electronics

MDPI

Article

Optimization Algorithm for Steel Surface Defect Detection Based on PP-YOLOE

Yi Qu, Boyu Wan, Cheng Wang *, Haijuan Ju, Jiabo Yu, Yakang Kong and Xiancong Chen

Fundamentals Department, Air Force Engineering University, Xi'an 710051, China; txkgrkgd2s138@163.com (Y.Q.); wby11968748741@163.com (B.W.); jhjcumtgx@163.com (H.J.); b2283216046@163.com (J.Y.); kongyakang@126.com (Y.K.); cxcaimath@163.com (X.C.)
* Correspondence: valid_01@163.com

Abstract: The fast and accurate detection of steel surface defects has become an important goal of research in various fields. As one of the most important and effective methods of detecting steel surface defects, the successive generations of YOLO algorithms have been widely used in these areas; however, for the detection of tiny targets, it still encounters difficulties. To solve this problem, the first modified PP-YOLOE algorithm for small targets is proposed. By introducing Coordinate Attention into the Backbone structure, we encode channel relationships and long-range dependencies using accurate positional information. This improves the performance and overall accuracy of small target detection while maintaining the model parameters. Additionally, simplifying the traditional PAN+FPN components into an optimized FPN feature pyramid structure allows the model to skip computationally expensive but less relevant processes for the steel surface defect dataset, effectively reducing the computational complexity of the model. The experimental results show that the overall average accuracy (mAP) of the improved PP-YOLOE algorithm is increased by 4.1%, the detection speed is increased by 2.06 FPS, and the accuracy of smaller targets (with a pixel area less than 322) that are more difficult to detect is significantly improved by 13.3% on average, as compared to the original algorithm. The detection performance is also higher than that of the mainstream target detection algorithms, such as SSD, YOLOv3, YOLOv4, and YOLOv5, and has a high application value in industrial detection.

Keywords: target detection; PP-YOLOE; attention mechanism; steel surface defect

Citation: Qu, Y.; Wan, B.; Wang, C.; Ju, H.; Yu, J.; Kong, Y.; Chen, X. Optimization Algorithm for Steel Surface Defect Detection Based on PP-YOLOE. *Electronics* **2023**, *12*, 4161. https://doi.org/10.3390/electronics12194161

Academic Editor: Beiwen Li

Received: 6 September 2023
Revised: 3 October 2023
Accepted: 5 October 2023
Published: 7 October 2023

1. Introduction

The process of industrial development is inseparable from the control of the performance of metal materials. Steel, as the most widely used alloy and the material with the highest production levels, has an irreplaceable role in transportation, machinery manufacturing, and the aerospace and national defense industry. With the rapid development of China's industrial sector and the changing market demand, the quality and performance requirements of steel in various fields are also increasing. However, in the process of steel production and manufacturing, factors such as production equipment and the technological level can lead to irregular defects such as cracks, scratches, pits, and patches on the steel surface [1], resulting in the different degrees of degradation of the mechanical properties, the corrosion resistance and wear resistance of the material in subsequent applications, and serious defects, which can even lead to safety-related accidents. Therefore, an accurate and efficient surface defect detection method has become an urgent requirement for current industrial development.

In recent years, with the development of deep learning, computer vision has been widely implemented in various fields, and the target detection for material surfaces is an important part of computer vision. Compared with the traditional manual detection method, this technology has the advantages of stability, high efficiency, and accuracy, and

the significant reduction in labor costs. However, in order to achieve high-accuracy target detection, a large amount of data training is required, and the application and optimization of various algorithms are inseparable.

Current deep learning-based target detection algorithms are mainly divided into one-stage and two-stage processes. One-stage target detection algorithms have a high detection speed but relatively low accuracy, and they include examples such as SSD [2] and YOLO [3]. Two-stage target detection algorithms have a higher detection accuracy but slow detection speeds. These include R-CNN [4], Fast R-CNN [5], Faster R-CNN [6], Mask R-CNN [7], etc. For general object surface defect detection, one-stage target detection algorithms are widely used because of their higher speeds and ability to meet the accuracy requirements. Li et al. [8] detected strip-steel surface defects by constructing a fully convolutional YOLO detection network, which provided an end-to-end detection solution for such detection and achieved a mAP of 97.55. Lin et al. [9] optimized the detection of steel surface defects by improving the SSD model to learn possible defects and proposed a deep residual network (ResNet) for defect classification. Zhang et al. [10] improved the newer PP-YOLOE algorithm model by adding a Coordinate Attention mechanism and replaced the atrous spatial pyramid pooling in the neck with spatial pyramid pooling, and a mAP of 74.2 and 76 was achieved using the models of both sizes of s/m, respectively. Ning et al. [11] realized the defect detection on the steel surface by using the improved YOLOv3 algorithm, which enhanced the robustness of the algorithm via data enhancement, improved the clustering analysis algorithm, added a new prediction box, and improved the detection effect. Kou et al. [12] improved the detection speed by adding a frameless mechanism to enhance the detection speed and designed a dense convolution module to extract richer feature information, and further improvements of the YOLOv3 algorithm were implemented to further enhance the detection accuracy. Li et al. [13] improved the YOLOv5 algorithm by using a K-means clustering algorithm to reset the preset anchor parameters to match the data samples. They also introduced the EAC-Net attention mechanism to enhance the feature extraction effect of the model and replaced the PANet structure in the Neck section with the BiFPN module to ensure a comprehensive integration of the features at all scales. Tested on a dataset of surface defects of aircraft engine components, this improvement not only increased the detection accuracy by 1.0%, but also reduced the inference time by 10.3%.

For the problem of steel surface defect detection, the difficulty lies mainly in the detection effect of small targets. The traditional manual detection method, which requires a significant level of work experience, is prone to the problem of missed detection under high intensity work [14]. As for the target detection algorithm, small targets, such as tiny scratches and cracks, are difficult to detect due to the problems of small target size, high background fusion, few features appearing in isolation, and difficulties in feature extraction after multiple downsampling instances [15]. Improving the detection accuracy of small targets has also been the focus of target detection algorithm research in recent years. Ihor et al. [16] introduced the importance and challenges of metal surface defect detection and proposed a method for metal surface defect detection using U-Net architecture, where they described the training process and optimization strategy and performed a detailed experimental evaluation of the U-Net-like architecture to derive the potential of the U-Net architecture for metal surface defect detection. Cai et al. [17] conducted a large number of experiments based on the R-CNN algorithm to explore the effect of different improvement methods for small target detection. Zhao et al. [18] introduced the feature pyramid network FPN in Faster R-CNN to carry out the degree scale fusion operation on the feature graph, so as to improve the detection ability of the network for defects in the small area of the steel surface. Zhang et al. [19] proposed an improved YOLOv5 algorithm, which added a micro-scale detection layer on the basis of the original algorithm and added a CBAM attention mechanism to control the feature information loss of defects, such as small target defects. Law et al. [20] proposed two variations of CornerNet: one is based on the attention mechanism, which does not require the pixel-level processing of target images. The other

is CornerNet-Squeeze, which introduces a more compact backbone network. Adding an attention mechanism can ensure the consistency of the detection speed insofar as possible, pursuing high accuracy and prioritizing detection accuracy; however, CornerNet-Squeeze pursues high real-time speeds, improving the accuracy as much as possible and prioritizing the detection speed. Extensive research on target detection has resulted in significant progress in detection accuracy and speed. In the context of the above-mentioned, this paper proposes an improved PP-YOLOE algorithm to solve the problem of the difficult detection of small targets by adding the Coordinate Attention mechanism and optimizing the Neck structure to achieve improvement in accuracy and speed.

