

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.

Article

Application Study of YOLOv5 Algorithm on Automotive Wheel Surface Defect Detection

Xiuye Xia, Zhenyu Zhang and Fucai Liu *

Engineering Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 066004, China; xxy245@stumail.ysu.edu.cn (X.X.); zzy_kz@stumail.ysu.edu.cn (Z.Z.)

* Correspondence: lfc_xb@263.net

Abstract: Surface defect detection is a crucial step in the process of automotive wheel production. However, the task possesses challenges due to complex background and a wide range of defect types. In order to detect the defects on the wheel surface accurately and quickly, this paper proposes a YOLOv5-based algorithm for automotive wheel surface defect detection. The algorithm trains and tests the YOLOv5s model using the self-created automotive wheel surface defect dataset, which contains four kinds of defects: linear, dotted, sludge, pinhole. The extensive experimental results demonstrate that the deep learning network trained by our method can achieve an average accuracy of 71.7% and 57.14 FPS. Our findings prove that this detection algorithm performs better than other common target detection algorithms and meets the real-time requirements of industrial applications.

Keywords: wheel surface defect detection; deep learning; YOLO; object detection; machine vision

1. Introduction

As an important component of automotive, the wheel surface may encompass defects due to various factors during the production process, which may affect its appearance, performance and safety[1]. Therefore, strict quality inspection is necessary before utilization, considering the safety of drivers and passengers. After our fieldwork at the factory, we found that wheel surface defect detection mainly relies on manual inspection. The highly repetitive work, higher labor intensity and the experience level of the workers all have an impact on the quality of the inspection [2]. Even skilled workers have to complete a wheel inspection within 20-30 seconds. Consequently, manual inspection is increasingly difficult to meet the requirement of fast, accurate and stable inspection. Therefore, there exists an immediate necessity to develop automated intelligent detection methods, such as the application of machine vision and artificial intelligence technologies to improve detection efficiency.

Target detection algorithms can be broadly categorized into traditional target detection algorithms and deep learning-based target detection algorithms. Traditional target detection algorithms have three components that detect the targets including image preprocessing, feature extraction, classification and recognition [1]. Patil et al. [3] used support vector machines and artificial neural networks to classify from various defect classes, achieving 92% accuracy in detecting weld defects. Liu et al. [4] proposed a semisupervised learning (SSL) framework and a multiclass Laplacian support vector machine (MCL SVM), which were applied to defect detection techniques to deal with the insufficient labeled samples. Jin et al. [5] presented a defect detection algorithm based on Relief algorithm and Adaboost-SVM to address the problems of poor generality and low accuracy of defect detection techniques, which is superior compared to other classifiers. Mao et al. [6] introduced an algorithm based on the histogram of oriented gradients (HOG) and support vector machine (SVM) to identify defects in transmission lines. The final method is able to achieve a recognition accuracy of 84.3% and a processing speed of 0.539s per image.

However, traditional target detection algorithms mainly execute manual feature extraction, which has limitations and is not very robust to variable features. With the development of artificial intelligence and the iteration of computer hardware, deep learning target detection algorithms gradually replace



traditional target detection algorithms by virtue of their strong learning ability, good adaptability to data sets, and high portability. Target detection algorithms can be broadly classified into two-stage target detection algorithms and one-stage target detection algorithms. Representative two-stage target detection algorithms are R-CNN [7], Fast R-CNN [8], and Faster R-CNN [9], while representative one-stage target detection algorithms are SSD [10] and YOLO [11]. Cheng et al. [12] achieved 78.25% accuracy by embedding a novel channel attention mechanism and adaptive spatial feature fusion (ASFF) module into RetinaNet. Zhao et al. [13] changed the network structure of the Faster R-CNN and replaced the convolutional network with deformable convolution network, achieving 75.2% mAP on the NEU-DET dataset. Zhi et al. [14] proposed an improved SSD chip defect inspection method that replaces VGGNet with ResNet while eliminating the first pooling layer; additionally, a prediction and deconvolution modules were included, and the hidden layer was pruned. The model achieves 78.2% mAP. Cui et al. [15] proposed SDDNet with the feature retaining block (FRB) and skip densely connected module (SDCM). Both modules achieved good results on publicly available datasets and are expected to be applied to industrial production. Li et al. [16] used the YOLOv4 algorithm to achieve 67.09% mAP on the NEU-DET dataset. Gao et al. [17] proposed a novel defect detection method. The feature collection and compression network (FCCN) with Gaussian weighted pooling (GWP) achieved an average accuracy of 80.2% and 33 FPS on NEU-DET. He et al. [18] proposed an inspection system for steel plate defect detection, which used a baseline convolutional neural network (CNN) with multilevel feature fusion network (MFN) and region proposal network (RPN). The system achieved 82.3% detection accuracy. Ying et al. [19] improved the YOLOv5s model by using the K-means++ algorithm and replacing the origin loss with focal loss. Besides, the efficient channel attention mechanism (ECA) was added, achieving 92.2% accuracy on the wire braided hose surface defect dataset.

