Mathematical Biosciences and Engineering

MBE, 21 (3): 3652–3667.
DOI: 10.3934/mbe.2024161
Received: 29 December 2023
Revised: 30 January 2024
Accepted: 02 February 2024
Published: 17 February 2024

http://www.aimspress.com/journal/MBE

Research article

A snail species identification method based on deep learning in food safety

Qiming Li* and Luoying Qiu

Department of Computer Science and Technology, Shanghai Maritime University, Shanghai 201306, China

* **Correspondence:** Email: qmli@shmtu.edu.cn; Tel: +86-021-38282823.

Abstract: In daily life, snail classification is an important mean to ensure food safety and prevent the occurrence of situations that toxic snails are mistakenly consumed. However, the current methods for snail classification are mostly based on manual labor, which is inefficient. Therefore, a snail detection and classification method based on improved YOLOv7 was proposed in this paper. First, in order to reduce the FLOPs of the model, the backbone of the original model was improved. Specifically, the original 3×3 regular convolution was replaced with 3×3 partial convolution, and the Conv2D_BN_SiLU module in the partial convolution was replaced with the Conv2D_BN_FReLU module. FReLU could enhance the model's representational capacity without increasing the number of parameters. Then, based on the specific features of snail images, in order to solve the problems of small and dense targets of diverse shapes, a receptive field enhancement module was added to the head to learn the different receptive fields of the feature maps and enhance the feature pyramid representation. In addition, the CIoU was replaced with the WIoU to make the model pay more attention to targets at the edge or difficult-to-regress accurate bounding boxes. Finally, the images of nine common types of snails were collected, including the *Pomacea canaliculata*, the Viviparidae, the Nassariidae, and so on. These images were then labeled using LabelImg software to create a snail image dataset. Experiments were conducted based on the dataset, and the results showed that the proposed method demonstrated the best performance compared to other state-of-the-art methods.

Keywords: Food safety; snail species identification; YOLOv7; FReLU; PConv; RFEM; WIoU

1. Introduction

Food not only has a profound impact on human health and nutrition but is also closely related to our identity, culture, and other information [1]. As the French gourmet Briat Savarin said, "Tell me what you eat, and I will tell you who you are." Therefore, research related to food [2,3] is always a hot research topic. Researchers from different fields have conducted research related to food from different perspectives, including food choice, food consumption, and food safety. We focus on the branch of food safety and conducts research on the detection and classification of a common food ingredient - snails, to avoid food poisoning incidents caused by accidental consumption.

Snails are a common food delicacy with rich nutrients, delicious taste and chewy texture, and highly popular in cuisine in multiple countries and regions around the world. However, there are many different species of snails, with over 25,000 known species. Most snail species are primarily used for ornamental purposes, and only a few species are edible, such as helix pomatia, garden snail, achatina fulica, white jade snail, and so on. On the other hand, there are also many inedible snail species, for example, nassarius contain tetrodotoxin, which is difficult to destroy at normal cooking temperatures and its minimum lethal dose to humans is approximately 2 mg [4]. The cone snail is a marine gastropod known for its beautiful appearance and dangerous toxicity. The venom of the cone snail is very complex and is generally referred to as conotoxin. According to research, the venom of a cone snail can cause ten deaths [5]. However, because the profits are too great, many illegal merchants use inedible snails as edible snails for sale. Therefore, designing a classification algorithm that can identify different species of snails has great practical significance.

Aimed at helping consumers quickly identify different species of snails and better safeguard their food safety, a multi-class classification method for snails based on YOLOv7 were proposed in this paper. The main improvements and innovations are as follows:

- The ELAN structure in the original backbone network is improved to the PELAN structure. First, the 3×3 convolution in the backbone network is replaced with 3×3 partial convolution (PConv) to reduce the calculation amount and parameter amount of the model. However, after the model is lightweight, it will inevitably lead to a decrease in accuracy; thus, the activation function in PELAN is changed to the funnel activation (FReLU). FReLU can enhance the representation ability of the model without increasing the number of model parameters, making the model lightweight while improving accuracy.
- A receptive field enhancer module (RFEM) is added to the head of the network model to learn the different receptive fields of the feature maps. Based on the different morphological features of the target in snail images, this module enhances the network's ability to extract features from dense and small targets. In other words, it helps the network to better understand the features of different sizes and shapes in the snail images, leading to more accurate identification of different snail species.
- The Weighted Intersection over Union (WIoU) loss function was used to optimize the model because it has a focusing mechanism that can achieve better performance in boundary box regression tasks and provide more accurate evaluation in the presence of imbalanced objects of different sizes.
- The most important of all, a dataset for snail multi-class identification is constructed. Based on common types of snails on the market, as well as some snails that are easily mistaken for others, determine the species of labeled objects in the dataset and annotate them manually. Finally, organize the dataset in the corresponding format for use as input data for the network.