2. Materials and Algorithms

2.1. Dataset

The experimental datasets involved in this paper all use the steel surface defect dataset (NEU-DET) produced by Kechen Song [21]'s team at Northeastern University, and one of the methods they used to collect the steel surface defects was to install two LED light sources symmetrically, tilted above the steel so as to be measured, and then place an industrial camera on the center axis of the two light sources to collect photos of the defects on the steel surface. In this process, due to the influence of the industrial camera hardware and the acquisition environment with uncertainties, the images captured using the camera must be pre-processed: firstly, according to the images captured using the camera, the whole dataset is corrected for distortion to eliminate the influence of the camera itself, and secondly, the image sensor CCD and CMOS capture image process will be affected by the sensor material properties, working environment, electronic components, and circuit structure, etc., thus introducing various noises. Therefore, high frequency noise is eliminated in this dataset and grayscale transformation is performed. A total of 1800 images were included in the dataset, and all images were grayscale images sized 200 $\times$ 200. There were six types of defects: Crazing, Inclusion, Patches, Pitted Surface, Rolled-in-scale, and Scratches. A total of 300 images were identified for each type of defect, and the labeling information of each type of defect was saved in an XML file with a total of 4189 bounding boxes.

The 1800 images in the dataset were divided into the training set, the validation set, and the test set with a ratio of 8:1:1. This resulted in a total of 1440 images in the training set, 180 images in the validation set, and 180 images in the test set. The images of various defects from the dataset are shown in Figure 1.

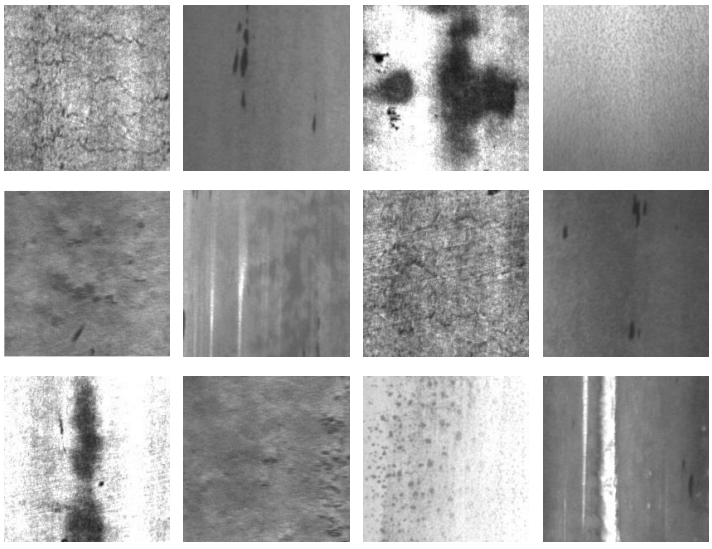


Figure 1. Some images in the dataset.

2.2. PP-YOLOE

In the current field of one-stage target detection, Yolox [22] introduces an advanced dynamic tag assignment method that is significantly more accurate than YOLOv5, which achieves the best balance of detection speed and detection accuracy with 50.1 mAP and 68 FPS when deployed and tested on hardware equipped with a Tesla V100 GPU, making it a masterpiece of the current YOLO family of networks. Inspired by YOLOX, Baidu proposed a new evolutionary version of PP-YOLOE [23], which is commonly used in industry, based on the previous SOTA model PP-YOLOv2.

PP-YOLOE is a single-stage anchor-free model based on PP-YOLOv2 [24], which surpasses many popular YOLO models. Four s/m/l/x volume-specific models are provided for different application scenarios, based on the anchor-free architecture, using powerful backbone and neck and introducing the CSPRepResStage, T-head, and dynamic label assignment algorithm TAL. It can be configured using a width and depth multiplier and avoids the use of special operators, such as deformable convolution or matrix nms, to make it easy to deploy on a wide variety of hardware and with a higher detection accuracy.

For the steel dataset in this paper, the training and testing images were grayscale images, and the image size was small. Most of the detection targets were also small targets, which are difficult to detect. After comparison, the subsequent PP-YOLOE-x model was used for training testing and improvement in this paper.

The model structure of PP-YOLOE is shown in Figure 2.

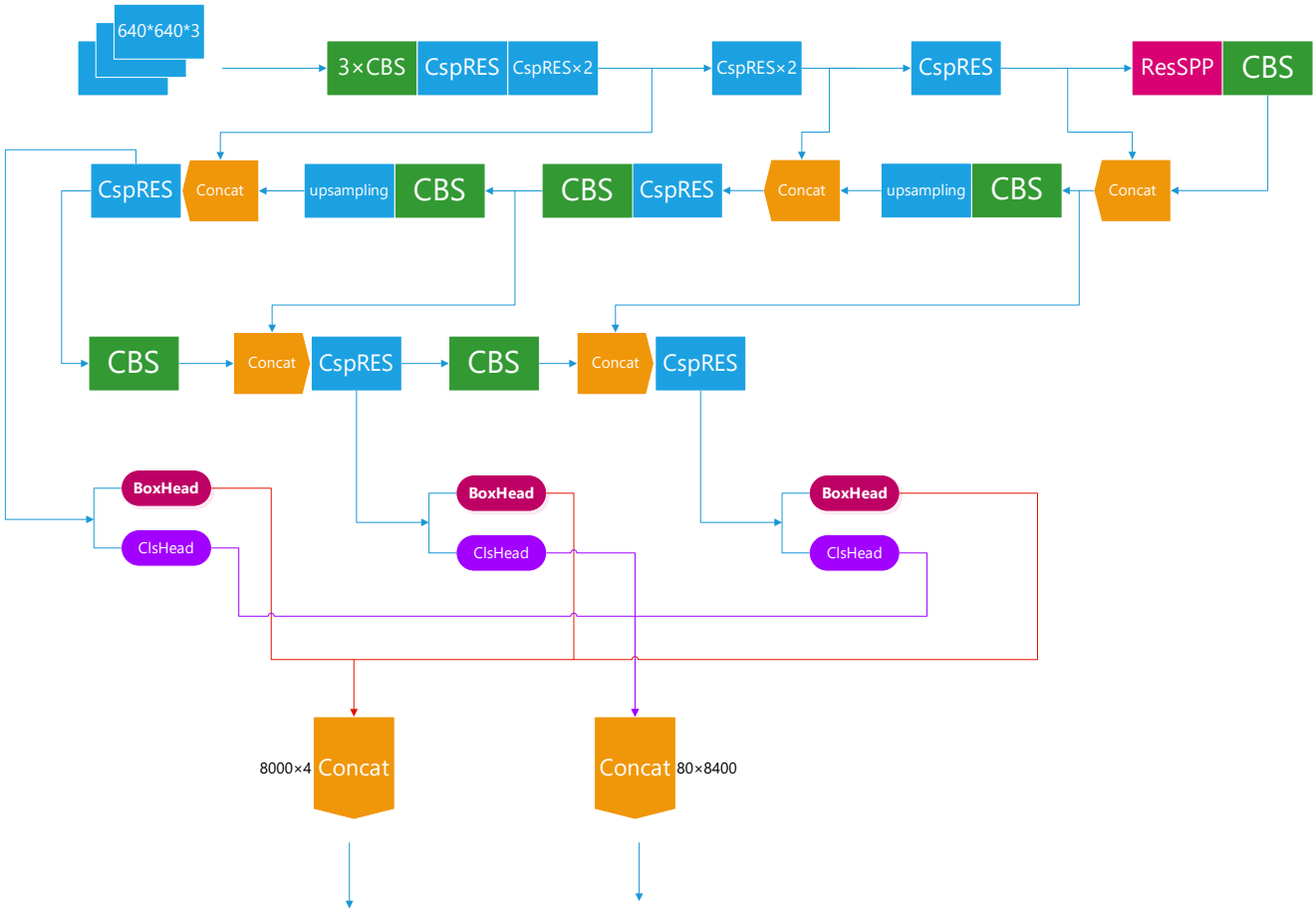


Figure 2. PP-YOLOE structure.

The algorithm consists of a scalable backbone and neck, Task Alignment Learning (TAL), an Efficient Task-aligned head with DFL and VFL (ET-head), and a SiLU activation function.