Most surface defects can be categorized into distinct types. In this paper, we mainly focus on four kind of defects: linear defects, sludge defects, dotted defects and pinhole defects, which are shown in Figure 1. Unlike traditional surface defects (such as steel surface defects), the detection of wheel surface defects presents more challenges due to the curved surface and complex background of the wheels. What's more, the hollowed-out parts under different shooting angles may cause interference. Among the four defects mentioned above, the area of dotted defects is relatively small and is difficult to be detected. Pinhole defects appear in clusters and are not fixed in shape and size. While traditional target detection algorithms are limited by their poor generalizability and other shortcomings to achieve better detection results. Based on the analysis above, the YOLOv5 algorithm was chosen as the target detection algorithm in this paper. The algorithm has demonstrate a good detection effect on small targets while owning fast detection speed and high detection accuracy. Our multiple experiments have shown that the YOLOv5 algorithm is able to achieve an average accuracy of 71.7% and 57.14 FPS on the self-built wheel surface defect dataset, which has fast detection speed and great improvement potential.

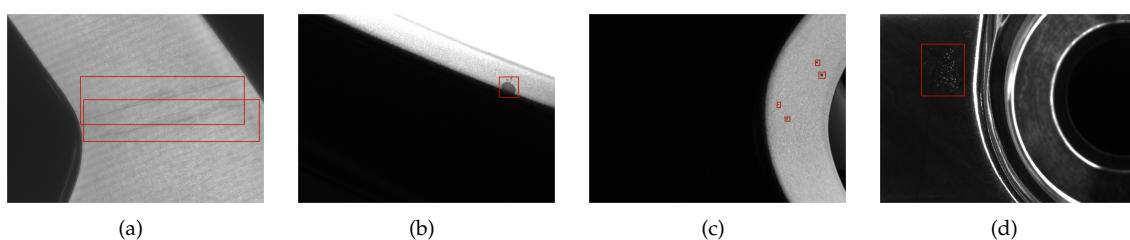


Figure 1. Different kinds of wheel surface defects. Defects have been marked in the pictures. (a) linear (b) sludge (c) dotted (d) pinhole.

2. Related work

2.1. Development of YOLO

The YOLO algorithm was first introduced in 2015. The algorithm treats the target detection problem as a regression problem and uses the entire image as input to the network. Besides, the location and class of the bounding box are obtained after the calculation of the network. However, the YOLO algorithm is less effective in detecting group targets and small targets that are close to each other. In 2017 the author Redmon proposed YOLO9000 and YOLOv2 [20] based on YOLOv1. The algorithm includes convolution with anchor frames, batch normalization, and a new feature extraction network called Darknet-19. These improvements significantly enhance the performance of YOLO compared to YOLOv1. In 2018, the authors proposed YOLOv3 [21] by replacing Darknet-19 with Darknet-53 on the basis of YOLOv2. This algorithm uses a feature pyramid network structure and is able to detect targets at multiple scales. In 2020, Alexey Bochkovskiy et al. [22] proposed YOLOv4 which combines Darknet-53 with CSPNet, adds an SPP pooling layer, and introduces Path Aggregation Network(PANet). YOLOv4 has significantly improved detection speed and accuracy, and can be run efficiently on common GPUs.