2. Related works

Before the development of computers, product classification mostly relied on manual and traditional mechanical processing methods, which were not only inefficient but also required a large amount of labor. However, using computer vision technology for product identification, positioning, and classification can not only improve production efficiency while ensuring product quality but also reduce the demand for labor. Before the development of deep learning, scholars often used traditional machine learning algorithms for classification. For example, the K-means clustering algorithm [6] is a very common method. It is an unsupervised learning technique used to identify clusters of data objects in a dataset. When implementing K-means clustering, it is important to consider the appropriate number of clusters and the characteristics of the data, as this method may has limitations in high-dimensional spaces and identifying clusters of varying densities. Support Vector Machines (SVM) [7] is another classic approach. It was proposed for binary classification problems and has successfully been applied to pattern recognition problems such as portrait recognition and text classification. Although SVM has achieved great success in solving binary classification problems, it is further required how to generalize SVM to many multi-valued classification problems in practical applications.

With the emergence of deep learning, deep learning-based algorithms have been widely used in product classification. For the study of fruits and vegetables, Mai et al. [8] proposed to add a multi-classifier fusion strategy to the Faster R-CNN network model to detect two small fruit data sets. In addition, the correlation coefficient is used to measure the diversity of classifiers, and a loss function with classifier correlation is introduced. Finally, the model achieved better detection results. However, the detection efficiency was relatively low due to the need for bounding box annotation. Liu et al. [9] conducted detection and recognition of fruit targets in complex backgrounds, using R-FCN (Region-based Fully Convolutional Networks) to improve the detection speed by reducing the time for extracting candidate regions, but there is room for improvement in detection accuracy. However, the above research only focuses on spherical objects, and there is room for further research on the detection and classification of non-spherical objects.

In the study of seafood, Villon et al. [10] conducted detection on underwater fish, using a CNN network for detection, with a detection rate of 94.6%, which was higher than the manual detection rate of 89.3%. Feng et al. [11] conducted classification research on shellfish and used an improved Faster R-CNN to recognize different shellfish under different scenes, increasing the recognition accuracy by nearly 4% compared to the original model. It can be seen that deep learning-based object detection algorithms have achieved ideal results in the field of seafood detection and classification. Wang et al. [12] used faster R-CNN for real-time detection of snails, but their goal was to analyze snail behavior in order to develop more effective pest control strategies. Borreta et al. [13] developed a Tiny-YOLOv4 snail recognition system using Raspberry Pi. Also, the system is used for agriculture and gardening, it refers only to four types of snails, and the dataset comprised only 200 images, making it prone to overfitting and low accuracy.

In a study using the YOLO algorithm for classification, Agorku et al. [14] used YOLO, SSD and EffientDet models to classify the presence of vessels and/or barges. Among them, the F1 score of YOLOv8 was 96%, which was higher than the other models. For public transportation agencies, Remote transportation planning efforts are also valuable. Cap et al. [15] proposed that in terms of plant disease diagnosis, YOLO-based systems generally provide advantages over classification-based systems, such as the ability to detect disease locations and superior classification performance. Haque et al. [16] proposed a rice leaf disease classification and detection method based on YOLOv5 deep learning to help farmers identify rice leaf diseases and improve the quality and quantity of crops. Pundhir et al. [17] used YOLOv3 to classify smokers and non-smokers in various environments and achieved a classification accuracy of 96.74%. Liu et al. [18] used YOLO instead of Mask-RCNN to detect whether everyone wears a mask in public places, making the efficiency and detection accuracy better than manual classification and improving classification performance. The above literature shows that the YOLO algorithm can also have good results in classification tasks.

3. Method

In order to solve the problem of snail classification and effectively identify the categories of poisonous snails, in this paper, an improved method is proposed based on YOLOv7 [19] network structure. The YOLOv7 algorithm is not specialized in detecting dense small targets as it is a unified detector that detects the entire image at once. This can result in smaller targets being ignored or falsely detected due to their low pixel count. Additionally, in order to improve the robustness of the algorithm, the snails in the images of the dataset constructed in this paper exhibit varying object states and significant size differences, making it challenging for traditional methods to accurately locate them. Therefore, three improvements to the overall network structure of YOLOv7 are proposed. The improved network structure is shown in Figure 1.

