applied sciences

MDPI

Article

Research on Lightweight Model for Rapid Identification of Chunky Food Based on Machine Vision

Zhongfeng Guo, Junlin Yang * and Siyi Liu

Liaoning Provincial Key Laboratory of Intelligent Manufacturing and Industrial Robots, Shenyang University of Technology, Shenyang 110870, China
* Correspondence: yang_jl@smail.sut.edu.cn

Abstract: To meet the demands of the food industry for automatic sorting of block-shaped foods using DELTA robots, a machine vision detection method capable of quickly identifying such foods needs to be studied. This paper proposes a lightweight model that incorporates the CBAM attention mechanism into the YOLOv5 model, replaces ordinary convolution with ghost convolution, and replaces the position loss function with SIoU loss. The resulting YOLOv5-GCS model achieves a mAP increase from 95.4% to 97.4%, and a reduction in parameter volume from 7.0 M to 6.2 M, compared to the YOLOv5 model. Furthermore, the first 17 layers of the MobileNetv3-large network are replaced with the CSPDarkNet53 network in YOLOv5-GCS, resulting in the YOLOv5-MGCS lightweight model, with a high FPS of 83, which is capable of fast identification of block-shaped foods.

Keywords: deep learning; YOLOv5 algorithm; lightweight model; machine vision

1. Introduction

With the continuous development of China's economy, the demand for food is constantly upgrading, and the domestic food industry is showing a rapid development trend. However, most food enterprises have low automation levels and high labor costs, which seriously affect their efficiency [1,2]. With the continuous development of robot technology, DELTA robots are widely used in high-repetition positions such as food sorting due to their high operating speed [3]. Combined with vision systems, they solve problems such as the low efficiency of manual sorting. Yoshinori Kuno et al. proposed a vision system for robots that combines with a control system to achieve recognition and grasping of batteries by SCARA robots [4]. Hosseininia et al. proposed a vision system for recognizing glass and ceramics to guide robots in polishing ceramics by combining it with a control system [5]. Xu et al. proposed the Light-YOLOv3 algorithm and applied it to robots [6]. This algorithm combines features such as the color, texture, and shape of fruits to design a lightweight module to replace the residual unit in YOLOv3, and uses an improved aggregation module to connect multiscale features for prediction. An experiment shows that the robot has a good detection effect in dense, backlit, long-distance, and special angle scenes under complex lighting conditions. Wang et al. applied the R-CNN algorithm to robots and found that this method can find scattered screws in real-time, realizing the automatic sorting and recovery of screws [7]. Zhang Lin et al. designed a visual medicine bag sorting system that combines a robot control system to complete sorting operations with parallel robots. An experiment shows that the system can efficiently complete visual recognition and sorting tasks [8]. Fang Haifeng et al. combined the vision system with DELTA robots to achieve the classification of plastic bottle garbage through color recognition [9].

According to the needs of enterprises, this paper proposes an improved model based on YOLOv5 to recognize and classify three types of block-shaped foods for automatic sorting by DELTA robots.

check for updates

Citation: Guo, Z.; Yang, J.; Liu, S. Research on Lightweight Model for Rapid Identification of Chunky Food Based on Machine Vision. *Appl. Sci.* **2023**, *13*, 8781. https://doi.org/10.3390/app13158781

Academic Editor: João M. F. Rodrigues

Received: 21 June 2023
Revised: 20 July 2023
Accepted: 25 July 2023
Published: 29 July 2023

2. Building the Dataset

2.1. Image Acquisition

To achieve the task of block-shaped food recognition, a dataset was compiled of images collected manually and from the internet. Three types of food were photographed from multiple angles in different backgrounds, resulting in 1540 manually collected images. As the light and background of our application scene for the DELTA robot sorting food on a conveyor belt is, in practice, relatively stable, the backgrounds of the pictures we selected were not complicated. The food had good lighting. An additional 509 images were collected from the internet, resulting in a dataset of 2049 images, as shown in Table 1.

Table 1. Dataset composition.

Category	Manually Collected	Internet Collected	Total
Mashu	532	210	742
Fantuan	496	156	652
Nuomiji	512	143	655
Total	1540	509	2049

2.2. Dataset Augmentation

Deep-learning-based object detection algorithms require a large number of images for training. When the number of image samples in the dataset is small, it often leads to problems such as model underfitting and poor robustness. Therefore, data augmentation techniques are used in this paper to increase the number of images [10].

(1) Geometric Transformation

Geometric transformation involves operations such as the translation, flipping, and scaling of images. The transformed images are shown in Figure 1.

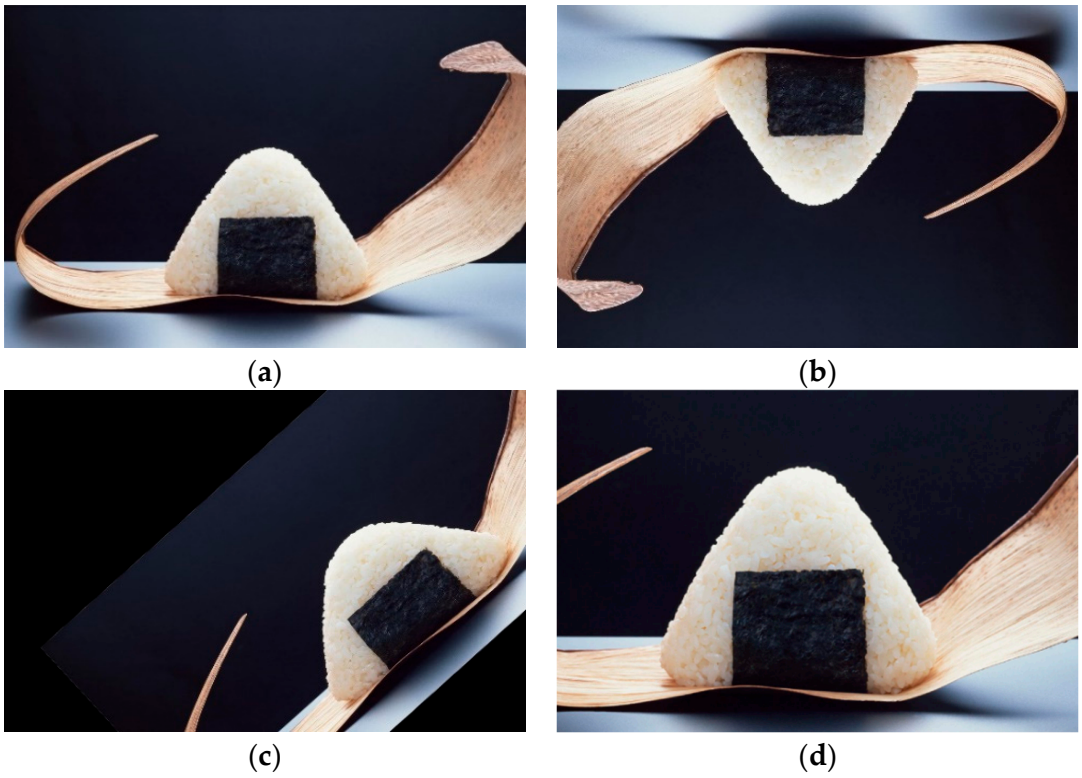


Figure 1. Image geometry transformation: (**a**) original image; (**b**) flipped image; (**c**) rotated image; (**d**) cropped image.

