

Article

High-Precision Detection Algorithm for Metal Workpiece Defects Based on Deep Learning

Xiujin Xu ¹, Gengming Zhang ¹, Wenhe Zheng ², Anbang Zhao ¹, Yi Zhong ¹ and Hongjun Wang ^{1,3,*}

¹ College of Engineering, South China Agricultural University, Guangzhou 510642, China; xiujin@stu.scau.edu.cn (X.X.); gengming_zhang@stu.scau.edu.cn (G.Z.); 20223171033@stu.scau.edu.cn (A.Z.); zy0328@stu.scau.edu.cn (Y.Z.)

² Wanhui Hardware Shenzhen Co., Ltd., Shenzhen 518118, China; justy.zheng@byd.com

³ Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China

* Correspondence: xtwjh@scau.edu.cn

Abstract: Computer vision technology is increasingly being widely applied in automated industrial production. However, the accuracy of workpiece detection is the bottleneck in the field of computer vision detection technology. Herein, a new object detection and classification deep learning algorithm called CSW-Yolov7 is proposed based on the improvement of the Yolov7 deep learning network. Firstly, the CotNet Transformer structure was combined to guide the learning of dynamic attention matrices and enhance visual representation capabilities. Afterwards, the parameter-free attention mechanism SimAM was introduced, effectively enhancing the detection accuracy without increasing computational complexity. Finally, using WIoUv3 as the loss function effectively mitigated many negative influences during training, thereby improving the model's accuracy faster. The experimental results manifested that the mAP@0.5 of CSW-Yolov7 reached 93.3%, outperforming other models. Further, this study also designed a polyhedral metal workpiece detection system. A large number of experiments were conducted in this system to verify the effectiveness and robustness of the proposed algorithm.



Citation: Xu, X.; Zhang, G.; Zheng, W.; Zhao, A.; Zhong, Y.; Wang, H. High-Precision Detection Algorithm for Metal Workpiece Defects Based on Deep Learning. *Machines* **2023**, *11*, 834. <https://doi.org/10.3390/machines11080834>

Academic Editor: Francisco J. G. Silva

Received: 3 August 2023

Revised: 11 August 2023

Accepted: 14 August 2023

Published: 16 August 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/).

Keywords: deep learning; neural networks; industrial inspection; defect detection

1. Introduction

In complex industrial production processes, defects such as collision damage, dents, wear, scratches, etc., can occur due to design and mechanical equipment failures, adverse working conditions, or human factors. In everyday use, products are also prone to corrosion and fatigue. These defects of different degrees increase the cost for companies, shorten the lifespan of products, and result in significant waste of resources, posing great risks to personal safety and social-economic development. Therefore, the capability of defect detection is the key to improving product quality without compromising production efficiency. In recent years, the development of machine-vision-based automated inspection methods has overcome the limitations of low accuracy, poor real-time performance, and high labor intensity associated with manual inspection [1]. This technology has emerged as a fast and reliable alternative for detecting various surface defects, offering significant advantages such as high automation, reliability, and objectivity. Machine-vision-based inspection has demonstrated strong adaptability to different environmental conditions and can operate continuously with high levels of precision and efficiency [2]. However, industrial visual defect detection methods are required to be equipped with characteristics such as high precision, high efficiency, and low cost [3]. This suggests that these requirements have also become important bottlenecks in the field of computer vision detection [4].

With the rapid development of deep learning technology, significant successes have been achieved in various fields such as object detection [5], intelligent robotics [6], saliency

detection [7], parking-lot sound event detection [8], smart-city safety sound event detection [9,10], and drone-blade fault diagnosis [11–14]. Deep learning is a machine learning method that uses multi-layer neural networks for automatic feature learning and pattern recognition. By combining low-level features to form higher-level representations of abstract attribute categories or features, deep learning algorithms can realize more accurate data understanding and analyses in terms of edges, shapes, and other abstract characteristics, thereby strengthening their effectiveness. As a result, numerous researchers have been exploring the utilization of deep learning techniques for defect detection in products to enhance product quality and production efficiency [15–18].

This article puts forward an improved deep learning neural network based on Yolov7, using the example of metal surface defects caused by locks. Through a large number of experiments, the proposed algorithm achieved high accuracy and efficiency in the process of automated detection. The surface defects of the lock body were determined as being diverse, covering almost all kinds of defects caused by metal mechanical processing, as shown in Figure 1. It is clear that the selected lock body presented a curved surface, and the uneven reflection of the curved surface caused excessive brightness in certain areas of the photos, ultimately affecting the detection results. Actually, this problem is also very common in metal workpieces [19]. Therefore, the selected lock body in this article has representative significance for metal workpieces. This study mainly classified surface defects of the selected lock body into the following categories, namely: bad stuff, freckles, scratches, poor contraction, bad cover, bump damage, and abrade.

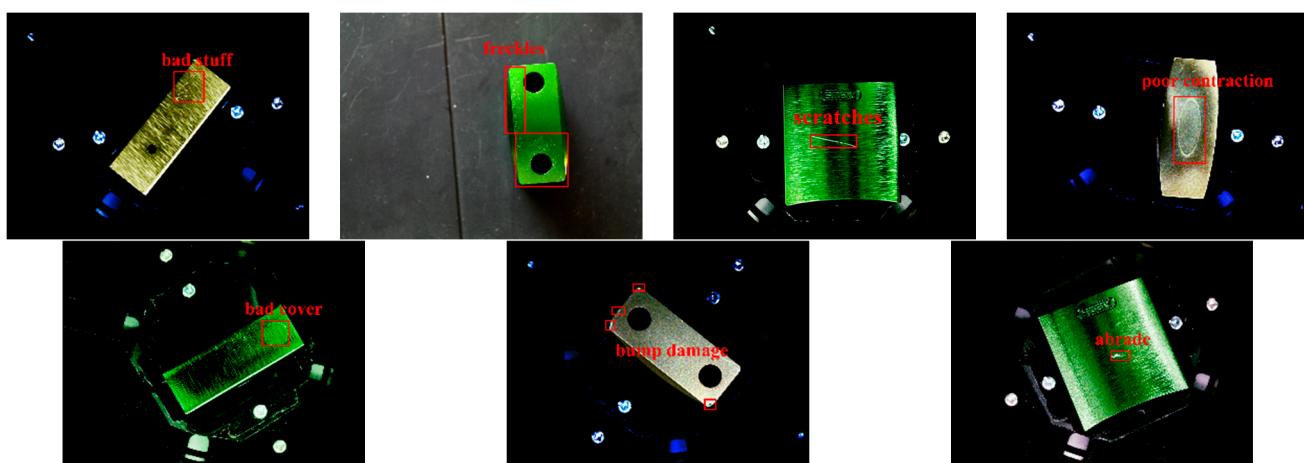


Figure 1. Examples of metal lock body surface defects.

In order to detect surface defects on metal lock bodies with high precision and efficiency, this paper establishes an advanced and effective deep learning network. Specifically, Yolov7 [20] was firstly combined with the CotNet Transformer structure [21], so as to combine the ability of the Transformer [22] to capture global information with the ability of CNNs to capture local information in proximity, thereby improving the feature representation capability of the network model. Based on this raised algorithm, the parameter-free attention mechanism SimAM [23] was then incorporated into the aforementioned network. Unlike existing channel/spatial attention modules, this module could derive 3D attention weights for feature maps without requiring additional parameters, effectively intensifying the detection accuracy. Finally, WIoUv3 [24] was adopted as the loss function to effectively mitigate a series of negative influences during the training process and allow for faster improvement in the accuracy of the model. In addition, a polyhedral metal component inspection system was also developed, including a detection platform and its operating software v1.0 that comply with actual production conditions. The improved network training model was applied to this system and its high efficiency and accuracy in automatically detecting surface defects on metal lock bodies were validated with numerous experiments.

