IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024EAP1101

Publicized:2024/10/18

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER *Special Section on LaTeX2ε Class File for the IEICE Transactions*

An Efficient Method for Sea Cucumber Recognition and Sorting Based on Improved YOLOv9 and RepViT

Meng HUANG[†] ***and*** **Honglei WEI**[†*], ***Nonmembers***

SUMMARY The automatic sorting system for sea cucumbers in food processing plants faces challenges such as high false detection rates, slow processing speeds, and sensitivity to light intensity variations. This paper presents a high-precision, high-efficiency real-time recognition and sorting method for sea cucumbers, based on YOLOv9 and the RepViT network. We improved the YOLOv9 model by introducing auxiliary training modules to help the model better understand the characteristics of sea cucumbers. Additionally, we used the lightweight RepViT network as the backbone to enhance the model's expressive power and computational efficiency while maintaining a low weight. We replaced the original CIoU loss function with the EIoU loss function to accelerate convergence. Experimental results show that our improved model achieves an accuracy of 98.33% in sea cucumber sorting, with an inference speed of 92.71 fps and a model size of only 42.53 MB, outperforming most detection models. Moreover, the average sorting speed for a single sea cucumber is just 0.92 seconds, meeting the production needs of food processing plants.

key words: *Sea Cucumber Classification, YOLOv9, RepViT Network, Object Detection, Auxiliary Training Head*

1. Introduction

Sea cucumbers are renowned for their high nutritional value, and with economic development, their production has steadily increased [1]. However, before these sea cucumbers are sold, it is necessary to screen out any defective ones. Currently, the sorting process is primarily conducted manually or with mechanical assistance [2]. Although manual sorting is reliable, it is slow, and prolonged operation can cause visual fatigue, increasing the likelihood of errors. Moreover, when mechanical equipment is used to assist in sea cucumber sorting, changes in external conditions can also lead to inaccurate identification [3]. Therefore, developing automated sorting equipment for sea cucumbers holds significant practical value in addressing these issues.

In the production and processing of sea cucumbers, defects often occur due to their varying sizes and shapes. Issues such as sea cucumbers breaking and stacking on top of each other complicate automated sorting. Traditional methods, including morphological techniques [4], template matching [5], and image processing [6], struggle with accurate identification and sorting, especially under varying lighting conditions. SAM (Segment Anything Models) [7] and Fast SAM [8] segmentation models are typically complex, requiring significant computational resources for training and inference, making them challenging to deploy in resource-constrained environments. To address these challenges, deep learning-based detection algorithms have shown remarkable success in identifying targets in complex settings [9]. Among the various object detection frameworks, the YOLO (You Only Look Once) series stands out for its accuracy and real-time performance, marking significant milestones in the field of object detection [10]. Compared to previous two-stage methods like R-CNN [11], YOLO models have progressively improved in detection precision, inference speed, and ease of deployment, making them well-suited for industrial automation applications. Song et al. [12] improved the detection method using YOLOv5, enhancing network detection efficiency. Qi et al. [13] applied YOLOv5 and PSPNet in identifying lychee picking positions. However, some methods still face issues with lower detection accuracy and slower inference speeds. For instance, Lyu et al. [14] focused on using the YOLOv5-CS model to detect and count green oranges in orchards. While they incorporated the CBAM (Convolutional Block Attention Module), it slightly impacted inference time and memory usage. Meng et al. [15] proposed a method utilizing attention mechanisms and weight fusion strategies to improve feature extraction efficiency. Yin et al. [16] developed an object detection and interpretation model based on gradient-weighted class activation mapping and reinforcement learning. Although this method performed well in remote sensing images, the Grad-CAM technique, which generates class activation maps through gradient information pooling, resulted in reduced spatial localization accuracy, affecting detection outcomes.

To address the high false detection rate, slow processing speed, and susceptibility to light intensity variations in sea cucumber sorting systems, we employ the YOLOv9 [17] model for identification. To enhance the model's expressiveness and computational efficiency while maintaining a lightweight structure, we use the RepViT (Reparameterization Vision Transformer) network as the Backbone. RepViT's reparameterization technology ensures efficient inference while preserving strong representational capabilities during training. Additionally, we introduce auxiliary training heads in the Head network to further improve the model's understanding of target features. The auxiliary training head is designed to help the model better capture and understand the characteristics of sea cucumbers. By providing additional supervision signals, it aims to improve detection accuracy and stability. Finally, we replace the original CIoU (Complete Intersection over Union) loss function with the EIoU (Efficient Intersection over Union) loss function.

[†]The author is with the Faculty Dalian Polytechnic University
[*]E-mail:weihl2005@163.com

EIoU offers a more precise measurement of the differences between predicted and actual bounding boxes, resulting in a more efficient training process and improved overall model performance.

