

Article

Lightweight Network-Based Surface Defect Detection Method for Steel Plates

Changqing Wang^{1,2,3}, Maoxuan Sun^{1,2,3}, Yuan Cao^{1,2,3,*}, Kunyu He^{1,2,3}, Bei Zhang^{1,2,3}, Zhonghao Cao^{1,2,3} and Meng Wang^{1,2,3}

¹ College of Electronics and Electrical Engineering, Henan Normal University, Xinxiang 453007, China

² Henan Key Laboratory of Optoelectronic Sensing Integrated Application, Xinxiang 453007, China

³ Henan Engineering Laboratory of Additive Intelligent Manufacturing, Xinxiang 453007, China

* Correspondence: xyuan_cao@163.com

Abstract: This article proposes a lightweight YOLO-ACG detection algorithm that balances accuracy and speed, which improves on the classification errors and missed detections present in existing steel plate defect detection algorithms. To highlight the key elements of the desired area of surface flaws in steel plates, a void space convolutional pyramid pooling model is applied to the backbone network. This model improves the fusion of high- and low-level semantic information by designing feature pyramid networks with embedded spatial attention. According to the experimental findings, the suggested detection algorithm enhances the mapped value by about 4% once compared to the YOLOv4-Ghost detection algorithm on the homemade data set. Additionally, the real-time detection speed reaches about 103FPS, which is about 7FPS faster than the YOLOv4-Ghost detection algorithm, and the detection capability of steel surface defects is significantly enhanced to meet the needs of real-time detection of realistic scenes in the mobile terminal.

Keywords: defect detection; lightweight; cavity spatial convolution; spatial attention



Citation: Wang, C.; Sun, M.; Cao, Y.; He, K.; Zhang, B.; Cao, Z.; Wang, M. Lightweight Network-Based Surface Defect Detection Method for Steel Plates. *Sustainability* **2023**, *15*, 3733. <https://doi.org/10.3390/su15043733>

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 18 November 2022

Revised: 9 February 2023

Accepted: 9 February 2023

Published: 17 February 2023



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

1. Introduction

With the rapid development of industrial automation technology, the study of automated [1,2] detection of defects in industrial production is receiving more and more attention. Due to the influence of various uncertainties, the surface of the steel plate in the production process will produce a variety of defects [3–7], such as scratches, deformation, welds, holes, etc. These defects [8–12] not only affect the integrity of the steel plate but also make a certain impact on the quality of the steel plate, so a more accurate detection of defects [13–16] on the surface of the steel plate is of paramount importance.

Conventional inspection methods use manual observation to detect defects, which is not only time-consuming and labor-intensive, but the results still do not meet the expected requirements. Based on the traditional industrial inspection methods proposed, the automated defect detection technology has been driven to a new level. Experts and scholars at home and abroad have conducted more profound research and practice on traditional machine vision in the detection of defects in steel plates. The enhanced BP detection algorithm was presented by Peng et al. [17] to detect flaws in steel plates. While this technique has a decent detection performance for flaws that are clear targets, it has a sluggish convergence rate and poor performance for small samples. Wang Yixin et al. [18] suggested a comparative detection approach utilizing machine vision; however, despite its high accuracy in recognizing faults in steel plates, it has a higher environmental impact and is incapable of detecting flaws in harsh conditions due to its difficulty with extracting feature images.

At this juncture, the accuracy of steel plate surface flaw detection [19] has increased due to the rapid development of deep learning technology in industrial inspection. Tian Siyang et al. investigated at timeframe instances of hot-rolled strip steel surface faults, identifying two faults,

watermarks and water droplets. The one-stage identification YOLOv2 [20] algorithm was developed and tested for a wide range of surface flaws on steel sheets, as well as against several interference effects caused by false defects. Although the approach can detect surface flaws in hot-tied steel sheets with an average mAP of 92.54%, the detection speed of just 14 FPS prevents real-time detection. Xu Qian et al. used a modified YOLOv3 [21] network structure for the detection of surface defects in steel plates, reducing the model size of YOLOv3 by using a lighter MobileNet [22] network model, adding a cavity convolutional neural network [23] to the network to improve the defect detection capability of steel plates, and adding the Inceptionv3 structure to the network to make the number of layers richer.

In this paper, a defect detection technique for YOLO-ACG is proposed. First, the model's detection accuracy and speed have significantly increased thanks to the use of GhostNet as the replacement for the backbone network of CSPMakenet53 in the YOLOv4 network. Secondly, by replacing the spatial convolutional pooling pyramids at different scales in the original YOLOv4-Ghost network with more accurate spatial convolutional pooling pyramids in the null space, the focus of the model on the significant regions of the feature map target is increased and the perceptual regions of the feature map are enhanced by combining the semantic information of the context. Finally, the pyramid network structure of feature fusion with spatial attention mechanism is embedded in the network design, and the loss of information at the edges of the feature map is addressed by using the FPN structure to connect the fusion of two channels from top to bottom, which facilitates the fusion of information at different scales. The experimental results demonstrate that the YOLO-ACG algorithm can detect surface flaws in steel sheets more quickly and accurately than other lightweight methods, meeting the expectations of industrial inspection. The article will be followed by a more detailed analysis of the YOLO-ACG algorithm in terms of network structure and experimental data.

2. Methodology

2.1. The YOLOv4 Backbone Network

The YOLO (You Only Look Once) algorithm was put forth as a ONE-STAGE target detection technique by Redmon et al. [24] in 2016. The fundamental idea behind the YOLO algorithm is to approach the object recognition problem as a regression problem and utilize a convolutional neural network [25] structure to directly forecast the bounding box and category probabilities from the input image. The fourth iteration of the YOLO algorithm, YOLOv4, employs a variety of algorithmic network architectures, including feature pyramid networks and complete convolutional networks. The CSPDarknet53 backbone network, which is seen in Figure 1, replaces the YOLOv3 algorithm's Darknet53 backbone network topology. Additionally, YOLOv4 use a Mish function for the activation function, logistic regression for the categorization of images, and a feature pyramid network for multi-scale target detection, all of which maintain a high accuracy rate and ensure real-time monitoring.

The YOLOv4 algorithm's backbone network is CSPDarknet53, which is also one of the best backbone networks. CSPDarknet53 generates three outputs, designated P3, P4, and P5, after applying convolutional layers 1×1 and 3×3 . In this process, P3 and P4 are convolved once for 1×1 and then input to the enhanced feature extraction network for feature fusion. P5 is convolved three times and input to the void pyramid pooling layer, with the pooling results being input to the enhanced feature extraction network for feature fusion.

2.2. GhostNet

Yolov4-ghost will replace the current YOLOv4 backbone network with the GhostNet module [26], making the network lighter and easier to deploy on mobile terminals. The head network uses a PAN (path aggregation network) network structure, while the backbone network consists of convolution, spatial pyramid pooling (SPP), and GhostBottleneck.

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
	CrossStagePartial			
1×	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
	Convolutional	128	3×3/2	64×64
	CrossStagePartial			
2×	Convolutional	64	1×1	
	Convolutional	64	3×3	
	Residual			64×64
	Convolutional	256	3×3/2	32×32
	CrossStagePartial			
8×	Convolutional	128	1×1	
	Convolutional	128	3×3	
	Residual			32×32
	Convolutional	512	3×3/2	16×16
	CrossStagePartial			
8×	Convolutional	256	1×1	
	Convolutional	256	3×3	
	Residual			16×16
	Convolutional	1024	3×3/2	8×8
	CrossStagePartial			
4×	Convolutional	512	1×1	
	Convolutional	512	3×3	
	Residual			8×8

Figure 1. CSPDarknet53 backbone network.

GhostBottleneck is replaced by GhostNet as the network’s hub. In place of other network modules, a plug-and-play reusable module called GhostBottleneck dramatically decreases the computational load and model volume of method models. Two Ghost modules are stacked to create a GhostBottleneck. The primary goal of the first Ghost module is to enable the addition of additional channels and dimensions to the feature extraction, usually in the form of an extension layer. The second Ghost module checks to see if the feature extraction dimension still matches its input after lowering the number of channels. The input and output of the two Ghost modules are finally connected. When the second Ghost module is utilized, the ReLU function is not. Because of the variation in the input data distribution between the front layer and the back layer following the activation function, constant matching is required, which reduces training efficiency.

2.3. Loss Function

Steel plates can have a wide variety of imperfections, so the algorithm used in the detection process needs to be extremely precise to identify the types and locations of flaws. The three components that make up the loss function are as follows: (1) confidence loss; (2) classification loss; and (3) bounding box regression loss.

$$\begin{aligned}
 loss_a = & \sum_{i=0}^{S^2} \sum_{j=0}^B W_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] \\
 -I_{noobj} & \sum_{i=0}^{S^2} \sum_{j=0}^B (1 - W_{ij}^{obj}) [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)]; C_i^j = P_{i,j} * IUO_{pred}^{truth}
 \end{aligned}
 \tag{1}$$

$$loss_b = \sum_{i=0}^{S^2} \sum_{j=0}^B W_{ij}^{obj} \sum_{c=1}^C [\hat{p}_i^j \log(p_i^j(c)) + (1 - \hat{p}_i^j(c)) \log(1 - p_i^j(c))]
 \tag{2}$$

$$loss_c = 1 - IOU + \frac{\rho(d, d^{st})}{c^2} + \alpha v; v = \frac{4}{\pi} (\arctan \frac{w^{st}}{h^{st}} - \arctan \frac{w}{h})^2
 \tag{3}$$

where W_{ij}^{obj} represents whether the j th anchor box is in the predicted i th grid, C_i^j is represented by the confidence that the j th bounding box in the i th grid has produced, $P_{i,j}$

represents the objective function's discriminant, \hat{C}_i^j shows the measurement's actual value; $\hat{P}_i^j(c)$ is represented as the expected likelihood that the j th bounding box in I grids belongs to c ; $P_i^j(c)$ is defined as the actual likelihood that j th bounding box in i th grid belongs to c ; d^{st}, d as the intersection of the true and anticipated boxes, ρ is represented by the location of the d^{st}, d centroid. In Figure 2, where the square box represents the prediction box and the rectangular box represents the real box, c is the diagonal distance between the minimum closed loops of the two boxes.

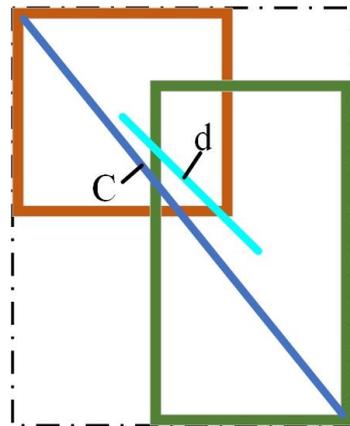


Figure 2. Anchor box.

3. Our Approach

3.1. YOLO-ACG Algorithm

This study proposes the three-part YOLO-ACG network model, which is based on the YOLOv4 algorithm and is visible in Figure 3. It includes a backbone network, a feature fusion network, and a detection head network. RGB images with a three-channel output are used as the input. The first step is to enlarge the feature scale of the image to 52×52 , 26×26 and 13×13 for image information screening and extraction through the P3, P4, and P5 levels of the backbone network. Second, the extracted results are sent to the CBM and CA attention mechanisms, and the ASPP module is added to the output of the extracted features to enhance the effective recognition of target defect differences by merging global and local characteristics with various perceptual fields. The CA attention mechanism then reinforces the output features to improve their location correlations and interactions across latitudes. The problem of the higher-layer network losing the feature information of the lower-layer network during the information extraction process is effectively resolved by fusing the extracted three-feature layers with various semantic information, passing the feature information, and allowing the feature information to enter the feature fusion network after the CA attention mechanism is finished. Finally, the non-maximal suppression algorithm (NMS), whose thresholds are used to further filter the redundant anchor frames during the NMS processing, is combined with the center distance factor of the prediction frame to create the final prediction frame.

3.2. Ghost Module

The redundancy of feature maps in neural convolutional networks is one of the most important features. When the feature maps are output for visualization, there are many outputs with very similar features, which can be obtained by simple linear transformations without complicated operations. As shown in Figure 4, the working principle of the standard convolution and Ghost modules are presented separately.

As shown in Figure 4a for standard convolution, although ordinary convolution extracts features by using a large number of convolutions, and thus, generates a feature map, the excessive number of convolution kernels with the number of channels generates redundant information and leads to an increase in computation. Additionally, the Ghost

module in Figure 4b separates the regular convolution into two parts, extracts the partial convolution to create the feature map, and then efficiently creates the whole feature map using straightforward linear operations.

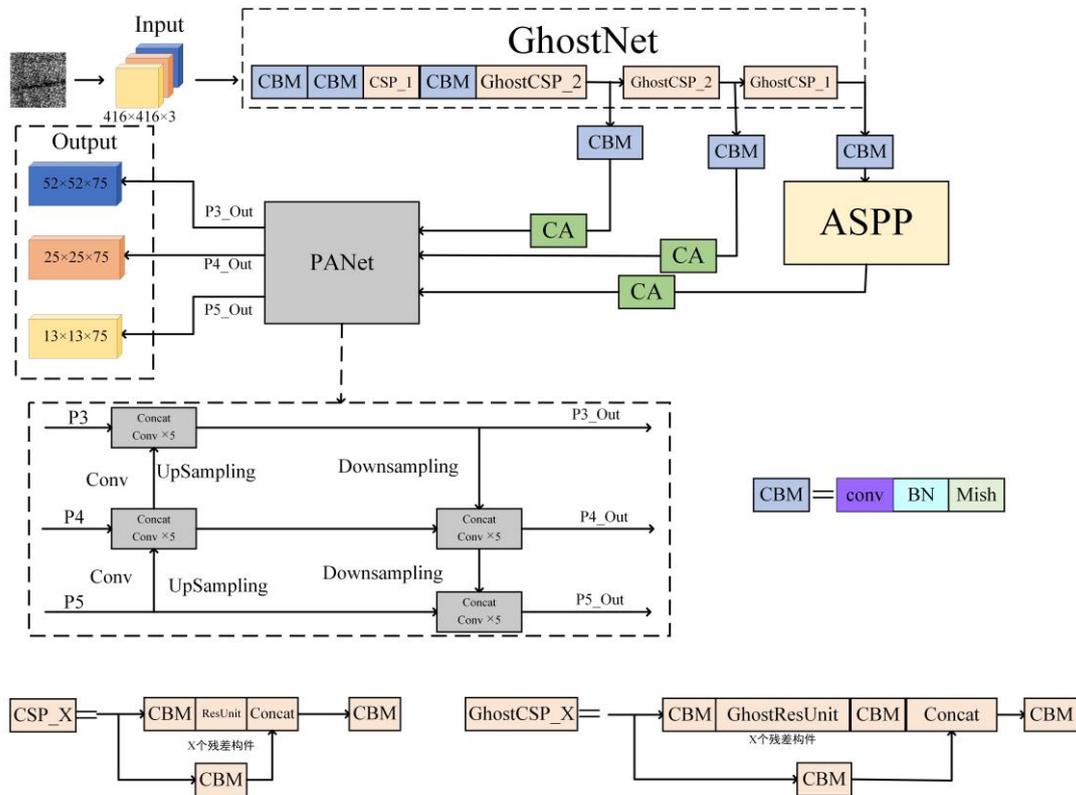


Figure 3. YOLO-ACG network model.

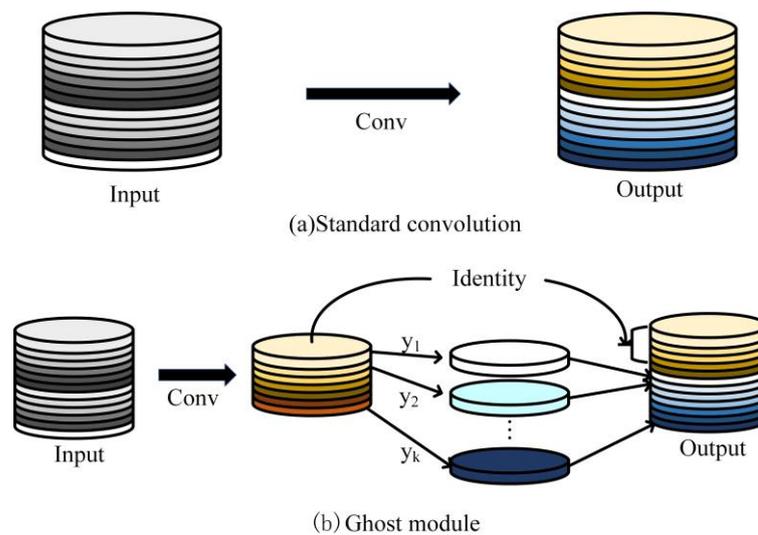


Figure 4. Standard convolution and Ghost module.

To reflect the benefits of Ghost convolution in convolutional computing, the input feature map's width, height, and channel count are all assumed to be w , h , and c , respectively. The output after one convolution is $n * h' * w'$, where k and d are the sizes of the linearly variable convolution kernel and standard convolution kernel, respectively. Equation (4) illustrates that the amount of convolution computation performed in the Ghost module serves as the denominator and the numerator of the equation, respectively.

Comparing the two convolutional calculations reveals that for the same parameters, the normal convolutional computation is s times larger than the convolutional computation in the Ghost module.

$$\begin{aligned} r_s &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s+c-1} \approx s \end{aligned} \quad (4)$$

3.3. Improved ASPP Module

The SPP [27] structure serves as the pyramidal pooling module for the whole YOLOv4 network. Large numbers of picture features must be stored in the SPP structure, and feature extraction from the feature map requires laborious multi-stage training that takes too long. The ASPP structure addresses the drawbacks of SPP by substituting the pooling process in the SPP structure with null convolution, which is typically utilized as the global feature extraction of the feature image and can be employed in the feature map with emphasis on the picture, preventing the loss of image data, although improving the semantic segmentation ability of the feature map and enhancing the perceptual field also results in the loss of information details on the edges of the feature map. The ASPP structure is a good answer to the aforementioned issue since it increases the perceptual field of the feature map without losing the finer details of the edge information.

The ASPP structure has two parts, the first part consists of an 1×1 convolutional layer and three 3×3 null convolutional layers with sampling rates of [6,12,18], respectively, whose convolutional kernels have a size of 256; the second part is a convolutional operation of 1×1 by global level pooling, and the same convolutional kernels also have a size of 256. Figure 5 depicts the ASPP module's structural layout. The method of null convolution and upsampling and the multiscale structure are used to realize the feature extraction of images in the environment of high resolution and perceptual field. This makes a significant improvement in the perceptual field of the feature image and the processing of the details of the edge of the feature image. The expansion rate is introduced in the convolution layer, expressed as the number of zero values in the convolution kernel.

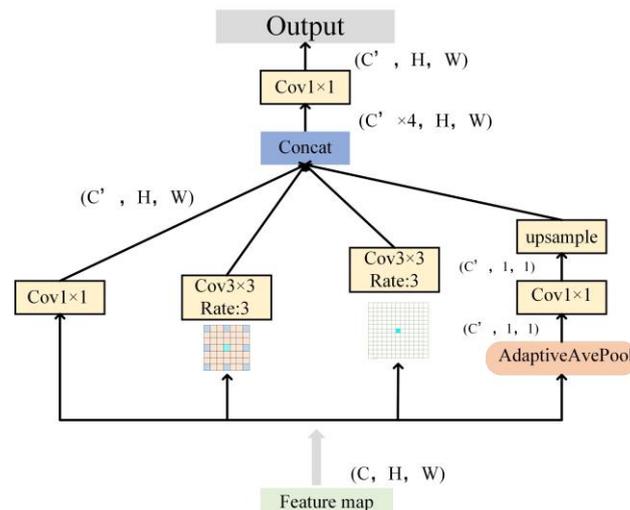


Figure 5. ASPP module.

3.4. CA Attention Mechanism Module

It was discovered throughout the target detection procedure that there was a lack of effectiveness in the detection of subtle faults. To solve this issue, the network was enhanced with the CA (coordinate attention) spatial attention mechanism [28], which increases the significance of the location relationship and cross-latitude interaction on the channel attention mechanism and improves the accuracy and sensitivity of the entire network model to the information and location of the defective targets. To solve this

issue, the CA (coordinate attention) spatial attention mechanism was incorporated into the network to increase the significance of its location relationship and cross-latitude interaction on the channel attention mechanism, making the entire network model more accurate and sensitive to the information and location of the defective targets. Figure 6 depicts the structure of the added CA attention network, which involves pooling the feature maps globally to obtain feature information in both directions. Sub-associative fusion and 1×1 convolutional transforms were then used to account for feature variation; finally, the integrated feature maps were divided into two feature maps with an equal number of channels by two 1×1 convolutions through transformation before being output by the added CA attention network. The CA module encodes the feature map's precise location to produce the width and height, such as the output concatenate for feature fusion, which is represented in Equation (5); Equation (6) represents the feature transformation of two independent features to make the input's dimensionality consistent; combining g^n and g^m to create a weight matrix in Equation (7) represents the outcome.

$$f = \beta(F([z^n, z^m])) \quad (5)$$

$$g^n = \delta(F_n(f^n)), g^m = \delta(F_m(f^m)) \quad (6)$$

$$y_a(i, j) = x_a(i, j) \times g_c^n(i) \times g_c^m(j) \quad (7)$$

where f denotes the mapped feature map, β denotes the nonlinear activation function, z^n and z^m denote the horizontal and vertical position relationship, g^n and g^m denote the feature map after the sigmoid output of two identical number of channels. Finally, x_a denotes the connected jump feature information.

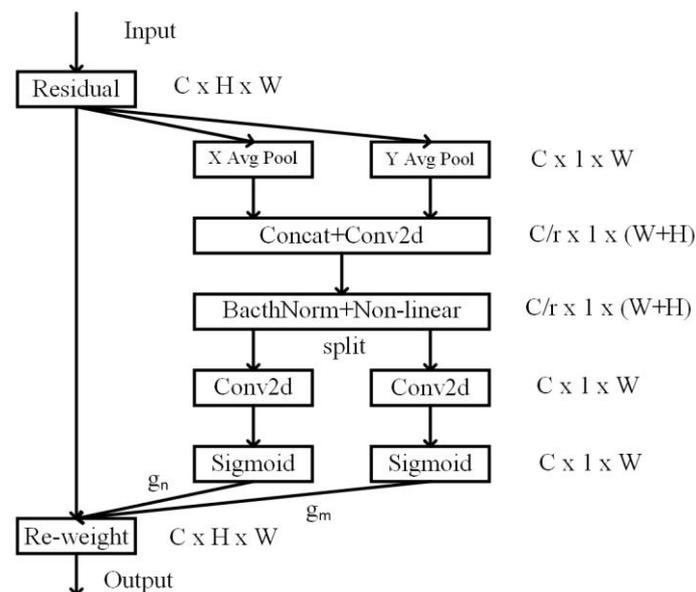


Figure 6. CA attention network.

4. Experimental Preparation

4.1. Test Environment

The experimental platform is Win10 OS, CPU is 12th Gen Intel(R) Core(TM) i7-12700KF 3.60 GHz; memory is 64 GB; GUP is NVIDIA GeForce RTX3090Ti. Pytorch 1.10.2 is used; the software runs in Anaconda 3.6; Cuda 10.0 and Cudnn 7.5 were installed to help speed up the GUP process, and Tensorflow 1.13.1, Opencv4.1, and Numpy 1.14.2 were installed in the environment. The auxiliary databases were installed to make the code run correctly.

4.2. Production of Data Set

In terms of data set, it was discovered that the three types of flaws that have the most impact on the quality of steel plate during the identification of defects in steel plate were weld, hole, and scratch. The number of feature photos offered by these three categories of steel plate defects was insufficient to meet the training needs, even though the existing data collection of the surface defects of German DAGM steel plate has a total of 10 types of steel plate defects. Therefore, the data set was expanded by combining the actual scene shooting with the public data set. A total of 4500 defect feature images were obtained from the entire self-made data set through on-site collection and selection of public data sets. Because it was discovered that the format and size of the feature map would influence the detection efficiency during the detection process, LabelImg labeling software was used to label the area of each type of image in proportion to the area to be labeled, and the length and width of the image were less than or equal to 3:1. This will help to better train the neural convolution network model. XML files should be used to store critical data from marked defect boxes in order to apply it to neural convolution network learning.

The target's proportion in the image varies slightly as a result of variations in the camera's viewing distance, and the model's capacity for adaptation is reduced by the various target sizes. The method of random scaling, clipping, and distribution of the logarithmic data set, which is more accurate for the training of the data set, was utilized to carry out the random splicing of photos in the preprocessing stage to tackle this problem. Figure 7 displays an illustration of data enhancement.

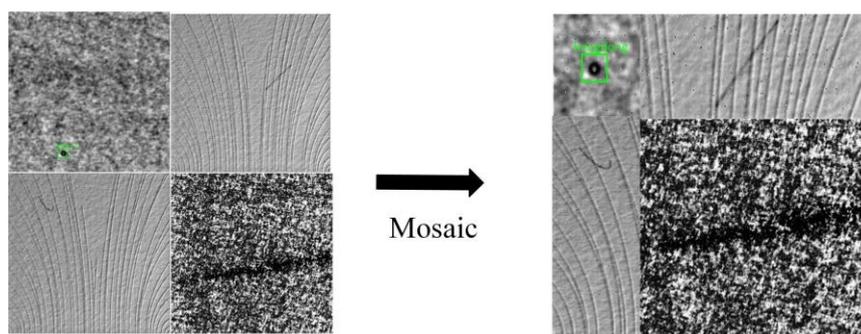


Figure 7. Example of data enhancement.

5. Results and Discussion

The self-made data set utilized in this experiment randomly separated the training set and test set at a ratio of 6:4. Several ablation experiments were then set up to assess the effects of each model improvement on the training effect and to choose the best model. The usefulness of the proposed algorithm was further confirmed by several sets of comparison experiments with the better steel defect detection methods already in use, and the superiority of the algorithm was assessed by the average detection accuracy (map) and the detection speed (FPS).

5.1. Training Model

The experiments in this paper were carried out in accordance with the predetermined parameters: the image input size was 416×416 ; the epoch was set to 300 rounds; the batch size was 128 for the first 50 rounds and 64 for the last 250 rounds; the learning rate was 1×10^{-2} for the first 50 rounds and 1×10^{-3} for the last 250 rounds; momentum was the amount of stochastic gradient descent in order to obtain a better convergence effect, momentum was set at 0.937. Momentum is the expression of the reduction in the learning efficiency of the initialization of the weights in terms of momentum. Figure 8 depicts the loss curve during the training phase.

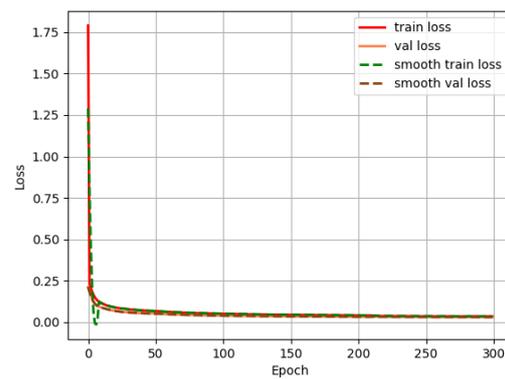


Figure 8. Loss curve during training.

It can be observed from the figure that the loss value decreases continuously as the epoch increases, and when the training proceeds to 50 rounds the loss curve is basically in a stable state without generating an overfitting situation, and the loss value of the YOLO-ACG algorithm converges to about 0.19 and 0.14 by increasing the accuracy of the recognition model, so the overall parameter setting of the algorithm is reasonable.

5.2. Comparison Experiment

The YOLO-ACG method is compared with current popular detection techniques, such as YOLOv4, YOLOv4-MobileNetv1, YOLOv4-MobileNetv2 [29], YOLOv4-MobileNetv3, YOLOv3-tiny and YOLOv4-tiny, in a self-made data set, as shown in Table 1, to confirm that the algorithm's improvement is more genuine and reliable.

Table 1. Data set to compare the experimental results.

Model	mAP	Model Size/MB	FPS
YOLOv4	96.35%	244.7	85.3
YOLOv4-MobileNetv1	88.39%	40.95	47.6
YOLOv4-MobileNetv2	89.52%	39.06	40.1
YOLOv4-MobileNetv3	89.75%	39.99	43.2
YOLO-ACG	92.49%	69.82	102.91

The chart shows that when employing a big network model, such as the YOLOv4 [30] detection method, the algorithm has a very high detection accuracy of 96.35% but the model's size is relatively large at 244.7 MB, making it difficult to deploy on mobile devices. Some lightweight models, Models 2 to 6 in Table 1, enhance the YOLO algorithm. The comparison results reveal that the lightweight model's size has been significantly decreased in comparison to YOLOv4, which is more readily implemented in mobile devices, but that the model's detection accuracy has been significantly reduced in comparison to the YOLOv4 algorithm.

In view of this, the YOLO-ACG network model proposed in this paper takes into account the computational speed of the model, the detection accuracy, and the size of the model. Its model size is about 1/4 times that of the YOLOv4 network model, although slightly higher than that of Table 2–6 models, about 1/3 times that of 2–6 network models. The suggested approach has unquestionable advantages in terms of operation speed, reaching up to 103FPS. It surpasses roughly 18FPS, as compared to YOLOv4. It exceeds by almost two times 2–6 models, realizing high-speed detection. The accuracy of the YOLOv4-ACG model is around 2% greater than that of the models in Table 2–6, although being about 4% lower than that of the YOLOv4 model. From the above three aspects, we can see that YOLO-ACG is more efficient when deployed on mobile devices.

5.3. Ablation Experiment

The ablation experiment is to improve different modules based on the YOLOv4 Ghost algorithm and use self-made data sets to conduct training and performance evaluation. Table 2 shows the comparison of evaluation results of all models.

Table 2. Training results of different algorithm models.

Experiment	SPP	ASPP	SE	ECA	CBAM	CA	mAP	FPS	Size/MB	Recall
1	✓						89.17%	95.88	43.63	67.34%
2	✓		✓				88.49%	96.53	44.25	71.42%
3	✓			✓			88.37%	95.17	44.61	71.91%
4	✓				✓		87.84%	96.01	44.26	70.42%
5	✓					✓	88.61%	97.88	43.84	68.77%
6		✓					91.64%	97.89	69.57	75.81%
7		✓	✓				91.09%	94.28	70.23	76.52%
8		✓		✓			90.49%	96.53	69.63	72.26%
9		✓			✓		89.76%	95.36	70.26	72.72%
10		✓				✓	92.49%	102.91	69.12	77.49%

As can be seen from the table, experiments conducted with the introduction of SPP and ASPP modules in the model reveal that the model size of the algorithm is approximately 1.5 times larger when the ASPP module is introduced than when the SPP module is introduced. In terms of accuracy, it is about 2% higher than the SPP module. In terms of recall rate, there is a significant improvement of approximately 8% compared to the SPP module. The detection speed is also improved compared with the introduction of the SPP module, which can reach about 98FPS. The comparison of the total ablation experiment reveals that, despite the model's size being only slightly larger—by about a third—than that of the method with the SPP module, it has far faster and more accurate detection. As a result, the ASPP module is elected as the algorithm's primary pooling layer. It can be shown from comparative tests 6–9 that the accuracy and speed of the algorithm in the detection algorithm with only the ASPP module are marginally improved, while the model size is slightly decreased, when compared to the introduction of ASPP module and the addition of SE [31], ECA, and CBAM. By comparing Experiments 6 to 10, it can be seen that Experiment 10's model size is only slightly smaller than Experiments 6 to 9's. In terms of recall, Experiment 10 exceeded Experiments 6 to 9 by about 2% to 5%, but experiment 10's precision is about 3% to 4% higher, and Experiment 10's speed is about 103 FPS, which is higher than Experiment 6 to 9's speed by about 6 to 9 FPS.

The aforementioned studies demonstrate that the revised YOLO-ACG algorithm approaches are efficient and increase the model's accuracy in the detection of steel plate surface flaws. In order to identify steel plate surface flaws more effectively, it possesses the qualities of quick detection speed, lightweight models, and ease of deployment in real-world situations.

6. Conclusions

This study suggests the YOLO-ACG method addresses the shortcoming of the YOLOv4 algorithm in handling flaw identification of steel plate data. From the three points below, the algorithm has been improved. First, lower the model size and substitute the present YOLOv4 method's backbone network with the lightweight Ghost module to make the algorithm simple to install on mobile devices. Then, to increase the maximum pooling efficiency of the YOLOv4 algorithm, the ASPP module is introduced to replace the maximum pooling layer. This considerably enhances the processing of the feature image's edge details and the feature image's receptive field. Finally, employing the pyramid feature fusion network CA module enables the enhancement of feature map fusion effectiveness in various scale spaces and further enhances the analysis of feature maps' edge information.

From the analysis of the experimental results, the proposed YOLO-ACG target detection algorithm applied to a homemade data set has a higher mAP of about 3% compared

to the existing YOLOv4-MobileNet algorithm model. In comparison with the size of the YOLOv4 algorithm model, the proposed algorithm is about 1/4 of the YOLOv4 algorithm model. The detection speed of the YOLO-ACG algorithm can reach about 103 FPS, which is twice as fast as the existing YOLOv4-MobileNet algorithm model. Therefore, YOLO-ACG target detection has significantly improved the ability to detect defects on the surface of steel plates and meet the mobile requirements for the real-time detection of realistic scenes.

Author Contributions: Conceptualization, C.W. and M.S.; methodology, C.W.; software, M.S.; validation, M.S.; formal analysis, C.W.; investigation, M.S.; resources, C.W.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, C.W., Y.C., B.Z., K.H., Z.C. and M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported by the National Natural Science Foundation of China (Fund Numbered 52177004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data sets used and analyzed in the current study are available from the corresponding author on reasonable request.