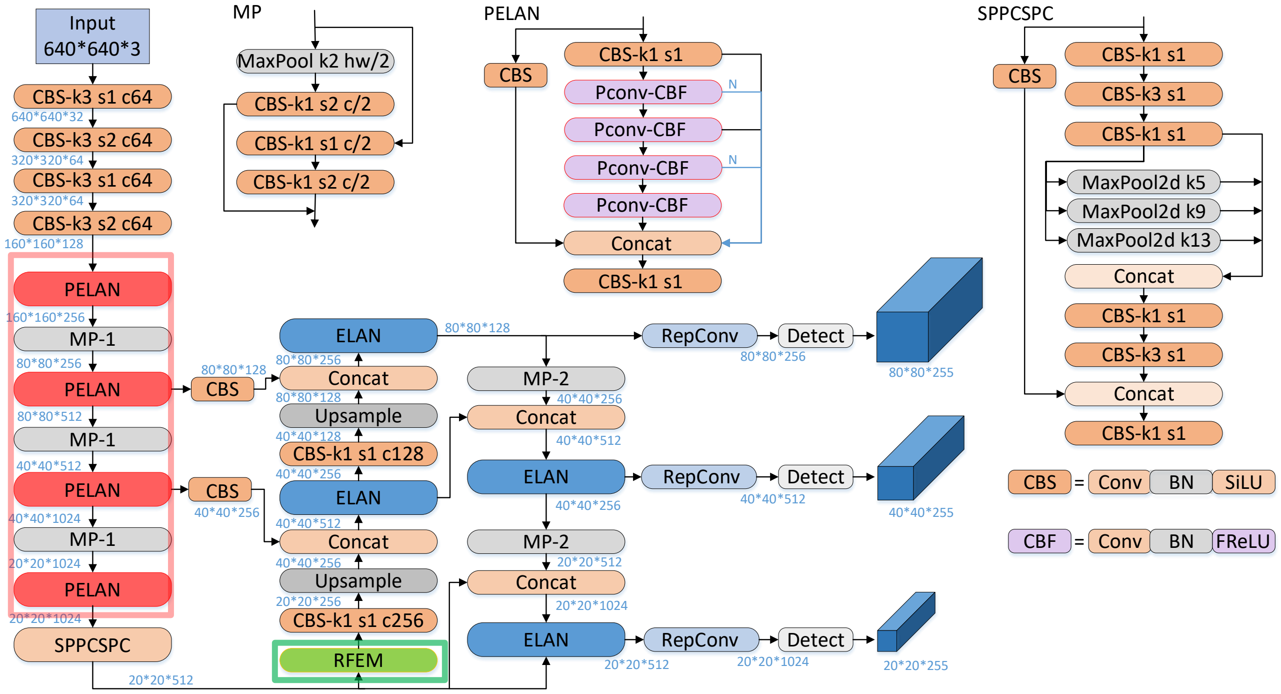

Figure 1. Our improvements to the overall structure of YOLOv7 and detailed structure of each module. The red box labeled PELAN is transformed from the ELAN structure and replaces the original CBS-k3-s1 with the PConv-CBF module. The green box labeled with RFEM is the added receptive field enhancer.

3.1. Improved Partial Convolution

PConv [20] is a technique used in convolutional neural networks (CNNs) for handling image boundaries and missing information. A filter is used to detect specific features in an image, where the filter is a matrix of size m×n, and different filters have different parameters. However, in traditional convolutional layers, the filter is applied to the entire input image, including the boundary pixels. This can lead to the loss of important information at the boundaries, thereby affecting the overall performance of the network. Therefore, in PConv, the filter is applied only to the valid pixels of the input image. This means that the filter does not extend beyond the borders of the input image and only operates on pixels that have valid information. The output of the PConv is then normalized by the number of valid pixels in the output. Furthermore, a re-evaluation of the computation speed of Depthwise Convolution (DWConv) [21] revealed that the main reason for the low FLOPs issue is frequent memory access. In contrast, PConv not only reduces computational redundancy but also decreases the number of memory accesses. It exploits redundancy in feature maps and systematically applies regular Conv only on a subset of input channels, without affecting the remaining channels. In fact, PConv has lower FLOPs compared to regular Conv. For an input $I \in R\hat{}(c\times h\times w)$, the FLOPs of regular Conv are

$$h \times w \times k^2 \times c^2 \tag{1}$$

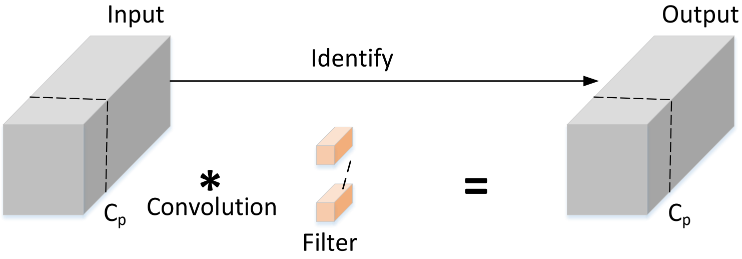

Figure 2. PConv removes redundant channels and processes only a few input channels to achieve speed and efficiency.

Figure 2 illustrates the working principle of PConv. It applies regular convolution for spatial feature extraction only to a portion of the input channels, while leaving the rest of the channels untouched. For continuous or conventional memory access, the first or last consecutive c_p channels are treated as representatives of the entire feature map for computation. In general, the default number of input and output feature map channels is the same. Thus, the FLOPs of PConv are only

$$h \times w \times k^2 \times {c_p}^2 \tag{2}$$

In the typical case of partial ratio r = c_p/c = 1/4, the FLOPs of PConv are only 1/16 of regular Conv. Additionally, PConv has a smaller memory access compared to regular convolution, where the memory access of regular Conv is

$$h \times w \times 2c + k^2 \times c^2 \approx h \times w \times 2c \tag{3}$$

and the memory access of PConv is

$$h \times w \times 2c_p + k^2 \times {c_p}^2 \approx h \times w \times 2c_p \tag{4}$$

which is only 1/4 of the memory access amount of regular convolution.