2. An Efficient Sea Cucumber Recognition Model Based on Improved YOLOv9 and RepViT

2.1 Fundamental Principles of the YOLOv9 Network

YOLOv9 continues the core philosophy of the YOLO series by treating object detection as a regression problem, predicting all object locations and categories through a single forward pass. Unlike traditional object detection algorithms, YOLOv9 incorporates the concept of PGI (Programmable Gradient Information), which helps generate reliable gradients via auxiliary reversible branches, addressing the issue of information loss during the feedforward process of deep neural networks. Additionally, YOLOv9 employs a Generalized ELAN (GELAN) structure designed to optimize parameters, reduce computational complexity, and enhance both accuracy and inference speed. By eliminating the need for candidate box generation and using a deeper network structure with richer feature fusion strategies, YOLOv9 significantly reduces computational load and improves detection speed. This allows the model to more accurately identify objects of varying scales and shapes. Therefore, this study chooses YOLOv9 as the foundation for model improvements.

2.2 Improved YOLOv9 Network Structure

The improved YOLOv9 network structure is illustrated in Figure 1. Enhancements to the YOLOv9 model focus on increasing detection accuracy and speed [18]. This includes the introduction of auxiliary training modules to better understand the characteristics of sea cucumbers. The model adopts the lightweight RepViT network as its backbone, balancing computational efficiency with enhanced expressive capability. Additionally, the original CIoU loss function is replaced with EIoU to expedite convergence. This study presents an adaptive refinement of the YOLOv9 model tailored for improved performance in sea cucumber sorting tasks.

2.2.1 Integrating the RepViT lightweight network

RepViT [19] is a lightweight CNN (Convolutional Neural Network) [20] designed for computer vision tasks. Inspired by RepVGG [21], it aims to maintain or enhance model performance while preserving its lightweight nature. The RepViT network structure, illustrated in Figure 2, consists of four stages. Each stage processes images at resolutions denoted as $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, and $\frac{H}{32} \times \frac{W}{32}$, with channel dimensions C_i, batch size B, and image size $H \times W$. The Stem module preprocesses input images. Stages 1 to 4 comprise multiple RepViTBlocks and optionally a RepViTSEBlock module. These include depth-wise separable convolution (3×3 DW), 1×1 convolution, a SE (Squeeze-and-Excitation) module, and a FFN (Feed-Forward Network). Each stage reduces spatial dimensions through downsampling. Additionally, the Pooling module performs global average pooling to further reduce spatial dimensions of feature maps. The FC module consists of fully connected layers for final category predictions.

The design of RepViT draws inspiration from RepVGG, utilizing structural reparameterization techniques to enhance the model's learning during training. It employs a multi-branch structure to boost model expressiveness during training, which is subsequently reparameterized into an equivalent single-branch structure during inference. This reduction in computational complexity enhances efficiency during inference, particularly beneficial for mobile devices by eliminating computation and memory costs associated with skip connections. The Backbone network, crucial for extracting image features, plays a pivotal role due to its streamlined and efficient structure, essential for achieving overall model lightweighting. Therefore, adopting RepViT as the Backbone network effectively reduces model size and computational complexity.

2.2.2 Introducing auxiliary training modules

The concept of auxiliary heads [22] was initially introduced by Rangi Lyu in NanoDet Plus as the AGM (Assign Guidance Module), aiming to address instability issues with auxiliary detection heads during the training of lightweight object detection models. It involves performing loss calculations in intermediate layers of the network to assist the training of auxiliary networks with detection features at different depths. The auxiliary training head provides additional gradient information through auxiliary classification tasks, aiding more effective gradient propagation to the shallower layers of the model. This alleviates gradient vanishing issues, allowing earlier layers to update their weights more effectively. As a result, this method enhances overall model convergence and training efficiency, improves generalization, reduces overfitting, promotes information exchange between different tasks, and ultimately enhances the overall performance of the model. In our study, we introduced an auxiliary training module in the YOLOv9 network, depicted in Figure 3. Here, Lead Head refers to the primary network used during training, while Aux Head denotes an auxiliary training network. The Lead Head refers to the primary network used during training for fine-grained classification, while the Aux Head assists in coarse-grained classification tasks.

2.2.3 Improving the Loss Function for Boundary Boxes

In the YOLOv9 model, CIoU is used as the loss function, computed as shown in equation (1). Here, $p(b, b^{gt})$ represents the distance between the centers of the predicted box b and the ground-truth box b^{gt}. Parameter c denotes the diagonal distance of the minimum enclosing rectangle around the predicted and ground-truth boxes. The term v as-