(2) Adding Noise

Noise refers to signal interference that occurs during image acquisition or transmission, and the most common types of noise are salt and pepper noise and Gaussian noise. Since noise is randomly distributed in the image, probability density functions are often used to model noise. Gaussian noise, also known as normal noise [11], has the following mathematical model:

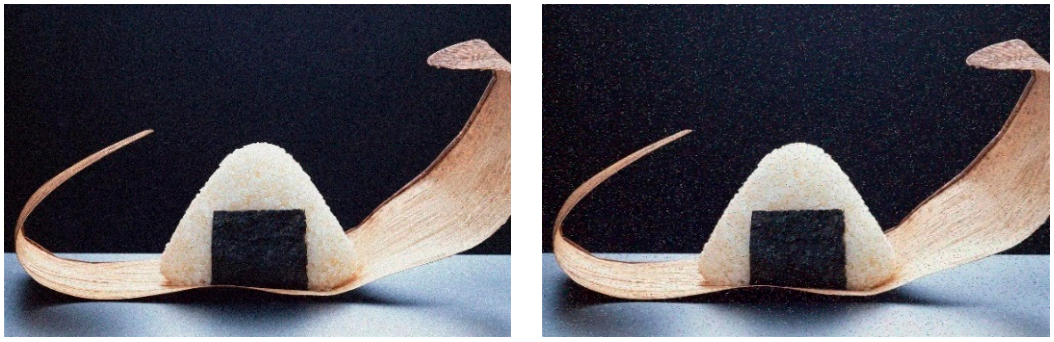
$$P(Z) = \frac{1}{\sqrt{z\pi\sigma}} exp\left\{ -\frac{(z-\mu)^2}{2\sigma^2} \right\} \quad (1)$$

where z is the gray value, μ represents the average value of z, and P(Z) is the probability density of the noise. Gaussian noise is often distributed around the mean, and as the difference between the gray value and the mean increases, the noise gradually decreases.

Salt and pepper noise, also known as impulse noise, has strong randomness and can be expressed by Equation (2).

$$p(z) = \begin{cases} p_a & z = a \\ p_b & z = b \\ 0 & else \end{cases} \quad (2)$$

where a and b are the gray values of salt and pepper noise. When a < b, the noise is represented by black dots, and when a > b, the noise is represented by white dots. The images after adding noise are shown below (Figure 2).

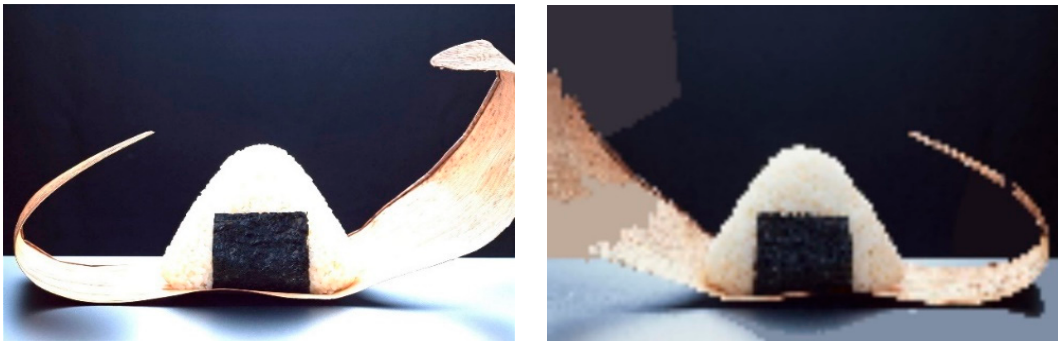


(**a**) (**b**)

Figure 2. Image after adding noise: (**a**) Gaussian noise; (**b**) pepper noise.

(3) Color Transformation

Unlike geometric transformation, color transformation changes the pixel's gray value without changing its coordinates. Common color transformations include changing brightness, changing contrast, and Gaussian blur. The transformed images are shown below (Figure 3).



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

Figure 3. Image color conversion: (**a**) changing brightness image; (**b**) Gaussian blur image.

(4) Cutout

Cutout is a data augmentation method proposed by Devries et al. in 2017 [12]. The main idea is to randomly crop a part of the image and fill the area with 0. Experimental results have shown that cutout is similar to the dropout regularization method in neural networks, which can prevent overfitting and improve the robustness of neural networks. It can also be used with other data augmentation operations to enhance the diversity of data. The images after the cutout operation are shown in Figure 4.

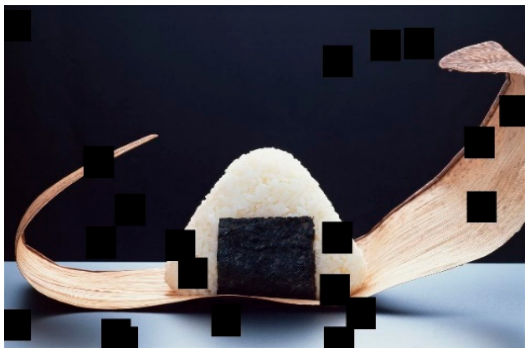


Figure 4. Cutout image enhancement.

After data augmentation, a total of 12,000 images were obtained and annotated for each image. The images in the training set, validation set, and test set were distributed in a ratio of 8:1:1.

3. Improved YOLOv5 Algorithm

3.1. YOLOv5 Algorithm

YOLOv5 is an improved version of YOLOv4 [13]. The structure of YOLOv5 is shown in Figure 5.

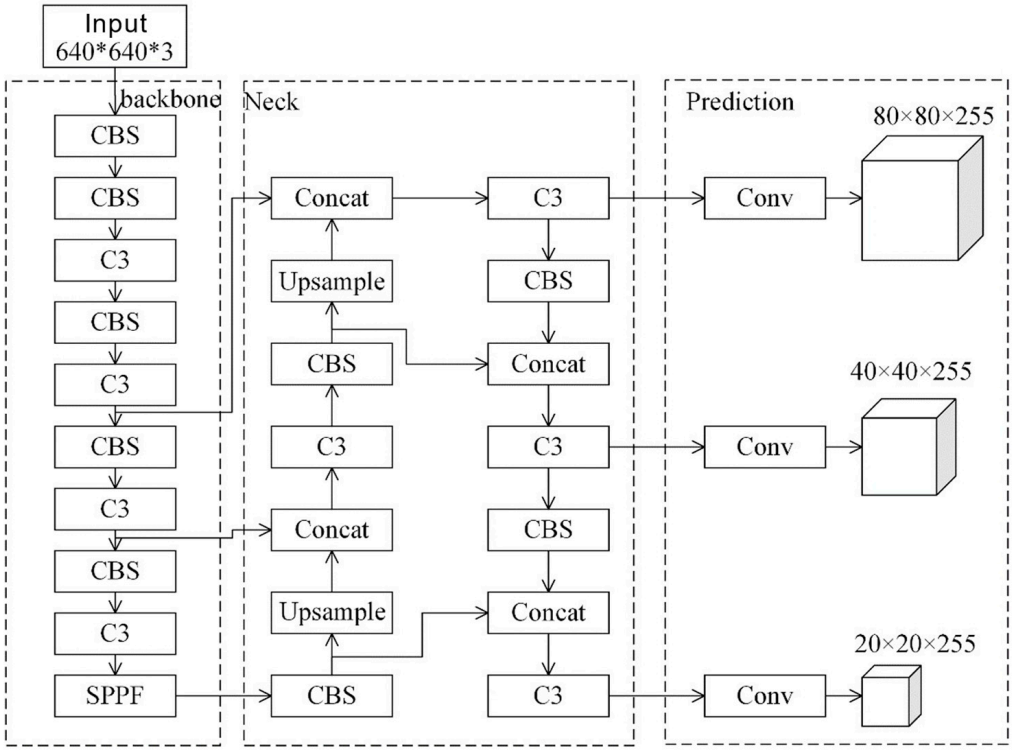


Figure 5. YOLOv5 structure.

According to the different depth and width of the network, YOLOv5 can be divided into five versions, among which YOLOv5n has the fastest detection speed, but the lowest detection accuracy. YOLOv5x has the highest detection accuracy, but the largest size and lower detection speed. In order to balance the detection accuracy and speed, YOLOv5s is used as the base model and improved in this paper.

3.2. Construction of YOLOv5-GCS Detection Model

3.2.1. CBAM Attention Module

Attention mechanisms are based on the study of human vision, where individuals selectively attend to specific information while disregarding other less important information due to limitations in their mental perception [14]. The attention mechanism in deep learning is similar to that of human vision, where important features are selectively focused on while disregarding irrelevant information. Incorporating attention mechanisms in neural networks can improve detection accuracy by addressing interference from the environment.