2.2. Feature Extraction Network

Convolutional neural networks are critical components in deep learning. After continuous improvement, classical neural networks such as AlexNet, VGGNet, GooLeNet, ResNet, and DenseNet [23] have emerged. AlexNet replaces the activation function of CNN with ReLU and uses Dropout to prevent overfitting. VGGNet has deeper layers and introduces the concept of the convolutional block. GooLeNet enhances the convolutional module function. ResNet adds the short connection structure to prevent the network from gradient disappearance and increases the network depth. DenseNet uses the dense connection to improve the backpropagation of gradient and make the network easier to train. In YOLOv5, the backbone network CSPDarknet53 added the Cross Stage Partial (CSP) module to Darknet53, which enhances the learning ability of convolutional neural network and reduces the computational bottleneck and memory cost.

2.3. Multi-Scale Features and Feature Fusion Network

In target detection, the scale variation of the target is a problem that cannot be ignored. There are different kinds of defects with different shapes in defect detection, some of which are slender (such as the linear defects in Figure 1 (a)) and some of which are small (such as the dotted defects in Figure 1 (c)), which pose a challenge to the localization and identification of target defects. In the process of feature extraction, the location information in the shallow features has higher resolution but less semantic information, while the deep features have stronger semantic information but lower resolution of location information. During the process of feature extraction, small target defects are easily disturbed by the background information and feature information is easily lost. The feature fusion module can well fuse the two to improve the detection effect of small targets. The feature pyramid network constructs a top-down structure with lateral connections [24], which fuses semantic features and location information to improve the performance of the network. Later, more feature fusion networks have been developed, such as bi-directional feature pyramid network (BiFPN) [25], PANet [26], etc.

3. Method

3.1. YOLOv5

YOLOv5 was released in June 2020. The YOLOv5 target detection model comprises four versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Despite their shared network structure, the models

differ only in terms of their depth and the width of their feature maps. Of the four versions, YOLOv5s has the smallest network depth and also has the fastest detection speed.

The YOLOv5 network structure comprises several components:

(1) CBS module: It consists of Conv+BN (Batch Normalization) + SiLU activation function, which is the basic component in the YOLOv5 network structure.

(2) Res unit module: It borrows from the residual structure of the ResNet network, which increases the depth of the network and enhances the learning ability of the network.

(3) C3 module: There are two types of C3 modules in YOLOv5 model: C3 1_X and C3 2_X. The former is only applied in the backbone and the latter is only applied in the neck. Each C3 1_X contains X Res units, while each C3 2_X contains 2*X Res units.

(4) Spatial pyramid pooling-fast (SPPF) module: The SPPF module cascades multiple small-size pooling kernels, which can fuse feature maps of different sensory fields, enriching the feature map expression capability and improving the operation speed.

There are two types of feature fusion in YOLOv5:

(1) Concat operation: The number of channels is merged. Only the number of features describing the image itself is increased, and the amount of information is not increased.

(2) Add operation: The feature maps are merged. Add can be seen as a special Concat, but with less computational effort.

The network structure of YOLOv5s is shown in Figure 2. The backbone part contains five CBS modules, while the convolution kernel size in each CBS module is 3*3 and the step size is 2, which plays the role of downsampling. If the input image size of the network is 640*640, the feature map changes as 640-320-160-80-40-20. The FPN+PAN structure is used in the neck part, where the FPN propagates top-down to add semantic features and the PAN propagates bottom-up to add positional features. The combination of the two improves the performance of the network. The prediction (head) part consists of three detectors, which output three feature maps respectively.