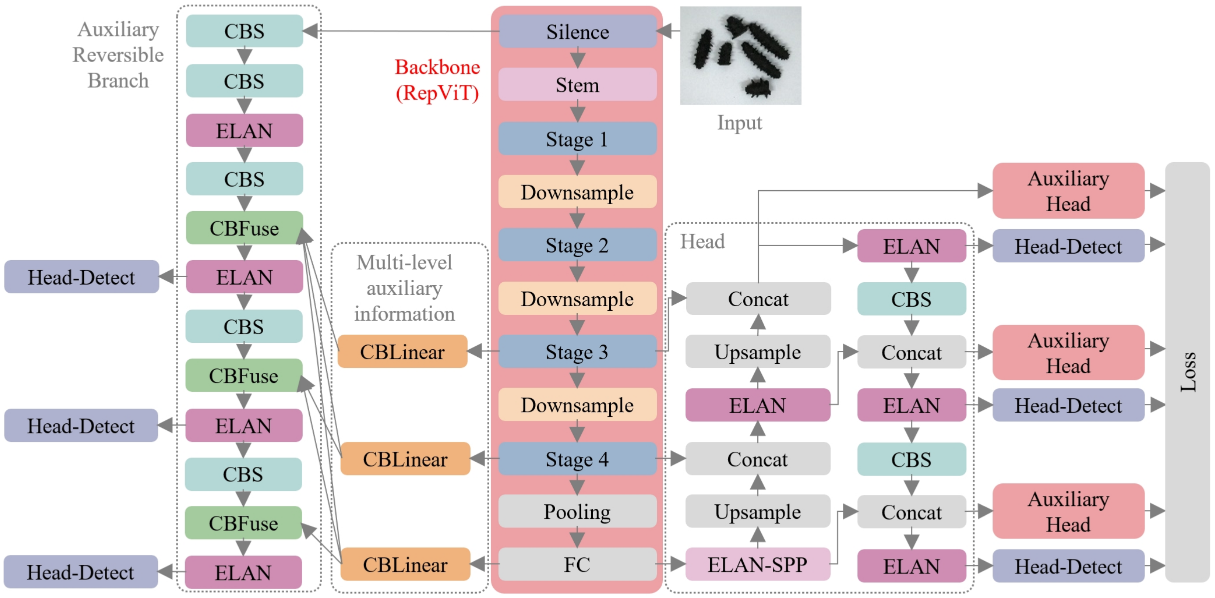


Fig. 1 Improved YOLOv9 Network

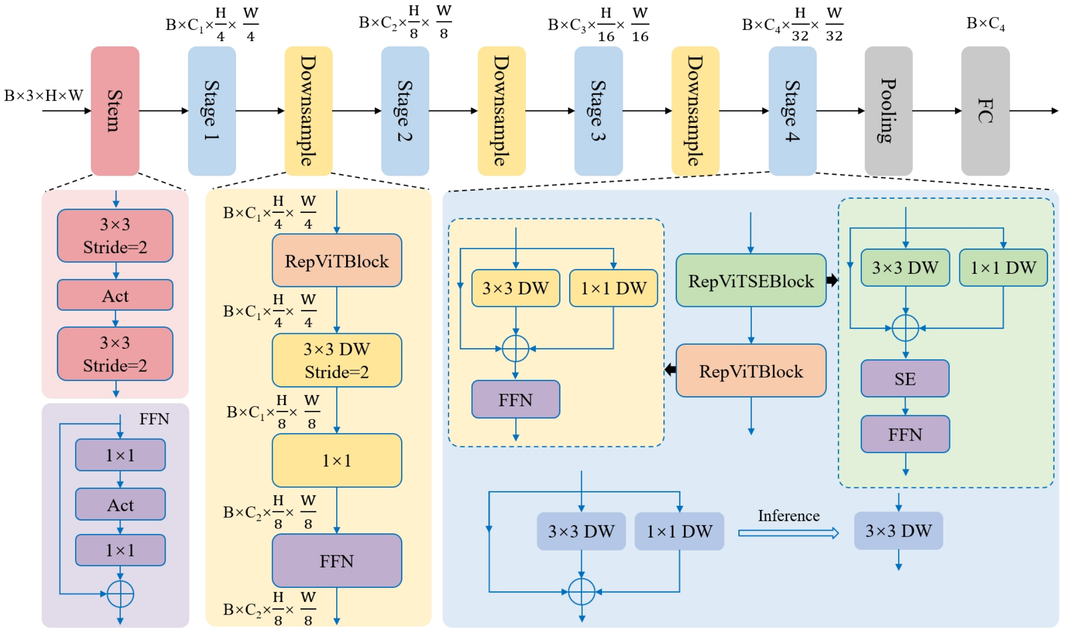


Fig. 2 The architecture of the RepViT network

sesses the consistency of aspect ratios between predicted and ground-truth boxes, with adjustment parameter a to balance this consistency's impact.

The CIoU metric improves training efficiency by calculating the differences between predicted and actual bounding boxes across more dimensions, yielding better results. However, its design concerning aspect ratio weights is inaccurate, neglecting the relationship between actual aspect ratio differences and confidence levels. This oversight hinders effective optimization of model similarity. On the other hand, EIoU [23] builds upon CIoU by separately addressing width and height losses in the original aspect ratio loss. Formula (2) illustrates the computation of EIoU, where w and h denote the width and height of the minimum enclosing rectangle of the predicted and actual boxes, respectively. EIoU resolves the issue in CIoU where bounding boxes may share the same aspect ratio but differ significantly in width and height, thereby enhancing regression accuracy and speeding up model convergence. Therefore, this paper adopts EIoU in place of CIoU for the boundary loss function in YOLOv9.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2}\alpha v \tag{1}$$