There are three main types of attention mechanisms in the visual domain: spatial, channel, and hybrid. The CBAM (convolutional block attention module) algorithm is a hybrid attention mechanism that contains two sub-modules, the channel attention module (CAM) and spatial attention module (SAM). This algorithm not only reduces computational effort but also locates important information more efficiently. Its structure is shown in Figure 6.

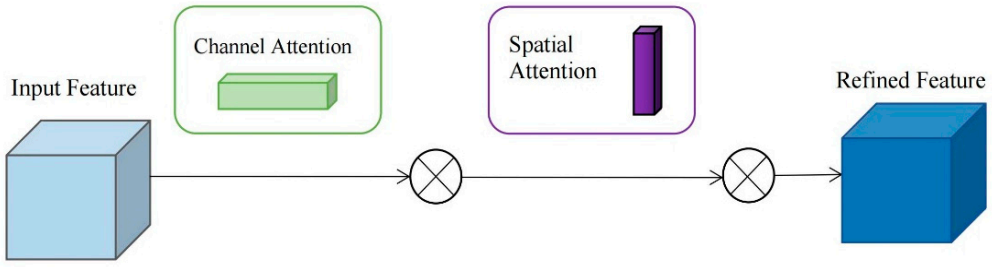


Figure 6. CBAM attention mechanism.

In this paper, we propose the integration of the CBAM module into the feature fusion layer of YOLOv5 [15]. The modified structure of the feature fusion layer, depicted in Figure 7, includes the CBAM module inserted after the C3 module and before the CBS. By leveraging the attention mechanism, the CBAM enhances the target features prior to the feature fusion operation. This enables the network to effectively suppress background noise, thereby enhancing the localization ability of the target and potentially reducing computation time while improving detection speed.

3.2.2. SIoU Loss Function

The YOLOv5 model employs the CIoU loss function, which does not consider the mismatch in orientation between the ground truth and predicted bounding boxes. This limitation leads to slow convergence and inefficiency. To address these issues, we propose the use of the SIoU loss function to replace the original loss function.

The SIoU loss function considers the coverage area, distance between center points, aspect ratio, and angle. The formula for the SIoU loss function is shown below [16]:

$$SIoU = IOU - \frac{\Delta + \Omega}{2} \tag{3}$$

$$L_{SIoU} = 1 - IOU + \frac{\Delta + \Omega}{2} \tag{4}$$

where Δ represents the distance loss function, and Ω represents the aspect ratio loss function. The distance loss function takes into account the angle loss. The expression for the angle loss is as follows:

$$\Lambda = 1 - 2sin^2\left(arcsin\left(\frac{C_h}{\sigma}\right) - \frac{\pi}{4}\right) \tag{5}$$

$$\alpha = arcsin\frac{C_h}{\sigma} \tag{6}$$

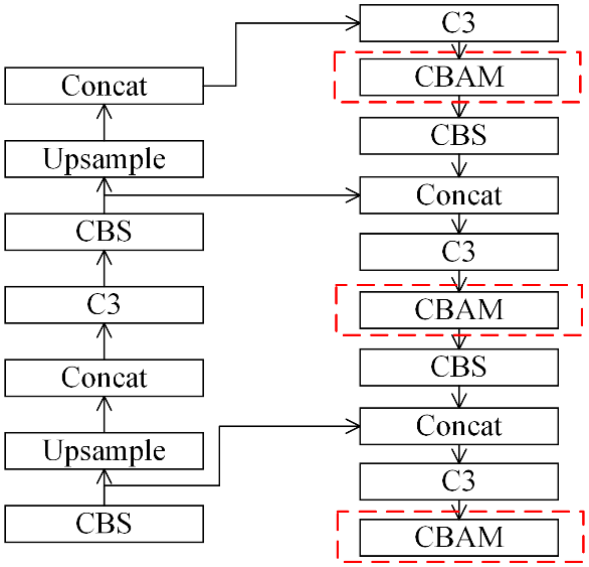


Figure 7. Feature fusion layer after adding CBAM.

Here, C_h represents the height difference between the ground truth and predicted bounding boxes, σ represents the distance between the centers of the two boxes, and α represents the angle between σ and the horizontal direction. The angle loss value is 0 when α is 0 or 90°. The angle penalty term for the SIoU is shown in Figure 8.

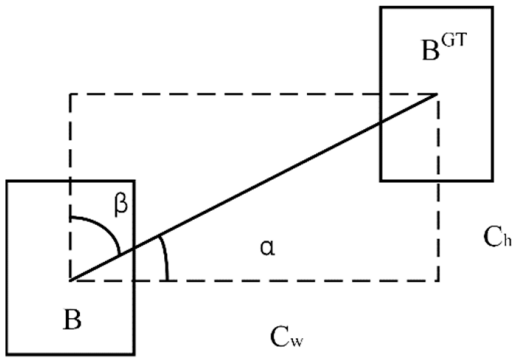


Figure 8. Angle penalty terms.

The expression for the distance loss is as follows:

$$\Delta = 2 - e^{-\gamma\rho_x} - e^{-\gamma\rho_y} \tag{7}$$

$$\gamma = 2 - \Lambda \tag{8}$$

where ρ_x and ρ_y represent the distance loss terms for the x and y coordinates of the center points of the ground truth and predicted bounding boxes. The closer the distance, the closer the value of the loss term is to 0. γ is influenced by the angle loss, and when the two boxes tend to be parallel, Λ tends to 0, and γ tends to 2. As a result, the proportion

of distance between the two boxes in the loss function decreases. When α tends to 45°, Λ tends to 1, and γ tends to 1, resulting in an increase in the proportion of distance between the two boxes in the loss function.

The expression for the aspect ratio loss is as follows:

$$\Omega = \left(1 - e^{-W_w}\right)^{\theta} + \left(1 - e^{-W_h}\right)^{\theta} \tag{9}$$

where

$$W_w = \frac{|w - w^{gt}|}{max(w, w^{gt})} \tag{10}$$

$$W_h = \frac{|h - h^{gt}|}{max(h, h^{gt})} \tag{11}$$

In Equation (9), θ is an adjustable parameter that controls the degree of attention to shape loss and needs to be selected based on experimental results. In Equations (10) and (11), (w, h) and (w^{gt}, h^{gt}) represent the width and height of the predicted and ground truth bounding boxes, respectively.

3.2.3. Ghost Convolution

GhostNet is a novel neural network architecture proposed by Han et al. in 2020 [17], which is based on the ghost convolution module. The main idea is to split the convolution into two steps. A comparison of normal convolution and ghost convolution is shown in Figure 9.

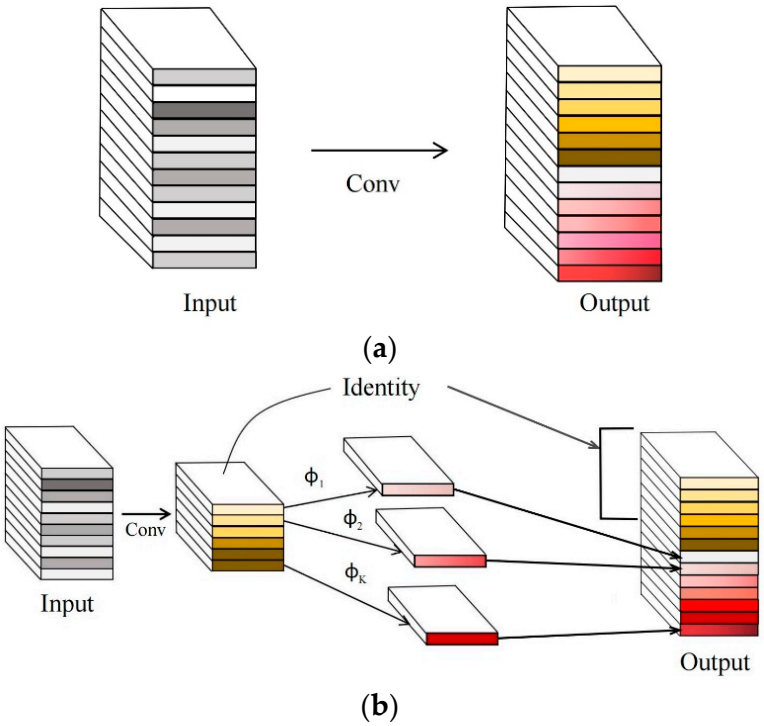


Figure 9. Comparison of normal convolution and ghost convolution: (**a**) normal convolution; (**b**) ghost convolution.