Furthermore, in PConv, since it utilizes regular convolution for spatial feature extraction, the activation function used in regular convolution is the the Sigmoid Linear Unit (SiLU) activation function, which is described in detail in reference [22]. Its specific expression is Eq.5.

$$f(x) = x \times sigmoid(x) \tag{5}$$

SiLU is a non-linear function used in neural networks that maps a real number to the interval (0,1). The SiLU function has been widely used in the field of deep learning to improve the accuracy and convergence speed of neural networks. However, compared to SiLU, FReLU has the better properties, which can enhance the network's response to small targets and provide some regularization to improve detection performance. In addition, FReLU has fewer parameters and better computational efficiency, which is very useful in situations where deep learning computational resources are limited. Therefore, in this paper, SiLU is replaced with FReLU, as studied in reference [23], and modify the original Conv2D_BN_SiLU module to the Conv2D_BN_FReLU module.

FReLU solves the problem of activation functions not depending on spatial conditions in previous versions by adding a spatial condition. It extends the ReLU (the expression of which is Eq.6) studied in reference [24], the PReLU mentioned in reference [25] to a visually parameterized ReLU with pixel-level modeling capabilities, improving the spatial sensitivity of the activation function. This improvement adds a negligible computational cost and is relatively easy to implement.

$$f(x) = max(x, 0) \tag{6}$$

$$f(x) = max[x, T(x)] \tag{7}$$

In Eq.7, $f(x)$ is the non-linear activation function and x is the independent variable. $T(x)$ represents a simple and efficient spatial context feature extractor. It uses a parameterized pooling window to enhance spatial attention, and its specific expression is Eq.8.

$$T(x) = x_{c,i,j}^{w} P_{c}^{w} \tag{8}$$

In Eq.8, $x_{c,i,j}^{w}$ represents a point on the two-dimensional space (i, j) at the center of the pixel in the c-th channel, and the parameterized pooling window takes the non-linear activation function on the c-th channel as the input pixel. P_{c}^{w} represents the coefficients shared by the pixels in the same channel on this parameter window.

Finally, the ablation experiment results show that the improvements for PConv are effective for most small and dense objects. It not only reduces computational complexity and decreases the model's size and memory access, but also effectively improves the model's mAP value when the FReLU activation function is applied.

3.2. Receptive Field Enhancer Module

Receptive Field Enhancer (RFE) [26] is a feature enhancement technique used in object detection. It can increase the size and number of receptive fields, thereby improving the performance of object detectors. Most object detectors shrink the input image to a smaller size and perform detection on these sizes. However, this may cause the detector to miss some small targets, and the role of RFE is to enable the detector to detect smaller targets without using higher resolution images. It enlarges the receptive field by convolving the feature map, allowing the detector to better detect targets of different sizes. The role of RFE is to optimize the performance of object detectors in order to improve the precision and recall of object detection.

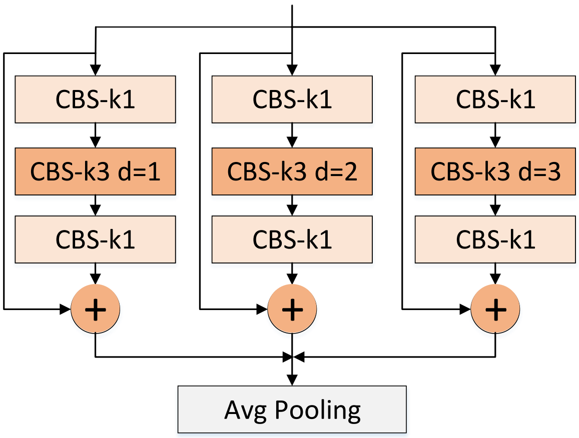

Figure 3. The detailed structure of RFE, which consists of 1×1 convolution layers, 3×3 convolution layers with different dilation rates, and average pooling layers.

Different sizes of receptive fields mean different abilities to capture long-distance dependencies, so the RFE module fully utilizes the advantages of receptive fields in the feature map using dilated convolutions [27]. As shown in Figure 3, the RFE module used in this paper can be divided into two parts: Multi branches based on dilated convolutions and the gathering and weighting layer. The dilation rates of the dilated convolutions in the multi-branch part are 1, 2, and 3, all using a fixed 3×3 convolution kernel size.

