 65.5 |
| mAP50-95 (%) | 32.5 | 33.8 | 33.1 | 33.2 | 32.5 | 32.0 | 33.2 | 33.6 |
| Params(M) | 2.87 | 2.87 | 3.33 | 3.62 | 3.33 | 4.08 | 3.62 | 4.08 |
| Flops(G) | 8.1 | 8.1 | 9.6 | 9.7 | 9.6 | 11.2 | 9.7 | 11.2 |
| FPS | 76.3 | 75.6 | 59.3 | 65.5 | 58.7 | 57.6 | 64.7 | 57.0 |
| P (%) | 65.2 | 68.4 | 70.7 | 65.9 | 67.6 | 65.7 | 68.1 | 66.9 |
| R(%) | 62.4 | 65.0 | 58.8 | 62.4 | 66.1 | 61.7 | 65.5 | 62.9 |

Table 5
The results of ablation experiments on the MTD dataset

| Method | Baseline(YOLOv8n) | | | | Our Models | | | |
|---|---|---|---|---|---|---|---|---|
| AIS | | √ | | | √ | | √ | √ |
| DyHeadBlock | | | √ | | √ | √ | | √ |
| EMABTK | | | | √ | | √ | √ | √ |
| mAP50(%) | 71.2 | 74.2 | 72.6 | 73.7 | 71.9 | 74.7 | 72.7 | 73.8 |
| mAP50-95 (%) | 48.7 | 51.5 | 49.1 | 51.5 | 50.2 | 50.4 | 51.4 | 50.3 |
| Params(M) | 2.87 | 2.87 | 3.32 | 3.62 | 3.32 | 4.08 | 3.62 | 4.08 |
| Flops(G) | 8.1 | 8.1 | 9.6 | 9.7 | 9.6 | 11.2 | 9.7 | 11.2 |
| FPS | 107.8 | 107.1 | 81.9 | 86.1 | 81.2 | 70.7 | 85.5 | 70.0 |
| P (%) | 80.8 | 70.6 | 78.9 | 72.5 | 79.0 | 82.2 | 78.3 | 84.8 |
| R(%) | 65.9 | 71.6 | 70.9 | 76.9 | 69.6 | 71.5 | 70.5 | 69.9 |

To gain a deeper understanding of the impact of multiple attention mechanisms on the model's focusing capability, we employed heatmap visualization techniques to analyze the attention distribution of the model, as shown in Fig. 8. The heatmap clearly illustrates that the model significantly reduces its focus on the background while concentrating more on the target objects after introducing attention mechanisms. This finding suggests that attention mechanisms effectively guide the model's attention towards regions that are more crucial for the final detection task, thereby enhancing the model's detection accuracy and efficiency.

We have observed a significant performance improvement through comparative analysis by incorporating multiple attention mechanisms. The visualized results from the heatmaps further validate the effectiveness of these attention mechanisms in enabling the model to focus more on critical information in the images, reducing false detections, and enhancing detection accuracy. These findings not only demonstrate the efficacy of our model design but also provide valuable insights for future research in the field of object detection.

## 4.4 Comparison with Other Algorithmic Detection Results

On the NEU-DET, GC10, and MTD, various object detection models exhibit different performance characteristics, as shown in Tables 6–8. Key metrics such as parameters, computational complexity, FPS, mAP, precision, and recall show significant differences among the YOLO series (including YOLOv6n, YOLOv6s, YOLOv5n, YOLOv5s, YOLOv3-tiny, YOLOv7-tiny, and the focus of this paper, YOLOv8-FCS), Faster-RCNN, and YOLOX. These differences are influenced by the design philosophies of different models, with one-stage models inclined towards optimizing speed and streamlining the process. In contrast, two-stage models prioritize improving detection accuracy.

On the NEU-DET dataset, YOLOv8-FCS stands out with an impressive mAP50 of 77.3% and mAP50-95 of 44.5%, demonstrating its efficiency and accuracy in handling challenging industrial images. In contrast, other YOLO series models, Faster-RCNN and YOLOX, exhibit competitive performance but fall short in precision, recall, or frame rate. On the GC10 dataset, where models face increased challenges, the overall mAP decreases. YOLOv8-FCS again proves its adaptability and superiority with a mAP50 of 65.5% and mA50-95 of 33.6%. Models like YOLOv6s and YOLOv5n demonstrate advancements in precision and recall, demonstrating their potential in handling complex environments. On the MTD dataset, YOLOv8-FCS performs exceptionally well, particularly with a mAP50 reaching 73.8%, showcasing its strong adaptability to multi-object detection tasks.

Table 6
The results of the comparative experiments on the NEU-DET dataset

| Methods | Params (M) | FLOPS (G) | FPS | mAP50 (%) | mAP50-95(%) | P(%) | R(%) |
|---|---|---|---|---|---|---|---|
| YOLOv6n[40] | 4.04 | 11.8 | 169.7 | 73.8 | 42.3 | 69.3 | 68.0 |
| YOLOv6s[40] | 15.54 | 44.0 | 153.1 | 73.8 | 41.4 | 69.1 | 69.6 |
| YOLOv5n[41] | 1.69 | 4.2 | 191.6 | 75.1 | 37.9 | 70.5 | 70.3 |
| YOLOv5s[41] | 6.7 | 15.8 | 178.6 | 77.0 | 40.7 | 74.4 | 70.7 |
| YOLOv3-tiny[17] | 8.28 | 12.9 | 434.0 | 74.4 | 36.7 | 73.8 | 67.3 |
| YOLOv7-tiny[19] | 5.74 | 13.1 | 78.6 | 73.2 | 36.4 | 73.1 | 66.5 |
| Faster-RCNN[9] | 41.37 | 23.1 | 21.1 | 74.5 | 39.1 | - | - |
| YOLOX-s[42] | 8.94 | 3.28 | 69.3 | 67.8 | 34.1 | - | - |
| YOLOv8-FCS | 4.08 | 11.2 | 93.0 | 77.3 | 44.5 | 71.7 | 72.7 |