In summary, this article contributes the following:

- (1) Introducing an advanced and effective deep learning neural network that builds upon the Yolov7 algorithm, utilizing the CotNet Transformer as the backbone, incorporating the parameter-free attention mechanism SinAM, and using WIoUv3 as the loss function.
- (2) To validate the efficiency and accuracy of the proposed algorithm, a polyhedral metal workpiece detection system was designed, including a detection platform and its operating software v1.0.
- (3) Providing an automated control solution for automated detection and sorting.

The structure of the remaining parts of this article is as follows. Section 2 provides a brief overview of related research on industrial defect detection methods. Section 3 presents a detailed description of the implementation of the proposed network architecture. The experiments conducted and the results obtained are presented in Section 4. Finally, in Section 5, the research findings and their significance are summarized, providing a conclusion to this study.

2. Related Work

Industrial defect detection has consistently remained as a prominent research topic undertaken within the realm of industrial vision. Machine vision algorithms offer a diverse range of methods for detecting defects, broadly classified into two categories: conventional approaches and deep learning methodologies. Cheetverikov et al. [25] effectively utilized these techniques to detect sudden flaws on fabric surfaces, where texture defects were analyzed by employing two fundamental structural characteristics, specifically consistency and local direction (anisotropy). Hou et al. [26] showcased that the precise recognition and partitioning of defects exposed on the surface of textures can be attained through the utilization of support vector machine classification methods relying on Gabor wavelet characteristics. Zheng et al. [27] proposed a VMD modulus optimization method based on maximum envelope kurtosis, which exhibits strong generalization and noise resistance. Cha et al. [28] introduced a visual detection technique for structures, with the help of the Faster Region Convolutional Neural Network (Faster RCNN) [29]. This method enables simultaneous and near-real-time identification of concrete fractures, medium- and high-level steel erosion, bolt erosion, as well as five distinct forms of steel delamination harm. With a resolution of 500×375 , this approach provides a relatively rapid speed of detection, averaging 0.03 s per image. However, industrial vision defect detection requires highly accurate methods capable of detecting subtle defects that are difficult for the human eye to observe. It is not only necessary to minimize false negatives and positives but also to be able to adjust the detection performance in a timely manner. According to the aforementioned literature, traditional methods are only suitable for defects with specific geometric features. Traditional methods require the manual design of feature descriptors for defects, which are appropriate for simple and rule-based industrial scenarios. The subtle nature and subjectivity of defects pose great challenges in the accurate description of defects using manual features. Moreover, due to the unknown and diverse nature of defects, multiple sets of defect templates need to be designed, but they cannot detect novel defects, only describing a limited range of defect types. When faced with complex and irregular data, traditional methods not only struggle to be applied but may also require complex post-processing procedures [3]. In most cases, on account of the complexity of mechanical processing, the types of defects in workpieces are diverse and traditional methods are no longer competent for defect detection.

In recent years, with the widespread application of deep learning in computer vision tasks [30–36], deep-learning-based industrial defect detection methods have rapidly developed and gradually become the mainstream. Due to the powerful feature extraction and representation capabilities of convolutional neural networks (CNN) for high-dimensional data, deep-learning-based methods can achieve the automatic learning of features that are difficult for humans to design. This not only saves the cost of manually designing

features but also significantly improves detection accuracy. Compared to traditional methods based on image processing and statistical learning, deep learning methods present advantages in handling complex industrial image data. Tao et al. [37] designed a novel cascaded autoencoder (CASAE) architecture for defect segmentation and localization. This approach satisfies the criteria for robustness and precision in identifying defects in metal materials. However, it is impractical for defects such as bad stuff and freckles owing to large numbers of labels needed, and a significant amount of manual labor required for creating and analyzing the dataset. Gao et al. [38] proposed a convolutional neural network with feature alignment trained in a hierarchical manner. The method introduces feature alignment, which maps unrecognizable defects to recognizable areas, and incorporates feature alignment into the training process using a hierarchical training strategy. Nevertheless, the network still needs improvement in recognizing small defects. Yoon et al. [39] presented a technique for real-time non-destructive testing for layered composite material defects by virtue of highly nonlinear solitary waves (HNSWs) in deep learning. The accuracy level of this technique exceeds 90%, highlighting the potential of real-time detection utilizing the proposed deep learning algorithm. However, this method is restricted to the layered detection of AS4/PEEK laminated composite materials, which has significant limitations. Wang et al. [40] put forward an unsupervised surface defect detection method based on a non-convex total variation (TV)-regularized kernelized Robust Principal Component Analysis (RPCA). However, this method is prone to false negatives in terms of detecting small-sized defects and requires further improvement. Yang et al. [41] came up with an effective unsupervised anomaly segmentation method that can detect and segment anomalies in small regions and constrained areas of an image. They designed a deep and efficient convolutional autoencoder to detect abnormal regions in images through fast feature reconstruction. However, this method struggles to detect those defect types without concave or convex surface features, such as freckles shown in Figure 1. Yang et al. [42] proposed a new method called a Multi-Scale Feature-Clustering-based Fully Convolutional Autoencoder (MS-FCAE) for the efficient and accurate detection of various types of texture defects with a small number of defect-free texture samples. But, this method is only applicable to different types of texture defects and lacks generalizability. Li et al. [43] introduced a novel automatic defect detection approach based on deep learning, namely You Only Look Once (YOLO)-Attention based on YOLOv4, which achieved fast and accurate defect detection in Web-based Augmented Assembly Manufacturing (WAAM). Wang et al. [44] presented an accurate object detector, ATT-YOLO (Attention-YOLO), which is oriented toward the problem of surface defect detection in electronics manufacturing. ATT-YOLO satisfies the requirements of surface defect detection and achieves the best tradeoff among lightweight YOLO-style object detectors.

In summary, there are various methods for detecting defects in metal workpieces, but the drawbacks of traditional methods are evident. Deep learning methods can be classified into two major categories: defect segmentation and object detection. However, based on the extensive literature reviewed, defect segmentation methods lack generalizability and are unable to achieve sufficient accuracy for detecting small-sized defects. On the other hand, object detection methods exhibit strong generalizability, an ease of dataset creation, high accuracy, and efficiency. Therefore, this paper proposes a deep learning network based on an improved YOLOv7 algorithm for the automated detection of surface defects on metal lock bodies.

3. Proposed Method

3.1. Yolov7 Algorithm Model

As a core technology in the domain of computer vision, object detection has gained broad utilization across various industrial sectors. Among them, the YOLO series algorithms have gradually become the preferred framework for most industrial applications due to their excellent overall performance [45–49]. However, in practical use, many algorithms fail to meet the requirements of industrial detection in terms of speed and accuracy.

There are various types of defects in locks and lock cores, with significant differences in shape and size. Consequently, it is necessary to adopt a deep learning algorithm with good robustness, as well as detection speed and accuracy, that can meet industrial requirements. This section introduces the improvement scheme of the Yolov7 object detection algorithm model. Figure 2 shows the modified network architecture.