When faced with similar and hard-to-distinguish objects, such as the Viviparidae and the Pomacea canaliculata in the dataset used in this paper, their features are similar and difficult to differentiate. Additionally, for small targets that appear densely in images, the preliminary improved algorithm is prone to false negatives or false positives during the detection process. However, after adding the RFE module to the preliminary improved model, it can more effectively handle snail images with different shapes, categories, and angles, and better capture the contextual information that is useful for object detection and target recognition.

In addition, to prevent the problems of gradient explosion and vanishing during training, residual connections were added. To ensure the effectiveness of this module for the model in this paper, the RFE module was added to the head of YOLOv7, in order to increase the receptive field of the feature map and improve the precision of the improved model for multi-scale object detection and recognition.

3.3. Weighted Intersection over Union

The loss function used by the YOLOv7 model is CIoU [28], which calculates the overlap area, center point distance, and aspect ratio (width-to-height ratio) of the predicted bounding box as three important factors. It is based on DIoU [29] and further improves the consistency of aspect ratios.

The introduction of WIoU [30] involves weighting the IoU by considering the area between the predicted and ground truth boxes, which addresses the potential bias issue in evaluating results using traditional IoU. In WIoU, the weight value for each object box depends on the degree of overlap with the ground truth box. The greater the overlap, the higher the weight of the object box, and vice versa. The function expression of WIoU is Eq.9, and the following is the calculation method:

(1) Calculate the IoU score between the predicted and ground truth boxes.

(2) Calculate the area between two boxes: Calculate the distance between the center points of the predicted and ground truth boxes and use this distance as the maximum distance between the two boxes to calculate the area between them.

(3) Based on the area between the two boxes, calculate the weight coefficient. This coefficient measures the relationship between the two boxes and can be used to weight the IoU score.

(4) By introducing the area and weight coefficient between the boxes, WIoU can more accurately evaluate the object detection results and avoid the bias problem of traditional IoU.

$$WIoU = \frac{\sum_{i=1}^{n} \omega_i IoU(b_i, g_i)}{\sum_{i=1}^{n} \omega_i} \tag{9}$$

$$\omega_i = \frac{v}{\left(1 - IoU(b_i, g_i)\right) + v} \tag{10}$$

$$v = \frac{4}{\pi^2}\left(tan^{-1}\frac{w}{h} + tan^{-1}\frac{w^{gt}}{h^{gt}}\right)^2 \tag{11}$$

Here, n represents the number of object boxes, b_i represents the coordinates of the i-th object box, g_i represents the coordinates of the ground truth box for the i-th object, $IoU(b_i, g_i)$ represents the IoU value between the i-th object box and the ground truth box, and ω_i represents the weight value, v is a parameter used to measure aspect ratio consistency.

Snails often exhibit different states (e.g., extending their heads while moving or retracting their heads and tails into their shells when startled). When faced with targets that undergo morphological transformations like these situations, different species of snails vary greatly in size. Therefore, compared to CIoU, WIoU can better evaluate detection results because it has a better focus mechanism, can better handle background noise and larger targets, and thus achieves better performance in bounding box regression tasks. It can also provide more accurate evaluations in the presence of imbalanced sizes of objects.

4. Experiment and analysis

4.1. Dataset

The image collection was carried out through on-site shooting and web crawling methods. To ensure the classification effect, the images of 9 different species of snails are collected. Part of them are five common edible snails in daily life, the other part are the snails more commonly seen in social news that are easy to be eaten by mistake, and there is another snail which is inedible and only for ornamental purposes.

(1) Data collection. The experimental dataset consists of two parts: web data and on-site data. The web data consists of images of 9 different kinds of snails obtained from the Internet using a web crawler tool, while the on-site data is obtained by cameras or mobile phones at places such as fruit markets and supermarkets.

(2) Data annotation. The image data was annotated using the LabelImg software and saved in XML format in PascalVOC format, with the file name matching the image name. Figure 4 provides some examples of dataset annotation.

(3) Data segmentation. Through the above two steps, 5550 images were obtained, which were

divided into training, validation, and testing subsets in an 8:1:1 ratio. Table 1 shows the statistical information of the dataset, including the number of image instances in the training, validation, and testing set splits.