In deep learning, a large number of redundant feature maps are typically generated to ensure a comprehensive understanding of the data by the network. However, many output features are similar, and only a simple linear transformation of one feature map is needed to obtain a new feature map. One feature map can be considered the "ghost" of another. Ghost convolution first uses a small number of convolutions to generate some feature maps, and then performs linear operations on these feature maps to obtain ghost

feature maps. Finally, the feature maps are concatenated by channel, which improves the detection speed while maintaining model accuracy.

Assuming the kernel size of the ghost convolution is d × d, the ratio of parameters between normal convolution and ghost convolution is as follows:

$$Rate = \frac{W' \cdot H' \cdot n \cdot k \cdot k \cdot C}{W' \cdot H' \cdot m \cdot k \cdot k \cdot C + W' \cdot H' \cdot (n-m) \cdot d \cdot d} = \frac{W' \cdot H' \cdot n \cdot k \cdot k \cdot C}{W' \cdot H' \cdot \frac{n}{s} \cdot k \cdot k \cdot C + W' \cdot H' \cdot (s-1) \cdot \frac{n}{s} \cdot d \cdot d} = \frac{k \cdot k \cdot C}{\frac{1}{s} \cdot k \cdot k \cdot C + \frac{(s-1)}{s} \cdot d \cdot d} \approx \frac{s \cdot C}{s + C - 1} \approx s \quad (12)$$

From the simplified result, it can be inferred that the parameter count of normal convolution is roughly s times that of ghost convolution. Therefore, replacing the normal convolution in the feature fusion layer of YOLOv5 with ghost convolution can improve the detection efficiency of the model.

3.3. Experiment Validation

3.3.1. Experimental Environment and Hyperparameter Settings

The experimental environment and hyperparameter settings are shown in Tables 2 and 3.

Table 2. Experimental environment.

Parameters	Configuration
Operating System	Ubuntu 18.04
CPU	Intel(R) Xeon(R) Platinum 8255C
GPU	RTX3080
Programming Languages	Python 3.8
Deep Learning Framework	Pytorch 1.9
Accelerated Environment	CUDA 11.0

Table 3. Hyperparameter settings.

Name	Numerical Value
Training image resolution	640 × 640 × 3
Epochs	200
Batch_size	16
Optimizer	SGD
Initial learning rate	0.01
Learning rate momentum (momentum)	0.937
Weight decay factor	0.0005

3.3.2. Comparison Experiment

Convergence Performance Analysis

To verify the convergence performance of YOLOv5-GCS, a comparison will be made between YOLOv5-GCS and the original model, and the performance of the models will be analyzed. The loss and mAP (mean average precision) curves of the original model and YOLOv5-GCS on the training set are shown in Figure 10.

The loss functions of both models start to decrease rapidly in the first 50 rounds of training and level off after 100 rounds. Notably, all three loss functions of the YOLOv5-GCS model are significantly smaller than those of YOLOv5s. A comparison of the mAP curves of YOLOv5-GCS and YOLOv5s is shown in Figure 10d, where the mAP of the YOLOv5-GCS model rapidly increases to 90% in the first 50 rounds of training and reaches around 97% after 100 rounds. The final results for the two models are 97.4% and 95.4%, respectively, proving that YOLOv5-GCS outperforms YOLOv5s in detection performance.

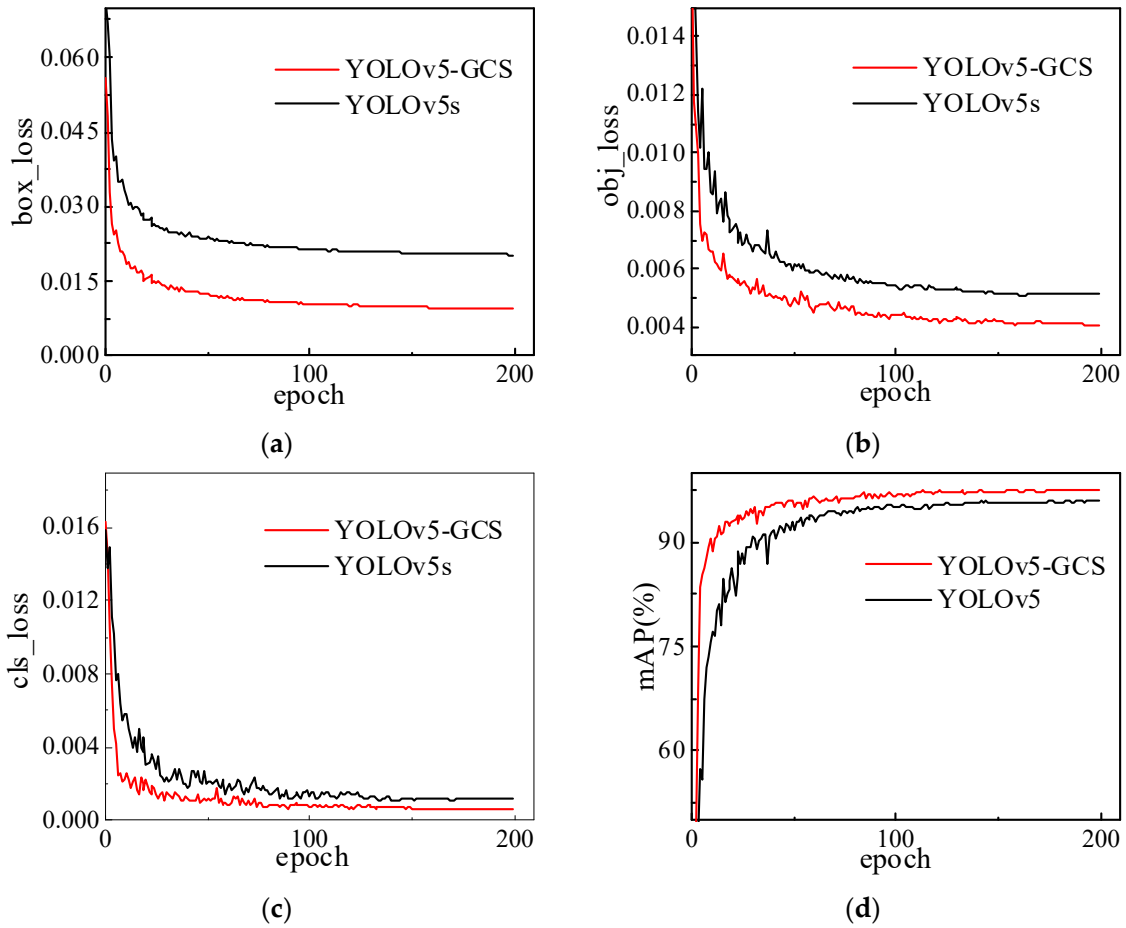


Figure 10. Loss and mAP curve charts: (**a**) box_loss; (**b**) obj_loss; (**c**) cls_loss; (**d**) mAP.

Classification Accuracy Analysis

After image preprocessing, we expanded the number of images to 12,000 with a large number of samples. Therefore, we divided the images into a training set, testing set, and validation set with the ratio of 8:1:1; we did not use the K-fold cross-validation method because it would increase the computational cost. The confusion matrices generated by YOLOv5s and YOLOv5-GCS are shown in Figure 11.

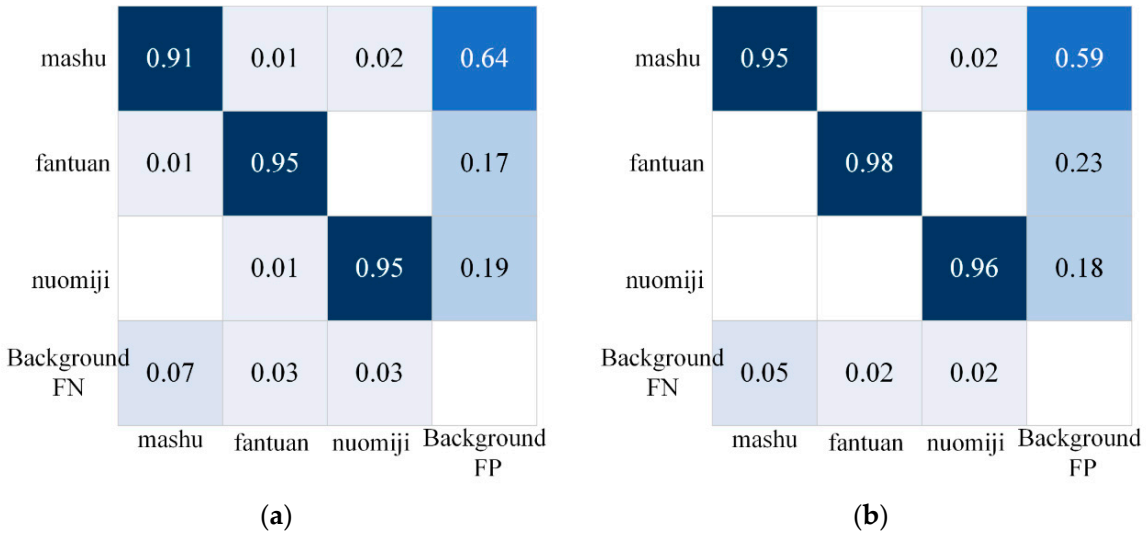


Figure 11. Confusion matrices generated by different models: (**a**) YOLOv5s; (**b**) YOLOv5-GCS.

From Figure 11a, the classification accuracies of mashu, fantuan, and nuomiji in YOLOv5s are 91%, 95%, and 95%, respectively. Among them, mashu has a 1% chance of being misidentified as fantuan and a 7% chance of being misidentified as background. Fantuan has a 1% chance of being misidentified as mashu and numiji, and a 3% chance of being identified as background. Nuomiji has a 2% chance of being identified as mashu and a 3% probability of being identified as background. This shows that the YOLOv5s model produces false and missed detections.

As shown in Figure 11b, the classification accuracies of mashu, fantuan, and nuomiji in the YOLOv5-GCS model are 95%, 98%, and 96%, respectively, which are 4%, 3%, and 1% better than YOLOv5s. Mashu and fantuan have no false detections, and nuomiji has a 2% chance of being falsely detected as mashu. The chances of several categories being recognized as background are reduced compared to YOLOv5s. In summary, YOLOv5-GCS can effectively improve the classification accuracy, reduce the probability of false detection and missing detection, and significantly improve the model performance.

Ablation Experiments

To verify the effects of the three improvements of CBAM, SIoU loss, and ghost convolution on the model, several sets of experiments are designed in this paper, and the experimental results are shown in Table 4. The accuracy and recall rates are improved after introducing SIoU loss, ghost convolution, and the CBAM attention mechanism in the network alone, and the number of model parameters is reduced after introducing ghost convolution. Adding both CBAM and SIoU loss to the model significantly improved the accuracy and recall, and increased the mAP by 1.4%. Adding CBAM and ghost convolution to the original model also improved the accuracy and recall of the model. Overall, compared with YOLOv5s, the YOLOv5-GCS model's precision, P, is improved by 2%, recall R by 1.4%, mAP by 2%, and the number of parameters by 0.8 M.

Table 4. Results of ablation experiments.

YOLOv5s	CBAM	SIoU	Ghost	P (%)	R (%)	mAP (%)	Number of Participants
√				92.7	93	95.4	7.0 M
√	√			94.3	93.7	96.6	7.2 M
√		√		92.9	93.5	96.0	7.0 M
√			√	93.5	93.7	96.3	6.0 M
√	√	√		94.3	93.8	96.8	7.2 M
√	√		√	94.4	94	97	6.2 M
√	√	√	√	94.7	94.4	97.4	6.2 M

Performance Analysis of Different Attention Mechanisms

To verify the effect of combining different attention mechanisms on the model, we introduced the feature fusion layer of the YOLOv5 algorithm into CBAM, SE, and CA for comparison experiments [18,19]. The mAP comparison of the three attention mechanisms with the original algorithm on the training set is shown in Figure 12.

The mAPs of all three attention mechanisms are higher than the original model, indicating that the introduction of attention mechanisms can improve the model's attention to the main features, enabling it to extract more effective information and improve the model performance. Among the three attention mechanisms, the CBAM attention mechanism improves the original model the most, and its effect is better than that of the CA and SE attention mechanisms.

A comparison of the performance of the three attention mechanisms on the validation set is shown in Table 5.

Based on the data in Table 5, it can be observed that the introduction of the SE, CA, and CBAM attention mechanisms into the model can improve mAP by 0.5%, 0.7%, and 1.2%, respectively. Therefore, it can be demonstrated that introducing the CBAM attention

mechanism into the YOLOv5s feature fusion layer improves the model performance more than other attention mechanisms.

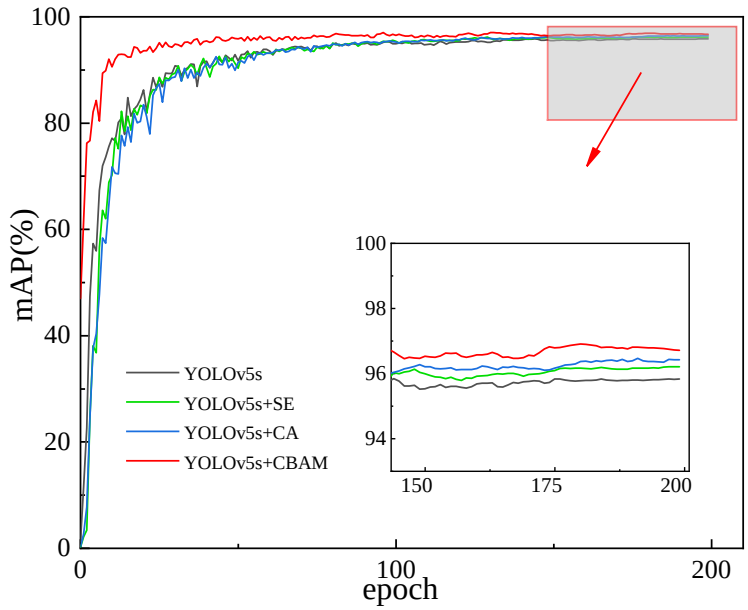


Figure 12. Comparison of the effect of different attention mechanisms on mAP.

Table 5. Performance comparison of three attention mechanisms.

Algorithm	P (%)	R (%)	mAP (%)	Number of Parameters (M)
YOLOv5s	92.7	93	95.4	7.0
YOLOv5s-SE	93	93.3	95.9	7.2
YOLOv5s-CA	92.9	93.4	96.1	7.2
YOLOv5s-CBAM	94.3	93.7	96.6	7.2

Comparison of Different Algorithms

To verify the superiority of the YOLOv5-GCS model, we compared it with several common target detection algorithms. The PR curves of each algorithm in the validation set are shown in Figure 13.