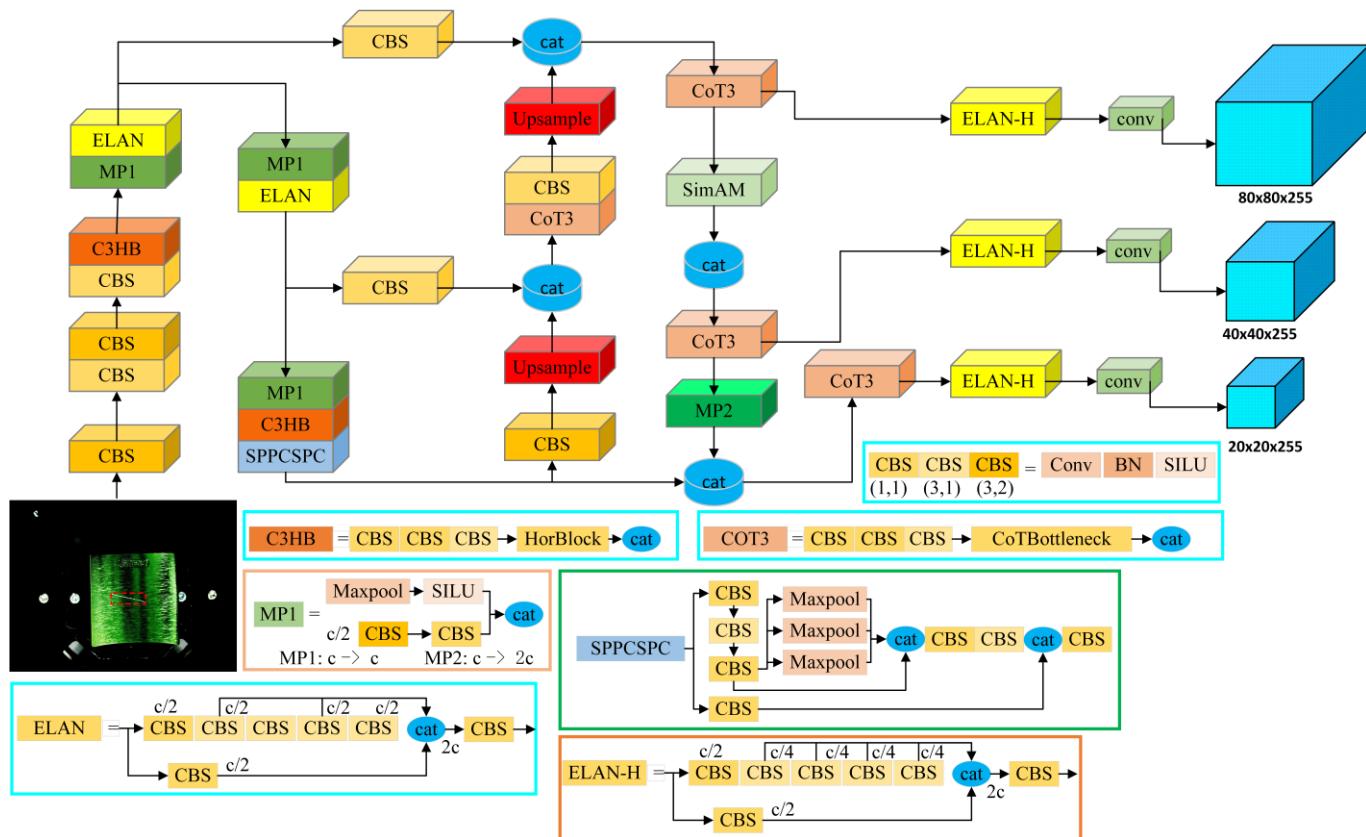


Figure 2. Overall structure of CSW-Yolov7.

3.2. Using CotNet Transformer as the Backbone

Most existing Transformer-based architecture designs directly operate on 2D feature maps, obtaining attention matrices by using self-attention (independent query points and all key pairs). However, the rich context between adjacent keys is not fully utilized. Li et al. [21] designed a novel Transformer module called a Contextual Transformer (CoT) for visual recognition, which completely takes advantage of the contextual information between input keys to guide the learning of dynamic attention matrices, thereby enhancing visual representation capability. Technically, CoT first encodes the input keys with a 3×3 convolution to generate the static contextual representation of the input. Furthermore, the encoded keys are concatenated with the input queries, and dynamic multi-head attention matrices are learned through two consecutive 1×1 convolutions. Finally, the learned attention matrices are multiplied by the input values to achieve the dynamic contextual representation of inputs, and the fusion of static and dynamic contextual representations is considered as the final output. In this paper, the Contextual Transformer structure was employed as the backbone of YOLOv7, resulting in a Transformer-style backbone network called Contextual Transformer Networks (CotNet). The principle of the Contextual Transformer module is illustrated in Figure 3.

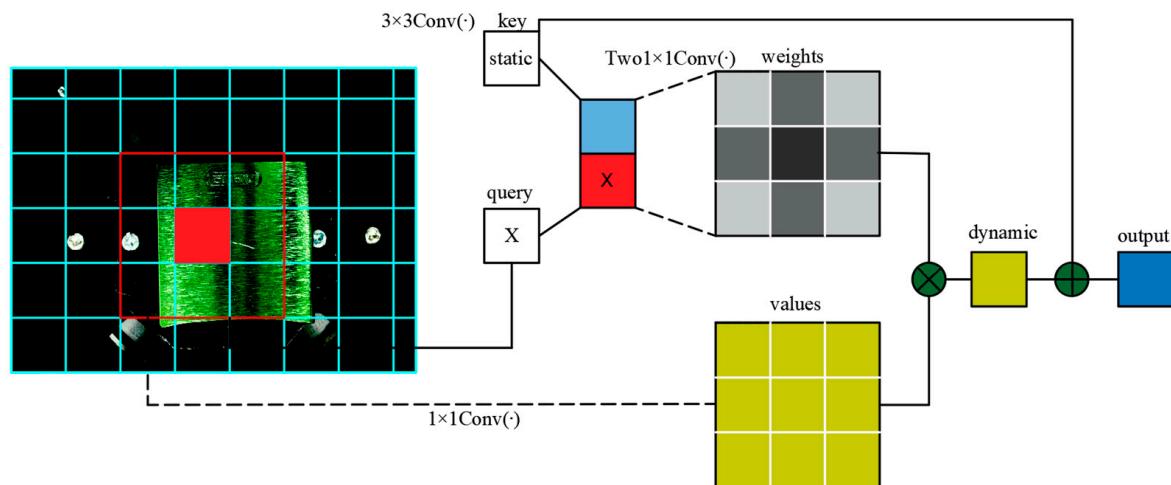


Figure 3. Contextual Transformer module principle.

The Contextual Transformer (CoT) module surpasses traditional self-attention mechanisms by leveraging the contextual information between input keys to facilitate self-attention learning and ultimately boost the network's representation capability. It combines context mining and self-attention learning, making full use of the contextual information between adjacent keys to promote self-attention learning and reinforce the expressive power of the aggregated feature maps. Figure 4 represents the structure diagram of the CoT module.

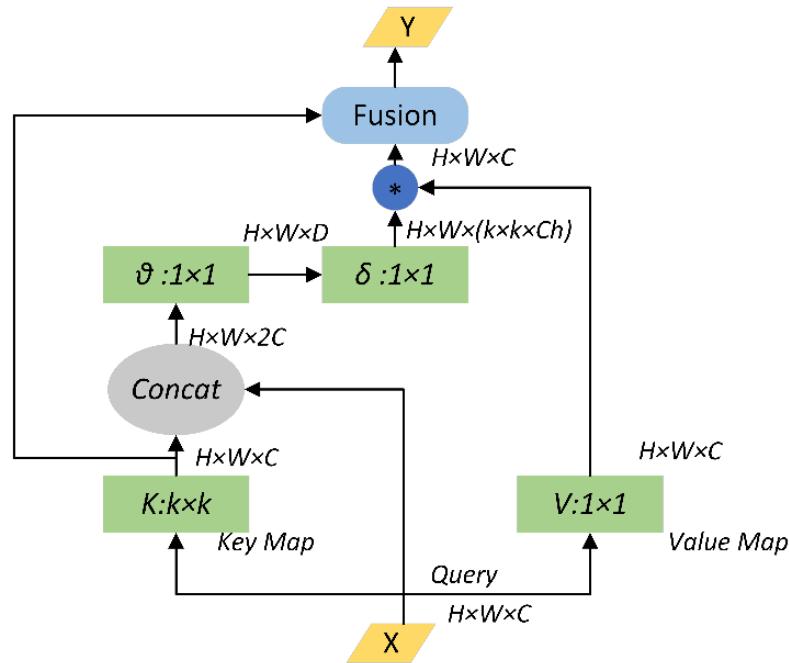


Figure 4. CoT module structure diagram. * denotes the local matrix multiplication, respectively.