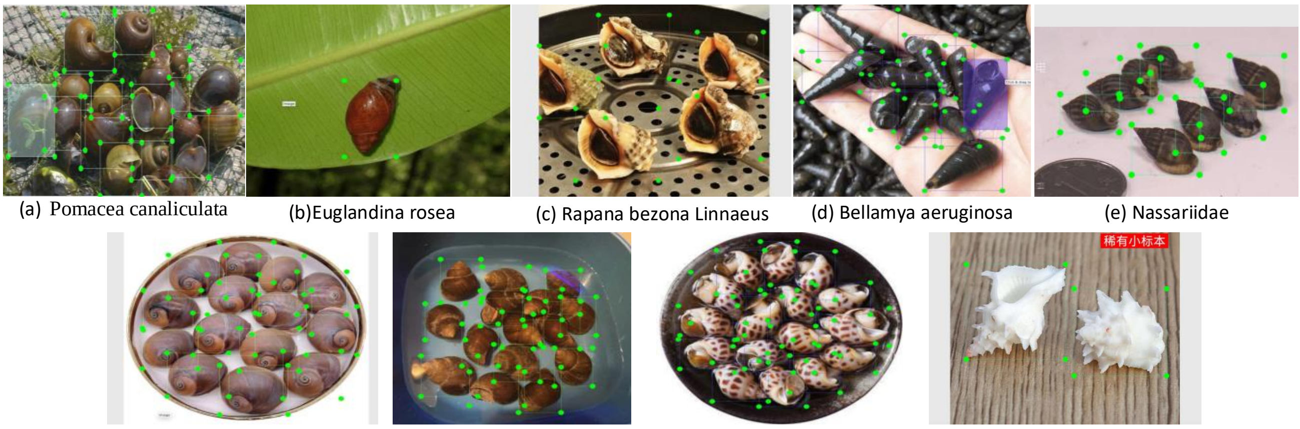

Figure 4. Example of dataset annotation. (a) *Pomacea canaliculata*, (b) *Euglandina rosea*, (c) *Rapana bezona* Linnaeus, (d) *Bellamya aeruginosa*, (e) Nassariidae, (f) *Turbo petholatus* L, (g) Viviparidae, (h) *Babylonia lutosa*, and (i) *Babelomurex kawanishii*.

Table 1. The details of dataset.

category	Pomacea canaliculata	Euglandina rosea	Rapana bezona Linnaeus	Bellamya aeruginosa	Nassariidae	Turbo petholatus L	Viviparidae	Babylonia lutosa	Babelomurex kawanishii
train	611	440	436	486	627	628	490	599	311
valid	66	49	47	32	78	78	45	80	19
test	76	25	17	19	65	64	56	73	8
sum	753	514	500	537	770	770	591	752	404

4.2. Experimental platform

The official YOLOv7 code was used to implement the experiments. All models were trained for 100 iterations with a batch size of 8. At each iteration, the validation split was evaluated. For each model, the training checkpoint that obtained the best mAP at an IOU threshold of 0.65 was selected for evaluating the test split. All experiments were run on a NVIDIA GeForce RTX 3090 GPU.

4.3. Evaluation metric

To evaluate the performance of the snail classification detection models, we use mean average precision (mAP), precision, recall, parameters, FLOPs and F1 score in reference [31].

Precision refers to the proportion of predicted positives that are true positives, while recall refers to the proportion of true positives that are correctly predicted. The formulas are Eq.12 and Eq.13.

$$P = TP/(TP + FP) \tag{12}$$

$$R = TP/(TP + FN) \tag{13}$$

where TP represents the number of true positive samples identified, FP represents the number of false positive samples identified, and FN represents the number of positive samples that were not identified.

AP represents the average precision for each class. mAP is the mean average precision for all classes, calculated using the Eq.14 and Eq.15

$$AP = \int_0^1 P(R)\, dR \tag{14}$$

$$mAP = \sum\nolimits_{i=1}^{N} AP_i/N \tag{15}$$

where R means the recall value, $P(R)$ is the precision when the recall value is R, and N represents the number of detected target categories.

The F1 score is a comprehensive indicator used to evaluate the performance of a classification model. It is the harmonic average of precision and recall. Specifically, the F1 score can be calculated by Eq.16.

$$F1 = 2 \cdot \mathrm{P}R/(P + R) \tag{16}$$

where P stands for precision and R stands for recall.

Parameter volume and computation volume (usually referred to as FLOPs) are two metrics used to evaluate model size and computational complexity, and their size is related to aspects such as model accuracy, runtime, and memory usage.

4.4. Ablation experiments.

To verify the effectiveness of each improvement module in the algorithm proposed in this paper, four ablation experiments were conducted under the same experimental conditions, and the experimental results are shown in the Table 2.

Table 2. Results of Ablation Experiment.

Model	PConv	+FReLU	RFE	WIoU	Params	GFLOPs	P	R	mAP	mAP@0.5:0.95	F1
YOLOv7	×	×	×	×	37.23M	105.3	0.843	0.852	0.901	0.661	0.847
model 1	√	×	×	×	**32.73M**	**85.0**	0.826	0.847	0.888	0.643	0.836
model 2	√	√	×	×	32.75M	85.2	0.878	0.874	0.915	0.695	0.875
model 3	√	√	√	×	34.46M	86.5	0.874	0.897	0.926	0.704	0.885
model 4	√	√	√	√	34.46M	86.5	**0.884**	**0.911**	**0.932**	**0.716**	**0.897**

As can be seen from Table 2, compared with the original model, although the model added with PConv has decreased in parameter amount and calculation amount, the decline in precision is also serious. However, the improved PConv using FReLU improved the network's expression ability and performance. Ignoring the number of parameters and calculations, the improved PConv increased the mAP of the model by 2.7%. Compared with the original model, the mAP increased by 1.4%, and F1 score increased by 2.8%, which shows that the improvements to the PConv module are effective. After adding REFM, the calculation and parameter quantity of the model increased slightly, but it was also within the acceptable range. Moreover, due to the expanded receptive field and the optimization of the performance of the object detector, the mAP and F1 score increased by 1.1% and

1%, respectively. After the final modification of IoU, the model's performance was further improved due to the focus mechanism of WIoU. The mAP increased by 0.6% and the F1 score increased by 1.2%. In the end, the mAP of the improved model reached 93.2%, which is 3.1% higher than the original model, and the F1 score reached 89.7%, which is 5% higher than the original model.