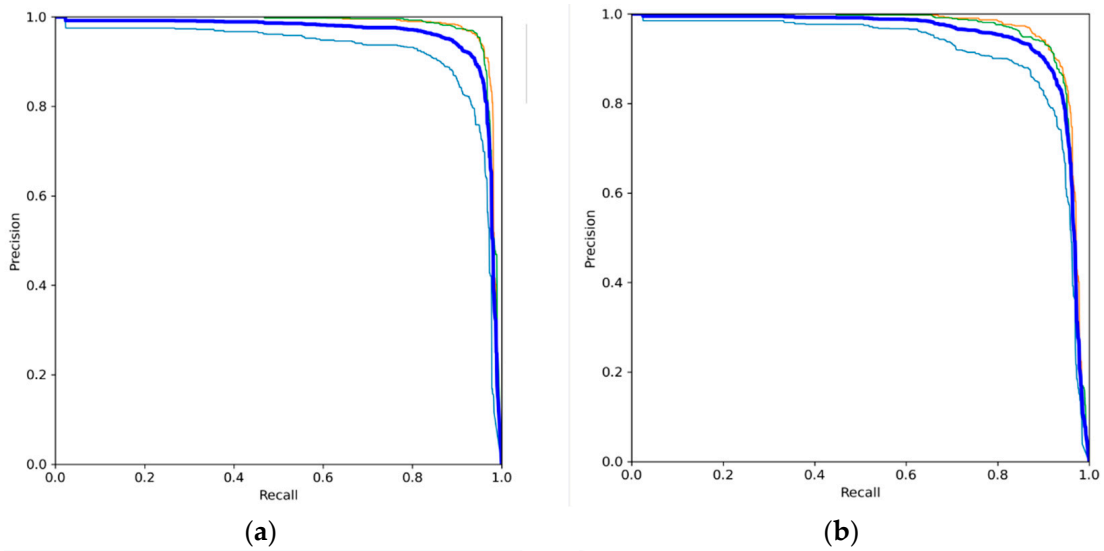
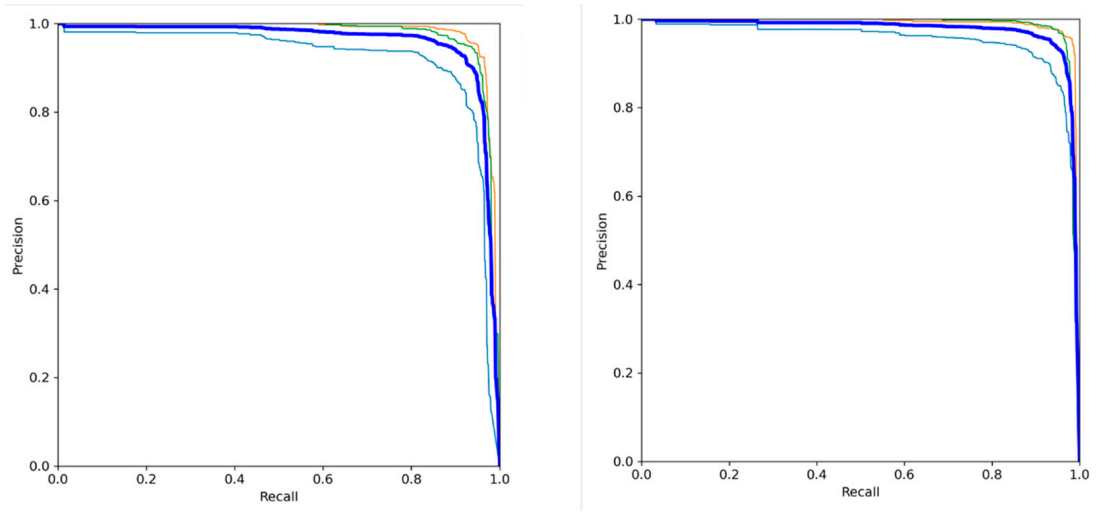


Figure 13. *Cont.*

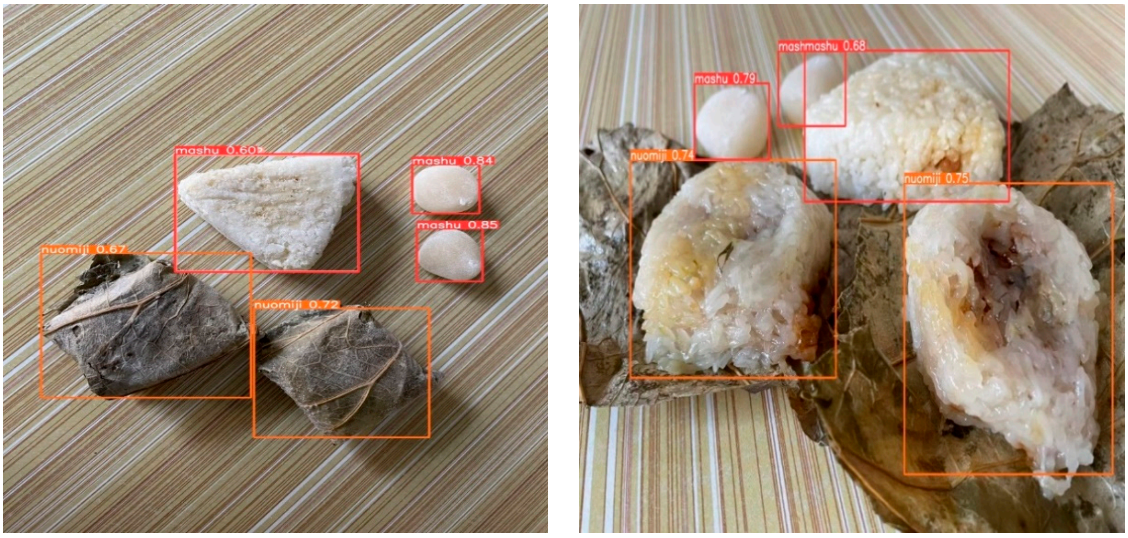


(**c**) (**d**)

Figure 13. PR curves of different algorithms in the validation set: (**a**) YOLOv5s; (**b**) YOLOv4; (**c**) YOLOv6s; (**d**) YOLOv5-GSC.

The performance of the model can be evaluated based on the area enclosed by the PR curve and the mAP value. As shown above, the area enclosed by the PR curve of the YOLOv4 algorithm is the smallest, while the area enclosed by the PR curve of YOLOv5-GCS is the largest, indicating that the model performance is optimal.

The detection effects of the different algorithms are shown in Figure 14. YOLOv5-GCS has a high correct recognition rate, with no missed detection or false detection, and the confidence level is higher than that of other models. This indicates that the improvement strategy proposed in this paper can effectively enhance the performance of YOLOv5s.



(**a**)

Figure 14. *Cont.*

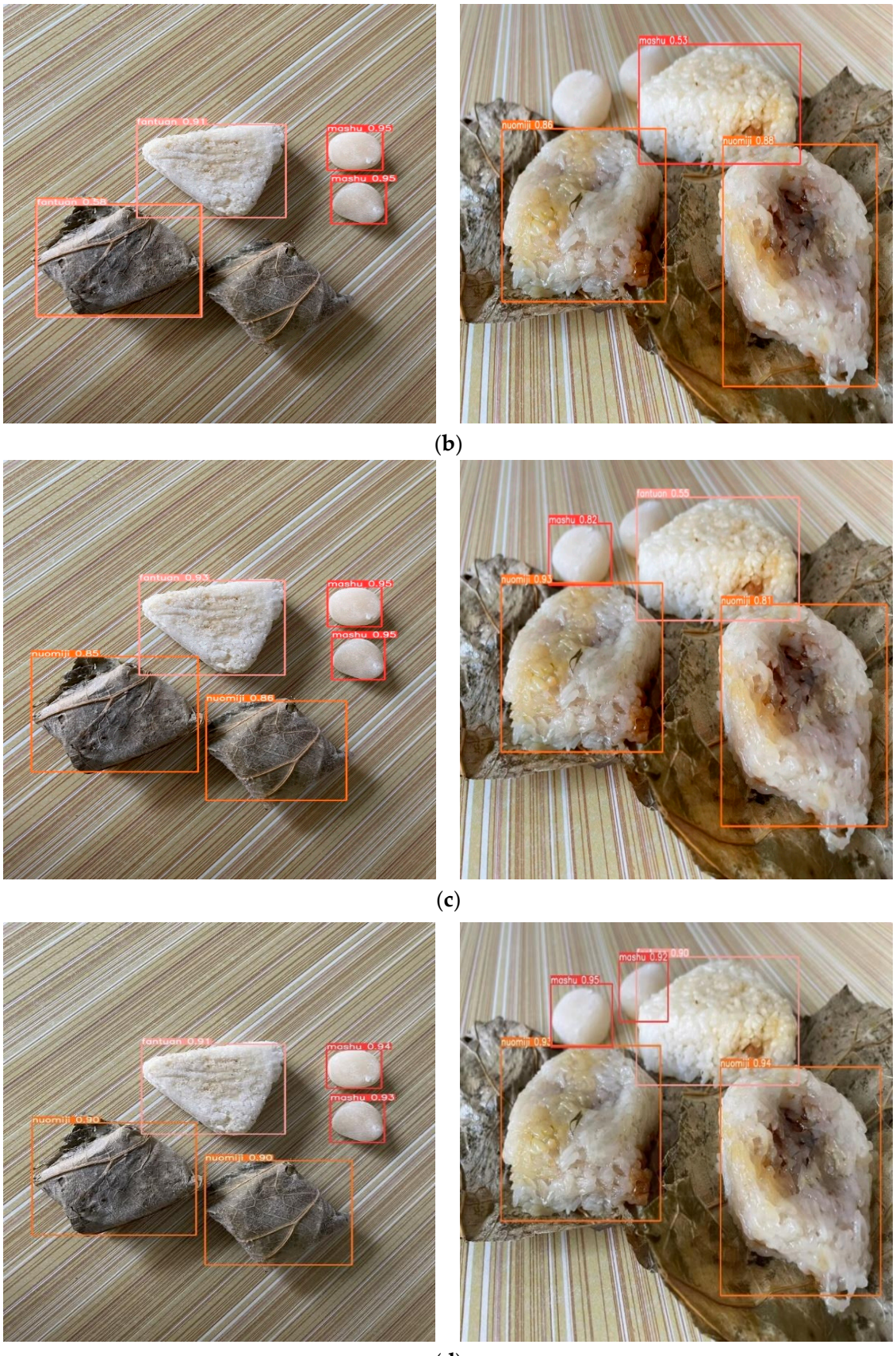


Figure 14. Comparison of the detection effect of each model: (**a**) YOLOv5s; (**b**) YOLOv4; (**c**) YOLOv6s; (**d**) YOLOv5-GSC.

4. Lightweight Model YOLOv5-MGCS

4.1. YOLOv5-MGCS Model

Due to the complex structure of the CSPDarkNet53 network in YOLOv5-GCS, the model has a large number of parameters and low FPS. To adapt to the high-speed sorting of DELTA robots, we improved the YOLOv5-GCS model by replacing the CSPDarkNet53 feature extraction network with the first 17 layers of the MobileNetv3-large network [20–22]. The feature fusion layer and detection head were kept unchanged, resulting in a lightweight model known as YOLOv5-MGCS. The network structure of YOLOv5-MGCS is as shown below (Figure 15).

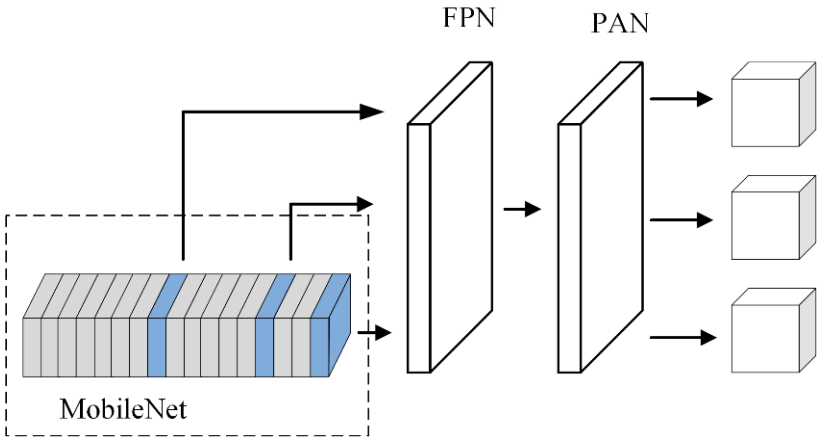


Figure 15. YOLO-MGCS structure.

4.2. Experimental Training and Analysis of Results

For the lightweight model, we used the same experimental environment and dataset as described above for training. The loss function of the YOLOv5-MGCS model with mAP on the training set is shown in Figure 16. The loss function decreases rapidly in the first 50 rounds, stabilizes after 100 rounds, and starts to converge after 150 rounds. In the mAP curve, the mAP rises rapidly to 90% in the first 50 rounds and starts to approach mAP values close to 96% after 100 rounds. The final mAP reached 96.5%.

To further verify the effect of the lightweight improvement strategy on the model performance and detection speed, we compared YOLOv5-MGCS with the YOLOv4, YOLOv5s, and YOLOv5-GCS models. As shown in Table 6, YOLOv4 has the lowest detection accuracy and the slowest detection speed, while YOLOv5s has the largest number of parameters. YOLOv5-GCS has the highest detection accuracy, of 97.4%, and a lower number of parameters (0.7 M less than YOLOv5s), with an improved FPS, from 55 to 60. Although the mAP is reduced in YOLOv5-MGCS compared to YOLOv5-GCS, the number of parameters is only 0.7 M less than YOLOv5s. Therefore, YOLOv5-MGCS meets the application requirements.

Table 6. Results of different detection models.

Model	mAP (%)	Number of Parameters (M)	FPS
YOLOv4	94.3	6.3	23
YOLOv5s	95.4	7.0	55
YOLOv5-GCS	97.4	6.3	60
YOLOv5-MGCS	96.5	3.3	83

The detection results of the lightweight model YOLOv5-MGCS are shown in the Figure 17 below.