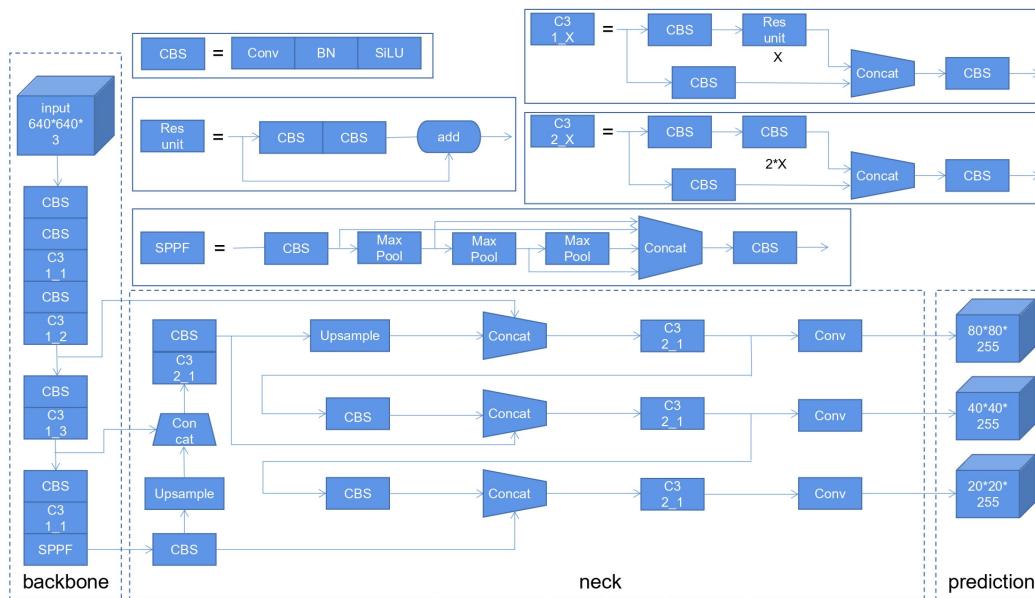


Figure 2. YOLOv5s network structure. The network takes the COCO dataset as input, with dimensions set to 640*640 pixels.

3.2. CIoU Loss

CIoU [27] is the improvement of IoU. The definition of IoU is as follows:

$$IoU = \frac{A \cap B}{A \cup B} \quad (1)$$

where we assume that A presents the prediction box and B presents the ground truth box. It is calculated as the ratio between the intersection and the union of A and B, obviously, the larger the value of IoU means the more accurate the model predicts. IoU has non-negativity and scale invariance, but IoU cannot reflect the distance between the boxes, so IoU is always 0 when the two boxes have no intersection. When IoU is equal but the overlap cases are different, IoU also cannot accurately reflect the magnitude of the overlap. As an improvement of IoU, CIoU considers three important factors at the same time: The overlapping area of the prediction box and the ground truth box, the centroid distance, and the aspect ratio. As a result, CIoU improves the convergence speed and enhances target-edge regression stability.

The definition of CIoU_Loss is as follows:

$$\begin{aligned} CIoU_Loss &= 1 - CIoU \\ &= 1 - \left(IoU - \frac{Distance_2^2}{Distance_C^2} \right. \\ &\quad \left. - \frac{\nu^2}{(1 - IoU) + \nu} \right) \end{aligned} \quad (2)$$

where Distance_2 is the Euclidean distance between the two center points of the prediction box and the ground truth box, and Distance_C is the diagonal distance of the minimum outer rectangle of the two boxes.

The definition of ν is as follows:

$$\nu = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w^p}{h^p} \right)^2 \quad (3)$$

where w^{gt} and h^{gt} are the width and height of the ground truth box, w^p and h^p are the width and height of the prediction box, respectively.

3.3. Loss Function

The loss function of YOLOv5 is divided into three parts, which are bounding box regression loss, target confidence loss and category loss. The loss function is formulated as follows:

$$L_{total} = L_{box} + L_{obj} + L_{cls} \quad (4)$$

$$L_{box} = \lambda_1 \frac{\sum_{i \in pos} (1 - CIoU_i)}{N_{pos}} \quad (5)$$

$$L_{obj} = -\lambda_2 \frac{\sum_{i=0}^N (o_i \ln(\hat{c}_i) + (1 - o_i) \ln(1 - \hat{c}_i))}{N} \quad (6)$$

$$L_{cls} = -\lambda_3 \frac{\sum_{i \in pos} \sum_{j \in cls} (O_{ij} \ln(\hat{p}_{ij}) + (1 - O_{ij}) \ln(1 - \hat{p}_{ij}))}{N_{pos}} \quad (7)$$

N_{pos} is the number of positive samples in Equation (5). In Equation (6), o_i is the IoU between the prediction box and the ground truth box, \hat{c}_i is the confidence of whether the prediction box contains the target after the activation function. N is the number of anchor box. In Equation (7), O_{ij} is the IoU between the prediction box and the ground truth box for the i -th positive sample responsible for the j -th class of targets, and \hat{p}_{ij} is the probability of the existence of the j -th class of targets in the i -th bounding box obtained by the activation function. $\lambda_1, \lambda_2, \lambda_3$ are the weight parameters.

4. Experiments

4.1. Datasets and Preprocessing

Our dataset was acquired from the factory field by placing the wheels in a black box and illuminating them with light from various angles. After cropping and filtering, we extracted the initial dataset that consisted of 400 grayscale images including four types of wheel surface defects: linear, dotted, sludge, and pinhole defects. There were 100 samples for each defect, totally 400 grayscale images, with the original resolution of 960*704. For object detection, each defect is manually labeled using labelimg to obtain a txt label file containing information of defect types and defect locations.

The shadows from different angles of illumination may affect the training, which put forward high requirements for the model. In order to prevent the model from falling into overfitting too early, we enhanced and expanded the dataset by using rotation of 90 degrees, 180 degrees, 270 degrees, horizontal flip and vertical flip. Some of the defects after augmentation are shown in Figure 3. The original 400 images were expanded to 2400 images. Before training, we divided the expanded dataset into training set, validation set and test set at a ratio of 7:2:1, containing 1680, 480 and 240 defective images, respectively. In addition, during training, we employed data augmentation operations (mosaic data enhancement and image panning) on the training set simultaneously to enhance the generalization of the model.

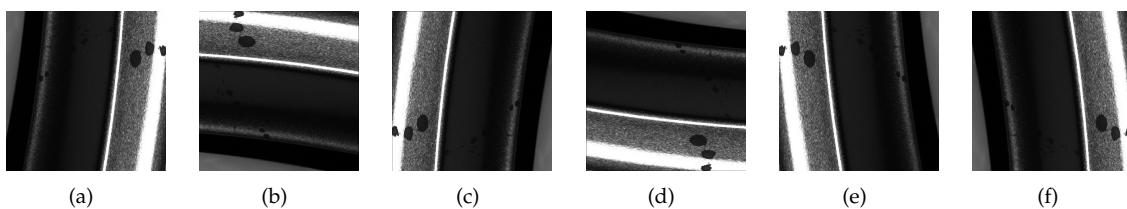


Figure 3. Data augmentation. (a) origin defect (b) rotate 90 degrees (c) rotate 180 degrees (d) rotate 270 degrees (e) horizontal flip (f) vertical flip.

4.2. Experimental Environment and Implementation Details

This experiment uses the Sugon remote computing platform with a 7285-32C-2.0GHz processor, 128G of memory, a 200Gb computing network, and four heterogeneous acceleration cards. Each experiment occupies 4 nodes, 32 CPUs and 4 DCU acceleration cards. The experiments are carried out on the PyTorch 1.10.0 deep learning framework and Python 3.7.9.

In the training phase, the main parameters to be set are learning rate, batch size and epoch. The learning rate determines whether the objective function can converge to a local minimum at the appropriate time. Batch size determines the number of samples chosen for one training, and a suitable batch size can make full use of the memory and make the direction of gradient descent more accurate. An epoch refers to a period where all samples complete one round of training, and many rounds of training are often needed for one training. Our model chooses Stochastic Gradient Descent (SGD) as the optimizer, and the value of momentum and weight decay is set to 0.937 and 0.0005. The initial parameters include a learning rate of 0.01, a batch size of 32, an epoch of 100, and input image size set at 960*960.

4.3. Evaluation Metrics

In order to evaluate the performance of the network, we introduce the following common evaluation metrics:

- (1) Precision and Recall

Precision represents how many of the samples with positive predictions are true positive samples, and recall represents how many of the original samples with positive predictions are correctly predicted. The formulas for accuracy and recall are shown below:

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (8)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (9)$$

where TP is the defect predicted by the model as a defect, FP is the background predicted by the model as a defect, FN is the defect predicted by the model as a background, and TN is the background predicted by the model as a background.

(2) AP and mAP

The Average Precision (AP) is a composite metric that combines accuracy and recall. The mean Average Precision (mAP) is the average of AP values for different categories of defects. The formulas for AP and mAP are shown below:

$$AP = \int_0^1 p(r)dr \quad (10)$$

$$mAP = \frac{\sum_{i=1}^K AP_i}{K} \quad (11)$$

where K is the number of categories. When multiple categories exist in the dataset, mAP is the average of the AP of each category.

(3) FPS

The algorithm's performance is also evaluated based on its speed, which is an important metric. FPS (frames per second) reflects the speed of the algorithm's target detection.

4.4. Experimental Results and Analysis

4.4.1. Experimental Results

The results of the YOLOv5 algorithm trained on a self-built dataset of wheel surface defects are shown in Figure 4. The obtained results demonstrate that sludge defects have the highest precision, linear defects and dotted defects have slightly worse precision, conversely, pinhole defects have the lowest precision. The likely reason behind this outcome is the large area of sludge defects with apparent features, as well as the large aspect ratio of the ground truth box, and dotted defects with a relatively single shape, so the precision is higher. Pinhole defects appear in clusters and have different shapes. What's more, the uncertainty edge of pinhole defects is the main reason for the low accuracy. Some detection results are shown in Figure 5.

The loss curve of the training set is shown in Figure 6. It can be seen from the figure that as the number of training rounds increases, the curve gradually converges, the loss of the network gradually decreases, and the model gradually stabilizes.

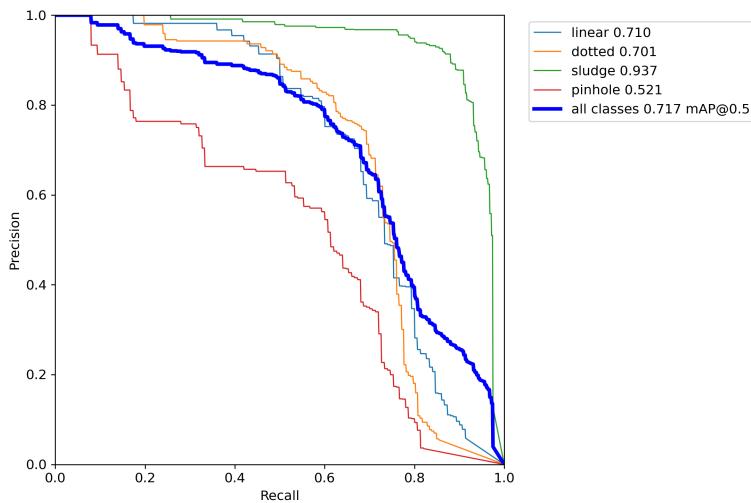


Figure 4. PR_curves of the training results. The curve's horizontal axis represents Recall, while the vertical axis represents Precision.

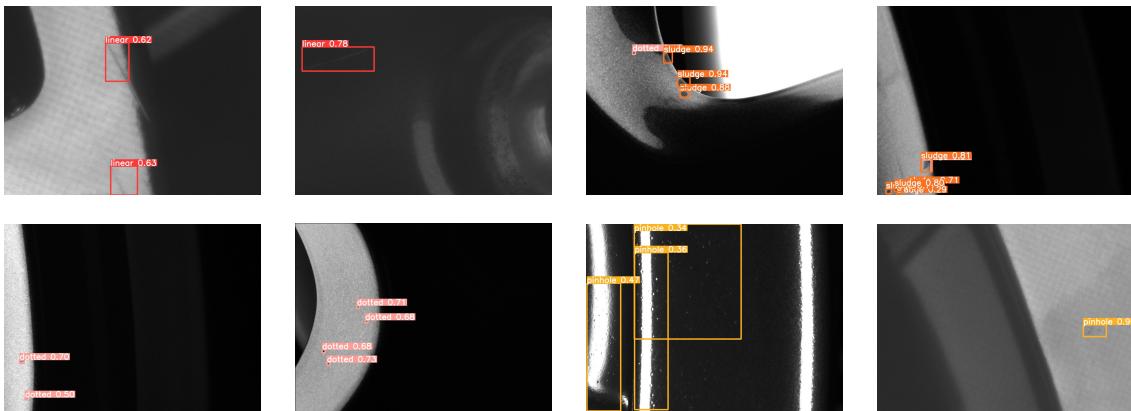


Figure 5. Some visual detection results on the self-built wheel surface defect dataset.

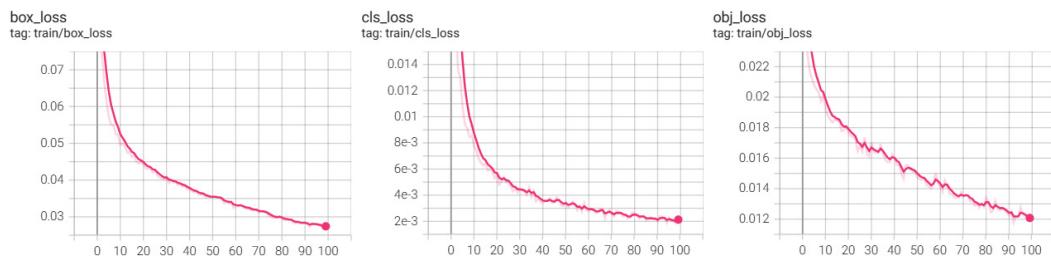


Figure 6. The training loss curve of YOLOv5s. The horizontal coordinate represents the number of epochs and the vertical coordinate represents the value of loss.

4.4.2. Algorithm Comparison

To demonstrate the superiority of the algorithm in this paper, we compared a number of different target detection algorithms utilizing the same dataset of wheel surface defects and identical parameter settings. Comparison of algorithm detection accuracy and detection speed is shown in Table 1.

Table 1. Accuracy and speed comparison of different algorithms on the same wheel surface defect dataset

Method	linear	dotted	sludge	pinhole	mAP	FPS
YOLOv5s	0.710	0.701	0.937	0.521	0.717	57.14
YOLOv4	0.630	0.820	0.740	0.370	0.640	10.52
SSD	0.540	0.150	0.400	0.370	0.360	11.09
Faster R-CNN	0.650	0.090	0.290	0.250	0.320	3.90

As can be seen from Table 1, among the four algorithms we studied, YOLOv5s acquires the highest accuracy on almost every defect class and also has the fastest speed. Although the YOLOv4 algorithm has the highest accuracy on point defects, it has a slightly lower average accuracy than YOLOv5s. Both SSD and Faster R-CNN are less effective in small target detection.

In summary, the YOLOv5 algorithm shows good performance on the wheel surface defect dataset used in this paper, with good adaptability to different defects and a high potential for improvement. The higher detection speed can also meet the demands of real-time industrial inspections.

5. Conclusion

Industrial surface defects are often inspected manually, which leads to inefficiencies, inaccuracy, and potential health risks for workers. To solve these issues, deep learning target detection algorithms like YOLOv5 are becoming increasingly popular in the industry due to their high generalizability and robustness. In this paper, the YOLOv5 algorithm is used for training and detecting the automotive wheel defects. The network trained on the self-built automotive wheel surface defect dataset achieves an average accuracy of 71.7% and FPS of 57.14, which is better than other common target detection algorithms. However, the algorithm still has some limitations on defects that appear in clusters with uncertain edge. To address this issue, we will explore more efficient data augmentation methods and higher quality datasets to improve the generalization ability of the model and solve the problem of insufficient data. We will also modify the network structure of the algorithm to improve the feature extraction ability. Better network parameters can also be chosen to improve the detection accuracy and choose better network parameters to improve detection accuracy while maintaining real-time requirements for industrial inspection.

Author Contributions: Conceptualization, X.X. and Z.Z.; methodology, X.X. and F.L.; validation, X.X.; investigation, X.X., Z.Z. and F.L.; writing—original draft preparation, X.X.; Writing—review and editing, X.X. and Z.Z.; supervision, F.L.; funding acquisition, F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the Joint Fund Project of the National Natural Science Foundation of China under Grant U22A2050, in part by the Natural Science Foundation of Hebei Province under Grant F2022203043.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Xing, J.; Jia, M. A convolutional neural network-based method for workpiece surface defect detection. *Measurement* **2021**, *176*, 109185. [\[CrossRef\]](#).
2. Chen, X.; Lv, J.; Fang, Y.; Du, S. Online Detection of Surface Defects Based on Improved YOLOV3. *Sensors* **2022**, *22*, 817. [\[CrossRef\]](#).

3. Patil, R.V.; Reddy, Y.; Thote, A.M. Multi-class weld defect detection and classification by support vector machine and artificial neural network. In *Modeling, Simulation and Optimization: Proceedings of CoMSO 2020*; Springer, 2021; pp. 429–438. [\[CrossRef\]](#).
4. Liu, L.; Guo, C.; Xiang, Y.; Tu, Y.; Wang, L.; Xuan, F.Z. A semisupervised learning framework for recognition and classification of defects in transient thermography detection. *IEEE Trans. Ind. Informat.* **2021**, *18*, 2632–2640. [\[CrossRef\]](#).
5. Jin, C.; Kong, X.; Chang, J.; Cheng, H.; Liu, X. Internal crack detection of castings: a study based on relief algorithm and Adaboost-SVM. *The International Journal of Advanced Manufacturing Technology* **2020**, *108*, 3313–3322. [\[CrossRef\]](#).
6. Mao, T.; Ren, L.; Yuan, F.; Li, C.; Zhang, L.; Zhang, M.; Chen, Y. Defect recognition method based on HOG and SVM for drone inspection images of power transmission line. In Proceedings of the 2019 international conference on high performance big data and intelligent systems (HPBD&IS). IEEE, 2019, pp. 254–257. [\[CrossRef\]](#).
7. 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, 2014, pp. 580–587.
8. Girshick, R. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **2015**, *28*.
10. 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 European conference on computer vision. Springer, 2016, pp. 21–37. [\[CrossRef\]](#).
11. 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, 2016, pp. 779–788.
12. Cheng, X.; Yu, J. RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–11. [\[CrossRef\]](#).
13. Zhao, W.; Chen, F.; Huang, H.; Li, D.; Cheng, W. A new steel defect detection algorithm based on deep learning. *Comput. Intell. Neurosci.* **2021**, *2021*. [\[CrossRef\]](#).
14. Zhi, L.; Bo, Z. Research and Application on the Improved SSD Chip Defect Inspection Algorithm. In Proceedings of the 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). IEEE, 2021, pp. 551–555. [\[CrossRef\]](#).
15. Cui, L.; Jiang, X.; Xu, M.; Li, W.; Lv, P.; Zhou, B. SDDNet: a fast and accurate network for surface defect detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [\[CrossRef\]](#).
16. Li, C.; Cai, J.; Qiu, S.; Liang, H. Defects detection of steel based on YOLOv4. In Proceedings of the 2021 China Automation Congress (CAC). IEEE, 2021, pp. 5836–5839. [\[CrossRef\]](#).
17. Gao, Y.; Lin, J.; Xie, J.; Ning, Z. A real-time defect detection method for digital signal processing of industrial inspection applications. *IEEE Trans. Ind. Informat.* **2020**, *17*, 3450–3459. [\[CrossRef\]](#).
18. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [\[CrossRef\]](#).
19. Ying, Z.; Lin, Z.; Wu, Z.; Liang, K.; Hu, X. A modified-YOLOv5s model for detection of wire braided hose defects. *Measurement* **2022**, *190*, 110683. [\[CrossRef\]](#).
20. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
21. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**.
22. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**.
23. Li, Q.; Luo, Z.; Chen, H.; Li, C. An overview of deeply optimized convolutional neural networks and research in surface defect classification of workpieces. *IEEE Access* **2022**. [\[CrossRef\]](#).
24. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
25. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

26. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
27. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE T. Cybern.* **2021**. [[CrossRef](#)].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.