According to Figure 4, the specific implementation principle is as follows. Given an input feature map X of size $H \times W \times C$, the keys, queries, and values are defined as $H = X$, $Q = X$, and $V = XW_v$, respectively. CoT first performs context encoding on all adjacent keys within a $k \times k$ neighborhood in the spatial dimension to obtain contextual keys $K^1 \in R^{H \times W \times C}$, which reflects the static contextual information between

local neighboring keys. Next, the contextual keys K^1 are concatenated with Q , passing through two consecutive 1×1 convolutions to obtain the attention matrix:

$$A = [K^1, Q]W_\theta W_\delta \quad (1)$$

The local attention matrix at each spatial position of A is learned based on the query features and contextual key features, rather than simple query–key pairs. By aggregating the contextual attention matrix A with the value V , a weighted feature map K^2 is obtained:

$$K^2 = V * A \quad (2)$$

The above feature map K^2 captures dynamic feature interactions between inputs. Therefore, the CoT module outputs the fusion of the static contextual representation K^1 and the dynamic contextual representation K^2 .

3.3. Introducing Parameter-Free Attention Mechanism

To further improve the detection accuracy of small defects, it is feasible and effective to introduce an attention mechanism that enhances the network’s focus on small targets without increasing the computational burden. As a consequence, this paper proposes the introduction of the parameter-free attention mechanism SimAM [23] in the Yolov7 network, directly estimating the three-dimensional weights (c). In each subgraph, the same color represents a single scalar used for each channel, spatial position, or point on the feature, as depicted in Figure 5.

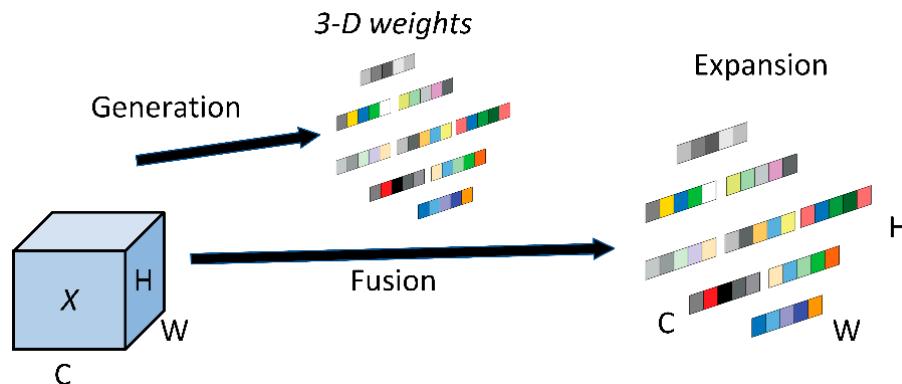


Figure 5. Basic principles of the SimAM module.

The SimAM module inspired by the attention mechanism in the human brain is a conceptually simple yet highly effective attention module. Unlike existing channel/spatial attention modules, this module can derive 3D attention weights from feature maps in the absence of additional parameters. Another advantage of this module is that most operations are based on the defined energy function, which avoids excessive structural adjustments.

3.4. Using $WIoUv3$ as the Loss Function

In object detection networks, the loss function is used to measure the difference between the model’s predicted results and true labels. Its main purpose is to guide the model in learning how to accurately predict the position and category of objects. Please refer to Figure 6 for details.

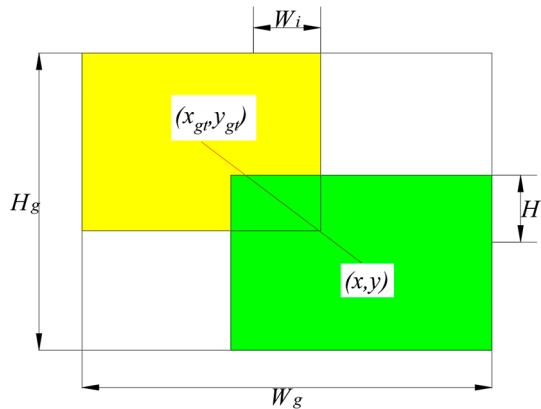


Figure 6. IoU calculation principle diagram.

Let the anchor box be denoted as $\vec{B} = [x \ y \ w \ h]$, and the target box as $\vec{B} = [x_{gt} \ y_{gt} \ w_{gt} \ h_{gt}]$. IoU (Intersection over Union) refers to a metric used to measure the degree of overlap between the predicted box and the ground truth box in object detection tasks. It can be defined as

$$L_{IoU} = 1 - IoU = 1 - \frac{W_i H_i}{S_u} \quad (3)$$

$$S_u = wh + w_{gt}h_{gt} - W_i H_i \quad (4)$$

Existing studies have considered various geometric factors related to bounding boxes and constructed penalty terms R_i to address this problem. Existing bounding box losses are all based on additive losses and abide by the following equation:

$$L_i = L_{IoU} + R_i \quad (5)$$

To achieve faster convergence during training and obtain higher accuracy under the same training conditions, this research proposed replacing the original loss function of Yolov7 with WIoUv3 [24]. Since it is difficult to avoid including low-quality examples in the training data, geometric metrics such as distance and aspect ratio can intensify the punishment for low-quality examples, leading to a decrease in the model's generalization performance. A good loss function should weaken the punishment of geometric metrics when the anchor box and target box overlap well, without excessively interfering with normal training, thereby enabling the model to display better generalization ability.

By constructing distance attention based on distance metrics, the following can be obtained:

$$L_{WIoUv1} = R_{WIoU} L_{IoU} \quad (6)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) \quad (7)$$

Here, W_g and H_g represent the size of the smallest enclosing box (as shown in Figure 6). To prevent the gradient of R_{WIoU} from hindering convergence, W_g and H_g are detached from the computation graph (denoted by superscript *).

The definition of the outlier score is used to describe the quality of anchor boxes. It is defined as

$$\beta = \frac{L_{IoU}^*}{L_{IoU}} \in [0, +\infty) \quad (8)$$

A smaller outlier score indicates a higher quality of the anchor box, and it is assigned a smaller gradient gain to focus the bounding box regression on anchor boxes of a normal quality. Anchor boxes with larger outlier scores are assigned smaller gradient gains, effectively preventing low-quality examples from generating large harmful gradients. Applying β to Formula (6) results in the following:

$$L_{WIoUv3} = rL_{WIoUv1}, r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \quad (9)$$

In Formula (9), both δ and α refer to hyperparameters. When using $WIoUv3$ as the loss function, in the middle and later stages of training, $WIoUv3$ assigns small gradient gains to low-quality anchor boxes to reduce harmful gradients. At the same time, $WIoUv3$ focuses on anchor boxes of a normal quality, improving the model's localization performance.

4. Experiments and Results

Section 3 introduces the basic principles of the proposed network in this paper. This section provides a detailed analysis of its experimental results.

4.1. Dataset and Experimental Setup

The experimental setup in this paper is based on the deep learning framework PyTorch, with the version of 1.10.1+cu102, Python 3.8, and was conducted on a Windows 10 operating system. The processor used was Intel(R) Core(TM) i7-10700 CPU@2.90 GHz, with 16 GB of memory. The graphics card model was NVIDIA GeForce RTX 2060, and CUDA 10.2 and CUDNN 7.6.5.32 were used to accelerate GPU computation. The specific configuration is organized in Table 1.

Table 1. Experimental environment settings.