The Backbone part of PP-YOLOE mainly uses the CSPRes structure, which is improved from the model ideas of RepVGG and CSP for ResNet and uses modules such as the SiLU activation function and Effective SE Attention.

Of these, RepVGG is improved on the basis of VGG by adding identity and residual branching to the Block of the VGG network, which is equivalent to applying the essence of the ResNet network to the VGG network; in the model inference stage, all network layers are converted to a 3×3 convolution via the Op fusion strategy, which facilitates the deployment and acceleration of the network. The basic structure is shown in Figure 3.

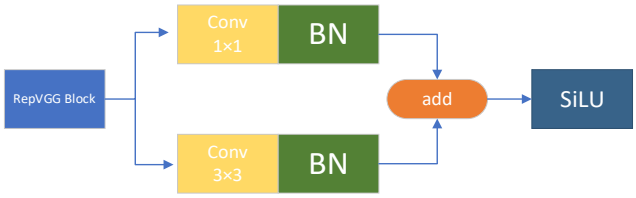


Figure 3. RepVGG structure.

In RepVGG, add refers to the residual connection. During the transition stage of converting simplified convolution layers to regular convolution layers by adding a depthwise convolution layer, residual connections are introduced. Specifically, for each simplified convolution layer, its output is added with the input of the layer, and then undergoes further transformation with a depthwise convolution layer. By using residual connections, RepVGG can maintain the hierarchical structure and functionality of the network, and it also makes it easier to optimize the model. Residual connections help alleviate the problem of gradient vanishing, reduce information loss, and improve network convergence.

BN and SiLU are consistent with the corresponding module structure in PP-YOLOE. BN, short for BatchNorm, is an algorithm frequently used in deep learning networks to accelerate neural network training, speed up convergence, and improve stability. SiLU is employed as the activation function for the grid, which is proposed as an activation function for neural network function approximation in reinforcement learning. Its objective is to introduce non-linearity to the neural network. SiLU is a weighted linear combination of Sigmoid, contained within the activation function Swish [25], and it is a special case of Swish with properties such as having no upper bound but a lower bound, being smooth, and non-monotonic. The relationship among them is described by Equations (1)–(3), where β is a constant or trainable parameter.

$$Sigmoid : \sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

$$SiLU(x) = x \cdot Sigmoid(x) \tag{2}$$

$$Swish(x) = x \cdot Sigmoid(\beta x) \tag{3}$$

The main idea of CSPNet is the Partial Dense Block, which can increase the gradient paths and balance the computation of each layer. Gradient paths refer to the paths through which gradients propagate from the output layer to the input layer of a model. These paths represent the relationships between the features and parameters in the model and their impact on the final object detection results. Through gradient paths, the model can calculate the gradients of each parameter using the backpropagation algorithm and update the parameter values based on the gradients, thereby optimizing the model to better adapt to the object detection task. Increasing the number of gradient paths can improve the training effectiveness and performance of the model as more gradient information can be propagated and utilized at different levels, enabling the model to better learn and capture the features and contextual information of the objects. The disadvantages of using explicit feature mapping replication for concatenation can be mitigated due to the cross-stage strategy. Balancing the computation of each layer, the underlying channel involved in the

dense layer operation is only half that of the original channel, and this method can reduce memory usage and alleviate the computational bottleneck almost by half. The structure of the CSPRes module, improved via the mutual integration of RepVGG, CSP, and ResNet, is shown in Figure 4.

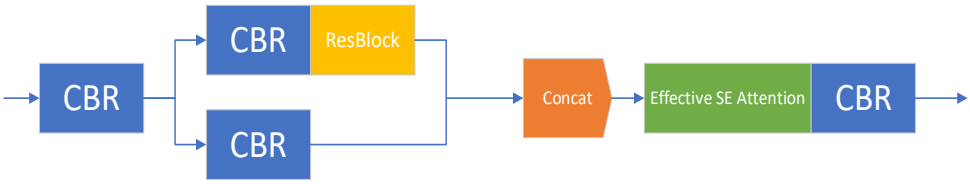


Figure 4. CSPRes structure.

In CSPRes, CBR refers to a basic block composed of a convolutional layer, a batch normalization layer, and a rectified linear unit (ReLU) layer. Specifically, the input of the CBR basic block is a feature map tensor. It undergoes feature extraction with a convolutional layer, followed by data standardization via a batch normalization layer, and finally undergoes a non-linear transformation with the ReLU activation function. The feature map tensor outputted by this basic block can be passed to subsequent CSP blocks or ResNet blocks for further feature extraction and processing. CBR basic blocks are important components of the CSPRes network as they can effectively extract features and reduce model complexity.

ResBlock refers to a residual block, which is a commonly used neural network layer structure typically composed of multiple convolutional layers, including convolution operations, batch normalization, and activation functions. The main feature of a resblock is the residual connection, which adds the input directly to the output of the residual block, retaining more original features during the information propagation process. Resblocks are designed to address the issues of gradient vanishing and difficulty in training models, while also increasing the depth and non-linear expression ability of the network. This allows the network to better capture complex features of the input data and improve the overall performance and accuracy.

Effective SE Attention (eSE) [26] is a channel attention module improved via the SE attention module. The original SE is a representative channel attention algorithm used in the CNN architecture, whose drawback is the reduction in dimensionality, which makes the channel information loss of SE more obvious, a larger computational burden, and more limited in some CV fields. eSE just replaces two fc layers in the original SE with one fc layer with channel number c to avoid channel information loss.

The ResSPP structure can input the same image with different sizes to obtain the same length of pooled features and can ignore the difference in the image size at the input to produce a fixed size output. This effectively avoids the problem of incomplete target segmentation and shape distortion caused by the R-CNN algorithm when processing the image with region cropping and scaling. Moreover, the SPP structure can avoid the convolutional neural network from repeatedly extracting features from images, thus increasing the speed of generating candidate frames and saving computational resources.

The other modules in Backbone are all relatively common, and the CBS module is Conv(3×3)+BN +SiLU and CBR is Conv +BN +ReLU.

The Neck of PP-YOLOE has a common FPN+PAN structure. FPN passes semantic information from a high latitude to a low latitude, which makes the large target clearer; and PAN passes semantic information from a low latitude to a high latitude, so that the small target is also clearer.

The Head part of PP-YOLOE adopts the T-Head of TOOD (Task-aligned one-stage object detection) [27], which is improved using the traditional one-stage detector by increasing the interaction between two tasks and enhancing the detector's ability to understand comparisons. The Cls Head and Loc Head are predominantly included and have a simple feature extractor and two task-aligned predictors (TAP). In the process of T-Head's

prediction of the classification and localization of FPN features, the T-Head first makes classification and localization predictions based on the FPN features, then TAL generates learning signals based on the task alignment metric calculated at each anchor point, and finally, T-Head automatically adjusts the classification probability and localization predictions based on the information transmitted back from TAL.

3. Methods

3.1. CA Attention Mechanism

Channel attention has shown significant improvement in the model performance in many applications, such as Squeeze-and-Excitation (SE) [28]. However, traditional channel attention typically only focuses on channel information, ignoring the target's position information, which makes it difficult to generate spatially selective attention maps. Coordinate Attention [29] is a new attention mechanism aimed at improving the performance and efficiency in mobile network design. Coordinate Attention embeds position information into channel attention, enabling the network to generate spatially selective attention maps, thus detecting small and difficult-to-detect targets more effectively [30]. It can be integrated into networks such as MobileNet and Efficient Net, improving the network performance and accuracy in tasks such as classification, detection, and segmentation.

The Block structure of Coordinate Attention is shown in Figure 5, which encodes channel relationships and long-term dependencies with precise location information, and the specific operation is divided into two steps: Coordinate information embedding and Coordinate Attention generation.