Table 3. The precisions of each species in ablation experiments.

Model	Pomacea canaliculata	Euglandina rosea	Rapana bezona Linnaeus	Bellamya aeruginosa	Nassariidae	Turbo petholatus L	Viviparidae	Babylonia lutosa	Babelomurex kawanishii
YOLOv7	0.907	0.81	0.815	**0.981**	0.937	0.946	0.846	0.887	0.984
model 1	0.899	0.867	0.708	0.971	0.934	0.922	0.841	0.873	0.980
model 2	0.927	0.746	0.885	0.971	0.967	0.942	0.879	**0.927**	**0.989**
model 3	0.913	0.854	**0.923**	0.970	**0.971**	**0.947**	0.873	0.907	0.988
model 4	**0.941**	**0.944**	0.870	0.971	0.966	0.938	**0.882**	0.895	0.987

As can be seen from Table 3, the identification precisions of the 9 species of snails have basically improved to varying degrees. After adding all three improvements to the original model, the identification precisions of *Pomacea canaliculata*, *Euglandina rosea*, *Rapana bezona* Linnaeus, Nassariidae, Viviparidae, *Babylonia lutosa* and *Babelomurex kawanishii* have been improved to varying degrees; compared with the original model, they have increased by 3.4%, 13.4%, 5.5%, 2.9%, 3.6%, 0.8%, and 0.3%, respectively. However, the precisions of Bellamya aeruginosa and Turbo petholatus L has decreased by 1% and 0.8% respectively, which is within the acceptable range. The improved model makes the identification precision of each species more balanced. The above experimental data shows the effectiveness of the improved model proposed in this article.

4.5. Comparative experiments

In order to further demonstrate the comprehensive performance of the algorithm proposed in this paper, we conducted comparative experiments with other algorithms on the dataset constructed in this paper, and the results are shown in Table 4. As can be seen from Table 4, the performance of the improved model in this article is better than other models. Although the parameter amount and calculation amount of the improved model are larger than other YOLO series algorithms, it has a greater advantage, the mAP is 4.7%, 1.7%, 3.1%, 1.3% and 4.8% higher than YOLOv5s, YOLOv6, YOLOv7, YOLOv8 and YOLOX algorithms, respectively, and the F1 score increased by 7.4%, 2.1%, 5%, 1.6% and 7.9%, respectively. This is because the model proposed in this article introduces more convolutional layers and adds a receptive field enhancer, which expands the receptive field, thus improving the perceptual and expressive capabilities of the model. Compared with Faster R-CNN, the Faster R-CNN model can better capture the details and contextual information of the target using a region proposal network (RPN) to generate candidate boxes. This makes Faster R-CNN more accurate when processing small targets or images with complex backgrounds. These characteristics all exist in the images in the dataset constructed in this article, so it has better performance than the original YOLOv7, but it is not as good as the improved algorithm in this article. Compared with the traditional SSD algorithm, the YOLO series of algorithms not only greatly reduces the calculation and the number of parameters, but also greatly improves the mAP and F1 score. This is because the

YOLO series algorithms can handle occlusion, scale change, different viewing angles, and other complex scenarios. By performing global perception on the entire image, YOLO can better capture the contextual information of the target, thereby improving the precision and robustness. In general, the algorithm in this paper achieves the best results in mAP and F1 score, whether compared with the YOLO series algorithms or compared with the traditional two-stage models.

Table 4. Results of Comparative Experiment.

Model	Backbone	Params	GFLOPs	mAP	F1
YOLOv5s	CSPDark-Net53	7.1M	16.5	0.885	0.823
YOLOv6[32]	EfficientRep	18.5M	45.3	0.915	0.876
YOLOv8	CSPDark-Net53	**3M**	**8.1**	0.919	0.881
YOLOX[33]	CSPDark-Net53	8.94M	26.78	0.884	0.818
Faster R-CNN[34]	VGG16	136.8M	402.0	0.911	0.874
SSD[35]	VGG16	24.2M	274.8	0.851	0.797
RetinaNet[36]	ResNet50	-	-	0.891	0.832
YOLOv7	-	37.24M	105.3	0.901	0.847
YOLOv7-Tiny	-	6.03M	13.2	0.852	0.803
Ours	-	34.46M	86.5	**0.932**	**0.897**

Table 5. The precisions of each species in comparative experiments.