Parameter	Configuration
Operating system	Windows 10
Deep learning framework	1.10.1+cu102
Programming language	Python 3.8
CUDA	CUDA10.2
GPU	NVIDIA GeForce RTX 2060
CPU	Intel(R) Core(TM) i7-10700 CPU@2.90 GHz

To make the experimental environment closer to the actual industrial production environment, this paper designs a detection platform for the automatic detection of polyhedrons, as shown in Figure 7, in which an industrial camera from Hikvision was used, based on a ring-shaped LED light source with an outer diameter of 70 mm and an inner diameter of 40 mm. Relevant parameters of the industrial camera and light source are summarized in Table 2. Approximately 20 photos of each lock at different angles were collected on the inspection platform, resulting in a total of 1210 raw images. Each raw image was enhanced through scaling, shifting, color inversion, Gaussian filtering, and other processing techniques, obtaining a total of 4840 images. The names and quantities of various defects are delineated in Table 3. During the collection process, the platform was placed in an indoor environment, and the camera exposure time was set to 150 ms. The model was trained using a ratio of training/validation/test set = 7:1.5:1.5. The hyperparameters involved during the training process are displayed in Table 4.

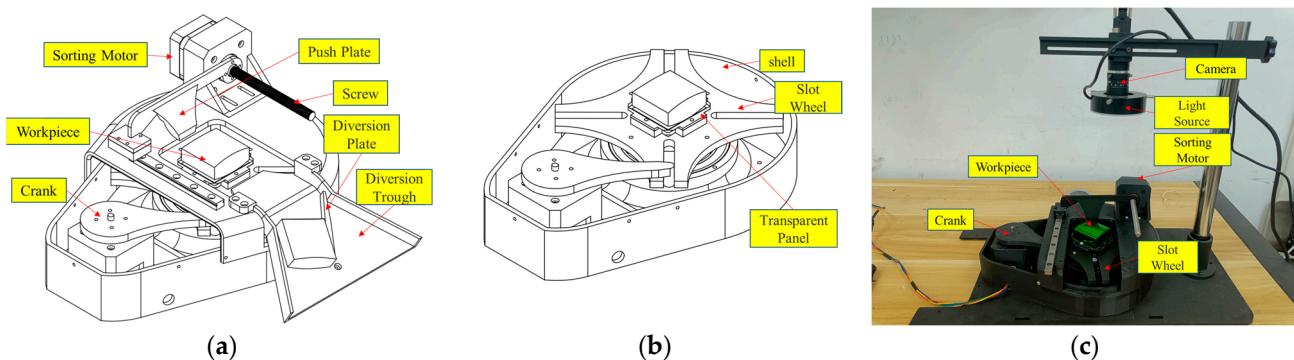


Figure 7. Polyhedron detection platform. (a) Automatic sorting platform model for polyhedrons. (b) Internal structure of the platform. (c) Detection platform. (Note: The area in the middle of the slot wheel where the workpiece is placed is transparent. In the actual detection platform, there should be three cameras. For the convenience of presentation, only one camera is shown in this paper. The other two cameras are installed directly below the transparent area to detect the underside of the workpiece, and on the side of the workpiece to detect the side view.).

Table 2. Relevant parameters of the industrial camera and light source.

Equipment	Parameter	Data
LED ring light	Item code	JHZM-A40-W
	Light source color	white
	Number of LEDs	48 shell LEDs
Industrial camera	Active pixels	5 million
	Type	multicolor
	Pixel size	2.2 $\mu\text{m} \times 2.2 \mu\text{m}$
	Frame rate/Resolution	31 @ 2592 \times 1944
Camera lens	Focal distance	12 mm
	Maximum image surface area	1/1.8" ($\varphi 9 \text{ mm}$)
	Aperture spectrum	F2.8–F16

Table 3. Dataset.

Total	Bad Stuff	Freckles	Scratches	Poor Contraction	Bad Cover	Bump Damage	Abrade
4840	880	320	1040	640	640	680	640

Table 4. Training parameters.

Parameter	Learning Rate	Batch Size	Epoch	Img. Size	Workers
Value	0.01	4	300	640	8

4.2. Evaluation Index

In this experiment, precision (P), recall (R), and average precision (AP) were utilized to assess the efficacy of surface anomaly detection. The equations for computing P , R , and AP are respectively specified in Formulas (10) to (12).

$$P = \frac{TP}{TP + FP} \quad (10)$$

$$R = \frac{TP}{TP + FN} \quad (11)$$

$$mAP = \frac{\sum_{i=1}^K AP_i}{K} \quad (12)$$

where TP represents the count of correctly identified true-positive samples as positive using the algorithm, FP represents the count of incorrectly identified false-positive samples as positive using the algorithm, and K refers to the count of defect categories. AP_i denotes the average precision of each defect category, which is defined as shown in Equation (13). Therefore, mAP represents the overall average precision across all defect categories.

$$AP = \int_0^1 P(R)d(R) \quad (13)$$

4.3. Analysis of Experimental Results

4.3.1. Ablation Experiment

In this study, ablation experiments were conducted to comprehensively validate the optimization effects of various enhancement modules. Specifically, multiple ablation experiments were carried out between Yolov7 (original), C-Yolov7 (CotNet), W-Yolov7 (WIoUv3), S-Yolov7 (SimAM), CS-Yolov7 (CotNet + SimAM), CW-Yolov7 (CotNet + WIoUv3), WS-Yolov7 (WIoUv3 + SimAM), and CSW-Yolov7 (CotNet + WIoUv3 + SimAM). The experimental results are summarized in Table 5. As mentioned in Section 1, there are seven types of defects in the metal lock body. For the convenience of data recording, this paper assigned sequential numbers one to seven to each defect type.

Table 5. Results of ablation experiments.

Model	mAP@0.5	AP						
		1	2	3	4	5	6	7
Yolov7 (original)	0.854	0.596	0.85	0.995	0.945	0.946	0.689	0.954
C-Yolov7 (CotNet)	0.873	0.587	0.775	0.976	0.996	0.967	0.826	0.985
S-Yolov7 (SimAM)	0.897	0.687	0.823	0.982	0.995	0.969	0.837	0.985
W-Yolov7 (WIoUv3)	0.905	0.721	0.797	0.986	0.996	0.996	0.851	0.985
CS-Yolov7 (CotNet + SimAM)	0.908	0.752	0.819	0.985	0.995	0.996	0.822	0.985
CW-Yolov7 (CotNet + WIoUv3)	0.923	0.748	0.842	0.994	0.996	0.996	0.898	0.985
WS-Yolov7 (SimAM + WIoUv3)	0.916	0.732	0.870	0.982	0.978	0.987	0.872	0.990
CSW-Yolov7 (CotNet + SimAM + WIoUv3)	0.933	0.772	0.869	0.994	0.995	0.995	0.91	0.993

From Table 5, it can be seen that the CotNet, SimAM, and WIoUv3 modules all contributed to the improvement in detection accuracy in Yolov7. The original Yolov7 achieved a mAP@0.5 of 87.1%. C-Yolov7, S-Yolov7, and W-Yolov7 were the networks with CotNet, SimAM, and WIoUv3 integrated, respectively, and their detection accuracy increased by 1.9%, 4.3%, and 5.1%, respectively. CS-Yolov7, CW-Yolov7, and WS-Yolov7 were networks obtained by combining two of the three modules, and their detection accuracy realized improvements of 5.3%, 6.9%, and 6.2%, respectively. Finally, CSW-Yolov7 incorporated all three modules into the original network, resulting in a significant increase in detection accuracy by 7.9% compared to the original network. The effectiveness and superiority of the proposed network are clearly demonstrated in this paper.

To visually observe the superiority of the proposed network in this paper, Figure 8 shows the change curves of mAP@0.5 and mAP@0.5:0.95 during the training process for both the original Yolov7 network and the improved CSW-Yolov7 network. It can be observed that the CSW-Yolov7 network realized a rapid increase in the values of mAP@0.5 and mAP@0.5:0.95 during the training process, ultimately achieving a significant improvement in accuracy.