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{2}$$

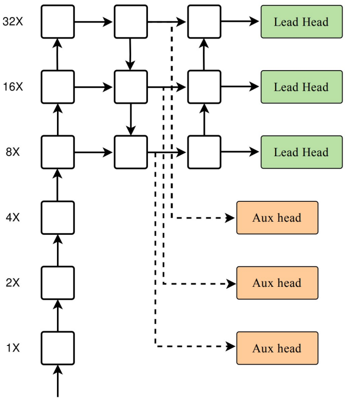


Fig. 3 Auxiliary Training Module

3. Experimental Design and Data Analysis

In this section, we describe the experimental procedure and model parameter configuration, providing a detailed analysis of the improved model's performance on our dataset. The results are compared with other models. Through a series of experiments, it is demonstrated that the efficient sea cucumber recognition model based on the improved YOLOv9 and RepViT significantly enhances detection accuracy and speed, while reducing the original model's size. This validates the model's effectiveness and superiority for factory detection applications.

3.1 Experimental Procedure and Parameter Settings

The experimental procedure is illustrated in Figure 4. The main steps include image acquisition, data augmentation, data annotation and partitioning, model training, and sea cucumber recognition and sorting. The experimental platform for sea cucumber recognition and sorting consists of a vision component, a robotic arm, and a control system. This setup ensures precise image capture and real-time execution of various accurate operations during the experiment. The detailed procedure is as follows:

Part 1: Image Acquisition. The primary aim of this study is to develop a highly specialized model for detecting defects in sea cucumbers within food processing plants. Due to the limited availability of publicly accessible datasets in this domain, we sought to enhance the model's practicality and generalization capabilities by accurately replicating the conditions of sea cucumbers in a factory setting. Recognizing that light intensity affects the sorting of sea cucumbers, we simulated conditions under strong light, normal light, low light, and dim light. This approach improves the model's detection accuracy throughout different times of the day. Sea cucumber samples were placed on the experimental platform shown in Figure 4, and an industrial camera was used to capture 900 images at various time intervals and light intensity levels. The resulting sea cucumber dataset is illustrated in Figure 5.

Part 2: Data Augmentation. To enhance the robustness and generalization ability of our model, we applied various augmentation techniques—cropping, Gaussian noise, and color jittering—to augment our dataset to a total of 1900 images. Cropping involves randomly cutting different regions of the images, enabling the model to adapt better to various perspectives and scales of objects. Gaussian noise introduces random noise into the images, mimicking sensor noise in real-world environments to enhance the model's stability in noisy conditions. Color jittering randomly adjusts brightness, contrast, saturation, and hue of the images, enabling the model to accommodate different lighting conditions and color variations. These augmentation techniques collectively aim to create a more diverse dataset, thereby improving the overall performance of the model.

Part 3: Data Annotation and Division. Data annotation was performed using the LabelImg software, as illustrated in Figure 6. Each sea cucumber without noticeable defects was labeled as "OK", while those showing signs of damage were labeled as "NG". Each annotated sample was enclosed within a bounding box, specifying its position in the image using coordinates. After annotation, all data were divided into training, validation, and test sets in an 8:1:1 ratio [24].

Part 4: Model Training Parameter Settings. Given our study focuses on individual sea cucumbers, we fine-tuned the pretrained YOLOv9 model on a custom dataset to enhance detection accuracy specifically for sea cucumbers. This optimization aims to improve overall model precision and robustness. Table 1 outlines the parameter settings used for model training.

Table 1 Experimental Parameters

Parameter	Configuration
CPU	AMD EPYC 7601
GPU	NVIDIA GeForceRTX3090
CUDA	11.8
Operating System	Windows11
Python	3.9.13
Torch	2.0.1
Momentum	0.937
Weight decay	0.0005
Batch size	64
Learning rate	0.001
Epochs	200
Confidence threshold [25]	0.2
Image size	1890×1417

Part 5: Recognition and Sorting of Sea Cucumbers. Drawing upon the efficient recognition and positioning capabilities of our neural network model described in this paper, precise coordinates of sea cucumbers within the field of view are obtained. This information guides the control system to