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

References

- Chen, K.; Li, H.; Li, C.; Zhao, X.; Wu, S.; Duan, Y.; Wang, J. An Automatic Defect Detection System for Petrochemical Pipeline Based on Cycle-GAN and YOLO v5. *Sensors* **2022**, *22*, 7907. [[CrossRef](#)] [[PubMed](#)]
- Dai, J.; Li, T.; Xuan, Z.; Feng, Z. Automated Defect Analysis System for Industrial Computerized Tomography Images of Solid Rocket Motor Grains Based on YOLO-V4 Model. *Electronics* **2022**, *11*, 3215. [[CrossRef](#)]
- Jung, H.; Rhee, J. Application of YOLO and ResNet in Heat Staking Process Inspection. *Sustainability* **2022**, *14*, 15892. [[CrossRef](#)]
- Zhao, Z.; Ge, Z.; Jia, M.; Yang, X.; Ding, R.; Zhou, Y. A Particleboard Surface Defect Detection Method Research Based on the Deep Learning Algorithm. *Sensors* **2022**, *22*, 7733. [[CrossRef](#)] [[PubMed](#)]
- 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](#)]
- Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated visual defect detection for flat steel surface: A survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644. [[CrossRef](#)]
- Shi, T.; Kong, J.; Wang, X.; Liu, Z.; Zheng, G. Improved Sobel algorithm for defect detection of rail surfaces with enhanced efficiency and accuracy. *J. Cent. South Univ.* **2016**, *23*, 2867–2875. [[CrossRef](#)]
- Thomas, B.G.; Jenkins, M.S.; Mahapatra, R.B. Investigation of strand surface defects using mould instrumentation and modelling. *Ironmak. Steelmak.* **2004**, *31*, 485–494. [[CrossRef](#)]
- Liu, Y.; Xu, K.; Wang, D. Online surface defect identification of cold rolled strips based on local binary pattern and extreme learning machine. *Metals* **2018**, *8*, 197. [[CrossRef](#)]
- Wang, Z.; Zhu, D. An accurate detection method for surface defects of complex components based on support vector machine and spreading algorithm. *Measurement* **2019**, *147*, 106886. [[CrossRef](#)]
- Kang, G.W.; Liu, H.B. Surface defects inspection of cold rolled strips based on neural network. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 8, pp. 5034–5037.
- Di, H.; Ke, X.; Peng, Z.; Zhou, D. Surface defect classification of steels with a new semi-supervised learning method. *Opt. Lasers Eng.* **2019**, *117*, 40–48. [[CrossRef](#)]
- Schlegl, T.; Seeböck, P.; Waldstein, S.M.; Schmidt-Erfurth, U.; Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 146–157.
- Lee, S.Y.; Tama, B.A.; Moon, S.J.; Lee, S. Steel surface defect diagnostics using deep convolutional neural network and class activation map. *Appl. Sci.* **2019**, *9*, 5449. [[CrossRef](#)]
- Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **2020**, *31*, 759–776. [[CrossRef](#)]
- Prappacher, N.; Bullmann, M.; Bohn, G.; Deinzer, F.; Linke, A. Defect detection on rolling element surface scans using neural image segmentation. *Appl. Sci.* **2020**, *10*, 3290. [[CrossRef](#)]
- Peng, K.; Zhang, X. Classification technology for automatic surface defects detection of steel strip based on improved BP algorithm. In Proceedings of the 2009 Fifth International Conference on Natural Computation, Tianjin, China, 14–16 August 2009; pp. 110–114.

18. Wang, Y.X.; Guang-Hui, Y.U.; Qiang, X.U. A Machine Vision Based Printing Defect Detection Technology for Product Packaging. *J. Jiangsu Univ. Technol.* **2019**, *25*, 7–14.
19. Wang, Z.Y. Research on steel plate surface defects detection method based on machine vision. *Comput. Modern* **2013**, *7*, 97–117.
20. Wang, L.; Wei, C.; Li, W.; Zhang, Y. Pedestrian detection based on YOLOv2 with pyramid pooling module in underground coal mine. *Comput. Eng. Appl.* **2018**, *55*, 133–139.
21. Yang, W.; Zhou, G.L.; Gu, Z.W.; Jiang, X.D.; Lu, Z.M. Safety Helmet Wearing Detection Based on an Improved Yolov3 Scheme. *Int. J. Innov. Comput. Inf. Control.* **2022**, *18*, 973–988.
22. Liu, M.; Mao, J. Deep Face Recognition Algorithm Based on Improved Mobilenet Algorithm. *Inf. Commun. Technol.* **2019**, *1*, 41–46.
23. Cheng, S.; Zhou, B. Recognition of Characters in Aluminum Wheel Back Cavity Based on Improved Convolution Neural Network. *Comput. Eng.* **2019**, *45*, 182–186.
24. 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, 26 June–1 July 2016; pp. 779–788.
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
26. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
27. Yang, K.; Jiao, Z.; Liang, J.; Lei, H.; Li, C.; Zhong, Z. An application case of object detection model based on Yolov3-SPP model pruning. In Proceedings of the 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 24–26 June 2022; pp. 578–582.
28. 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, Virtual, 19–25 June 2021; pp. 13713–13722.
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
30. Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
31. 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.

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.