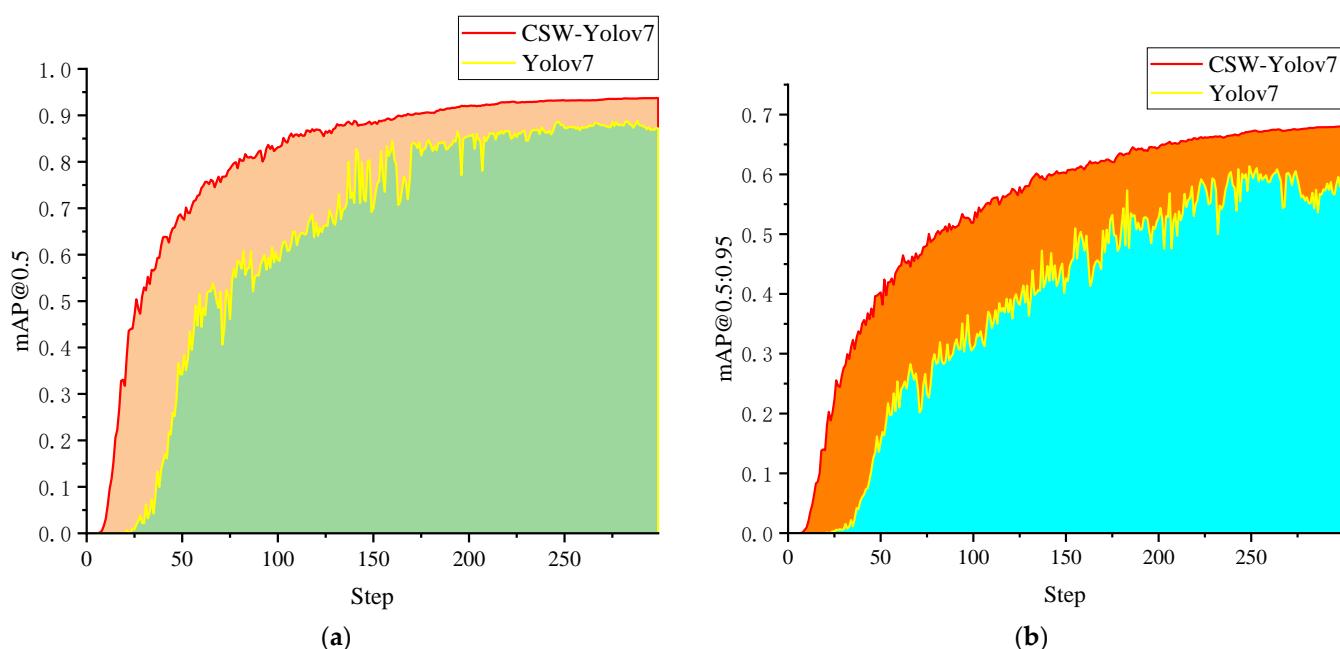


Figure 8. Comparison of accuracy before and after improvement in YOLOv7. (a) Curve of mAP@0.5 changes. (b) Curve of mAP@0.5:0.95 changes.

To demonstrate the superiority of the raised algorithm compared to other algorithms, comparative experiments were conducted on other YOLO series algorithms in this study. Under the same experimental conditions, the same dataset was applied for training, and six metrics were recorded: mAP@0.5, parameters, GFLOPS, inference, NMS, and preprocess. The results are summed up in Table 6.

Table 6. Performance comparison of object detection algorithms.

Version	mAP@0.5	Parameters	GFLOPS	Speed (Time: ms)		
				Inference	NMS	Preprocess
YOLOv5s	0.884	7.23 M	16.6	5.1	1.0	0.20
YOLOv6	0.916	17.19 M	44.08	7.10	1.19	0.20
YOLOv7	0.854	37.23 M	105.2	10	1.1	11.2
Yolov8	0.928	11.13 M	28.5	5.6	0.5	0.3
CSW-Yolov7	0.933	33.57 M	40.3	8.4	1.1	9.5

According to Table 6, it can be observed that the proposed algorithm in this study achieved the highest accuracy. The overall detection time could be represented by the sum of three metrics: inference, non-maximum suppression (NMS), and preprocess. The detection time of the network was set as 19.1 ms. Based on the comparison in Table 6 and considering its application in industrial automation production, CSW-YOLOv7 was proven to be the optimal choice.

4.3.2. Practical Application of the Algorithm

As shown in Figure 7, this study simulated a real industrial production environment. The actual detection process is explained as follows. Firstly, the workpiece (a metal lock in this study) was placed in the middle of the slot wheel on the detection platform. Then, three industrial cameras worked simultaneously during the actual detection. If a defect was detected with any of the cameras, the sorting motor would push the workpiece out, indicating it as a defective product. Otherwise, if no defects were detected after the crank rotated for four turns, it indicated that all six sides of the workpiece were defect-free,

including four sides. Subsequently, the sorting motor would push out the workpiece, which was considered as a qualified product. Finally, the ejected workpieces would enter the sorting device (as shown in Figure 7a, the flow control device consists of a diversion plate, a diversion trough, and a servo motor, with the servo motor being obstructed and not shown). The specific workflow is illustrated in Figure 9.

In addition, to facilitate the real-time control of the detection platform and observe the detection results, a user interface (UI) was designed for the metal automated sorting system, as shown in Figure 10. Through extensive experiments, this detection system demonstrated high accuracy and efficiency in automatic sorting.

After numerous experiments, the algorithm introduced in this article was applied to the system. The sorting success rate of the system depending on the detection accuracy of the algorithm was confirmed to meet the industrial inspection requirements for defects such as scratches and bad covers. However, for small-sized defects with irregular distribution such as bad stuff, the sorting success rate of this system still needs improvement.

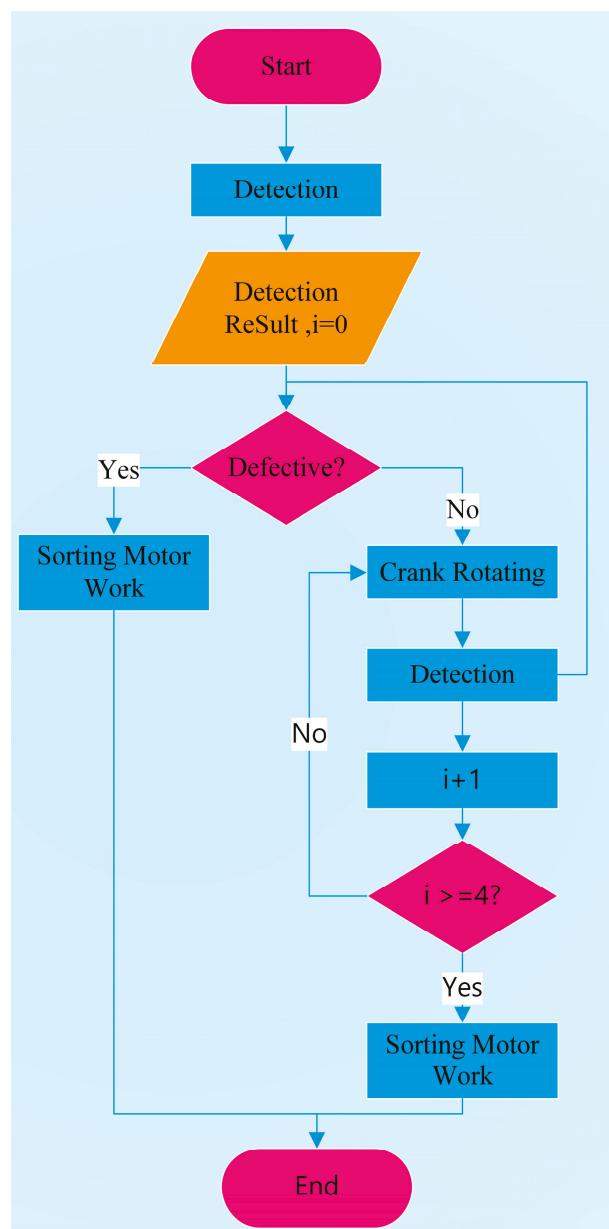


Figure 9. Workflow of the detection platform.