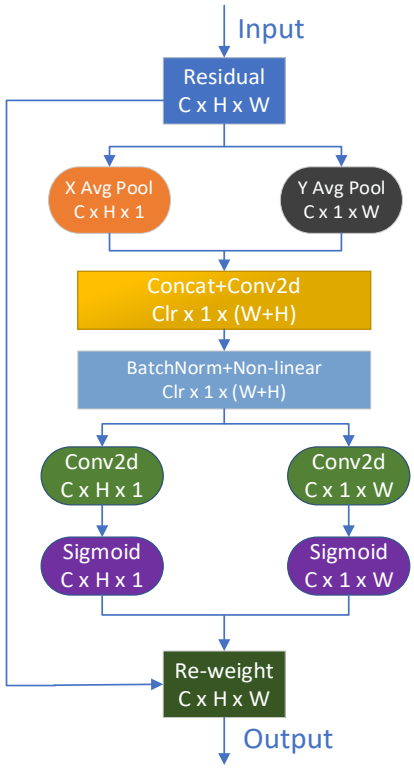


Figure 5. Coordinate Attention mechanism.

- Coordinate Information Embedding

The traditional global pooling method cannot accurately preserve the global spatial information as it is compressed into a channel descriptor. In order to obtain accurate target location information, it is necessary to decompose the global pooling and transform it into a one-to-one feature encoding operation.

Given an input X, which is an intermediate feature tensor with values directly obtained from a convolutional layer with a fixed kernel size, the compression step of the *c*-th channel for achieving decomposed global pooling can be represented as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j) \tag{4}$$

In the equation, z_c is the output associated with the *c*-th channel; and x_c refers to the *c*-th channel in the X tensor, which is a sub-tensor in X with a size of $1 \times H \times W$. Specifically, given the input X, two pooling kernels with sizes (H,1) or (1,W) are first used to encode each channel along the horizontal and vertical coordinates, respectively. This produces the output for the *c*-th channel with height h and width w, as shown in the following equation.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le j \le H} x_c(h, j) \tag{5}$$

$$z_c^w(\omega) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j, \omega) \tag{6}$$

The two transformations mentioned above aggregate features along the directions of the two spatial dimensions. Through such transformations, a pair of direction-aware feature maps are obtained. Compared to a single feature vector, the CA Block can preserve precise spatial directional information, capture long-range dependencies along one spatial direction, and retain accurate positional information along another spatial direction. This is advantageous for the algorithm network to more accurately locate the target.

- Coordinate Attention Generation

In order to fully utilize the information generated by embedding the coordinate information and effectively capture the relationships between channels, the following transformation is performed again to complete the generation of Coordinate Attention:

First, the information embedding is subjected to the Concatenate operation and subsequently transformed with a 1×1 convolutional transformation function F1, as shown in Equation (7):

$$f = \delta(F_1([z^h, z^w])) \tag{7}$$

In the equation, $[z^h, z^w]$ represents the Concatenate operation along the spatial dimensions. δ denotes a non-linear activation function, and f refers to an intermediate feature map that encodes spatial information in the horizontal and vertical directions. Then, f is decomposed into two separate tensors along the spatial dimensions.

$$f^h \in R^{C/r \times H} \tag{8}$$

$$f^w \in R^{C/r \times W} \tag{9}$$

In the equation, r is the reduction ratio that controls the size of the SE Block. Then, using two additional 1×1 convolutional transformations, F_h and F_w, f^h and f^w are separately transformed into tensors with the same number of channels as X, as shown in Equations (10) and (11).

$$g^h = \sigma(F_h(f^h)) \tag{10}$$

$$g^w = \sigma(F_w(f^w)) \tag{11}$$

where σ is the activation function of the Sigmoid. In order to reduce the complexity and computational overhead of the model, the number of channels is usually reduced here by using an appropriate scaling ratio r. The output g^h and g^w are then expanded as attention

weights. Then, the outputs g^h and g^w are expanded as attention weights, respectively. Finally, the output of the Coordinate Attention Block can be written as:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{12}$$

The Coordinate Attention mechanism is added within Backbone, and the three image outputs of the CSPResNet structure are connected to the inputs of the Coordinate Attention module. After obtaining the position information via this module, the images are output from within the CA module in their original state and enter the Neck part for subsequent training.

3.2. Neck Structure Optimization

The improved Neck structure uses FPN to construct a laterally connected feature pyramid, replacing the original FPN+PAN structure in PP-YOLOE, as shown in Figure 6.

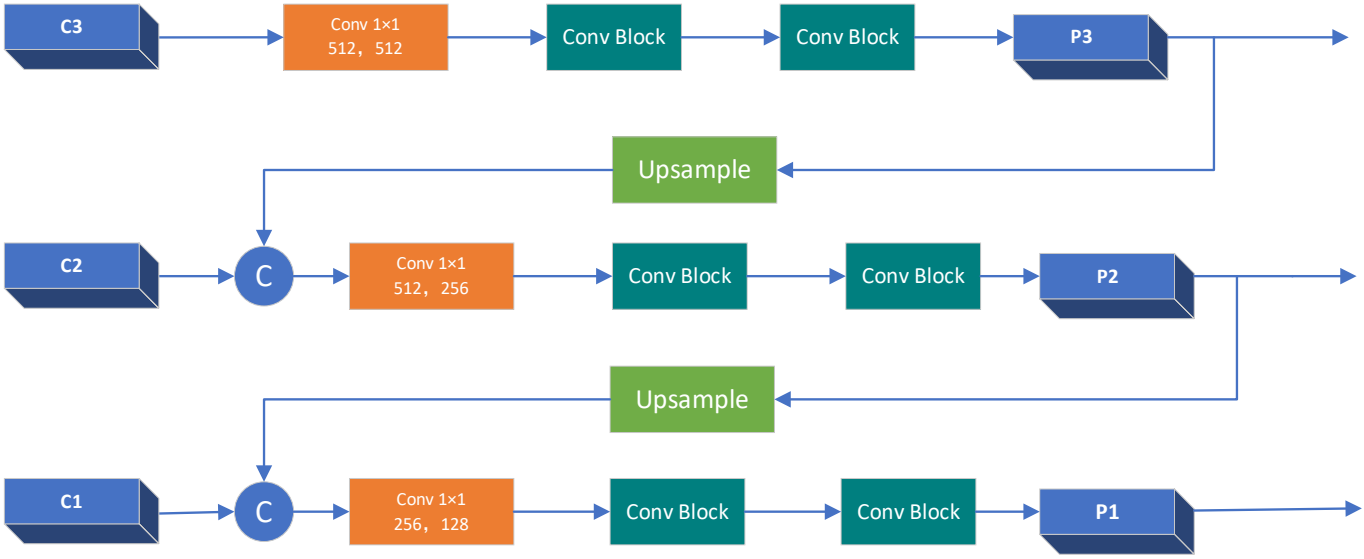


Figure 6. Improved Neck structure.

The three output stages of Backbone are denoted as C1, C2, and C3, which are input into the FPN module in the Neck structure and output as P1, P2, and P3 after operation. For the input image with the resolution size $W \times H$, the output Pn resolution size is $(W/2^n) \times (H/2^n)$.

Since the target size of the detection image varies, different sizes and classes of the targets have different features, and the feature pyramid structure can thus be used to obtain feature maps of different sizes using shallow and deep features to distinguish between the simple and complex detection targets, where the performance of the algorithm can be significantly improved and more robust semantic information can be obtained using this feature pyramid structure.

The new Neck structure will connect the three outputs of Backbone and completely replace the Neck part of the original algorithm, capturing all the output images into the FPN pyramid structure for feature recognition and then the unified output to the Head part. This structure not only guarantees the accuracy of recognition but also greatly reduces the amount of model computation.