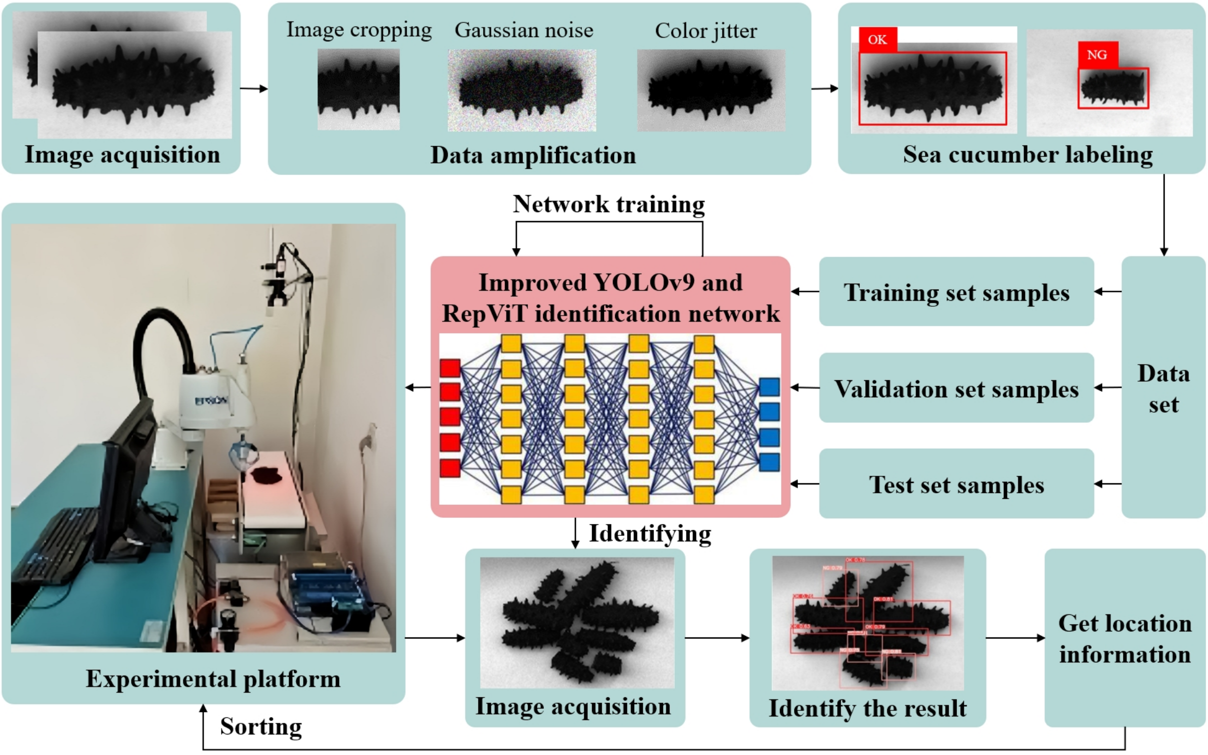
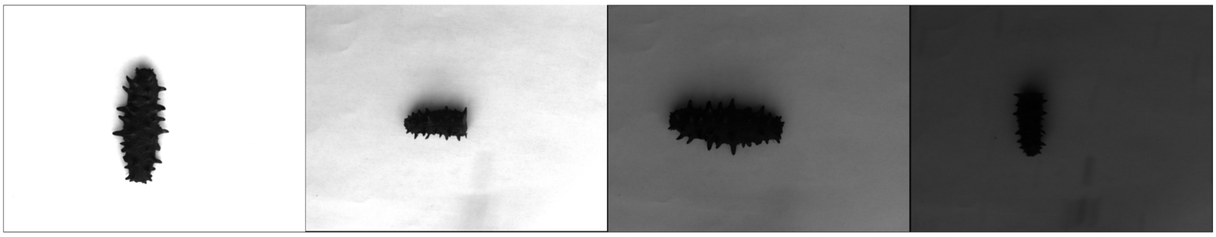
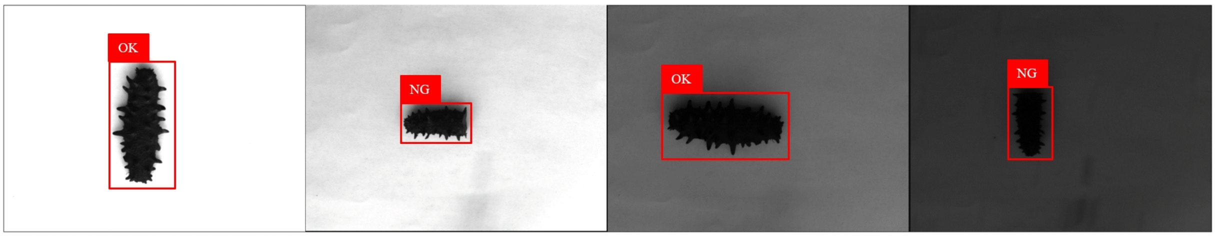


Fig. 4 Experimental Procedure



(a) (b) (c) (d)

Fig. 5 Sea Cucumber Dataset. (a) High-quality sea cucumber under strong light, (b) Low-quality sea cucumber under normal light, (c) High-quality sea cucumber under low light, (d) Low-quality sea cucumber under dim light.



(a) (b) (c) (d)

Fig. 6 Data annotation. Panels (a) and (c) depict high-quality sea cucumbers, while (b) and (d) show low-quality ones.

operate robotic arms with precision for grasping actions, thereby achieving automated sorting of sea cucumbers.

3.2 Methods for Evaluating Algorithm Performance

After training, it is necessary to evaluate the model's ability to accurately detect objects. In this experiment, we assess the performance of the algorithm using three key metrics: $Precision$, $Recall$, mAP (Mean Average Precision) [26], and FPS (Frames Per Second). Here, TP denotes true positives, FP denotes false positives, and FN denotes false negatives. AP_j represents the average precision for defect class j, where j ranges from 1 to n. $FrameNum$ is the number of frames, and $ElapsedTime$ is the sum of image preprocessing time, inference time, and post-processing time. N_{OK} represents the number of correctly sorted sea cucumbers, while N_{ALL} is the total number of sea cucumbers.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$mAP = \frac{\Sigma_{j=1}^{n} AP_j}{n} \tag{5}$$

$$FPS = \frac{FrameNum}{ElapsedTime} \tag{6}$$

$$Rate = (\frac{N_{OK}}{N_{ALL}} \times 100\%) \tag{7}$$

3.3 Ablation Experiments

This study evaluates the impact of introducing three key modules on network performance through ablation experiments: the RepViT network, the auxiliary training module, and the EIoU loss function. As shown in Table 2, these experiments were conducted using the same dataset, training parameters, and equipment. The table reveals that incorporating the RepViT network led to a slight decrease in $Precision$ by 2.74%, $Recall$ by 3.15%, and mAP by 2.21%. However, when both the RepViT network and auxiliary training module were included, $Precision$ increased by 3.42%, $Recall$ by 3.79%, and mAP by 4.90%. The integration of the RepViT network, auxiliary training module, and EIoU function resulted in even more significant improvements, with $Precision$, $Recall$, and mAP increasing by 4.72%, 5.47%, and 5.78%, respectively. These findings highlight the critical role of these three key modules in enhancing the performance of YOLOv9.

3.4 Comparative Experiments

3.4.1 Comparative Experiments with Different Loss Functions

To further confirm whether the EIoU loss function can accelerate model convergence, we compare the performance of GIoU, DIoU, and EIoU. We conducted all experiments using the same dataset and training parameters. The results are shown in Table 3.

The experimental results demonstrate that the EIoU loss function significantly improves $Precision$ and $Recall$. Compared to GIoU, $Precision$ and $Recall$ increased by 3.33% and 3.79%, respectively. Compared to DIoU, the increases were 1.76% and 1.53%. However, in terms of mAP, EIoU showed a decrease of 1.49% compared to DIoU. Despite the YOLOv9 model's overall mAP reduction after introducing EIoU, the improvements in $Precision$ and $Recall$ are notable, validating the effectiveness of EIoU. These experiments support the conclusion that EIoU is an effective choice for enhancing $Precision$ and $Recall$, despite its slight impact on overall mAP.

3.4.2 Comparative Analysis of Different Algorithms

Using the enhanced YOLOv9 network, comparative experiments were conducted with Faster-RCNN, DETR, ViDT, PP-YOLO, YOLOv5, and YOLOv8 under identical datasets and training parameters, as shown in Table 4. From the table, it is evident that the improved YOLOv9 network outperforms other algorithms in both $Precision$ and $Recall$. While its mAP value is slightly lower compared to YOLOv8 (by 1.84%), YOLOv9 excels in detection accuracy. Moreover, the inference speed of the enhanced YOLOv9 network reaches 92.71 frames per second, with a model size of only 42.53MB. This makes it more suitable for real-time detection in factory environments compared to other advanced models. These comparative experiments underscore the effectiveness of the YOLOv9 model enhancement, demonstrating not only improved network performance but also outstanding detection capabilities.

3.5 Recognition and Sorting Experiment

To ensure the practicality of our model, we tested its performance in a real food processing factory. The testing process ran from 8:00 AM to 8:00 PM, with sea cucumber recognition and sorting on the actual production line every two hours. Meanwhile, we recorded the accuracy of each sorting attempt and calculated the average time to sort a single sea cucumber, as shown in Table 5. Throughout the day, the average sorting accuracy for sea cucumbers reached an impressive 97.01%, with each sea cucumber being sorted in as little as 0.92 seconds. The stability of our model ensures the system effectively meets the practical requirements of sorting.

Furthermore, Figure 7 illustrates the detection results of the improved YOLOv9 model compared to its earlier version. Randomly placing sea cucumbers on the experimental platform, we varied the camera aperture to simulate different levels of daylight intensity throughout the day. ”OK” denotes high-quality sea cucumbers, while ”NG” signifies poor-quality ones. By comparing the performance before and after the model improvement, we observed that the enhanced YOLOv9 model effectively identifies both high and low-quality sea cucumbers even under low light conditions.

4. Conclusion

In response to challenges such as high misidentification rates,

Table 2 Ablation Experiments/%

Model	*Precision*	*Recall*	*mAP*@0.5
YOLOv9	93.61	94.20	91.34
YOLOv9+RepViT	90.87	91.05	89.13
(Relative improvement)	(-2.74)	(-3.15)	(-2.21)
YOLOv9+RepViT+Aux head	97.03	97.99	96.24
(Relative improvement)	(+3.42)	(+3.79)	(+4.90)
YOLOv9+RepViT+Aux head+EIoU	**98.33**	**99.67**	**97.12**
(Relative improvement)	**(+4.72)**	**(+5.47)**	**(+5.78)**