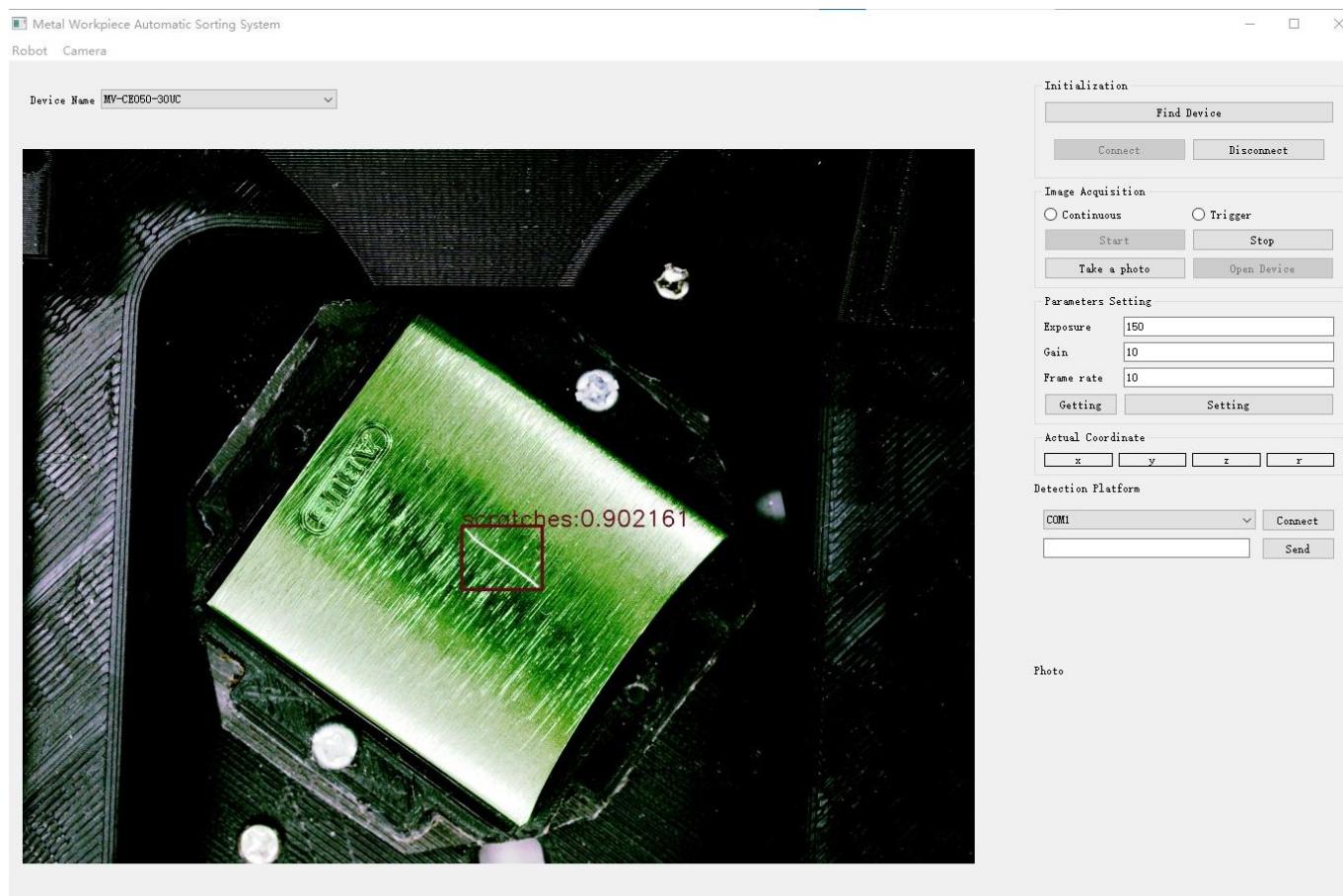


Figure 10. Metal workpieces' automatic sorting system.

5. Discussion

This study proposed an efficient and accurate improved Yolov7 network called CSW-Yolov7. The CotNet Transformer module was adopted as the backbone, the parameter-free attention mechanism SimAM was introduced, and the WIoUv3 was used as the loss function. By combining these three important modules, the mAP@0.5 value of the CSW-Yolov7 network was improved by 6.9% in contrast with the original network. Compared to other Yolo series networks, the proposed network also demonstrated significant superiority. Additionally, this study designed a system for the industrial automation detection and sorting of metal parts, which exhibited strong practicality and reliability through extensive experiments.

However, this study still exposes some limitations that can be addressed in future research directions. Firstly, the proposed algorithm failed to achieve an AP@0.5 value exceeding 80% when detecting small-sized and irregularly distributed defects such as bad stuff. In this regard, further improvements are needed for such defects. Additionally, the improved network realized a significantly lower GFLOPS value compared to the original network. Lastly, the network proposed in this paper was only validated for the detection of metal workpieces. Future research can explore its application in other fields such as construction and agriculture.

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

Funding: This research was supported in part by the Laboratory of Lingnan Modern Agriculture Project under Grant NT2021009 and Guangzhou Science and Technology Project (2023B01J0046).

Data Availability Statement: Not applicable.

Acknowledgments: We sincerely acknowledge the funding support of the project. We are also grateful for the efforts of all our colleagues.