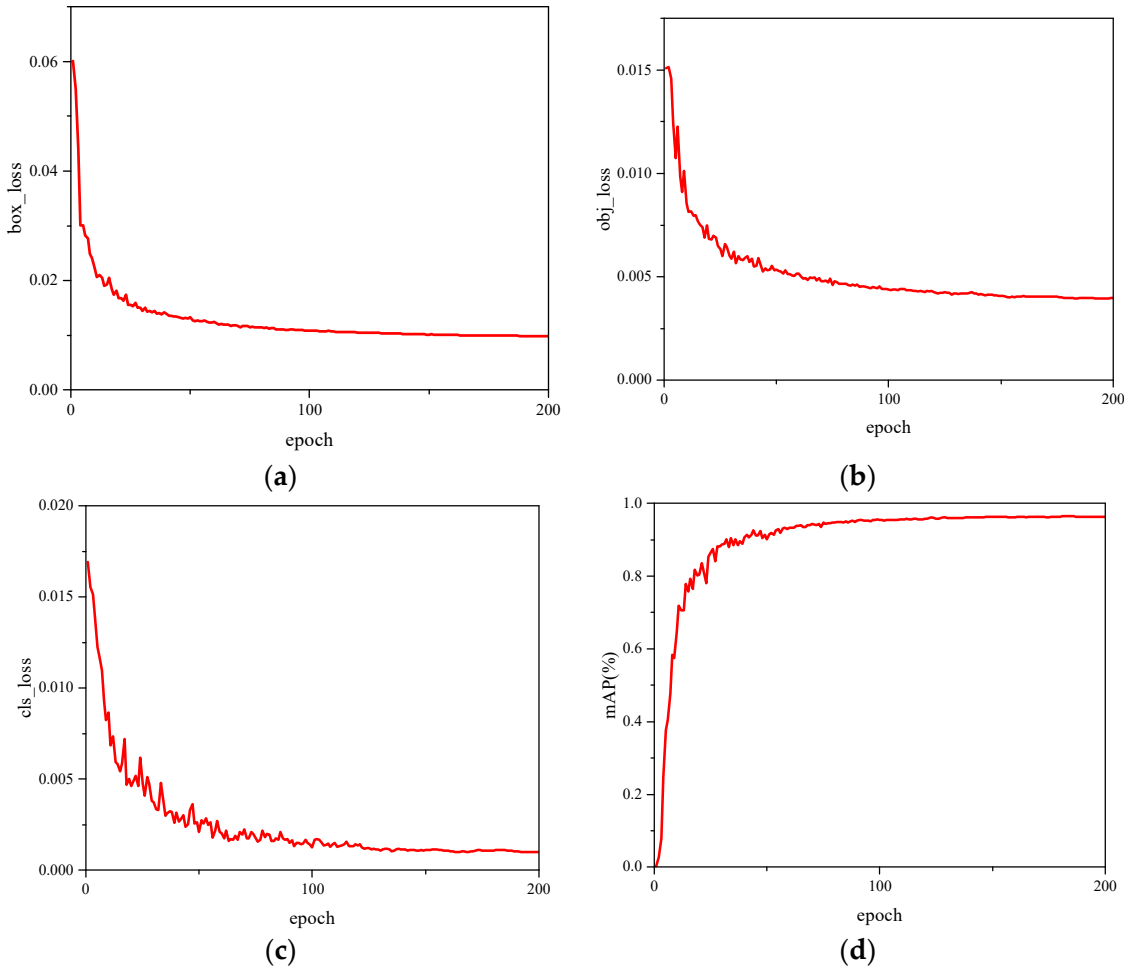


Figure 16. Loss and mAP curves for the lightweight model: (**a**) box_loss; (**b**) obj_loss; (**c**) cls_loss; (**d**) mAP.

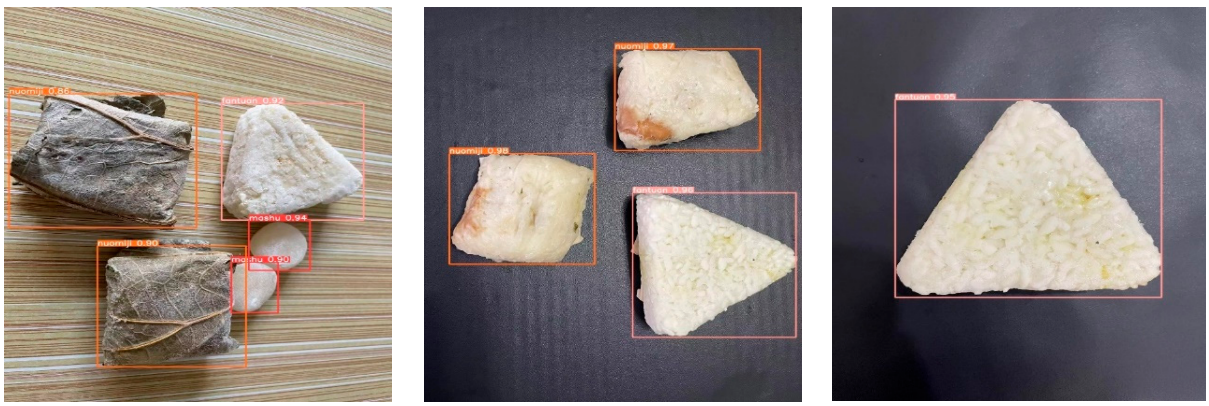


Figure 17. The detection effect of the lightweight model.

5. Conclusions

Since the DELTA robot is working at a high speed, it requires image processing equipment with a high detection speed to give feedback to the DELATA robot. This paper proposes the YOLOv5-MGCS model, which is based on the YOLOv5 model but has been improved and designed to meet the needs of enterprise applications. The specific improvements are as follows:

(1) The CBAM attention mechanism is added to the feature fusion network of YOLOv5s, and the normal convolution is replaced with the ghost convolution module. Addition-

ally, the position loss function in YOLOv5s is replaced with SIoU loss. The improved YOLOv5-GCS model detects block food significantly better than YOLOv5s, with a mAP value improved from 95.8% to 97.5%, and a reduction in the number of model parameters from 7 M to 6.3 M.

(2) A lightweight model, YOLOv5-MGCS, is proposed, where the first 17 layers of the MobileNetv3-large network are selected to replace the CSPDarkNet53 network in YOLOv5-GCS. The FPS value of the improved model YOLOv5-MGCS is up to 83, which can meet the demand of real-time detection. The number of parameters has been changed from 7.0 M to 3.3 M to reduce the CPU computing burden.

In conclusion, the proposed YOLOv5-MGCS model has achieved significant improvements in detection accuracy and detection speed, making it suitable for practical applications in the food industry.

Author Contributions: Z.G.: conception of the study, proposition of the theory and method, supervision; J.Y.: literature search, figures, manuscript preparation and writing, programming, testing of exiting code components; S.L.: data collection. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Liaoning Provincial Education Department Project (grant no. LJKZ0114).

Institutional Review Board Statement: The study did not require ethical approval.

Informed Consent Statement: The study did not involve humans.

Data Availability Statement: Not applicable.

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

References

1. Yan, Z.; Wang, L.; Sun, Z. Study of High-speed Auto-sorting System for Food Production. *Packag. Eng.* **2009**, *30*, 16–18.
2. Gong, C.; Lan, L.; Xiong, J. Design Analysis and Implementation of Delta Parallel Robot. *J. Mech. Transm.* **2014**, *38*, 61–63.
3. Zhang, W. Control Technique and Kinematic Calibration of Delta Robot Based on Computer Vision. Ph.D. Thesis, Tianjin University, Tianjin, China, 2012; pp. 25–35.
4. Kuno, Y.; Numagami, H.; Ishikawa, M.; Hoshino, H.; Nakamura, Y.; Kidode, M. Robot vision implementation by high-speed image processor TOSPIX: Battery inspection. *Robotica* **1983**, *1*, 223–230. [CrossRef]
5. Hosseininia, S.J.; Khalili, K.; Emam, S.M. Flexible Automation in Porcelain Edge Polishing Using Machine Vision. *Procedia Technol.* **2016**, *22*, 562–569. [CrossRef]
6. Xu, Z.; Jia, R.; Sun, H.; Liu, Q.; Cui, Z. Light-YOLOv3: Fast method for detecting green mangoes in complex scenes using picking robots. *Appl. Intell.* **2020**, *50*, 4670–4687. [CrossRef]
7. Wang, Z.; Li, H.; Zhang, X. Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach. *Autom. Constr.* **2019**, *97*, 220–228. [CrossRef]
8. Zhang, L.; Li, Y.; Qin, B.; Shang, D. Design of the Medicine Bag Sorting System Based on Machine Vision and Parallel Robot. *Mach. Tool Hydraul.* **2019**, *47*, 68–71.
9. Fang, H.; Xu, K.; Wan, X.; Kai, W. Design of Recyclable Garbage Sorting System Based on Parallel Robot. *Modul. Mach. Tool Autom. Manuf. Tech.* **2020**, 24–27+32. [CrossRef]
10. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
11. Luisier, F.; Blu, T.; Unser, M. Image denoising in mixed Poisson-Gaussian noise. *IEEE Trans. Image Process.* **2010**, *20*, 696–708. [CrossRef] [PubMed]
12. Devries, T.; Taylor, G.W. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv* **2017**, arXiv:1708.04552.
13. Bochkovskiy, A.; Wang, C.Y.; Liaom, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. Zhang, S.; Wang, H.; Ran, X. Light traffic sign detection algorithm based on YOLOv5. *Electron. Meas. Technol.* **2022**, *45*, 129–135.
15. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
16. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
17. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.

18. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE Press: New York, NY, USA, 2018; pp. 7132–7141.
19. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, arXiv:2103.02907.
20. Howard, A.G. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2020**, arXiv:1704.04861.
21. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and Linear bottlenecks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
22. Howard, A.; Sandler, M.; Chu, G. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.