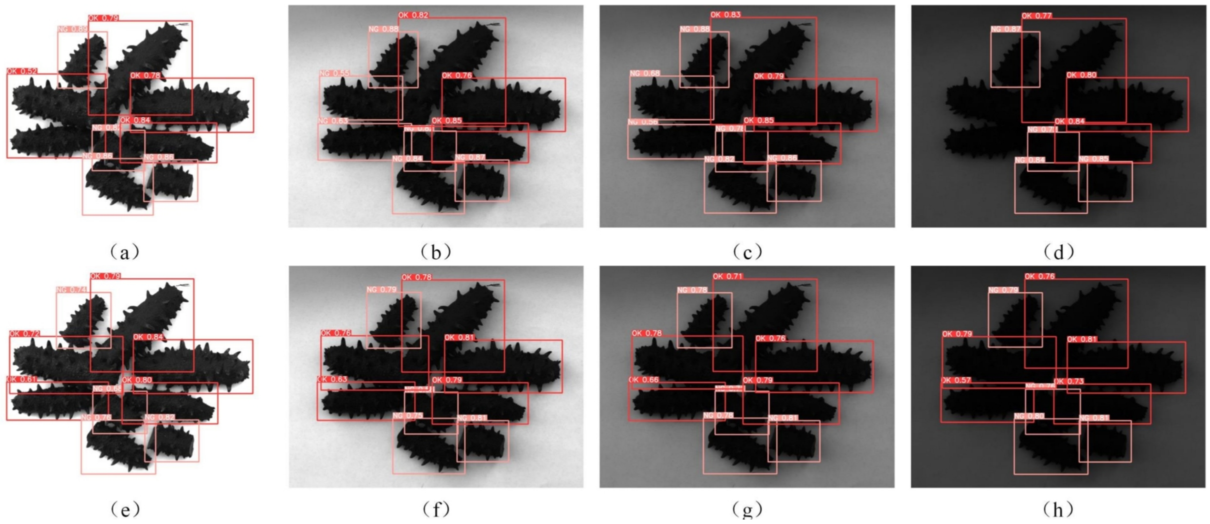


Fig. 7 Evaluating Detection Performance. Panels (a)-(d) depict the detection results using YOLOv9, while panels (e)-(h) show the results after enhancements were applied to YOLOv9.

Table 3 Comparison of Different Loss Functions/%

Model	*Precision*	*Recall*	*mAP*@0.5
YOLOv9	93.61	94.20	91.34
YOLOv9+GIoU	91.68	92.17	89.94
YOLOv9+DIoU	93.25	94.43	**95.04**
YOLOv9+EIoU	**95.01**	**95.96**	93.55

slow processing speeds, and sensitivity to lighting variations in sea cucumber recognition within food processing factories, this paper proposes an efficient sea cucumber recognition and sorting method based on an enhanced YOLOv9 and RepViT framework. By integrating the RepViT network into YOLOv9, introducing auxiliary training heads, and incorporating the EIoU loss function, our approach aims to enhance detection capabilities while keeping the model lightweight. The improved model is poised to significantly boost production efficiency, reduce costs, and ensure product quality, thereby driving the food processing industry towards greater automation and intelligence. Future optimizations in model architecture hold promise for achieving higher accuracy and faster processing speeds. Moreover, adapting the sea cucumber sorting system to diverse real-world scenarios will mitigate misidentification due to varying light intensities and environmental complexity, offering a more efficient and economical solution for the food processing sector.

References

[1] H. Hou, S. Shao, Y. Zhang, H. Kang, C. Qin, X. Sun, and S. Zhang, “Life cycle assessment of sea cucumber production: A case study, China,” J. Clean. Prod., vol.213, pp.158–164, Mar. 2019.

[2] X. Jiang and C. Xue, “The Pretreatment Technology of Raw Sea Cucumber and New Processing Technology of Salted Sea Cucumber,” in Advances in Sea Cucumber Processing Technology and Product Development, C. Xue, ed. Springer International Publishing, Cham, 2023, pp.145–169.

[3] Z. Ren, F. Fang, N. Yan, and Y. Wu, “State of the Art in Defect Detection Based on Machine Vision,” Int. J. Precis. Eng. Manuf.-Green Technol., vol.9, no. 2, pp.661–691, Mar. 2022.

[4] V. Hemamalini, S. Rajarajeswari, S. Nachiyappan, M. Sambath, T. Devi, B. K. Singh, and A. Raghuvanshi, “Food Quality Inspection and Grading Using Efficient Image Segmentation and Machine Learning-Based System,” J. Food Qual., vol.2022, p.e5262294, Feb. 2022.

[5] J. Li, W. Xu, P. Shi, Y. Zhang, and Q. Hu, “LNIFT: Locally Normalized Image for Rotation Invariant Multimodal Feature Matching,” IEEE Trans. Geosci. Remote Sens., vol.60, pp.1–14, 2022.

[6] A. S. Zamani, L. Anand, K. P. Rane, P. Prabhu, A. M. Buttar, H. Pallathadka, A. Raghuvanshi, and B. N. Dugbakie, “Performance of Machine Learning and Image Processing in Plant Leaf Disease Detection,” J. Food Qual., vol.2022, p.e1598796, Apr. 2022.

[7] Alexander Kirillov et al., Segment anything. arXiv preprint, arXiv:2304.02643, 2023.

[8] Xu Zhao et al., Fast Segment Anything. arXiv preprint, arXiv:2306.12156, 2023.

[9] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, “Deep learning-based object detection in augmented reality: A systematic

Table 4 Comparison of Various Algorithms

Model	*Precision*/%	*Recall*/%	*mAP*@0.5/%	*FPS*/(Frames·s-1)	*Size*/MB
Faster-RCNN	55.47	59.28	56.27	19.81	141.47
DETR	91.83	91.34	89.91	33.81	126.07
ViDT	80.11	84.34	79.94	20.80	134.28
PP-YOLO	53.28	56.77	54.96	48.44	174.72
YOLOv5	90.61	92.72	89.34	86.33	66.43
YOLOv8	97.23	98.41	**98.96**	70.42	83.70
Our YOLOv9	**98.33**	**99.67**	97.12	**92.71**	**42.53**

Table 5 Recognition and Sorting

Experiment Number	Time	*Rate*/%	*SingleAverageTime*
1	08:00	98.97	1.01s
2	10:00	99.01	0.92s
3	12:00	98.76	1.03s
4	14:00	96.83	1.13s
5	16:00	96.24	1.21s
6	18:00	94.80	1.04s
7	20:00	94.41	1.14s

review," Comput. Ind., vol.139, p.103661, Aug. 2022.

[10] Z. Zou et al., Object Detection in 20 Years: A Survey, Proc. IEEE, vol. 111, no. 3, pp. 257-276, 2023. doi:10.1109/JPROC.2023.3238524.

[11] D. Wang and D. He, "Fusion of Mask RCNN and attention mechanism for instance segmentation of apples under complex background," Comput. Electron. Agric., vol.196, p.106864, May 2022.

[12] Q. Song, S. Li, Q. Bai, J. Yang, X. Zhang, Z. Li, and Z. Duan, "Object Detection Method for Grasping Robot Based on Improved YOLOv5," Micromachines, vol.12, no. 11, p.1273, Nov. 2021.

[13] X. Qi, J. Dong, Y. Lan, and H. Zhu, "Method for Identifying Litchi Picking Position Based on YOLOv5 and PSPNet," Remote Sens., vol.14, no. 9, p.2004, Jan. 2022.

[14] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, and S. Liu, "Green Citrus Detection and Counting in Orchards Based on YOLOv5-CS and AI Edge System," Sensors, vol.22, no. 2, p.576, Jan. 2022.

[15] X. Meng, X. Wang, S. Yin, and H. Li, "Few-shot image classification algorithm based on attention mechanism and weight fusion," J. Eng. Appl. Sci., vol.70, no. 1, p.14, Mar. 2023.

[16] S. Yin, L. Wang, M. Shafiq, L. Teng, A. A. Laghari, and M. F. Khan, "G2Grad-CAMRL: An Object Detection and Interpretation Model Based on Gradient-Weighted Class Activation Mapping and Reinforcement Learning in Remote Sensing Images," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol.16, pp.3583–3598, 2023.

[17] Wang C Y, Yeh I H, Liao H Y M. YOLOv9: Learning what you want to learn using programmable gradient information. arXiv 2024[J]. arXiv preprint arXiv:2402.13616.

[18] H. Wang, S. Zhang, S. Zhao, Q. Wang, D. Li, and R. Zhao, "Real-time detection and tracking of fish abnormal behavior based on improved YOLOV5 and SiamRPN++," Comput. Electron. Agric., vol.192, p.106512, Jan. 2022.

[19] Wang A, Chen H, Lin Z, et al. Repvit: Revisiting mobile cnn from vit perspective. arXiv 2023[J]. arXiv preprint arXiv:2307.09283.

[20] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobileformer: Bridging mobilenet and transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5270–5279, 2022. 1, 2, 3.

[21] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 13733–13742, 2021. 1, 2, 4, 5.

[22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," https://arxiv.org/abs/2207.02696v1, accessed May 16. 2023.

[23] Z. Yang, X. Wang, and J. Li, "EIoU: An Improved Vehicle Detection Algorithm Based on VehicleNet Neural Network," J. Phys. Conf. Ser., vol.1924, no. 1, p.012001, May 2021.

[24] X. Yu, T. W. Kuan, Y. Zhang, and T. Yan, "YOLOv5 for SDSB Distant Tiny Object Detection," 2022 10th International Conference on Orange Technology (ICOT), Shanghai, China, pp.1–4, Nov. 2022.

[25] S. Kumar and C. Kumar, "Deep Learning based Target detection and Recognition using YOLO V5 algorithms from UAVs surveillance feeds," 2023 International Conference for Advancement in Technology (ICONAT), Goa, India, pp.1–5, Jan. 2023.

[26] Y. Xie, J. Jiang, H. Bao, P. Zhai, Y. Zhao, X. Zhou, and G. Jiang, "Recognition of big mammal species in airborne thermal imaging based on YOLO V5 algorithm," Integr. Zool., vol.18, no. 2, pp.333–352, 2023.

Meng Huang Master degree in Dalian Polytechnic University, main research direction is machine vision and artificial intelligence.

Honglei Wei PhD, Associate professor, research direction is machine vision and mechatronics system design.