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

References

1. Rasheed, A.; Zafar, B.; Rasheed, A.; Ali, N.; Sajid, M.; Dar, S.H.; Habib, U.; Shehryar, T.; Mahmood, M.T. Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review. *Math. Probl. Eng.* **2020**, *2020*, 8189403. [[CrossRef](#)]
2. Yang, J.; Li, S.; Wang, Z.; Dong, H.; Wang, J.; Tang, S. Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges. *Materials* **2020**, *13*, 5755. [[CrossRef](#)] [[PubMed](#)]
3. Chen, Y.; Ding, Y.; Zhao, F.; Zhang, E.; Wu, Z.; Shao, L. Surface Defect Detection Methods for Industrial Products: A Review. *Appl. Sci.* **2021**, *11*, 7657. [[CrossRef](#)]
4. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [[CrossRef](#)]
5. Khan, F.; Salahuddin, S.; Javidnia, H. Deep Learning-Based Monocular Depth Estimation Methods—A State-of-the-Art Review. *Sensors* **2020**, *20*, 2272. [[CrossRef](#)]
6. Yang, G.; Yang, J.; Sheng, W.; Junior, F.E.F.; Li, S. Convolutional Neural Network-Based Embarrassing Situation Detection under Camera for Social Robot in Smart Homes. *Sensors* **2018**, *18*, 1530. [[CrossRef](#)]
7. Borji, A.; Cheng, M.; Jiang, H.; Li, J. Salient Object Detection: A Benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [[CrossRef](#)]
8. Ciaburro, G. Sound Event Detection in Underground Parking Garage Using Convolutional Neural Network. *Big Data Cogn. Comput.* **2020**, *4*, 20. [[CrossRef](#)]
9. Ciaburro, G.; Iannace, G. Improving Smart Cities Safety Using Sound Events Detection Based on Deep Neural Network Algorithms. *Informatics* **2020**, *7*, 23. [[CrossRef](#)]
10. Costa, D.G.; Vasques, F.; Portugal, P.; Aguiar, A. A Distributed Multi-Tier Emergency Alerting System Exploiting Sensors-Based Event Detection to Support Smart City Applications. *Sensors* **2020**, *20*, 170. [[CrossRef](#)] [[PubMed](#)]
11. Iannace, G.; Ciaburro, G.; Trematerra, A. Fault Diagnosis for UAV Blades Using Artificial Neural Network. *Robotics* **2019**, *8*, 59. [[CrossRef](#)]
12. Peng, L.; Liu, J. Detection and analysis of large-scale WT blade surface cracks based on UAV-taken images. *IET Image Process.* **2018**, *12*, 2059–2064. [[CrossRef](#)]
13. Saied, M.; Lussier, B.; Fantoni, I.; Shraim, H.; Francis, C. Fault Diagnosis and Fault-Tolerant Control of an Octorotor UAV using motors speeds measurements. *IFAC-PapersOnLine* **2017**, *50*, 5263–5268. [[CrossRef](#)]
14. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends Signal Process.* **2014**, *7*, 197–387. [[CrossRef](#)]
15. Tao, X.; Wang, Z.; Zhang, Z.; Zhang, D.; Xu, D.; Gong, X.; Zhang, L. Wire Defect Recognition of Spring-Wire Socket Using Multitask Convolutional Neural Networks. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2018**, *8*, 689–698. [[CrossRef](#)]
16. Jiang, J.; Chen, Z.; He, K. A feature-based method of rapidly detecting global exact symmetries in CAD models. *Comput. Aided Des.* **2013**, *45*, 1081–1094. [[CrossRef](#)]
17. Cheng, J.C.P.; Wang, M. Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Autom. Constr.* **2018**, *95*, 155–171. [[CrossRef](#)]
18. Bergmann, P.; Löwe, S.; Fauser, M.; Sattlegger, D.; Steger, C. Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. *arXiv* **2019**, arXiv:1807.02011.
19. Fang, X.; Luo, Q.; Zhou, B.; Li, C.; Tian, L. Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials. *Sensors* **2020**, *20*, 5136. [[CrossRef](#)]
20. Wang, C.; Bochkovskiy, A.; Liao, H.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
21. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual Transformer Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 1489–1500. [[CrossRef](#)] [[PubMed](#)]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, A.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Long Beach, CA, USA, 2017; pp. 6000–6010.
23. Yang, L.; Zhang, R.; Li, L.; Xie, X. SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Virtual, 18–24 July 2021.
24. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
25. Chetverikov, D.; Hanbury, A. Finding defects in texture using regularity and local orientation. *Pattern Recogn.* **2002**, *35*, 2165–2180. [[CrossRef](#)]

26. Hou, Z.; Parker, J.M. Texture Defect Detection Using Support Vector Machines with Adaptive Gabor Wavelet Features. In Proceedings of the 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1, Breckenridge, CO, USA, 5–7 January 2005; pp. 275–280.
27. Zheng, S.; Zhong, Q.; Chen, X.; Peng, L.; Cui, G. The Rail Surface Defects Recognition via Operating Service Rail Vehicle Vibrations. *Machines* **2022**, *10*, 796. [[CrossRef](#)]
28. Cha, Y.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [[CrossRef](#)]
29. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
30. Zhang, Y.; Cai, W.; Fan, S.; Song, R.; Jin, J. Object Detection Based on YOLOv5 and GhostNet for Orchard Pests. *Information* **2022**, *13*, 548. [[CrossRef](#)]
31. Ye, W.; Ren, J.; Zhang, A.A.; Lu, C. Automatic pixel-level crack detection with multi-scale feature fusion for slab tracks. *Comput.-Aided Civ. Infrastruct. Eng.* **2023**, *1*–18. [[CrossRef](#)]
32. Zhang, W.; Liu, W.; Li, L.; Jiao, H.; Li, Y.; Guo, L.; Xu, J. A framework for the efficient enhancement of non-uniform illumination underwater image using convolution neural network. *Comput. Graph.* **2023**, *112*, 60–71. [[CrossRef](#)]
33. Qiao, Y.; Shao, M.; Liu, H.; Shang, K. Mutual channel prior guided dual-domain interaction network for single image raindrop removal. *Comput. Graph.* **2023**, *112*, 132–142. [[CrossRef](#)]
34. Koulali, I.; Eskil, M.T. Unsupervised textile defect detection using convolutional neural networks. *Appl. Soft Comput.* **2021**, *113*, 107913. [[CrossRef](#)]
35. Wang, J.; Yu, L.; Yang, J.; Dong, H. DBA_SSD: A Novel End-to-End Object Detection Algorithm Applied to Plant Disease Detection. *Information* **2021**, *12*, 474. [[CrossRef](#)]
36. Panboonyuen, T.; Thongbai, S.; Wongweeranimit, W.; Santitamnont, P.; Suphan, K.; Charoenphon, C. Object Detection of Road Assets Using Transformer-Based YOLOX with Feature Pyramid Decoder on Thai Highway Panorama. *Information* **2022**, *13*, 5. [[CrossRef](#)]
37. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks. *Appl. Sci.* **2018**, *8*, 1575. [[CrossRef](#)]
38. Gao, Y.; Gao, L.; Li, X. A hierarchical training-convolutional neural network with feature alignment for steel surface defect recognition. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102507. [[CrossRef](#)]
39. Yoon, S.; Song-Kyoo Kim, A.; Cantwell, W.J.; Yeun, C.Y.; Cho, C.; Byon, Y.; Kim, T. Defect detection in composites by deep learning using solitary waves. *Int. J. Mech. Sci.* **2023**, *239*, 107882. [[CrossRef](#)]
40. Wang, J.; Xu, G.; Li, C.; Wang, Z.; Yan, F. Surface Defects Detection Using Non-convex Total Variation Regularized RPCA With Kernelization. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [[CrossRef](#)]
41. Yang, J.; Shi, Y.; Qi, Z. DFR: Deep Feature Reconstruction for Unsupervised Anomaly Segmentation. *arXiv* **2020**, arXiv:2012.07122.
42. Yang, H.; Chen, Y.; Song, K.; Yin, Z. Multiscale Feature-Clustering-Based Fully Convolutional Autoencoder for Fast Accurate Visual Inspection of Texture Surface Defects. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1450–1467. [[CrossRef](#)]
43. Li, W.; Zhang, H.; Wang, G.; Xiong, G.; Zhao, M.; Li, G.; Li, R. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102470. [[CrossRef](#)]
44. Wang, J.; Dai, H.; Chen, T.; Liu, H.; Zhang, X.; Zhong, Q.; Lu, R. Toward surface defect detection in electronics manufacturing by an accurate and lightweight YOLO-style object detector. *Sci. Rep.* **2023**, *13*, 7062. [[CrossRef](#)]
45. Zhang, Y.; Ni, Q. A Novel Weld-Seam Defect Detection Algorithm Based on the S-YOLO Model. *Axioms* **2023**, *12*, 697. [[CrossRef](#)]
46. Zhou, M.; Lu, W.; Xia, J.; Wang, Y. Defect Detection in Steel Using a Hybrid Attention Network. *Sensors* **2023**, *23*, 6982. [[CrossRef](#)] [[PubMed](#)]
47. Chen, C.; Zheng, Z.; Xu, T.; Guo, S.; Feng, S.; Yao, W.; Lan, Y. YOLO-Based UAV Technology: A Review of the Research and Its Applications. *Drones* **2023**, *7*, 190. [[CrossRef](#)]
48. Li, X.; Wang, Q.; Yang, X.; Wang, K.; Zhang, H. Track Fastener Defect Detection Model Based on Improved YOLOv5s. *Sensors* **2023**, *23*, 6457. [[CrossRef](#)] [[PubMed](#)]
49. Chen, Y.; Liu, H.; Chen, J.; Hu, J.; Zheng, E. Insu-YOLO: An Insulator Defect Detection Algorithm Based on Multiscale Feature Fusion. *Electronics* **2023**, *12*, 3210. [[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.