Table 7
The results of the comparative experiments on the GC10 dataset

| Methods | Params (M) | FLOPS (G) | FPS | mAP50 (%) | mAP50-95(%) | P(%) | R(%) |
|---|---|---|---|---|---|---|---|
| YOLOv6n[40] | 4.04 | 11.8 | 116.0 | 62.3 | 31.1 | 68.0 | 59.2 |
| YOLOv6s[40] | 15.54 | 44.0 | 90.6 | 65.0 | 32.5 | 72.3 | 60.7 |
| YOLOv5n[41] | 1.69 | 4.2 | 108.7 | 64.8 | 32.8 | 65.3 | 62.6 |
| YOLOv5s[41] | 6.7 | 15.8 | 87.9 | 64.0 | 32.5 | 65.0 | 63.1 |
| YOLOv3-tiny[17] | 8.29 | 12.9 | 115.4 | 57.7 | 27.4 | 56.2 | 59.0 |
| YOLOv7-tiny[19] | 5.75 | 13.1 | 55.9 | 62.3 | 31.0 | 59.1 | 63.7 |
| Faster-RCNN[9] | 41.39 | 90.9 | 20.9 | 65.5 | 32.2 | - | - |
| YOLOX-s[42] | 8.94 | 26.78 | 55.9 | 57.0 | 27.7 | - | - |
| YOLOv8-FCS | 4.08 | 11.2 | 57.0 | 65.5 | 33.6 | 66.9 | 62.9 |

Table 8
The results of the comparative experiments on the MTD dataset

| Methods | Params (M) | FLOPS (G) | FPS | mAP50 (%) | mAP50-95(%) | P(%) | R(%) |
|---|---|---|---|---|---|---|---|
| YOLOv6n[40] | 4.04 | 11.8 | 121.1 | 70.2 | 49.2 | 82.0 | 63.5 |
| YOLOv6s[40] | 15.54 | 44.0 | 85.8 | 71.5 | 50.5 | 74.6 | 72.5 |
| YOLOv5n[41] | 1.68 | 4.1 | 168.4 | 66.4 | 42.1 | 72.1 | 64.0 |
| YOLOv5s[41] | 6.7 | 15.8 | 149.0 | 71.6 | 47.2 | 72.0 | 72.5 |
| YOLOv3-tiny[17] | 8.27 | 12.9 | 356.7 | 70.4 | 46.4 | 78.1 | 64.6 |
| YOLOv7-tiny[19] | 5.74 | 13.1 | 57.8 | 69.1 | 45.6 | 78.2 | 66.9 |
| Faster-RCNN[9] | 41.37 | 90.9 | 37.9 | 66.3 | 43.1 | - | - |
| YOLOX-s[42] | 8.94 | 26.77 | 19.0 | 68.8 | 44.8 | - | - |
| YOLOv8-FCS | 4.08 | 11.2 | 70.0 | 73.8 | 50.3 | 84.8 | 69.9 |

The YOLOv8-FCS model significantly enhances detection accuracy while maintaining efficiency, thanks to its lower parameters, computational complexity, and outstanding FPS, mAP, precision, and recall performance. These results demonstrate that YOLOv8-FCS is a powerful visual detection model that excels in various tasks and environments. Furthermore, the success of YOLOv8-FCS further validates the potential and prospects of one-stage detection models in deep learning. Meanwhile, other YOLO series models, Faster RCNN, YOLOX, and others, also demonstrate their robust functionality and application potential in their respective domains. However, in direct comparison with YOLOv8-FCS, there is still room for improvement in specific vital metrics. Overall, the performance of these models not only reflects the latest advancements in object detection technology and provides valuable insights for future research and applications.

Figure 9 showcases a comparative analysis of partial detection results between the YOLOv8-FCS and YOLOv8 models across three distinct datasets (NEU-DET, GC10, and MTD dataset). Through this comparison, we can visually observe the performance improvement brought by the model enhancements, particularly in reducing missed detections and false positives.

Figure 9 provides a visual depiction that allows us to discern that the YOLOv8-FCS model attains a remarkable decrease in false negative rate on three datasets after incorporating multiple attention mechanisms. This progress can be attributed to the ability of attention mechanisms to help the model focus more on the target regions, thereby enhancing the detection capability of small objects or objects in complex backgrounds. Notably, when confronted with the MTD dataset, which frequently encompass minor or subtle defects, the enhanced model showcases a heightened proficiency in recognizing these targets, indicating a strengthened capacity for handling challenging scenarios.

## 4.5 Qualitative Results

Figure 10 depicts the qualitative results of the YOLOv8-FCS algorithm on three datasets. The figure provides compelling evidence that the YOLOv8-FCS model achieves accurate recognition and precise localization of defects in steel surface images.

# 5 Conclusion

In summary, to address the limitations of previous methods for steel surface defect detection, such as low detection accuracy, slow speed, low level of intelligence, and insufficient utilization of image information, we have made improvements to the YOLOv8 model and proposed the YOLOv8-FCS model to achieve efficient and accurate detection of steel surface defects. We have also introduced a single-channel adversarial input preprocessing method and incorporated multiple attention modules to enhance the network's representation capability and detection performance. The results on three open-source datasets demonstrate that the detection performance of YOLOv8-FCS surpasses that of YOLOv8 and other models. The model achieves a

detection speed of 93 frames per second, with a mAP of 77.3% on the NEU-DET dataset, effectively balancing detection accuracy and efficiency. The visualization results through heatmaps further confirm that the proposed YOLOv8-FCS algorithm exhibits superior expressive power compared to other algorithms.

This paper proposes an algorithm that is a general model, enabling easy application to other detection tasks, such as medical image detection and intelligent monitoring. Engineers and researchers can customize and improve the model according to their specific tasks and adapt it to precise detection requirements. Furthermore, there is room for improvement in the single-channel adversarial input preprocessing method proposed in this paper. While the AIS method enhances the network's detection performance, it does come at the cost of preprocessing time. Interested researchers can delve into the exploration of acceleration algorithms to optimize this process, which is one of our future research directions. Looking ahead, we plan to apply the AIS method to image classification and image segmentation domains to study its effectiveness in other areas. We will also continue to extend the application of the YOLOv8-FCS model to detection tasks in different fields.

## Declarations

# Disclosure of Interest

**s.** The authors have no competing interests to declare that are relevant to the content of this article.

### Competing Interests

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rongsheng Lu reports financial support was provided by National Natural Science Foundation of China (NSFC) Grant No. 51875164). Rongsheng Lu reports was provided by National key Research and Development Program of China (No. 2018YFB2003801)

# Author Contribution

Bingtao Hu and Dahang Wan wrote the main manuscript text and prepared figures. Rongsheng Lu did a review of the paper and provided funding. Bingtao Hu , Dahang Wan, Sailei Wang and Jiajie Yin did the experiments for the paper. All authors reviewed the manuscript.

# Acknowledgement

This work was supported by the National Key Research and Development Program of China (No. 2023YFF0715502); Anhui Provincial Key Research and Development Project (No. 202304a05020013).

## References

1. Usamentiaga R, Lema DG, Pedrayes OD, Garcia DF (2022) Automated Surface Defect Detection in Metals: A Comparative Review of Object Detection and Semantic Segmentation Using Deep Learning. IEEE Trans Ind Applicat 58:4203–4213. https://doi.org/10.1109/TIA.2022.3151560
2. Batsuuri S, Ahn J, Ko J (2012) Steel surface defects detection and classification using SIFT and voting strategy. 6:161–166
3. Qinghe H, Jiazhuo X, Weidong C (2009) Yang Dalei Application of artificial neural networks to strip steel surface defect diagnosis. In: 2009 Chinese Control and Decision Conference. IEEE, Guilin, China, pp 2476–2479
4. Martins LAO, Padua FLC, Almeida PEM (2010) Automatic detection of surface defects on rolled steel using Computer Vision and Artificial Neural Networks. In: IECON 2010–36th Annual Conference on IEEE Industrial Electronics Society. IEEE, Glendale, AZ, pp 1081–1086

5. Peng K, Zhang X (2009) Classification Technology for Automatic Surface Defects Detection of Steel Strip Based on Improved BP Algorithm. In: 2009 Fifth International Conference on Natural Computation. IEEE, Tianjian, China, pp 110–114
6. Boudiaf A, Benlahmidi S, Harrar K, Zaghdoudi R (2022) Classification of Surface Defects on Steel Strip Images using Convolution Neural Network and Support Vector Machine. J Fail Anal Preven 22:531–541. https://doi.org/10.1007/s11668-022-01344-6
7. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation
8. Girshick R (2015) Fast R-CNN
9. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans Pattern Anal Mach Intell 39:1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
10. Wang S, Xia X, Ye L, Yang B (2021) Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks. Metals 11:388. https://doi.org/10.3390/met11030388
11. Hou X, Liu M, Zhang S et al (2023) CANet: Contextual Information and Spatial Attention Based Network for Detecting Small Defects in Manufacturing Industry. Pattern Recogn 140:109558. https://doi.org/10.1016/j.patcog.2023.109558
12. Ren Q, Geng J, Li J (2018) Slighter Faster R-CNN for real-time detection of steel strip surface defects. 2018 Chinese Automation Congress (CAC). IEEE, Xi'an, China, pp 2173–2178
13. Shi X, Zhou S, Tai Y et al (2022) An Improved Faster R-CNN for Steel Surface Defect Detection. In: 2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP). IEEE, Shanghai, China, pp 1–5
14. He Y, Song K, Meng Q, Yan Y (2020) An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. IEEE Trans Instrum Meas 69:1493–1504. https://doi.org/10.1109/TIM.2019.2915404
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You Only Look Once. Unified, Real-Time Object Detection
16. Redmon J, Farhadi A YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and, Recognition P (2017) (CVPR). IEEE, Honolulu, HI, pp 6517–6525
17. Redmon J, Farhadi A (2018) YOLOv3: An Incremental Improvement
18. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4. Optimal Speed and Accuracy of Object Detection
19. Wang C-Y, Bochkovskiy A, Liao H-YM (2023) YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Vancouver, BC, Canada, pp 7464–7475
20. Liu W, Anguelov D, Erhan D et al (2016) SSD: Single Shot MultiBox Detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp 21–37
21. Fu C-Y, Liu W, Ranga A et al DSSD: Deconvolutional Single Shot Detector
22. Zhao C, Shu X, Yan X et al (2023) RDD-YOLO: A modified YOLO for detection of steel surface defects. Measurement 214:112776. https://doi.org/10.1016/j.measurement.2023.112776
23. Qian X, Wang X, Yang S, Lei J (2022) LFF-YOLO: A YOLO Algorithm With Lightweight Feature Fusion Network for Multi-Scale Defect Detection. IEEE Access 10:130339–130349. https://doi.org/10.1109/ACCESS.2022.3227205
24. Liu Y, Yu L, Zhang Q (2023) An Improved YOLOv5 Detection Method for Strip Surface Defect. In: 2023 28th International Conference on Automation and Computing (ICAC). pp 1–7
25. Yang N, Guo W (2022) Application of Improved YOLOv5 Model for Strip Surface Defect Detection. In: 2022 Global Reliability and Prognostics and Health Management (PHM-Yantai). pp 1–5
26. Tang L, Cai LC, Cheng K et al (2023) Improved Yolov5n strip surface defect detection algorithm. In: 2023 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS). pp 1–5
27. Yu B, Chen W, Wang W (2023) Research on Industrial Non-Destructive Testing Technology Based on Improved YOLOv5s. In: 2023 12th International Conference of Information and Communication Technology (ICTech). pp 435–440
28. Chen H, Du Y, Fu Y et al (2023) DCAM-Net: A Rapid Detection Network for Strip Steel Surface Defects Based on Deformable Convolution and Attention Mechanism. IEEE Trans Instrum Meas 72:1–12. https://doi.org/10.1109/TIM.2023.3238698

29. Selvaraju RR, Cogswell M, Das A et al (2020) Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. Int J Comput Vis 128:336–359. https://doi.org/10.1007/s11263-019-01228-7
30. Hu J, Shen L, Albanie S et al (2019) Squeeze-and-Excitation Networks
31. Woo S, Park J, Lee J-Y, Kweon IS (2018) CBAM: Convolutional Block Attention Module. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer Vision – ECCV 2018. Springer International Publishing, Cham, pp 3–19
32. Wang Q, Wu B, Zhu P et al (2020) ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA, pp 11531–11539
33. Krig S (2016) Image Pre-Processing. In: Krig S (ed) Computer Vision Metrics: Textbook Edition. Springer International Publishing, Cham, pp 35–74
34. Demant C, Garnica C, Streicher-Abel B (2013) Overview: Image Preprocessing. In: Demant C, Streicher-Abel B, Garnica C (eds) Industrial Image Processing: Visual Quality Control in Manufacturing. Springer, Berlin, Heidelberg, pp 25–63
35. Ouyang D, He S, Zhang G et al (2023) Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In: ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Rhodes Island, Greece, pp 1–5
36. Dai X, Chen Y, Xiao B et al (2021) Dynamic Head. Unifying Object Detection Heads with Attentions
37. He Y, Song K, Meng Q, Yan Y (2020) An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. IEEE Trans Instrum Meas 69:1493–1504. https://doi.org/10.1109/TIM.2019.2915404
38. Lv X, Duan F, Jiang J et al (2020) Deep Metallic Surface Defect Detection: The New Benchmark and Detection Network. Sensors 20:1562. https://doi.org/10.3390/s20061562
39. Huang Y, Qiu C, Guo Y et al Surface Defect Saliency of Magnetic Tile
40. Li C, Li L, Jiang H et al (2022) YOLOv6: A Single-. Stage Object Detection Framework for Industrial Applications
41. GitHub - ultralytics/yolov5 YOLOv5  in PyTorch > ONNX > CoreML > TFLite. https://github.com/ultralytics/yolov5. Accessed 13 Mar 2024
42. Ge Z, Liu S, Wang F et al (2021) YOLOX: Exceeding YOLO Series in 2021

## Figures

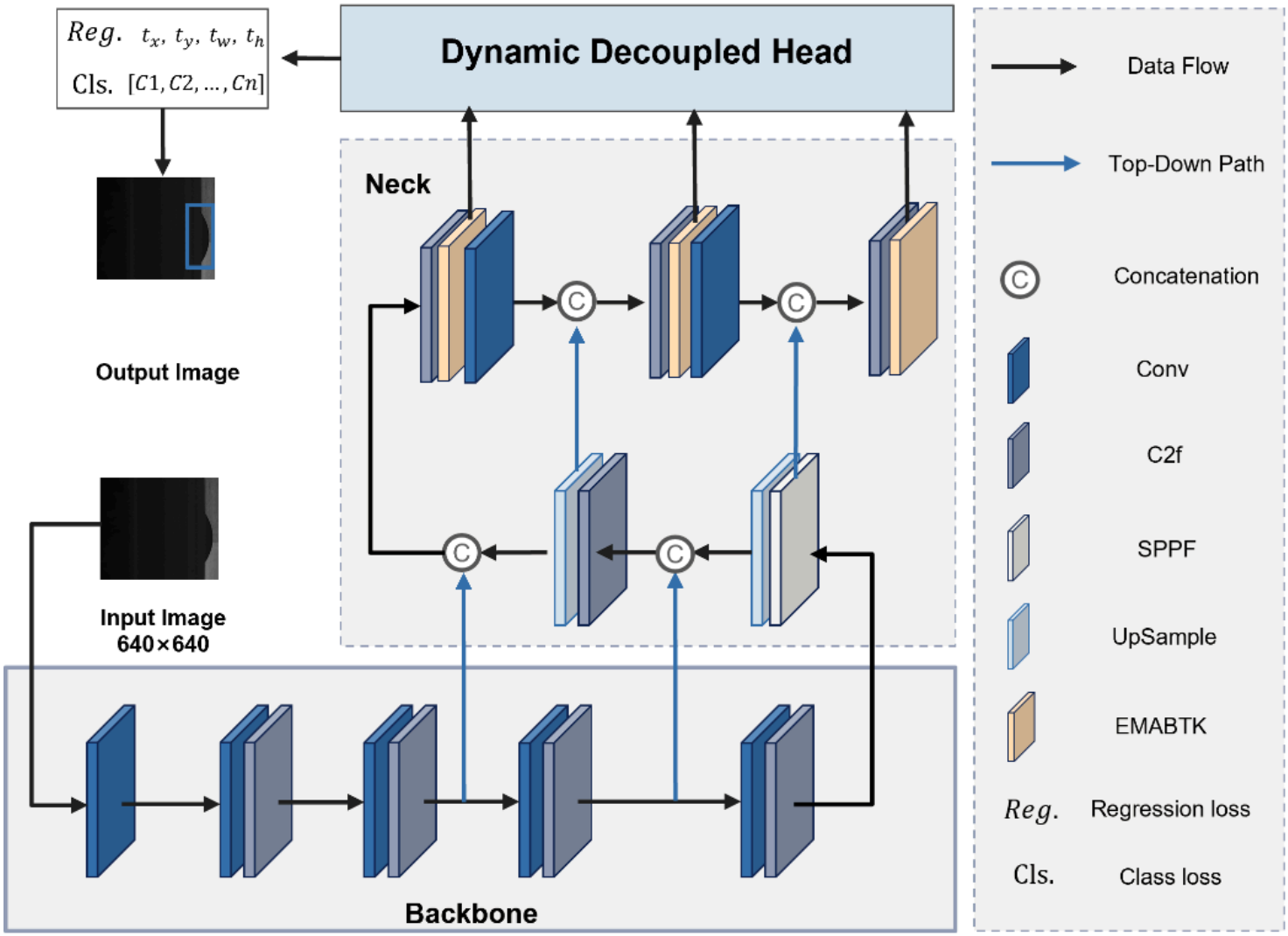



Figure 1

Overall network structure of YOLOv8-FCS

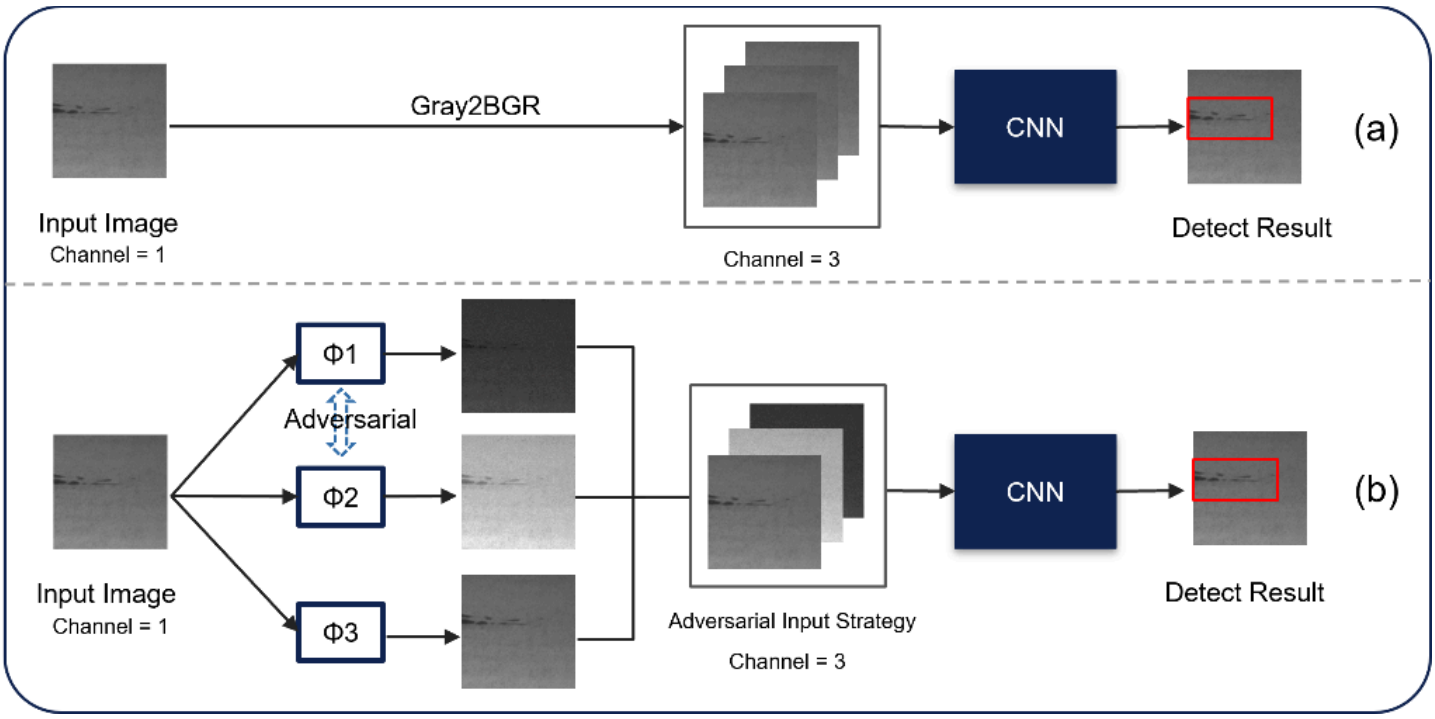



**Figure 2**

Comparison between single-channel image adversarial input and ordinary grayscale image input

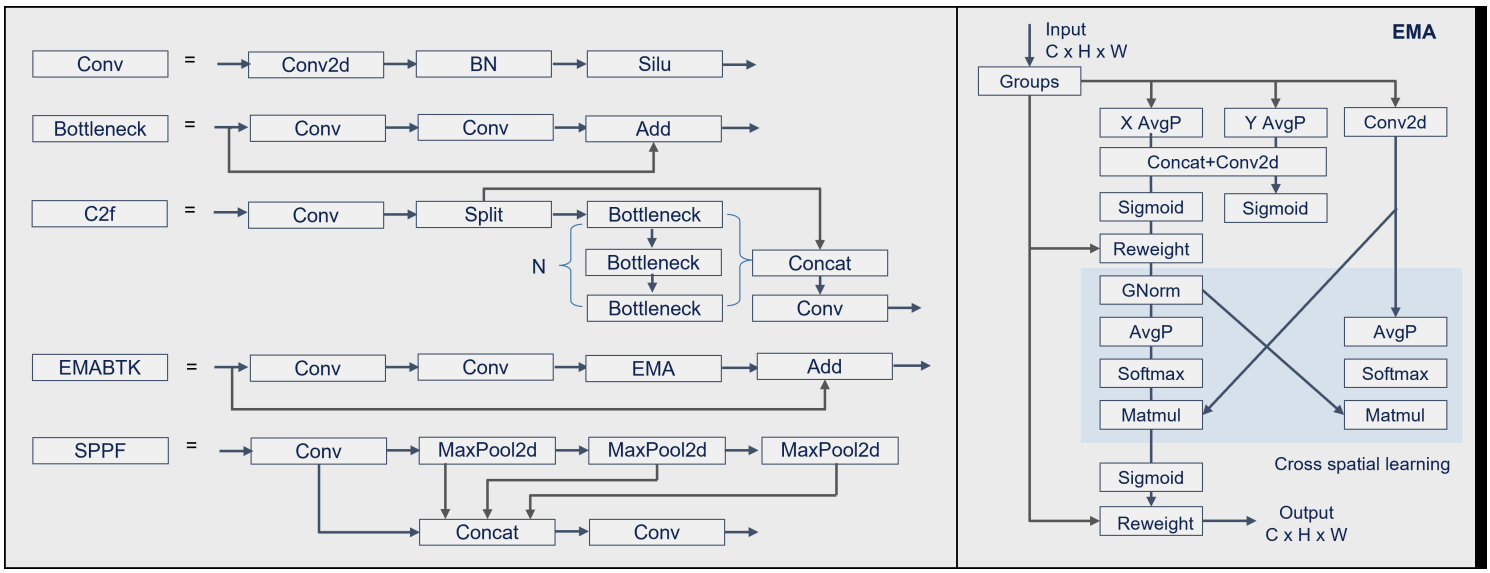



**Figure 3**

Structure of each module of YOLOv8-FCS

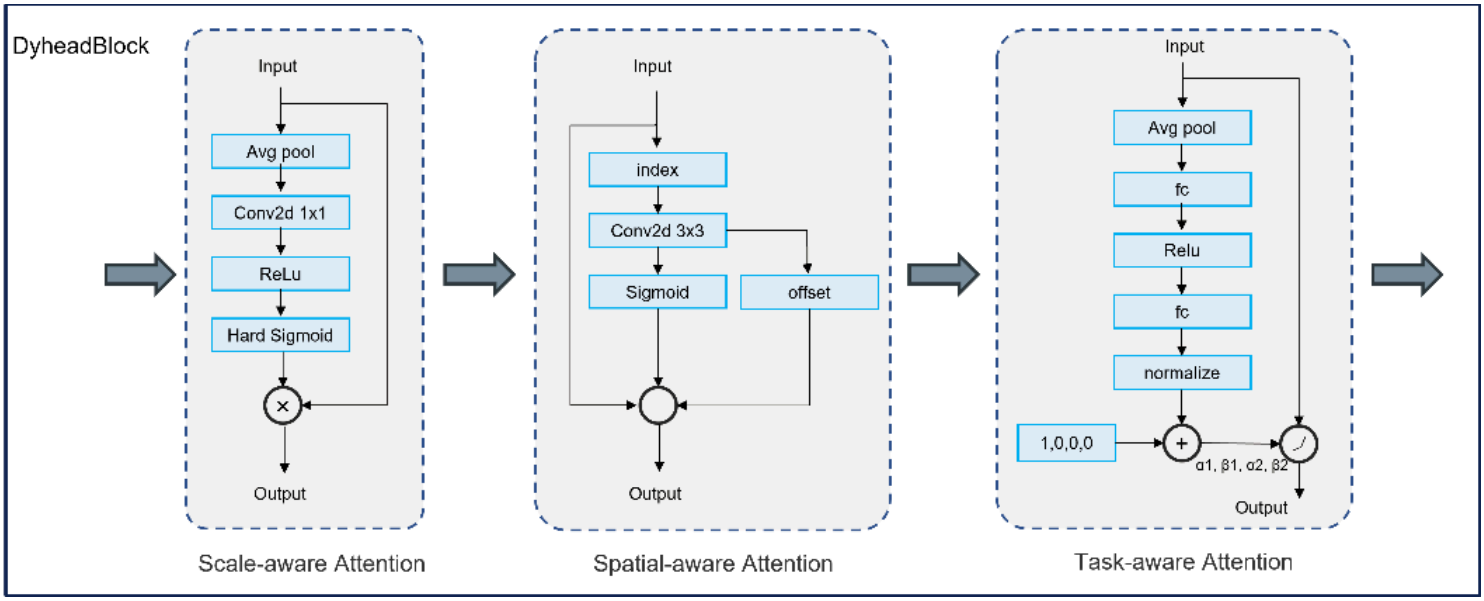



**Figure 4**

Structure diagram of the DyHeadBlock module

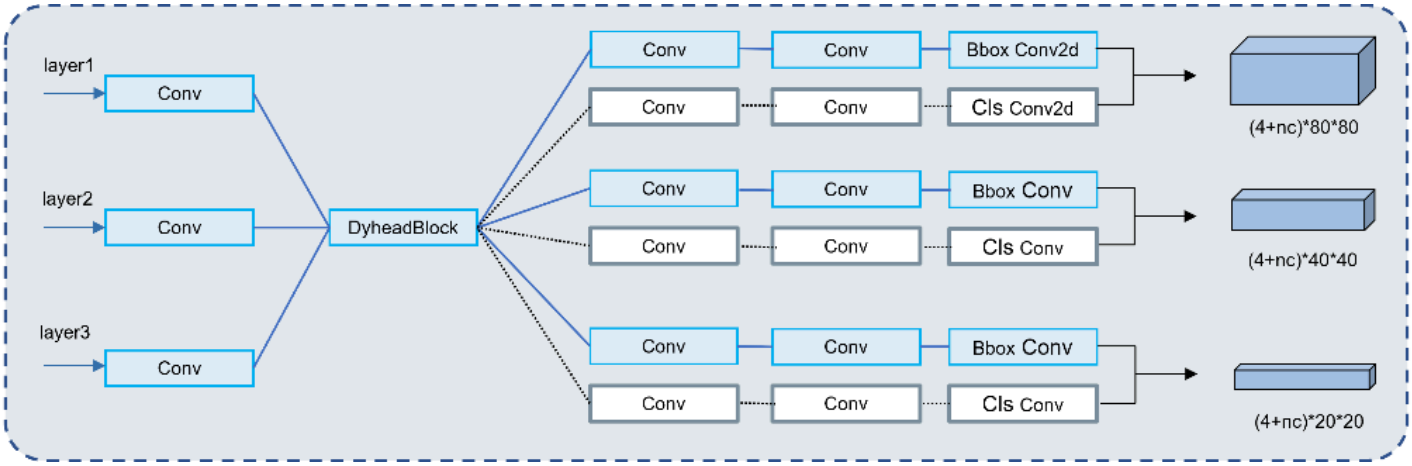
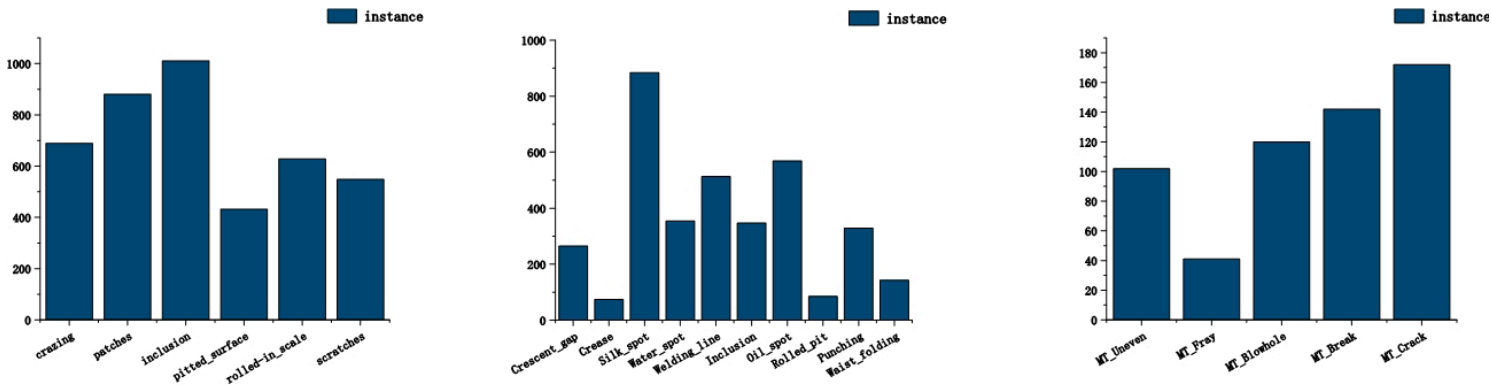


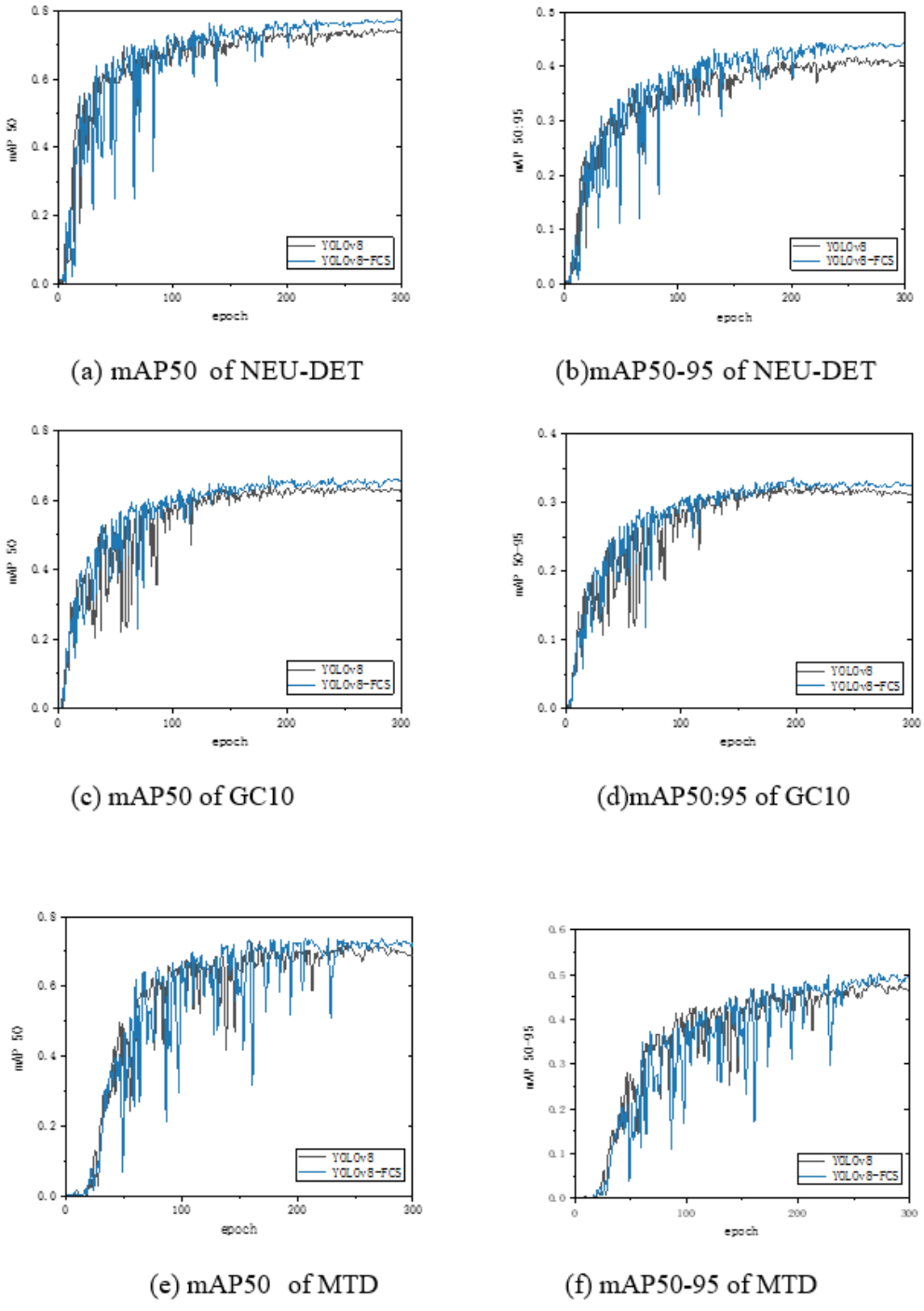

**Figure 5**

Dynamic Decoupled Head



## Figure 6

Distribution of categories in the three datasets




## Figure 7

The comparisons of mAP curve on three datasets

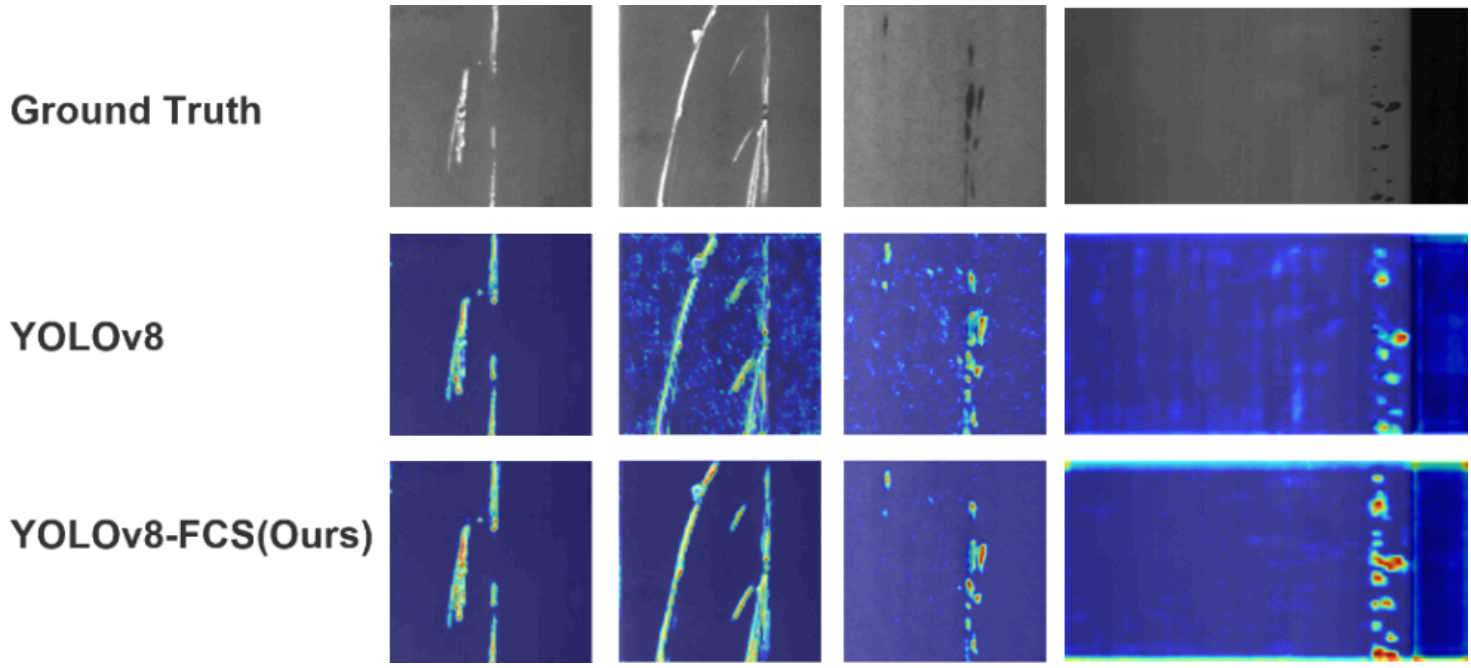



**Figure 8**

Heatmap visualization of detection results

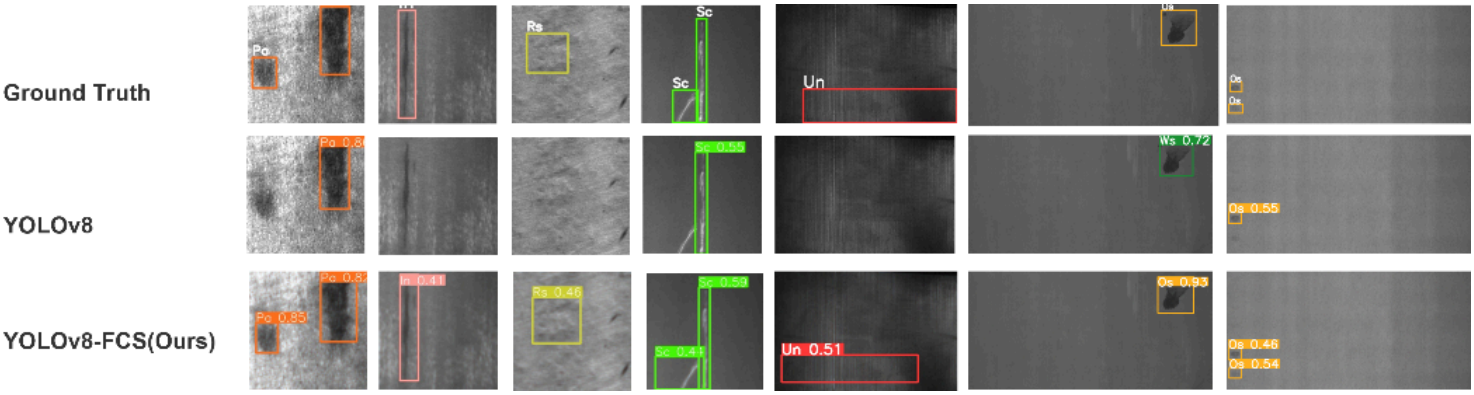



**Figure 9**

Comparison of partial detection results among different models

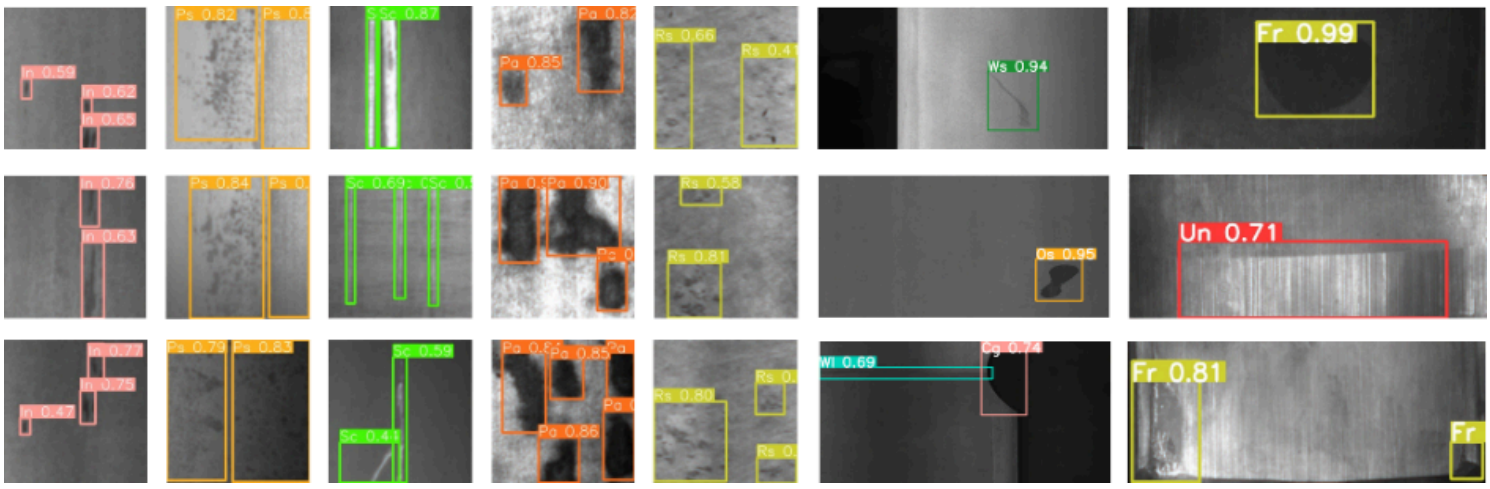


**Figure 10**

Partial detection results of YOLOv8-FCS on three datasets

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Highlights.docx