Model	Pomacea canaliculata	Euglandina rosea	Rapana bezona Linnaeus	Bellamya aeruginosa	Nassariidae	Turbo petholatus L	Viviparidae	Babylonia lutosa	Babelomurex kawanishii
YOLOv5s	0.926	0.725	0.731	0.968	0.948	0.934	0.846	0.898	0.988
YOLOv6	0.904	0.897	0.866	0.972	0.944	0.926	0.844	0.900	0.982
YOLOv8	0.893	0.908	**0.876**	0.976	0.946	0.924	0.852	0.901	**0.994**
YOLOX	0.917	0.74	0.732	0.968	0.948	0.930	0.849	0.888	0.984
Faster R-CNN	0.887	0.858	0.799	0.962	0.931	0.927	0.851	**0.909**	0.985
SSD	0.902	0.657	0.689	0.941	0.915	0.898	0.822	0.862	0.973
RetinaNet	0.903	0.788	0.803	0.972	0.918	0.934	0.837	0.888	0.976
YOLOv7	0.907	0.81	0.815	**0.981**	0.937	**0.946**	0.846	0.887	0.984
YOLOv7-Tiny	0.892	0.626	0.698	0.948	0.911	0.934	0.815	0.860	0.982
Ours	**0.941**	**0.944**	0.870	0.971	**0.966**	0.938	**0.882**	0.895	0.987

As can be seen from Table 5, in terms of each species, the average precisions of *Rapana bezona* Linnaeus, Viviparidae, and *Babylonia lutosa* are slightly inferior to other species. The improved network model in this paper has the best precision in the identification of *Pomacea canaliculata*, *Euglandina rosea*, Nassariidae and Viviparidae species. Among them, *Pomacea canaliculata*, *Euglandina rosea*, and Nassariidae are the most common species of inedible snails that can cause poisoning, which is also in line with the central theme of food safety discussed in this paper. The identification precision of the remaining inedible *Babelomurex kawanishii* is only 0.7% less than the best data obtained by YOLOv8. Although the average precisions of other species have not reached the best, they are within an acceptable range. Overall, the improved model in this article is better than other mainstream models in snail species identification on the dataset constructed in this article.

5. Conclusions

In this paper, food safety is taken as the background, and a series of studies on snail species identification are conducted in response to the social incidents of poisoning caused by accidentally eating poisonous snails in daily life. However, there are challenges such as dense small targets, multi-scale targets, and category imbalance in the snail species identification task based on computer vision. Therefore, an improved model based on YOLOv7 for snail species identification is proposed in this paper. First, since there is no public data set that can be used directly, the images of 9 different kinds of snails are collected for manual annotation and a dedicated data set for snail species identification is constructed. Second, in order to improve algorithm performance without increasing the number of calculations and parameters as much as possible, the backbone network of original model is replaced with an improved PConv. In addition, due to the characteristics of small and dense targets in the snail images, a receptive field enhancer is added, and the loss function is replaced. Experimental results show that the model proposed in this article has better precision and practicability than other mainstream models. In subsequent research, we will continue to optimize the model so that it can be deployed on edge devices with limited computing and storage resources without reducing the performance.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in creating this article.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. W. Min, L. Liu, Y. Liu, M. Luo, S. Jiang, A survey on food image recognition, *Chin. J. Comput.*, **45** (2022), 542–566. https://doi.org/10.11897/SP.J.1016.2022.00542
2. S. Sajadmanesh, S. Jafarzadeh, S. A. Ossia, H. R. Rabiee, H. Haddadi, Y. Mejova, et al., Kissing cuisines: Exploring worldwide culinary habits on the web, in *Proceedings of the 26th international conference on world wide web companion*. (2017), 1013–1021. https://doi.org/10.1145/3041021.3055137
3. J. Chung, J. Chung, W. Oh, Y. Yoo, W. G. Lee, H. Bang, A glasses-type wearable device for monitoring the patterns of food intake and facial activity, *Sci. Rep.*, **7** (2017), 41690. https://doi.org/10.1038/srep41690
4. J. Yao, W. Jin, C. Gong, B. Jia, S. Sui, Y. Liang, et al., Distribution characteristics of tetrodotoxin in the coastal waters of China, *Mar. Environ. Sci.*, **40** (2021), 161–166. https://doi.org/10.12111/j.mes.20190299

5. Q. Chen, C. Peng, F. Zhao, J. Li, B. Gao, Study on genetic diversity of cone snail in south china sea based on COI sequences, *Genomics Appl. Biol.*, **39** (2020), 3955–3960. https://doi.org/10.13417/j.gab.039.003955
6. J. MacQueen. Some methods for classification and analysis of multivariate observations, in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, **1** (1967), 281–297.
7. C. Cortes, V. Vapnik. Support-vector networks, *Mach. Learn.*, **20** (1995), 273–297. https://doi.org/10.1023/A:1022627411411
8. X. