processes

MDPI

*Article*

# Flying Steel Detection in Wire Rod Production Based on Improved You Only Look Once v8

**Yifan Lu [1], Fei Zhang [1,*], Xiaozhan Li [1], Jian Zhang [1], Xiong Xiao [1], Lijun Wang [2] and Xiaofei Xiang [1]**

[1] National Engineering Research Center for Advanced Rolling and Intelligent Manufacturing, University of Science and Technology Beijing, Beijing 100083, China; luyifan200010@163.com (Y.L.); lixiaozhan@nercar.ustb.edu.cn (X.L.); m202321365@xs.ustb.edu.cn (J.Z.); docxxiao@ustb.edu.cn (X.X.); xiangxf@nercar.ustb.edu.cn (X.X.)

[2] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China; wanglj@ustb.edu.cn

* Correspondence: zhangfei@ustb.edu.cn

## Abstract

In the process of high-speed wire rod production, flying steel accidents may occur due to various reasons. Current detection methods relying on sensors like hardware make debugging complex as well as limit real-time and accuracy. These methods are complicated to debug, and the real-time and accuracy of detection are poor. Therefore, this paper proposes a flying steel detection method based on improved You Only Look Once v8 (YOLOv8), which can realize high-precision flying steel detection based on machine vision through the monitoring video of the production site. Firstly, the Omni-dimensional Dynamic Convolution (ODConv) is added to the backbone network to improve the feature extraction ability of the input image. Then, a lightweight C2f-PCCA_RVB module is proposed to be integrated into the neck network, so as to carry out the lightweight design of the neck network. Finally, the Efficient Multi-Scale Attention (EMA) module is added to the neck network to fuse the context information of different scales and improve the feature extraction ability. The experimental results show that the average accuracy (mAP@0.5) of the flying steel detection method based on the improved YOLOv8 is 99.1%, and the latency is reduced to 2.5 ms, which can realize the real-time accurate detection of the flying steel.

**Keywords:** object detection; YOLOv8; flying steel accident; safety monitoring; deep learning

check for updates

Academic Editor: Changsoo Je

Received: 30 April 2025
Revised: 2 July 2025
Accepted: 16 July 2025
Published: 18 July 2025

**Citation:** Lu, Y.; Zhang, F.; Li, X.; Zhang, J.; Xiao, X.; Wang, L.; Xiang, X. Flying Steel Detection in Wire Rod Production Based on Improved You Only Look Once v8. *Processes* **2025**, *13*, 2297. https://doi.org/10.3390/pr13072297

**Copyright:** © 2025 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

As a common product in steel production, the wire rod is mostly produced by single-line, high-speed rolling [1]. Flying steel accidents are a major hidden danger, threatening the safety of wire rod rolling, especially during the high-speed rolling process of small-sized wire rod products where flying steel accidents occur frequently, seriously affecting production efficiency and operational safety [2]. The evolution mechanism of the flying steel accident presents the following characteristics: When the wire rod accidentally breaks free from the roll gap constraint, its front end hits the obstacle and stops suddenly, while the subsequent wire rod still maintains a high-speed feed state due to the continuous action of the rolling force. Under such circumstances of uneven force distribution in the front and back, the wires in a high-temperature state will fly out of control in large areas from the rolling track, thus causing a flying steel accident. There are many causes of flying steel accidents, such as poor contact between the roll and billet, deviation of the rolling line, fluctuations in billet size parameters, improper setting of rolling mill parameters, and so

on [3,4]. If the flying steel accident is not detected and dealt with in time, it may damage the equipment, lead to production suspension, and even cause personal injury [5].

However, the existing detection methods for production faults such as flying steel mainly rely on rolling signals, thermal detection component signals, and manual observation. de Sena et al. proposed a strategy to automatically diagnose gearbox faults using the current signal of the induction motor [6]. Tang et al. used a hybrid fault diagnosis method based on expert experience and data-driven methods to detect equipment faults in high-speed wire finishing mills by installing wireless vibration sensors at the production site [7]. Shi et al. installed multi-source sensors on the rolling mill to collect data and proposed a new deep learning method based on improved one-dimensional convolutional neural network and improved two-dimensional convolutional neural network for real-time health monitoring of the rolling mill [8]. Yue et al. leveraged multi-sensor information fusion and enhanced diagnostic methods based on Deep Belief Networks to monitor the health status of rolling mills under limited datasets [9].

The fault detection method that relies on rolling signals and thermal component signals necessitates the halting of production for the automated reconfiguration of the rolling line, leading to extended shutdowns for debugging, installation, and maintenance purposes. Meanwhile, manual observation methods suffer from untimely detection and delayed response. Current research on detecting flying steel with machine vision technology is limited, and the accuracy of these detections is still inadequate. The random and unpredictable shape of flying steel introduces additional complexities to its detection, and enhancing the accuracy and real-time capabilities of this detection remains a significant challenge in the realm of flying steel detection.

You Only Look Once (YOLO), an object detection algorithm pioneered by Redmon et al. in 2016, stands out for its rapid detection speed, making it ideal for real-time applications [10]. The anchor mechanism enables YOLO to detect objects across multiple scales of feature maps simultaneously, thereby improving the algorithm's ability to adapt to objects of different sizes [11]. The detection accuracy is higher than the traditional region-based object detection algorithm. At present, YOLO is mostly used in the detection of steel surface defects in the process of steel production monitoring. Xie et al. proposed C2f_LMSMC, which combines lightweight, multi-scale, mixed convolution LMSMC and efficient global attention mechanism to improve YOLOv8 for steel surface defect detection [12]. Wang et al. embedded location information into channel attention by adding a coordinated attention mechanism CA to the backbone network, which solved the problem of location information loss caused by global pooling, and they used a decoupled head detector to improve the accuracy of steel surface defect detection [13]. Wang et al. proposed a regression loss function that accounts for aspect ratio and scale, and they incorporated a focusing loss to improve the YOLO model [14]. This approach further addresses the issue of sample imbalance in the steel pipe defect dataset and significantly enhances the accuracy of steel pipe defect detection.

However, the above method cannot meet the requirements of high accuracy and real-time performance of flying steel detection at the same time. Therefore, this paper introduces an online detection approach for flying steel utilizing YOLOv8, which addresses the limitations of conventional detection techniques, enables swift and precise detection, and improves the efficiency of emergency handling, thereby reducing production losses and ensuring production safety. The main contributions of this paper are as follows:

(1) In order to improve the accuracy of flying steel, the ODConv is added to the YOLOv8 model to enhance the feature extraction ability of the model to the input data and efficiently capture the feature representations of the wire rod.

(2) Improve the network lightweight module C2f-PCCA_RVB to lighten the neck network and improve the detection speed of the model.
(3) The EMA module is added to the neck network, so that the model integrates global context information in the feature extraction process, thereby improving the feature extraction ability and detection accuracy of the model.

## 2. Theoretical Foundations

### *2.1. YOLOv8*

YOLOv8 is a version of the YOLO series proposed in 2023, which has made many improvements and innovations based on YOLOv5 [15]. YOLOv8 introduces a new backbone network, detection head, and loss function, thereby improving performance and flexibility. Its network structure is shown in Figure 1. In addition, it also supports multiple tasks such as image classification, object detection, and instance segmentation, and has achieved significant performance improvements on the COCO Val 2017 dataset [16]. However, compared with YOLOv5, some models of YOLOv8 have increased in the number of parameters and calculations, which may lead to slower inference speed.

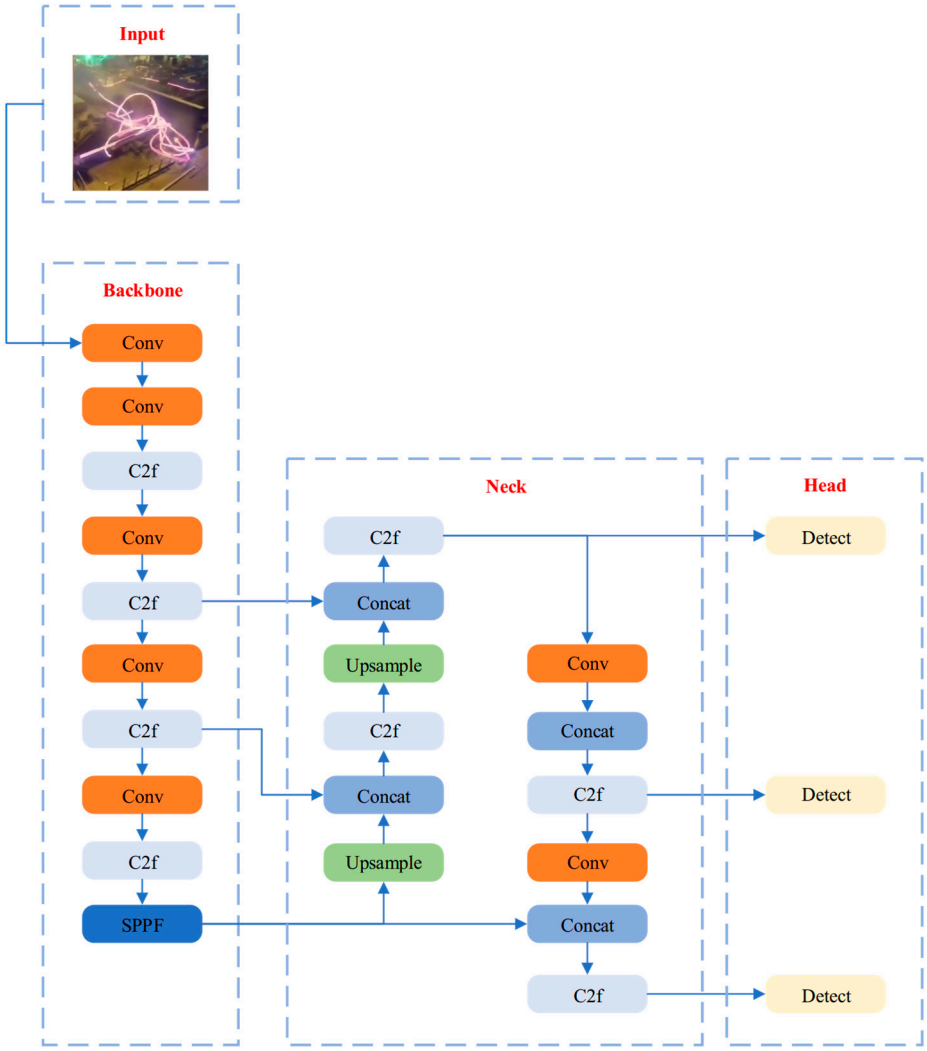



**Figure 1.** YOLOv8 network structure diagram [17].

Backbone: The structure includes standard convolutions (Conv), CSPDarknet53 to 2-Stage FPN (C2f), and Spatial Pyramid Pooling (SPPF), which are responsible for feature extraction. Residual connections and bottleneck structures are used to reduce the size

of the network and improve performance. YOLOv8 uses the C2f module with richer gradient flow as the basic component unit in the Backbone part. Relative to the C3 module in YOLOv5, the C2f module boasts reduced parameter counts while enhancing feature extraction performance.

Neck: Mainly responsible for feature fusion and augmentation, in order to better detect objects of different scales. Path Aggregation Network (PAN) modules are used to aggregate the paths of different levels of features to enhance the ability of feature expression.

Head: Responsible for forecasting the location, dimensions, and classification of the bounding box. YOLOv8 introduces a novel detection head architecture known as a decoupled head, which effectively segregates the classification process from the detection head. The anchor base mechanism is changed to an anchor-free mechanism, which reduces the super parameter setting of the anchor frame and simplifies the training process.

Other improvements include the following:

(1) Adaptive NMS: An adaptive threshold to reduce missed detection and false detection.
(2) Automatic mixing accuracy training: Speeds up the training speed and reduces memory usage.

### 2.2. RepViTBlock

By incorporating a multi-head, self-attention module, the lightweight vision Transformer (ViTs) demonstrates outstanding global modeling capability and performance. However, it exhibits limited feature extraction ability at the image pixel level and requires substantial computational resources [18]. On the other hand, Convolutional Neural Network (CNN) models effectively capture local features in images through convolution operations while reducing parameter count through parameter sharing. They excel in tasks such as image recognition and object detection. Nonetheless, CNN also presents certain limitations including a lack of global information perception and loss of input data location information due to the use of convolution operations. To address their respective limitations, a growing number of researchers are delving into hybrid models of CNN and Transformer to take advantage of the strengths of both architectures [19].

Tsinghua scholar Wang et al. gradually improved the standard lightweight CNNs by integrating the efficient architecture design of lightweight ViTs, resulting in a new pure lightweight model called RepViT [20]. By modifying the block structure of MobileNetV3-L, they separated the token mixer and channel mixer to obtain the block structure RepViTBlock of lightweight ViTs. RepViTBlock takes into consideration both latency on mobile devices and top-1 accuracy on ImageNet, leading to lower latency and higher performance. They introduced a multi-branch topology for deep convolution during training using structural reparameterization to enhance performance. During the inference phase, it is possible to consolidate the deep convolutional multi-branch architecture into a single-branch configuration. This integration helps to mitigate the extra computational and memory expenses that arise from the presence of multiple branches.

### 2.3. PConv

In recent years, numerous researchers have utilized Depthwise Convolution (DWConv) in applying it to neural network models for lightweighting purposes [21–23]. DWConv breaks down standard convolutions into deep convolutions and point convolutions using a $1 \times 1$ kernel size [24]. This approach effectively minimizes both the number of parameters and the computational burden in the models. Nevertheless, during efforts to decrease floating point operations (FLOPs), DWConv and GSConv frequently encounter heightened memory access-related side effects. In contrast, Partial Convolution (PConv) exclusively conducts its operation on specific segments within input feature maps instead of em-

ploying traditional full-scale convolutions [25]. Therefore, PConv reduces computational redundancy and minimizes memory access compared with DWConv or GSConv.

### *2.4. CA*

In recent years, the attention mechanism module has been widely adopted in computer vision to enhance feature extraction and improve neural network performance. Hu et al. introduced the Squeeze-and-Excitation (SE) module, comprising two components: Squeeze, which involves global pooling to compress features; and Excitation, which utilizes a two-layer, fully connected structure to derive channel weights in the feature map for input into subsequent network layers [26]. While the SE module considers inter-channel dependencies, it overlooks spatial correlation and performs suboptimally with a small number of channels. Woo et al. proposed the Convolutional Block Attention Module (CBAM), which addresses both channel and spatial information but entails high computational complexity and increased computing resources [27].

In light of the aforementioned limitations, Hou et al. introduced a novel Attention mechanism for mobile networks, namely Coordinate Attention (CA), which incorporates location information into channel attention [28]. The CA module is tailored to augment the expressive capacity of learning features in mobile networks and can produce a tensor of the same size after transforming any intermediate feature tensor in the network.

By incorporating coordinate data and creating attention based on coordinates, CA is capable of acquiring information across different channels, as well as details that are sensitive to direction and position. Lightweight and flexible CA can be seamlessly integrated into other modules to enhance feature representation. Figure 2 shows the structure comparison of SE and CA.

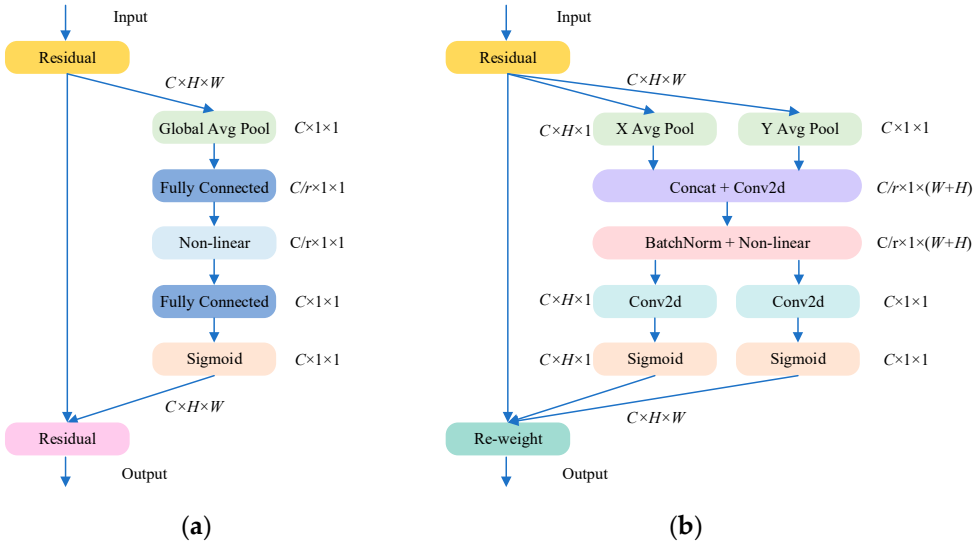



**Figure 2.** Comparison of network structure between SE and CA. (**a**) SE; (**b**) CA.

## 3. Flying Steel Detection Model

To enhance the speed and accuracy of flying steel inspection, we have made some improvements to YOLOv8. In the input position of the YOLOv8 model backbone, the ODConv is added to replace the original ordinary convolution, which increases the input dependence for the model input and enhances the feature extraction ability of the model for the input data. The improved network lightweight module C2f-PCCA_RVB is used to replace the original C2f structure of the neck network, and the neck network is lightweight in design to improve the detection speed of the model. Finally, an EMA module is added to the neck network to fuse the global context information, thereby improving the feature

extraction ability and detection accuracy of the model. The overall model structure is shown in Figure 3. The specific algorithm principle and improvement process are described in the subsequent sections.

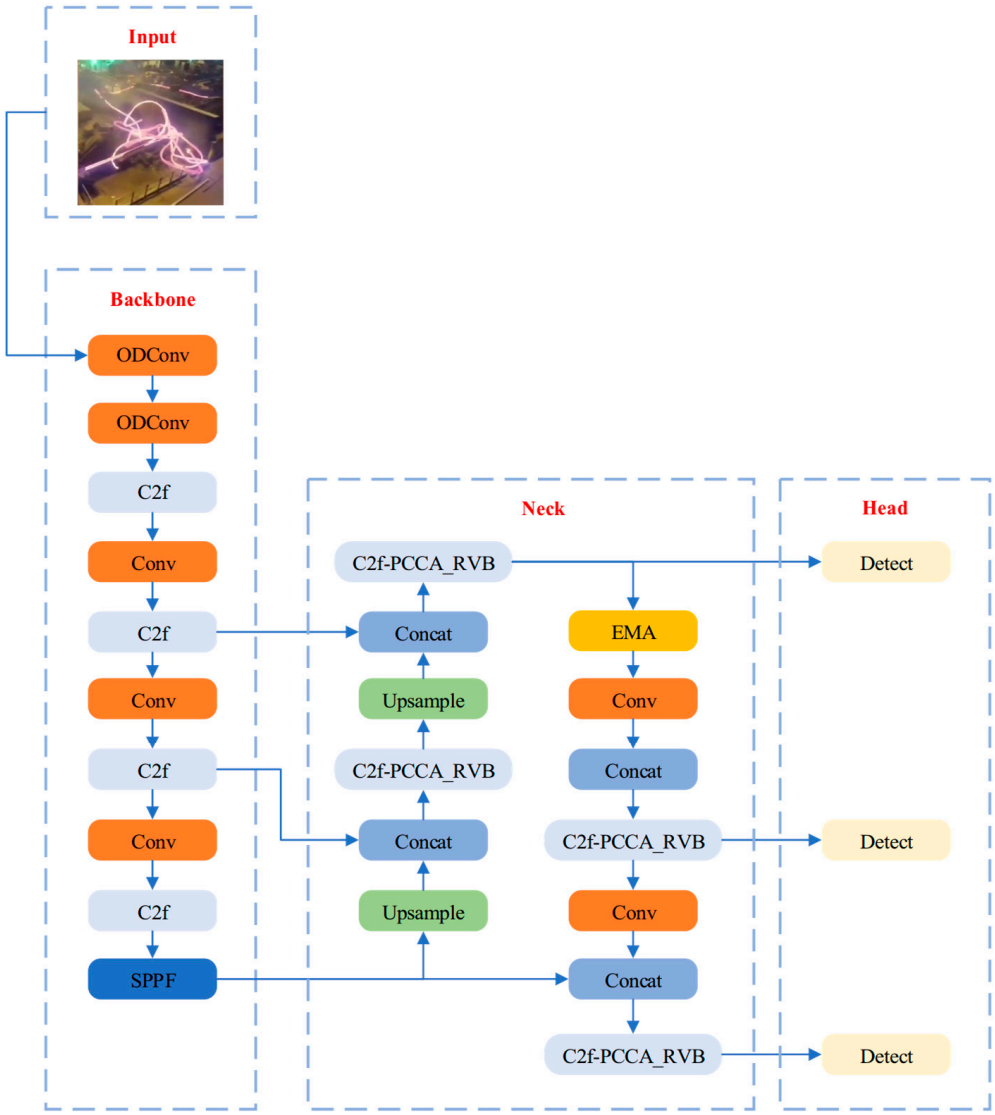



**Figure 3.** Overall structure of OEC-YOLOv8 [17].

### 3.1. ODConv

In the YOLOv8 training process, the image data is initially processed by Conv to extract features. However, the traditional convolution is constrained by a single static kernel, which runs independently of the input samples, resulting in limited feature extraction capabilities. Traditional dynamic convolution only assigns kernel dynamic attributes based on the number of kernels, but it ignores three key dimensions—the space size of each kernel, the number of input channels, and the number of output channels [29]. In contrast, ODConv uses a novel multi-dimensional attention mechanism and parallel strategy to obtain comprehensive knowledge of all four dimensions in the kernel space of any given layer [30]. This enables ODConv to dynamically adjust its feature extraction method according to the specific features in the incoming data, emphasizing relevant information while suppressing irrelevant details to produce more effective representation. Therefore, this paper uses ODConv instead of the original two standard convolutions at the input position of YOLOv8.

### *3.2. C2f-PCCA_RVB*

The C2f module, a vital constituent of YOLOv8, is employed within both the backbone and neck architectures. Its implementation facilitates the creation of lightweight models while simultaneously enabling the extraction of more comprehensive gradient flow information. The C2f module bifurcates the input data, with one branch directly transforming input features into output through convolutional layers. This operation aids in extracting features of varying levels and abstraction within the input data. Meanwhile, the other branch undergoes processing by multiple Bottleneck modules, enhancing network nonlinearity and representation capabilities for complex data modeling. In the end, the integration of features is realized by merging the features from various branches along the channel axis, thereby enhancing the comprehensive representation of features.

The RepViTBlock is a new lightweight module with higher accuracy and detection efficiency. Its structure mainly includes DWConv, channel attention mechanism module SE, and Conv. In order to solve the limitations of the DWConv and SE modules, we use the PConv and CA modules to further improve RepViTBlock and propose PCCA_RVB. In order to lighten the neck network, the C2f module is lightened. The C2f module is improved by using PCCA_RVB, and C2f-PCCA_RVB is proposed and applied to the neck network of the model. Figure 4 shows the structure of RepViTBlock versus PCCA_RVB. Figure 5 shows the structure comparison of C2f and C2F-PCCA_RVB.

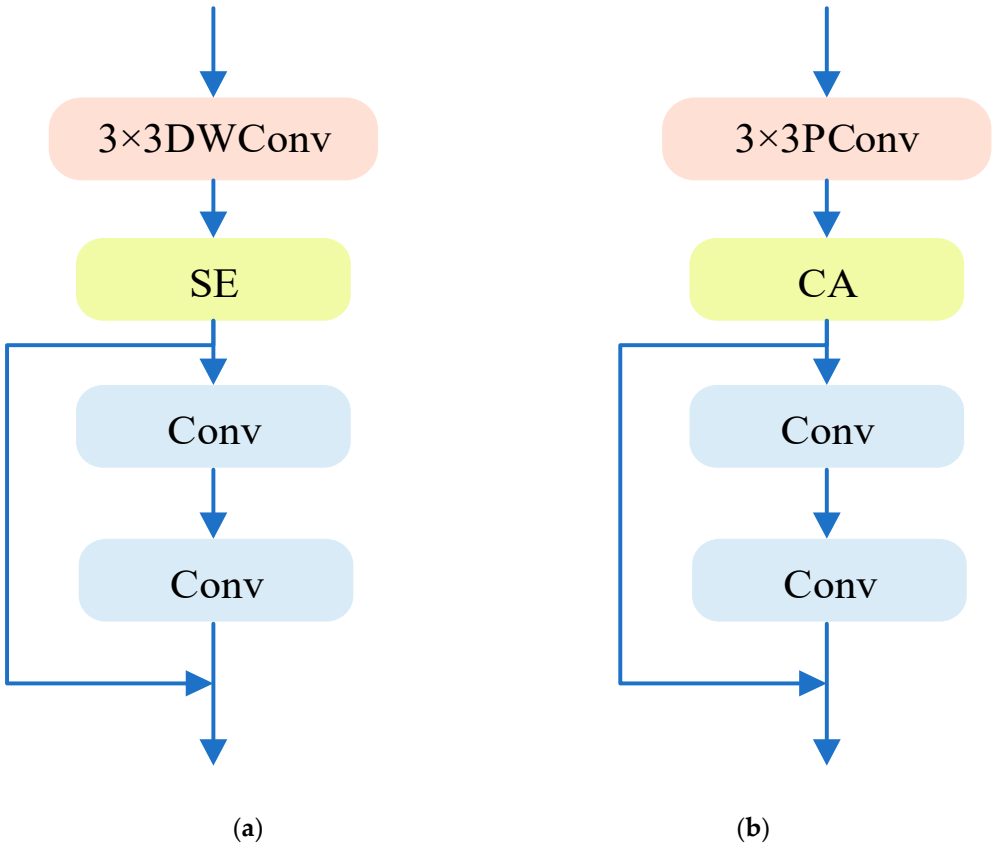



**Figure 4.** Comparison of network structure between RepViTBlock and PCCA_RVB. (**a**) RepViTBlock; (**b**) PCCA_RVB.

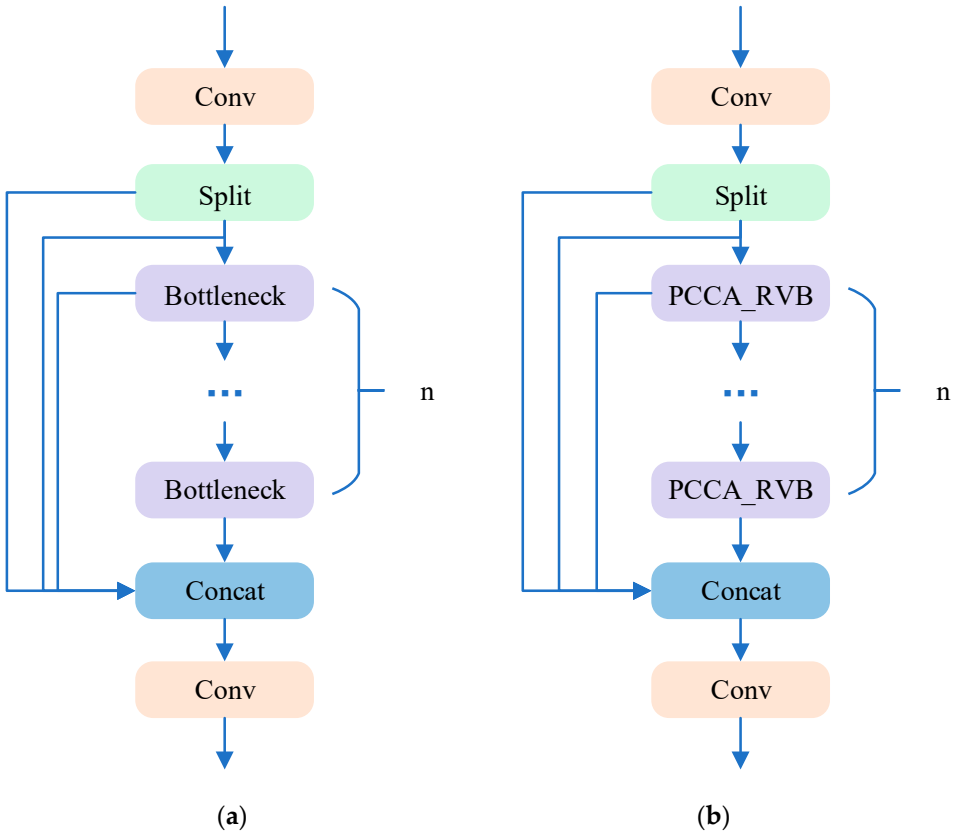



**Figure 5.** Comparison of network structure between C2f and C2f-PCCA_RVB. (**a**) C2f; (**b**) C2f-PCCA_RVB.

### *3.3. EMA*

EMA is an attentional mechanism module based on the cross-space learning method, aimed at reducing computational overhead while preserving channel-specific information [31]. EMA uses convolutional nuclei of different sizes to collect multi-scale spatial information at the same processing stage to improve the feature expression capability of the network. The parallelization structure is used to extract the attention weight of the group feature map, retain the information on each channel, and reduce the computational overhead. By using a cross-spatial learning approach, precise location information is embedded into EMA to deal with short- and long-term dependencies and integrate global context information [32].

By introducing EMA into YOLOv8's neck network, the network is able to generate better pixel-level attention to advanced feature maps. The EMA module is placed after the neck network C2f module, and the effectiveness of this improvement is proved by ablation experiments.

## 4. Experiments

### *4.1. Datasets and Preprocessing*

In the research on flying steel detection, the construction of an experimental dataset serves as the foundation for accomplishing the object detection task. Given that flying steel accidents are low-probability events, the image data directly collected from the production site is inherently limited. Therefore, the data sources are divided into two directions:

1. Sampling from production site monitoring videos: Image data is extracted by systematically sampling frames from monitoring videos recorded at the production site.

2. Internet-based data collection: Flying steel data is gathered from online resources to expand the diversity of the dataset.

This dual-source data collection strategy is designed to ensure the richness and representativeness of the dataset, while enhancing the model's adaptability to varied scenarios. Sample image data for flying steel detection is illustrated in Figure 6.



**Figure 6.** Sample image data of flying steel detection.

Labelimg is a graphical image annotation tool that is widely used in computer vision tasks. The images were labeled using Labelimg, with the labeling categories being "flying_steel" and "normal_steel." The two categories are, respectively, the target wire objects of the flying steel accident and the wire objects under normal working conditions. An example of the image labeling diagram is shown in Figure 7. After the annotation is completed, the annotated data can be exported in a universal format. The annotations in this paper are exported in the PASCAL VOC format. Subsequently, in accordance with the model's data specifications, the dataset is partitioned into a training set and a validation set at a ratio of 4:1.

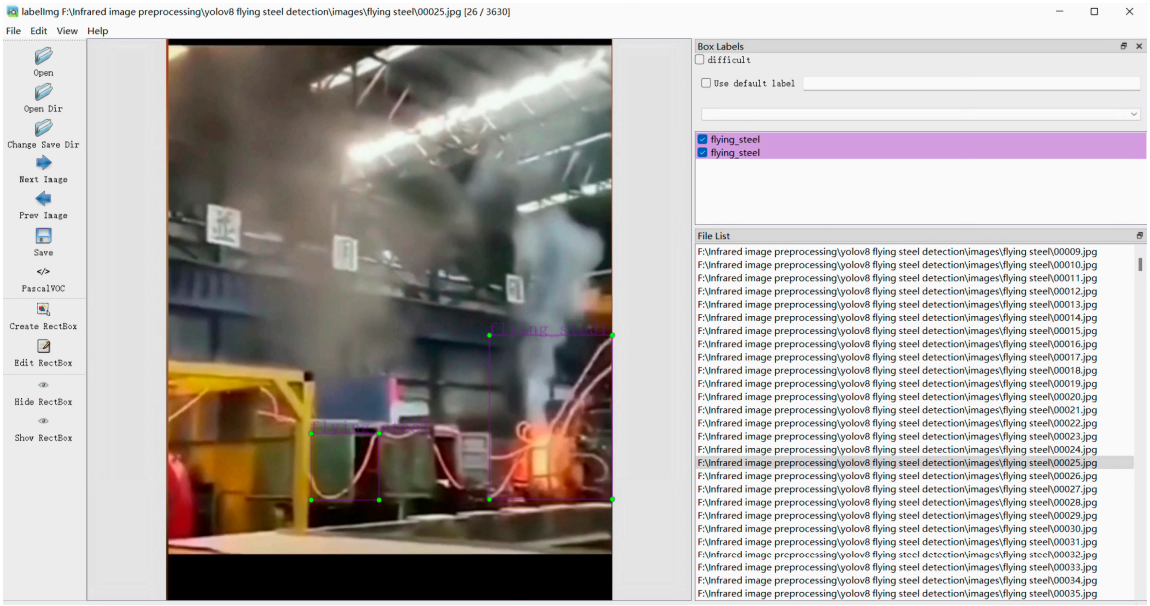



**Figure 7.** Sample image of image data annotation.

Image data cleaning and data preprocessing are important steps in computer vision tasks, which directly affect the model performance. Through data cleaning, damaged, unclear, or poor-quality images are removed to ensure that each image is not only relevant to flying steel but also meets the quality standards required for model training. A total of 2021 bar and wire images have been collected in this dataset. The number of labels in the specific dataset is shown in Table 1. The strong, random shape features of flying steel bring difficulties to model feature learning, so we collect more flying steel data so that the model can fully learn and identify the characteristics of flying steel. The validity of the flying steel dataset will be reflected by the verification results of the model.

**Table 1.** Number of labels.

| Label Class | Number of Labels |
|---|---|
| Flying_steel | 1692 |
| Normal_steel | 842 |

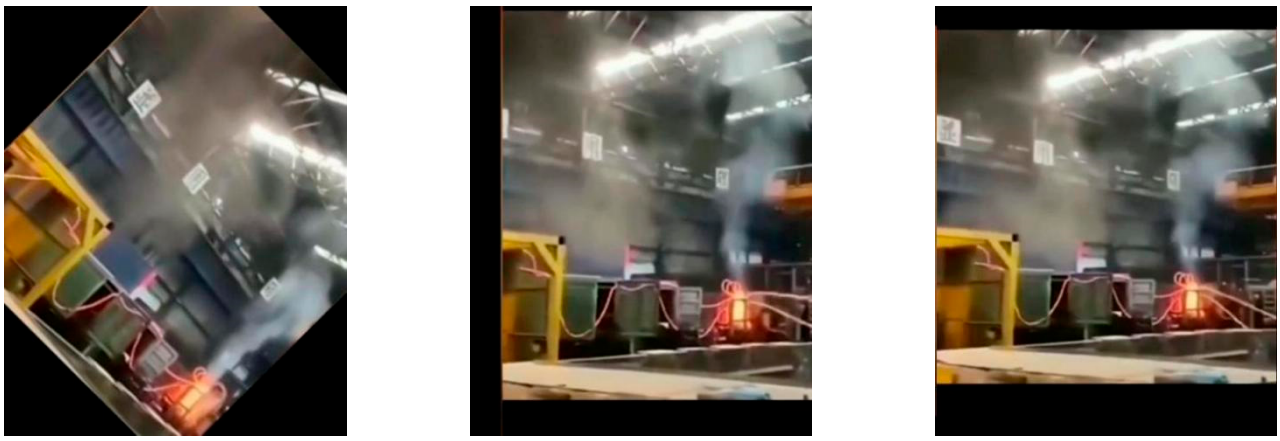
Due to the limited data sources and the limitation of data collection, data augmentation was carried out to improve the overall quality of the dataset. Data augmentation methods include the following:

Random rotation: Rotation range: $[-20^\circ, 20^\circ]$.

Random translation: Shift range: $[-80, 80]$.

Random stretch: Stretch range: $[0.5, 1.5]$.

The image after preprocessing is shown in Figure 8. It should be noted that the detection dataset for this flying steel includes image samples that are affected by environmental factors such as water vapor and occlusion. These disturbances in the image samples can enhance the reliability of the detection results for flying steel.



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

**Figure 8.** Image geometry transformation example diagram. (**a**) Random rotation; (**b**) Random translation; (**c**) Random stretch.

### *4.2. Experimental Parameter Setting and Model Training*

The YOLOv8 model is available in multiple versions. YOLOv8n is the most compact version [15]. It has fewer parameters and faster detection speed, making it more suitable for real-time detection of flying steel. While other YOLOv8 variants may outperform YOLOv8n in terms of overall performance, their detection speed lags behind. The hyperparameter settings of the specific model can be found in Table 2.

In order to ensure the accuracy of the experiment and the consistency of the experimental conditions, the experiment was tested on the same server, and the specific environmental

information is shown in Table 3. In this experiment, the YOLO algorithm employs a fixed-size image of 640 × 640 pixels as input, with the training process set to 300 epochs. Last.pt and best.pt are saved during YOLOv8 model training. If the training is interrupted, the previous training can be continued through last.pt. By continuously saving best.pt, the model over-fitting can be avoided, so the larger epochs set in this experiment ensure that the final result of the training can accurately reflect the model performance.

**Table 2.** Hyperparameter settings of network training [17,33].

| Parameter | Settings |
|---|---|
| Image size | 640 × 640 |
| Initial learning rate | 0.01 |
| Final learning rate | 0.01 |
| Batch size | 4 |
| Epoch | 300 |
| Momentum | 0.937 |
| Weight_decay | 0.0005 |
| Warmup_epochs | 3.0 |
| Warmup_momentum | 0.8 |
| Warmup_bias_lr | 0.1 |

**Table 3.** Environment configuration information for model training.

| Name | Configuration Information |
|---|---|
| Operating System | Windows 11 |
| Development Language | Python 3.8.5 |
| Framework | Pytorch 2.0.0 + CUDA11.8 |
| GPU | NVIDIA GeForce RTX 3060 (12 GB) |
| CPU | Intel 12th Gen Core i7-1360P |
| Memory Size | 16 GB |

## 5. Experimental Results and Analysis

### *5.1. Training Results and Analysis*

Figure 9 illustrates the contrast in loss function trajectories between the standard YOLOv8n method (depicted in blue) and the enhanced algorithm introduced in this study (shown in orange). According to the comparison of loss curves, it was found that the improved YOLOv8n converges faster than the original model, and the final loss value is smaller. The improved model achieved lower loss values in a shorter time, which reduced training time and improved training efficiency. The decrease in loss value indicates an improvement in the predictive ability of the improved model, which can better fit the training data and make more accurate predictions on new data.

Figure 10 shows the PR curve of the model in this paper. The PR curve reflects the relationship between the accuracy and recall rate of the model at different thresholds. Precision measures the proportion of samples that the model predicts as positive examples but are actually positive examples. Recall measures the proportion of samples that are predicted as positive examples by the model among all true examples. The PR curve of this model is close to the upper right corner of the coordinate graph, which indicates that the model has both a high recall rate and high accuracy. This suggests that when identifying flying_steel, the model can find all flying_steel samples more accurately and avoid misjudging normal_steel samples as flying_steel as much as possible.

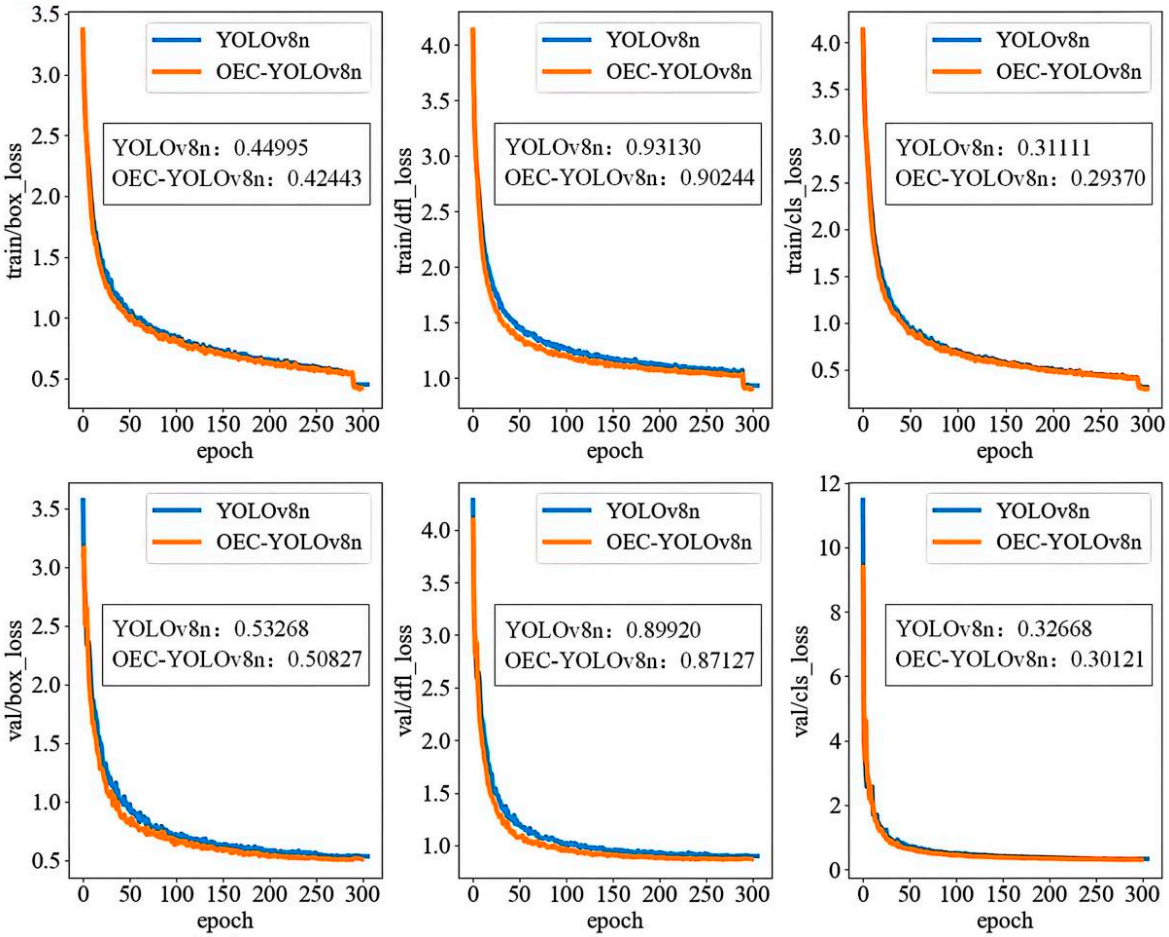



**Figure 9.** Comparison of model training loss function curves.

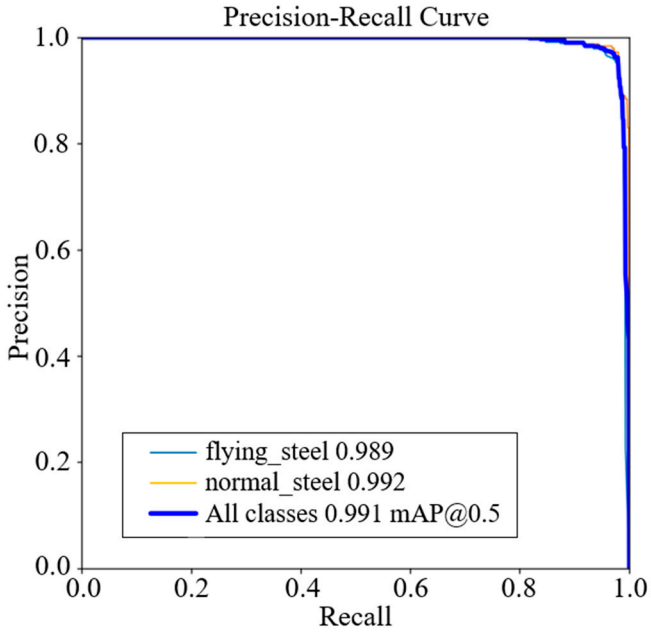



**Figure 10.** PR curve.

Figure 11 is the confusion matrix after the training of the OEC-YOLOv8 model. The confusion matrix provides a comprehensive overview of the outcomes from classification tasks, comparing the prediction results of the model with the actual labels. It can be

seen that the model in this paper can correctly classify the majority of flying_steel and normal_steel samples, and the number of misclassified samples is relatively small. This indicates that the OEC-YOLOv8 model has accurate classification, small deviation, and strong generalization ability. However, the confusion matrix reveals that the model exhibits certain limitations in classifying the background category, specifically manifested as a relatively high false positive rate.

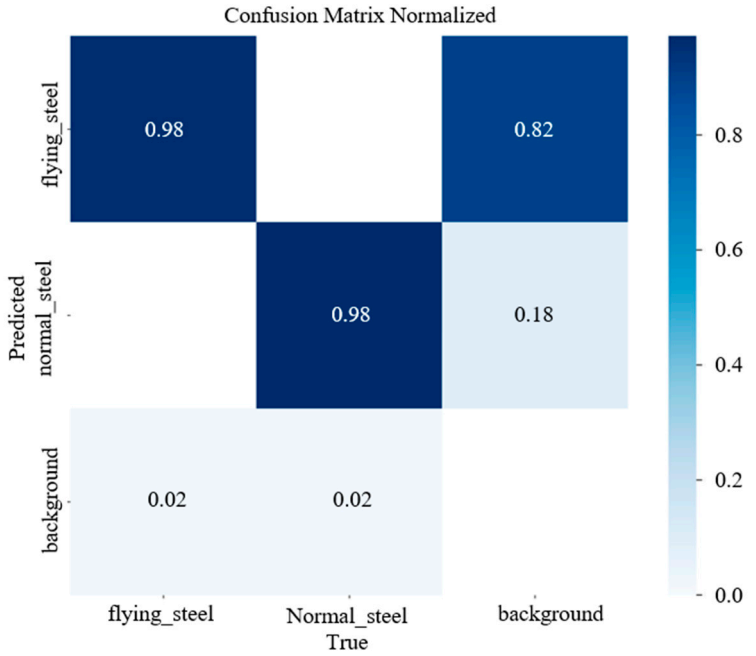



**Figure 11.** Confusion matrix.

### 5.2. Ablation Experiments

The purpose of the ablation experiment is to demonstrate the significance of each improvement within this algorithm. To evaluate the extent of enhancement achieved by each modification method in this paper, ablation experiments were conducted to validate the model. The verification results are shown in Table 4.

**Table 4.** Ablation experimental results.

| ODConv | C2f-PCCA_RVB | EMA | mAP@0.5 | Detection Time (ms) |
|---|---|---|---|---|
| | | | 0.983 | 2.7 |
| √ | | | 0.985 | 2.8 |
| √ | √ | | 0.988 | 2.4 |

Firstly, the ODConv module is employed to substitute two conventional convolutions located at the input stage of the algorithm. Then, C2f-PCCA_RVB is used to replace the original C2f module in the neck network of the algorithm to lighten the neck network. Finally, the network incorporates an EMA module to enhance its feature extraction capabilities.

The mean Average Precision (mAP) metric comprehensively evaluates a model's detection performance across different Intersection over Union (IoU) thresholds. Here, IoU measures the degree of overlap between the predicted bounding boxes and the actual ground-truth annotations. mAP@0.5 denotes the average precision at an IoU threshold of 0.5. mAP@0.5~0.95 represents the average precision across IoU thresholds ranging from 0.5 to 0.95 in 0.05 increments. A higher mAP value signifies superior detection performance

under diverse IoU requirements, offering a more comprehensive reflection of the model's detection accuracy.

The mAP@0.5 value of the original YOLOv8 algorithm reaches 98.3%, and the detection time is 2.7 ms. The detection time is the time required for the flying steel detection model to process one frame of an image. After adding the ODConv module, the mAP@0.5 value of the algorithm is increased by 0.4%. The addition of C2f-PCCA_RVB exhibits the most critical lightweight improvement and increases the mAP@0.5 value by 0.2%. Ultimately, the application of the EMA algorithm yields a 0.3% improvement in the model's mAP@0.5 value. After synthesizing all the improvements, the mAP@0.5 value of the algorithm is 99.1%, and the detection time is 2.5 ms.

### 5.3. Comparative Experiment

This experiment compares the improved model of this paper with the historical version of the YOLO model. In order to ensure the value of comparative experiments, the selected comparative models are the models with the smallest number of parameters in each version of history. In this experiment, the number of parameters, computational complexity (FLOPs), detection time, mAP@0.5, and mAP@0.5~0.95 were selected as evaluation indicators to verify the performance of the model in the training process. The experimental results are shown in Table 5. Parameters and FLOPs indicate the reaction model size. Detection speed reflects the number of image frames that the model can handle within a unit of time. mAP@0.5 and mAP@0.5~0.95 indicate the reaction accuracy of models.

**Table 5.** Experimental results of accuracy comparison between OEC-YOLOv8 and YOLO series models.

| Method | Precision | Recall | mAP@ 0.5 | mAP@ 0.5~0.95 |
|---|---|---|---|---|
| YOLOv3-tiny | 0.972 | 0.961 | 0.978 | 0.766 |
| YOLOv4-csp | 0.949 | 0.947 | 0.974 | 0.799 |
| YOLOv5n | 0.978 | 0.962 | 0.981 | 0.804 |
| YOLOv6n | 0.966 | 0.967 | 0.979 | 0.811 |
| YOLOv7-tiny | 0.953 | 0.959 | 0.978 | 0.822 |
| YOLOv8n | 0.982 | 0.968 | 0.983 | 0.830 |
| YOLOv9t | 0.982 | 0.970 | 0.985 | 0.837 |
| YOLOv10n | 0.984 | 0.973 | 0.987 | 0.845 |
| YOLOv11n | 0.983 | 0.974 | 0.989 | 0.849 |
| YOLOv12n | 0.986 | 0.976 | 0.990 | 0.856 |
| OEC-YOLOv8n | 0.985 | 0.977 | 0.991 | 0.864 |

The experimental results of the accuracy index comparison between the OEC-YOLOv8 model and the YOLO series model are shown in Table 5.

In the accuracy comparison experiment with the historical version of the YOLO model in Table 5, the OEC-YOLOv8n model performed outstandingly in multiple indicators. Its accuracy is 0.3% higher than that of YOLOv8n, and the recall rate is 0.9% higher. mAP@0.5 and mAP@0.5~0.95 metrics are both higher than those of the other models. When compared to YOLOv8n, the mAP@0.5 value demonstrates an increase of 0.8%, while the mAP@0.5~0.95 value shows a 3.4% enhancement. Moreover, compared with the updated YOLOv9~YOLOv12, OEC-YOLOv8 still performs well and outperformed YOLOv12 in three accuracy metrics: recall, mAP@0.5, and mAP@0.5~0.95. It can also be intuitively seen from the mAP@0.5 comparison of different YOLO algorithms in Figure 9 that the average precision value of the OEC-YOLOv8 algorithm is higher and the detection accuracy is significantly improved.

Figure 12 is the mAP@0.5 comparison diagram of different YOLO algorithms. Through the image, it can be seen more intuitively that the OEC-YOLOv8 algorithm has higher average accuracy value and higher detection accuracy than other YOLO algorithms. This shows that the improved model detection accuracy has been improved.

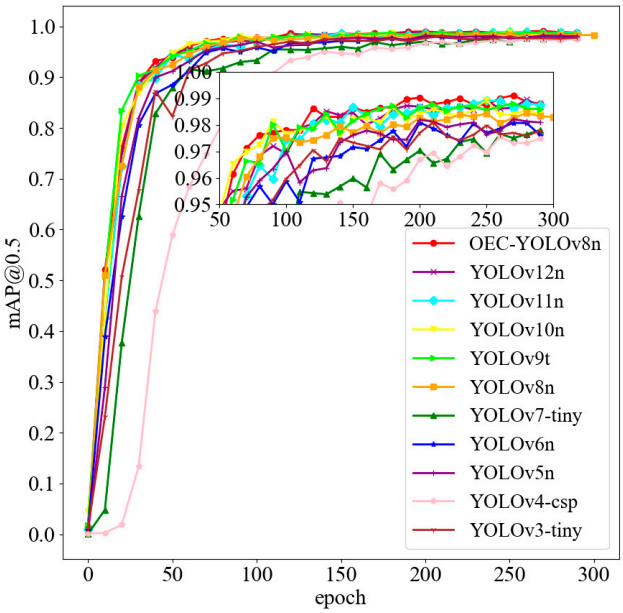



**Figure 12.** Comparison of mAP@0.5 curves between OEC-YOLOv8 and YOLO series models.

Table 6 shows the experimental results of the comparison experiment on the detection speed between the OEC-YOLOV8 model and the YOLO series models.

**Table 6.** The experimental results of OEC-YOLOv8 and YOLO series model detection speed comparison.

| Method | Parameters (M) | FLOPs (G) | Detection Speed (FPS) | Detection Time (ms) |
|---|---|---|---|---|
| YOLOv3-tiny | 103.8 | 283.3 | 61.35 | 16.3 |
| YOLOv4-csp | 52.5 | 52.5 | 68.97 | 14.5 |
| YOLOv5n | 2.6 | 7.8 | 384.62 | 2.6 |
| YOLOv6n | 4.7 | 11.4 | 263.16 | 3.8 |
| YOLOv7-tiny | 6.2 | 13.8 | 192.30 | 5.2 |
| YOLOv8n | 3.2 | 8.7 | 370.37 | 2.7 |
| YOLOv9t | 2.0 | 7.7 | 384.62 | 2.6 |
| YOLOv10n | 2.3 | 6.7 | 357.14 | 2.8 |
| YOLOv11n | 2.6 | 6.5 | 416.67 | 2.4 |
| YOLOv12n | 2.6 | 6.5 | 400.00 | 2.5 |
| OEC-YOLOv8n | 2.8 | 7.8 | 400.00 | 2.5 |

The results of the detection speed comparison experiment in Table 6 show that the OEC-YOLOv8n model performs well in terms of the number of parameters and FLOPs. Compared with YOLOv8n, the number of parameters is reduced by 0.4 M and the FLOPs is decreased. This means that the model has a reduced demand for computing power and is easier to deploy. Its detection time is shortened to 2.5 ms and the detection speed reaches 400 FPS. Compared with the updated YOLO series model, the detection speed and detection time of OEC-YOLOv8 is the same as that of YOLOv12, which can meet the requirements of rapid detection of flying steel accidents.

Table 7 shows the comparison results of the OEC-YOLOv8 model with other common object detection models in four precision metrics: accuracy, recall, mAP@0.5, and mAP@0.5~0.95.

**Table 7.** Experimental results of accuracy comparison between OEC-YOLOv8 and other object detection models.

| Method | Precision | Recall | mAP@0.5 | mAP@0.5~0.95 |
|---|---|---|---|---|
| Faster R-CNN | 0.966 | 0.957 | 0.975 | 0.786 |
| SSD | 0.958 | 0.964 | 0.980 | 0.788 |
| MobileNetv3-small | 0.971 | 0.973 | 0.985 | 0.813 |
| ShuffleNetv2-0.5x | 0.963 | 0.962 | 0.982 | 0.806 |
| OEC-YOLOv8n | 0.985 | 0.977 | 0.991 | 0.864 |

Figure 13 shows a comparison of the mAP@0.5 curves of OEC-YOLOv8 with other object detection models.

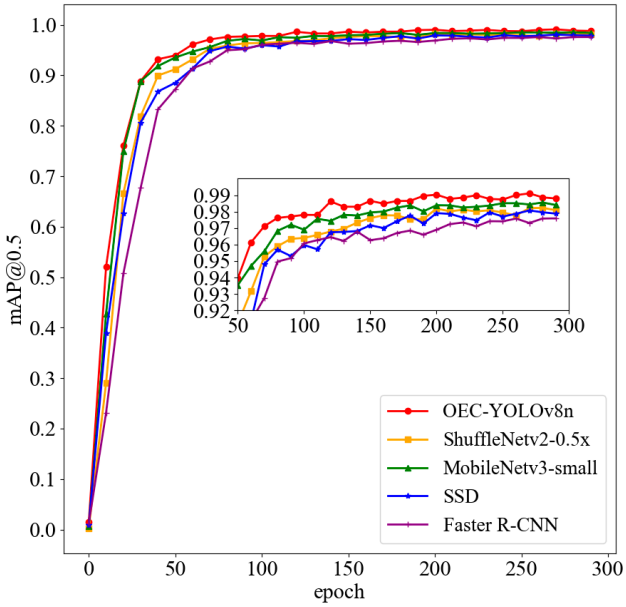



**Figure 13.** Comparison of mAP@0.5 curves between OEC-YOLOv8 and other object detection models.

Table 8 makes a detailed comparison of the performance of the OEC-YOLOV8 model with other target models in terms of detection speed.

**Table 8.** Experimental results of the comparison of detection speeds between OEC-YOLOv8 and other object detection models.

| Method | Parameters (M) | FLOPs (G) | Detection Speed (FPS) | Detection Time (ms) |
|---|---|---|---|---|
| Faster R-CNN | 45.1 | 148.9 | 64.94 | 15.4 |
| SSD | 23.0 | 28.5 | 178.57 | 5.6 |
| MobileNetv3-small | 3.2 | 68.6 | 232.56 | 4.3 |
| ShuffleNetv2-0.5x | 1.2 | 149.6 | 312.50 | 3.2 |
| OEC-YOLOv8n | 2.8 | 7.8 | 400 | 2.5 |

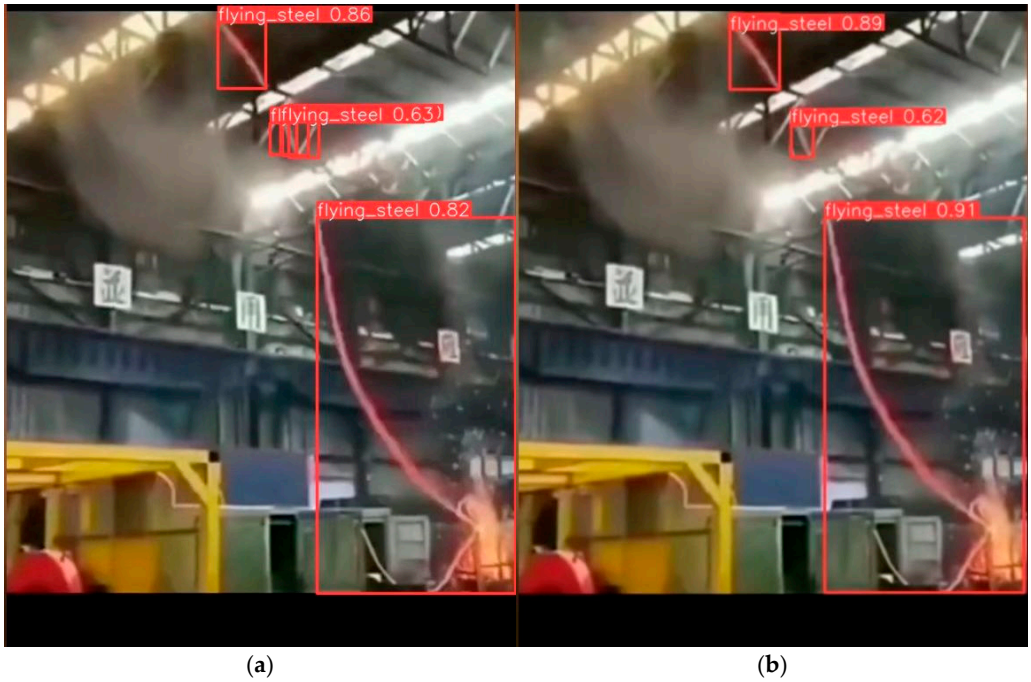
The comparative experimental results of OEC-YOLOv8n in Tables 7 and 8 with other common object detection models show that the OEC-YOLOv8n model has advantages in both detection accuracy and speed. In terms of accuracy, OEC-YOLOv8n has an accuracy of 0.985, a recall rate of 0.977, a high statistic of 0.991 for mAP@0.5, and 0.864 for

mAP@0.5~0.95, all of which are superior to the other comparison models. And from the curve comparison results in Figure 10, it can be seen that the accuracy index mAP@0.5 of OEC-YOLOv8 is generally higher than that of other object detection models.

In terms of detection speed, the FLOPs of the OEC-YOLOv8 model is only 7.8 G, which is lower than that of other object detection models. Moreover, the detection speed reached 400 FPS and the detection time was 2.5 ms, which was the fastest among all the comparison models. The parameter quantity of the OEC-YOLOv8 model is 2.8 M. Although it is at a medium level among all the comparison models, and compared with other models, the increase in the parameter quantity does not bring excessive computational burden.

In general, the OEC-YOLOv8 model performs better than other comparison models in terms of detection accuracy and detection speed. This means that the OEC-YOLOv8 model can identify the accident of flying steel more quickly and accurately, as well as gain valuable time for subsequent emergency treatment, so as to effectively reduce the loss caused by production accidents and provide strong technical support for production safety monitoring.

Figure 14 shows the comparison of OEC-YOLOv8n and YOLOv8n object detection results. Compared with YOLOv8n, the object detection results of the OEC-YOLOv8n model in the test diagram do not overlap the target box. In a complex environment with water vapor interference and equipment occlusion, it can better distinguish the interference objects in the flying steel and the surrounding environment, and the object detection results are more accurate. Moreover, the confidence of the OEC-YOLOv8n object detection results is higher, indicating that the detection results are more reliable and more suitable for production sites with complex environments. In conclusion, although the confusion matrix shows certain limitations in background detection, the object detection results indicate that the OEC-YOLOv8n model still has high accuracy and robustness in practical applications.




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

**Figure 14.** Comparison of object detection results of OEC-YOLOv8n and YOLOv8n. (**a**) YOLOv8n object detection result diagram; (**b**) OEC-YOLOv8n object detection result diagram.

Figure 15 is the experimental result diagram of the OEC-YOLOv8 model verification. The OEC-YOLOv8 model demonstrates precise discrimination between flying steel and normal steel, while accurately pinpointing the target's location within images. It can be

applied to high-speed wire rod mills through real-time sampling of monitoring videos to detect whether flying steel accidents occur, as well as display the test results in the secondary production control system, both accurately and intuitively. If the flying steel accident is detected, an alarm signal can be sent in time.



**Figure 15.** The experimental result diagram of the OEC-YOLOv8 model verification.

## 6. Conclusions and Discussion

Aiming at the problems and difficulties of the existing flying steel detection methods, this paper proposes a flying steel detection algorithm based on OEC-YOLOv8 to realize high-precision flying steel detection based on machine vision. To begin with, ODConv is incorporated into the backbone network to endow the network with input-dependent characteristics, thereby enhancing the feature extraction capability for input images. Then, a lightweight C2f-PCCA_RVB module is proposed to give the neck network a lightweight design. Finally, the EMA module is integrated into the neck network, where global contextual information across various scales is merged to further boost the model's feature extraction efficiency.

The final experimental results show that the average accuracy (mAP@0.5) of flying steel detection based on OEC-YOLOv8 reaches 99.1%, and the detection time reaches 2.5 ms, which can realize the real-time accurate detection of flying steel. Compared with the historical version of the YOLO series and other object detection algorithms, the algorithm has higher accuracy and faster detection speed, shows better performance, and achieves good experimental results. Compared with the latest YOLOv12, OEC-YOLOv8 outperforms YOLOv12 in three accuracy metrics and achieves the same detection speed.

Unfortunately, the current experimental conditions are limited. The method in this paper is only verified under laboratory conditions and has not been verified by practical application in the industrial field. In theory, the flying steel detection model can be carried on a separate server, and its operation will not affect the operation of the industrial production control system. Furthermore, the dataset size is constrained due to the limited availability of data sources. The confusion matrix thus shows certain limitations. Future research can further collect and label data, as well as use more data preprocessing methods to construct a richer and more complete flying steel detection dataset, which further improves the generalization ability and robustness of the algorithm. Artificial intelligence technology is developing rapidly. By combining the OEC-YOLOv8 model and other artificial intelligence technologies, continuous improvement and optimization can be carried out to further study

and make technological breakthroughs in the detection of flying steel. The flying steel detection algorithm based on the object detection algorithm is not necessarily the optimal solution to the safety monitoring problem. Perhaps an unsupervised learning method can be considered to overcome the problem that abnormal data is difficult to collect.

**Author Contributions:** Conceptualization, Y.L. and F.Z.; methodology, Y.L.; software, Y.L.; validation, Y.L., X.L. and J.Z.; formal analysis, X.X. (Xiong Xiao); investigation, L.W.; resources, X.X. (Xiaofei Xiang); data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L.; visualization, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key Research and Development Program of China, grant number 2022YFB3304000.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author due to privacy restrictions.

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

## References

1. Wu, D.; Xiao, F.; Wang, B.; Liu, J.; Liao, B. Investigation on grain refinement and precipitation strengthening applied in high speed wire rod containing vanadium. *Mater. Sci. Eng.* **2014**, *592*, 102–110. [CrossRef]
2. Causes and Treatment of Piled Steel in the Rolling Process of High-Speed Wire Rod. Available online: https://lmmrolls.com/causes-and-treatment-of-piled-steel-in-the-rolling-process-of-high-speed-wire-rod/ (accessed on 15 July 2022).
3. Wang, Y.; Ding, Z. Cause analysis and countermeasures of steel piling accident in hot rolling coiler. *Metall. Power* **2018**, *21*, 9–12. [CrossRef]
4. High-Speed Wire Rod Roughing Mill Unit Steel Accumulation. Available online: https://lmm-rollingmill.com/blog/analysis-and-treatment-of-pile-up-accidents-in-high-speed-wire-rod-roughing-mill/ (accessed on 28 December 2023).
5. Safety and Health in the Steel Industry: Data Report 2024. Available online: https://worldsteel.org/safety-and-health/safety-and-health-in-the-steel-industry-data-reports/safety-and-health-in-the-steel-industry-data-report-2024/ (accessed on 25 March 2025).
6. de Sena, A.P.C.; de Freitas, I.S.; Filho, A.C.; Sobrinho, C.A.N. Fuzzy diagnostics for gearbox failures based on induction motor current and wavelet entropy. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 265. [CrossRef]
7. Tang, N.; Zhang, Q.; Wang, C.; Gao, L. Hybrid Fault Diagnosis for High Speed Wire Rod Finishing Mill. In Proceedings of the 2023 6th International Symposium on Autonomous Systems (ISAS), Nanjing, China, 23–25 June 2023; pp. 1–6. [CrossRef]
8. Shi, P.; Yue, Y.; Hao, G.; Hua, C. A novel multi-source sensing data fusion driven method for detecting rolling mill health states under imbalanced and limited datasets. *Mech. Syst. Signal Process.* **2022**, *171*, 108903. [CrossRef]
9. Yue, Y.; Shi, P.; Tian, J.; Xu, X.; Hua, C. Rolling mill health states diagnosing method based on multi-sensor information fusion and improved DBNs under limited datasets. *ISA Trans.* **2022**, *134*, 529–547. [CrossRef] [PubMed]
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
11. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. CoRR. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
12. Xie, W.; Sun, X.; Ma, W. A lightweight multi-scale feature fusion steel surface defect detection model based on YOLOv8. *Meas. Sci. Technol.* **2024**, *35*, 5. [CrossRef]
13. Wang, B.; Wang, M.; Yang, J.; Luo, H. YOLOv5-CD: Strip steel surface defect detection method based on coordinate attention and a decoupled head. *Meas. Sens.* **2023**, *30*, 100909. [CrossRef]
14. Wang, L.; Song, C.; Wan, G.; Cui, S. A surface defect detection method for steel pipe based on improved YOLO. *Math. Biosci. Eng.* **2024**, *21*, 3016–3036. [CrossRef] [PubMed]
15. Varghese, R.; Sambath, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Tamil Nadu, India, 18–19 April 2024. [CrossRef]
16. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312. [CrossRef]

17. Ding, B.; Zhang, Y.; Ma, S. A Lightweight Real-Time Infrared Object Detection Model Based on YOLOv8 for Unmanned Aerial Vehicles. *Drones* **2024**, *8*, 479. [CrossRef]
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
19. Khan, A.; Rauf, Z.; Sohail, A.; Rehman, A.; Asif, H.; Asif, A.; Farooq, U. A survey of the vision transformers and their CNN-transformer based variants. *arXiv* **2023**, arXiv:2305.09880. [CrossRef]
20. Wang, A.; Chen, H.; Lin, Z.; Han, J.; Ding, G. RepViT: Revisiting Mobile CNN From ViT Perspective. *arXiv* **2023**, arXiv:2307.09283. [CrossRef]
21. Gao, H.; Liu, S.; van der Maaten, L.; Weinberger, K.Q. CondenseNet: An Efficient DenseNet Using Learned Group Convolutions. *arXiv* **2017**, arXiv:1711.09224. [CrossRef]
22. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. *arXiv* **2018**, arXiv:1807.11626. [CrossRef]
23. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946. [CrossRef]
24. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
25. Chen, J.; Kao, S.; He, H.; Wen, S.; Lee, C. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031. [CrossRef]
26. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; p. 99. [CrossRef]
27. Woo, S.; Zark, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521. [CrossRef]
28. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717. [CrossRef]
29. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic Convolution: Attention Over Convolution Kernels. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11027–11036. [CrossRef]
30. Li, C.; Zhou, A.; Yao, A. Omni-Dimensional Dynamic Convolution. *arXiv* **2022**, arXiv:2209.07947. [CrossRef]
31. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. *arXiv* **2023**, arXiv:2305.13563. [CrossRef]
32. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. *arXiv* **2019**, arXiv:1809.02983. [CrossRef]
33. Liu, L.; Li, P.; Wang, D.; Zhu, S. A wind turbine damage detection algorithm designed based on YOLOv8. *Appl. Soft Comput.* **2024**, *154*, 111364. [CrossRef]


**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.