3.3. The Three Improved Models

A total of three models are designed in this paper for comparison with the original PP-YOLOE model:

1. PP YOLOE-C

The PP-YOLOE-C model adds the Coordinate Attention mechanism to the Backbone structure of the original PP-YOLOE model. To ensure that Coordinate Attention can utilize the features extracted via the convolutional layers, it is placed after the convolutional operations in the backbone network, specifically after the CspRES and ResSPP modules. This arrangement allows the image outputted via the backbone network to pass through the Coordinate Attention module before undergoing subsequent operations, thereby obtaining more accurate positional information.

2. PP YOLOE-N

The PP-YOLOE-N model replaces the traditional PAN+FPN type Neck structure in the original PP-YOLOE with an improved FPN type Neck structure. The operational pathway is modified by deleting the output before the ResSPP+CBS structure in the Backbone network. Instead, the output pathway is divided into three paths. Subsequently, the improved Neck structure processes and the complete outputs from the Backbone, which then feeds them into the Head part. This modification reduces the overall computational workload of the model while ensuring accuracy.

3. PP YOLOE-CN

While adding the Coordinate Attention mechanism to the Backbone structure of the original PP-YOLOE model, the original Neck structure is replaced by the improved FPN Neck structure. During the operation, the target image first obtains the location information via the Coordinate Attention module, then enters the improved Neck structure for feature extraction and, finally, inputs at the Head for subsequent operation output. The improved model structure is shown in Figure 7.

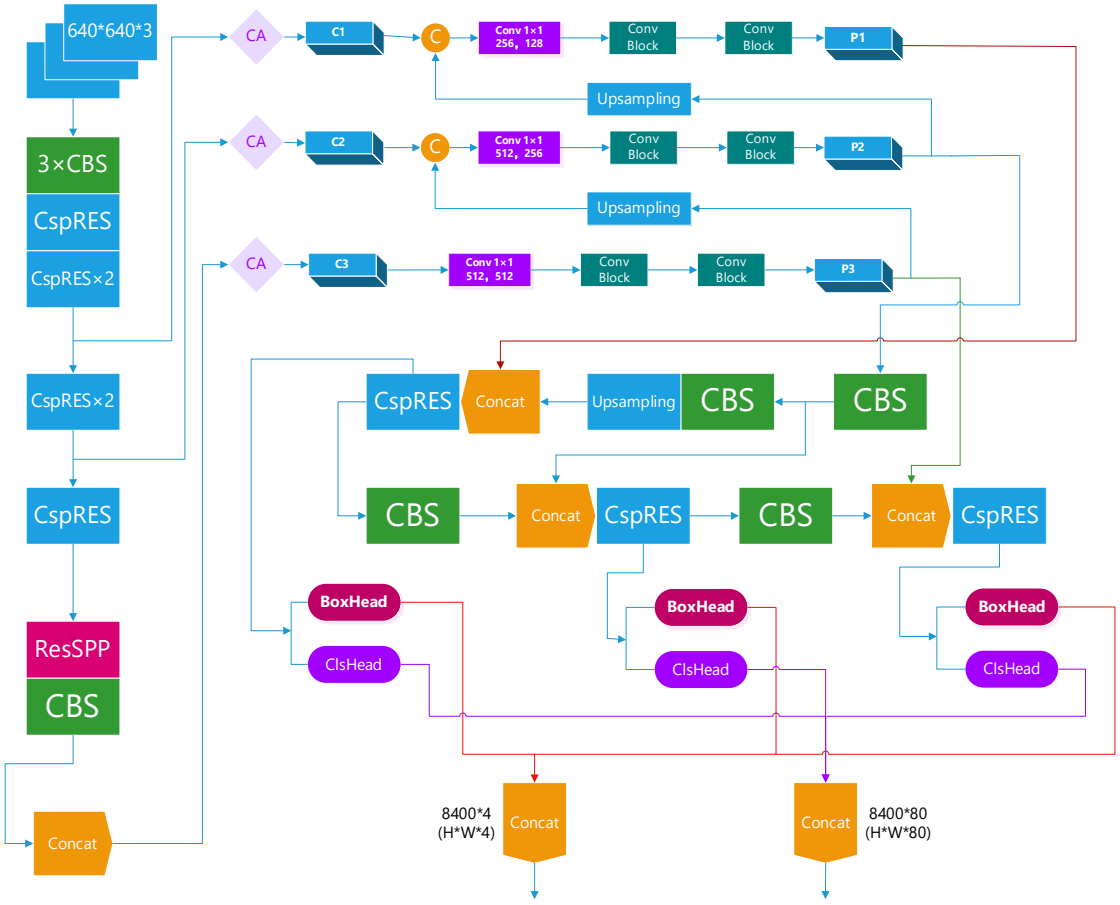


Figure 7. PP-YOLO-CN structure.

4. Experiment

The experimental environment in this paper is based on the Linux operating system with a 160 GB running memory, using PaddlePaddle 2.2.0 as the deep learning framework, Python version 3.8, and CUDA version 11.2. CPU is Intel(R) Xeon Gold 6330 CPU @2.00 GHz. GPU is NVIDIA GeForce RTX 3090 24 G.

4.1. Experimental Evaluation Criteria

This paper analyzes the performance changes in the improved model by comparing the accuracy parameters and model parameters of the modified network with the original version.

The primary indicator for the accuracy parameter is AP@0.5 [31], which refers to the average precision of the model at an IoU threshold of 0.5, effectively measuring the detection performance of the model across all defect categories. In addition to this, we extracted three parameters, namely AP_{small}, AP_{medium}, and AP_{large}. These parameters represent the detection accuracy of small (pixel area less than 32^2), medium (pixel area greater than 32^2 and less than 96^2), and large (pixel area greater than 96^2) targets within the IoU threshold range of 0.5 to 0.95. To further validate the detection performance of the improved algorithm on different defect categories, we also extracted the detection accuracy of the improved algorithm for each individual defect category.

Model parameters refer to the average inference speed and weight volume. Average inference speed, also known as FPS (Frames Per Second), refers to the number of images that a GPU can infer per second. The processing time for each image includes the image preprocessing time, inference time, and post-processing time. The weight volume refers to the size of the weight file generated after model training. It represents the total amount of weight parameters used in the model. Weight parameters are learned using optimization algorithms during the training process and are used to represent the connection weights of the neural network in the model. These weight parameters store the knowledge and feature representation capability of the model.

4.2. Training Process

The PP-YOLOE algorithm uses an anchorless box mechanism, which places one anchor point on each pixel used as the upper and lower boundaries of the three detection heads, assigns ground truths to the corresponding feature maps, determines the center of the bounding box via calculation, and selects the nearest pixel point around it as a positive sample. For the target dataset and server performance, the initial learning rate is set to 0.005, the warm-up mechanism that has been proven effective in many applications is used [32], the warm-up period is set to 5, the work–number is set to 14, the original PP-YOLOE structure batch size is set to 60, and the subsequent training is adjusted to the optimum according to the model size and redundancy of video memory. The x-version of the original PP-YOLOE model and the three improved models are trained and tested in turn to compare and analyze the parameters and performance of each model.

4.3. Training Loss

During the training process, a total of four sets of loss function values of PP-YOLOE-CN are extracted to reflect the performance of the target detection algorithm proposed in this paper. Figure 8 shows the change curve of these four groups of loss function values during training.

Loss_cls represents the classification loss, which is used to reflect the consistency between the prediction result and the actual label. In defect classification tasks, Cross Entropy Loss (CEL) is usually used to calculate Loss_cls. As shown in Equation (13), where N represents the number of defective image samples trained, C represents the number of defective categories, $y_{i,j}$ is the label (0 or 1) of the ith sample belonging to the jth category, and $P_{i,j}$ is the probability that the sample of the prediction model belongs to the jth category.

The smaller the cross entropy loss (Loss_cls), the more accurate the prediction results of the model.

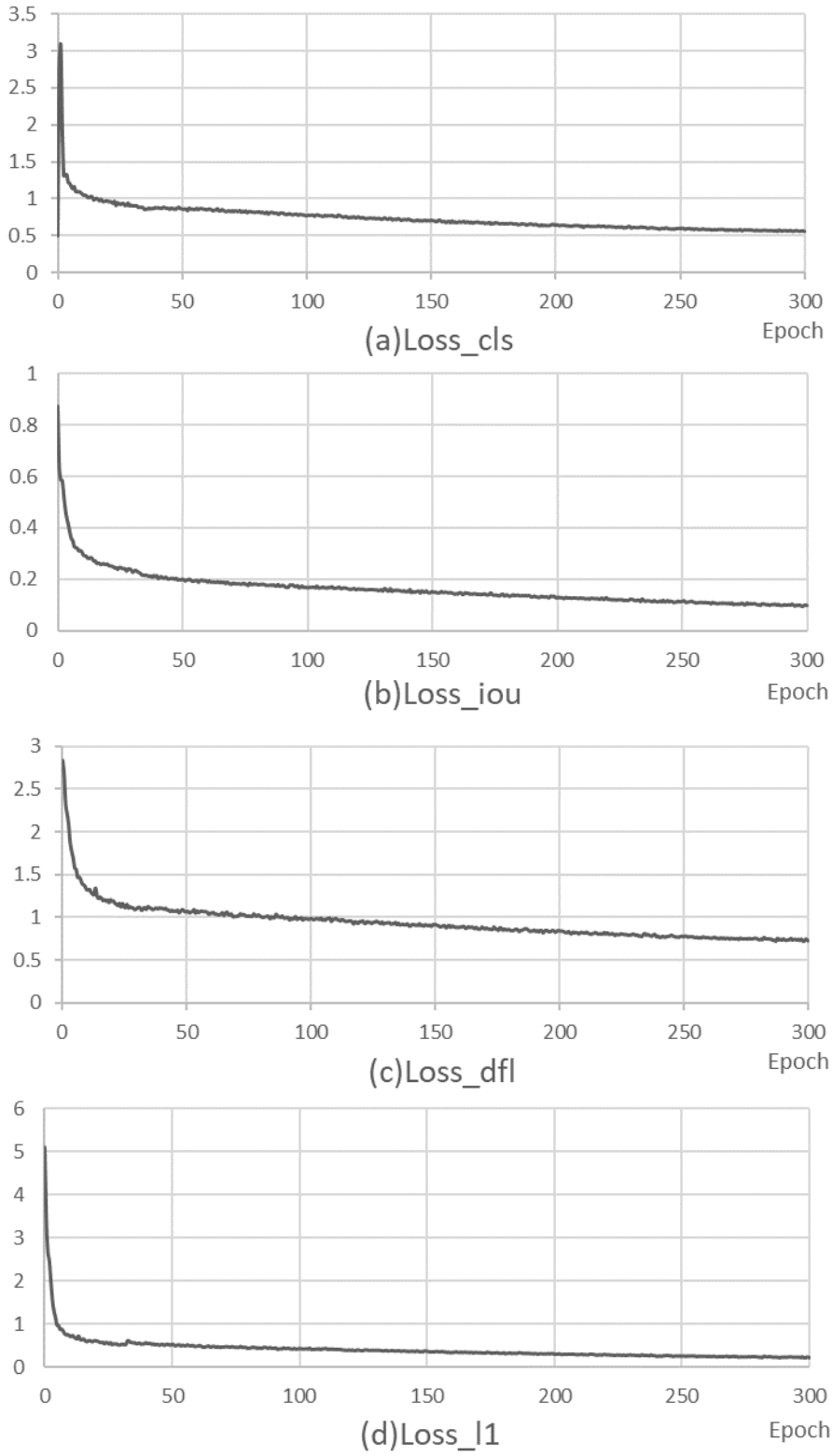


Figure 8. PP-YOLOE-CN model training process loss.

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j}\log(p_{i,j}) \tag{13}$$

Loss_iou(intersection over union loss) is the cross-ratio loss between the predicted bounding box and the true bounding box, which is used to measure the degree of overlap between the predicted and true values of the bounding box. As shown in Equation (14), where $area(P)$ is the predicted location of the defect target and $area(G)$ is the true location of the defect target. By observing the change curve of Loss_iou in Figure 8b, the change in the degree of overlap between the predicted and true defect locations can be reflected.

$$L_{iou} = 1 - \frac{area(P) \cap area(G)}{area(P) \cup area(G)} \tag{14}$$

Loss_dfl is the Dual Focal Loss, which consists of two Focal Losses: one for processing positive samples, and the other for processing negative samples. It calculates correction coefficients using the offsets and gradients in the defect feature map and applies them to the positions of the original detection frames so that they are closer to the real target positions, thus further reducing the defect detection error and solving the problem of imbalance between difficult and easy samples in the one-stage target detection algorithm.

loss_l1 is the Mean Absolute Error Loss, which is used to calculate the absolute error between the predicted value and the true value. As shown in Equation (15), N represents the number of defect samples, y_i is the true value of the ith defect sample, and $\hat{y}_i$ is the value predicted by the model. The smaller the loss_l1 is, the closer the result predicted via the response detection model is to the true value.

$$L_{l1} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{15}$$

The number of training iterations is set to 300, and it can be seen from Figure 8 that Loss_iou, Loss_dfl, and Loss_cls decrease rapidly from 0 to 40 iterations, and then shift to a slow decrease. Among them, Loss_cls exists a local overfitting phenomenon in the early stage of training, and there is a large increase in a short period of time; Loss_l1 decreases sharply from 0 to 20 iterations, and then shifts to a slow decrease. After 300 iterations, the loss value tends to stabilize.

5. Results

5.1. Experimental Results

To determine the impact of each optimization structure on the network performance, ablation experiments are conducted in this paper to train four models, namely PP-YOLOE, PP-YOLOE-C, PP-YOLOE-N, and PP-YOLOE-CN, to obtain different models for the testing performance on this dataset, as shown in Table 1 and Figure 9.

Table 1. Comparison of model detection performance.

Model	AP@0.5	mAP(0.5~0.95)			FPS	Weight/MB
		AP_{small}	AP_{medum}	AP_{large}		
PP-YOLOE	72.3	33.1	37.2	40.5	11.28	386.58
PP-YOLOE-C	74.3	43.0	38.3	44.0	10.7	387.0
PP-YOLOE-N	71.9	38.5	33.5	41.5	14.11	232.39
PP-YOLOE-CN	76.4	46.4	37.1	43.7	13.34	232.8

By analyzing the above experimental results, it can be seen that the improved C model improves the mAP by 2.0% compared with the original model, but the FPS decreased by 0.58. The introduction of the Coordinate Attention mechanism significantly improves the detection accuracy of the defective targets in the model, but the FPS decreases by a small amount, which is due to the addition of new modules in Backbone, increasing the overall computing power of the model. Observing the data in the above table, the Coordinate Attention mechanism has the most obvious improvement on the detection of small targets,

with mAP increasing from 33.1% to 43.0%, which is an improvement of nearly 10 percentage points, while the detection accuracy of medium and large targets also improves by 1.1 and 3.5 percentage points, respectively. Additionally, for Crazing, Inclusion, Patches, Rolled-in-scale, and Scratches (five categories of defective targets), the detection effect displays an obvious improvement. It can be determined that when Coordinate Attention is added to the network, the increase in computation has a small impact on the whole model and the decrease in the detection speed is not significant, but the detection effect of the model is significantly improved.

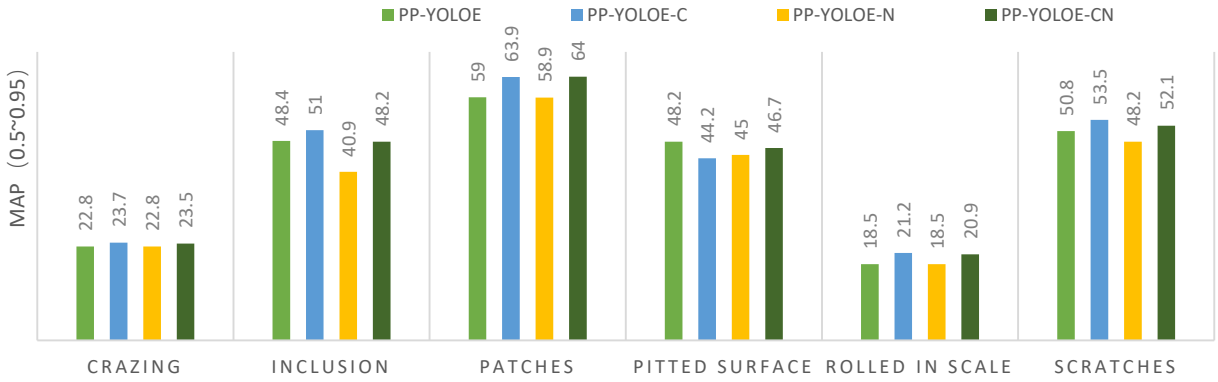


Figure 9. Comparison of the detection performance of different defect categories.

The mAP of the improved N model is reduced by 0.4% compared with the original model, but significantly reduces the inference time and improves 2.83 FPS. It can be seen that the simplification of the Neck of the structure can effectively reduce the overall level of the model operation and ensure that the accuracy does not decrease too much. After the structure optimization, the detection effect of the model for small and large targets is still improved by 5.4 and 1.0 percentage points, respectively, and the detection effect of the Crazing and Rolled-in-scale defects does not decrease, and the detection effect of the Patches defects is also improved. This indicates that the simplification of the Neck structure is a good fit for this dataset and has a significant optimization effect.

The improved CN model improves 4.1% mAP and 2.06 FPS compared to the original model, and the weight volume becomes reduced significantly. By combining the C and N models, the CN model combines the advantage of improving both the detection accuracy and the inference speed. According to the observed data, this improvement has the most obvious effect on small target defect detection, increasing by 13.3 percentage points, and on the large target, increasing by 3.2 percentage points. The detection accuracy of the four categories of Crazing, Patches, Rolled-in-scale, and Scratches improved by 0.7%, 5%, 2.4%, and 1.3%, respectively. For the pitted surface class target, the accuracy is reduced by 1.5% because the defects in this class are generally larger in area and the main defect points are more dispersed, while the improvement direction of the CN model is mainly for small targets, which cannot be perfectly fitted to such defect features, resulting in the decrease in the accuracy for this class target.

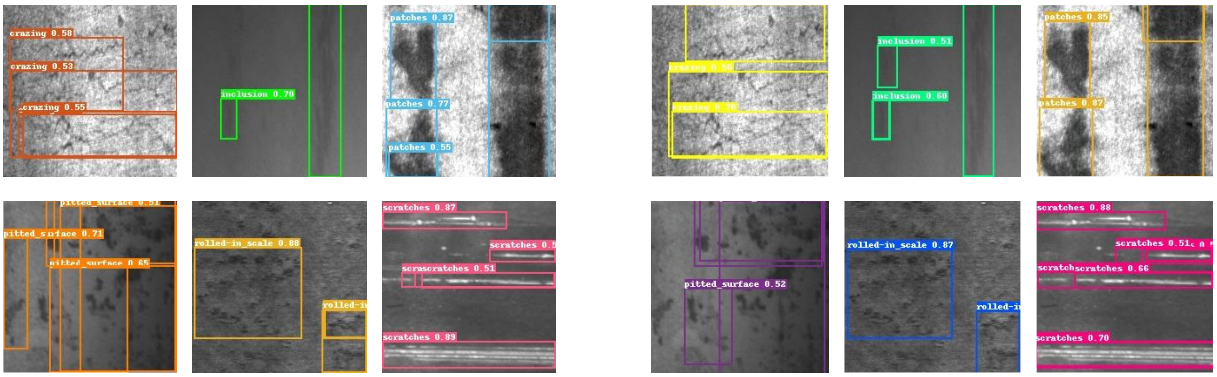
The four models, including the original PP-YOLOE model, have the best effect on the identification of Patches defects and the poorest effect on the identification of Crazing and Rolled-in-scale defects. The study of the characteristics of the six categories of defects shows that most of the defects in the Crazing and Rolled-in-scale categories belong to the large targets. In the process of model calculation, such targets cannot match the corresponding anchor due to the low downsampling multiplier, and thus, cannot participate in the regression, and these two types of defects have a high degree of integration with the background, where the model cannot obtain sufficient negative samples, resulting in their low accuracy.

The above analysis shows that the addition of the Coordinate Attention mechanism within the model can effectively enhance the detection performance of the model and

significantly improve the accuracy of small target detection, as simplifying the structure of the Neck reduces the overall computation of the model, significantly improves the speed of image detection without affecting the accuracy required for testing, and also reduces the size of the weight file by 39.8%, which is more conducive to the deployment of lightweight hardware.

5.2. Detection Effect

In order to visualize the improvement of the improved model on the detection of target defects, the 180 images of the test set were detected using four models, and the recognition effect of PP-YOLOE and the improved PP-YOLOE-CN on some of the test images is shown in Figure 10.



(**a**) PP-YOLOE part of the test picture (**b**) PP-YOLOE-CN part of the test picture

Figure 10. Improved algorithm part of the detection image comparison.

Figure 10a shows the target detection of the PP-YOLOE model, and Figure 10b shows the target detection of the improved PP-YOLOE-CN model. By comparison, the original model is prone to positioning errors for targets in the Crazing category, misses the detection of targets in the Inclusion and Scratches categories, and repeats the tracing frames and the detection of the same target as multiple targets in the Patches, Pitted surface, and Rolled-in-scale categories. Compared with the original PP-YOLOE, the improved CN model is more sensitive to more difficult targets, which can effectively avoid the occurrence of target misdetection, and the detection strategy for multiple targets in the immediate vicinity and superimposed targets is more intelligent, significantly reducing the cases of repeated frame tracing and the segmentation of the same target and the recognition of multiple targets as the same object, effectively avoiding the detection error problem of the original model.

5.3. Performance Comparison

In order to further verify the detection performance of the PP-YOLOE-CN algorithm on this dataset, this paper tested four mainstream, one-stage target detection algorithms, namely SSD, YOLOv3, YOLOv4, and YOLOv5, under the same experimental conditions and compared the performance with the improved model proposed in this paper. The results are shown in Table 2 and Figure 11.

According to the above comparison results, the proposed PP-YOLOE-CN model in this paper leverages the advantages of the C model and N model, effectively improving the overall recognition accuracy compared to several mainstream target detection algorithms. The recognition accuracy of targets of various sizes in this dataset has been improved to varying degrees, especially for small targets where the recognition performance has been significantly enhanced, achieving optimized detection results. Additionally, the recognition accuracy of six different defective targets has also been improved, with more significant improvements observed for the challenging Crazing and Rolled-in-scale category targets, and a noticeable 3.3% increase in average precision for the Patches category targets, which outperforms other mainstream algorithms.

Table 2. Performance comparison with mainstream algorithms.

Model	AP@0.5	mAP(0.5~0.95)		
		AP_{small}	AP_{medum}	AP_{large}
SSD	21.1	9.1	11.3	14.5
YOLOv3	59.7	28.8	33.6	37.1
YOLOv4	61.3	31.3	30.0	36.9
YOLOv5	72.9	34.2	35.4	41.7
PP-YOLOE	72.3	33.1	37.2	40.5
PP-YOLOE-C	74.3	43.0	38.3	44.0
PP-YOLOE-N	71.9	38.5	33.5	41.5
PP-YOLOE-CN	76.4	46.4	37.1	43.7

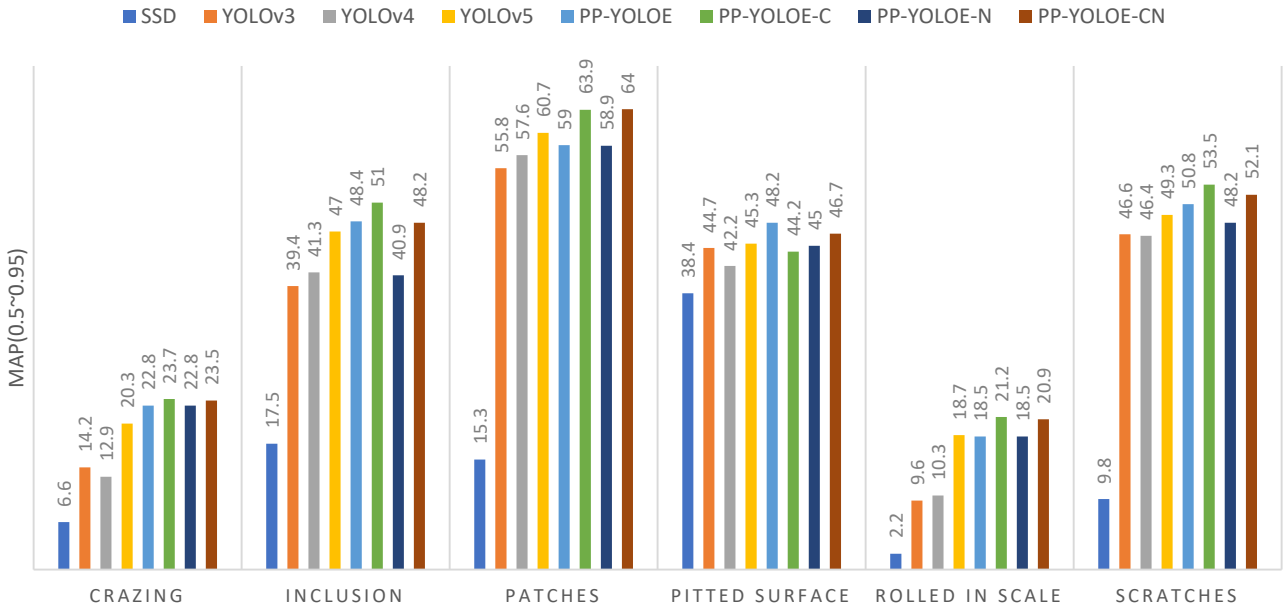


Figure 11. Comparison of detection performance with mainstream algorithms for different defect classes.

6. Conclusions

The first improved PP-YOLOE optimization algorithm for small targets is proposed to address the problems of the low detection accuracy of existing target detection, with the detection speed not meeting industrial requirements and an insufficient small target detection capability. By adding the Coordinate Attention mechanism, the detection accuracy of all kinds of targets is improved, and the detection accuracy for small targets is significantly improved. By simplifying part of the Neck structure, the total calculation of the model is reduced and the detection speed is significantly improved. Compared with the original PP-YOLOE model, the improved optimization algorithm increases the overall mAP by 4.1% and the mAP for small targets by 13.3%, while improving 2.06 FPS.

In the future, with the support of experimental equipment and working conditions, in terms of preliminary work, it is planned to enhance the model learning effect by expanding the number of datasets and data enhancement, since, for machine learning, the amount of data is always the most important feature, and a special dataset related to the defects is being prepared for collection. In terms of the training process, detection can be further improved by increasing the number of iterations, optimizing the learning rate adjustment, etc. In terms of the training process, the detection accuracy can be further improved by increasing the number of iterations and optimizing the learning rate adjustment. In terms of the model grid structure, the existing structure can be further optimized by improving the tracing mechanism and adding a transformer to further improve the detection effect of

the difficult targets, optimizing the loss function to accelerate the convergence accuracy and speed, choosing a smaller s/m model to further reduce the number of operations and increase the training and detection speed, and reduce the size of the weight file to achieve portable hardware deployment.

Author Contributions: Conceptualization, C.W.; methodology, H.J.; software, Y.Q. and B.W.; validation, Y.Q.; formal analysis, Y.Q. and B.W.; investigation, H.J. and B.W.; resources, H.J. and X.C.; data curation, Y.Q.; writing—original draft preparation, Y.Q. and Y.K.; writing—review and editing, C.W.; visualization, Y.Q. and J.Y.; supervision, C.W. and B.W.; project administration, C.W. and H.J.; funding acquisition, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Basic Research Program of Shaanxi, Program number 2023-JC-QN-0696.

Data Availability Statement: The data that support the findings of this research are openly available at http://faculty.neu.edu.cn/songkechen/zh_CN/zhym/263269/list/index.htm (accessed on 15 September 2023).

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

References

1. Qu, E.; Cui, Y.; Xu, S.; Sun, H. Saliency defect detection in strip steel by improved Gabor filter. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2017**, *45*, 12–17.
2. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016.
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149. [CrossRef] [PubMed]
7. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
8. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network. *IFAC-Pap.* **2018**, *51*, 76–81. [CrossRef]
9. Lin, C.Y.; Chen, C.H.; Yang, C.Y.; Akhyar, F.; Hsu, C.Y.; Ng, H.F. (Eds.) *Cascading Convolutional Neural Network for Steel Surface Defect Detection*; Springer: Berlin/Heidelberg, Germany, 2019.
10. Zhang, Y.; Liu, X.; Guo, J.; Zhou, P. Surface Defect Detection of Strip-Steel Based on an Improved PP-YOLOE-m Detection Network. *Electronics* **2022**, *11*, 2603. [CrossRef]
11. Ning, Z.; Mi, Z. Research on Surface Defect Detection Algorithm of Strip Steel Based on Improved YOLOV3. *J. Phys. Conf. Ser.* **2021**, *1907*, 012015. [CrossRef]
12. Kou, X.; Liu, S.; Cheng, K.; Qian, Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement* **2021**, *182*, 109454. [CrossRef]
13. Li, X.; Wang, C.; Ju, H.; Li, Z. Surface defect detection model for aero-engine components based on improved YOLOv5. *Appl. Sci.* **2022**, *12*, 7235. [CrossRef]
14. Li, D.; Li, Y.; Xie, Q.; Wu, Y.; Yu, Z.; Wang, J. Tiny defect detection in high-resolution aero-engine blade images via a coarse-to-fine framework. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–12. [CrossRef]
15. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS—Improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
16. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
17. Konovalenko, I.; Maruschak, P.; Brezinova, J.; Prentkovskis, O.; Brezina, J. Research of U-Net-Based CNN Architectures for Metal Surface Defect Detection. *Machines* **2022**, *10*, 19. [CrossRef]
18. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W.J.H.L. A New Steel Defect Detection Algorithm Based on Deep Learning. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–13. [CrossRef] [PubMed]
19. Zhang, R.; Wen, C. SOD-YOLO: A Small Target Defect Detection Algorithm for Wind Turbine Blades Based on Improved YOLOv5. *Adv. Theory Simul.* **2022**, *5*, 7. [CrossRef]

20. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. Cornernet-Lite: Efficient Keypoint Based Object Detection. *arXiv* **2019**, arXiv:1904.08900.
21. He, Y.; Song, K.C.; Meng, Q.G.; Yan, Y.H. An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1493–1504. [CrossRef]
22. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding Yolo Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
23. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
24. Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X. PP-YOLOv2: A practical object detector. *arXiv* **2021**, arXiv:2104.10419.
25. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
26. Lee, Y.; Park, J. Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13906–13915.
27. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 3490–3499.
28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
29. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, BC, Canada, 11–17 October 2021; pp. 13713–13722.
30. Zha, M.; Qian, W.; Yi, W.; Hua, J. A lightweight YOLOv4-Based forestry pest detection method using coordinate attention and feature fusion. *Entropy* **2021**, *23*, 1587. [CrossRef] [PubMed]
31. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–308. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.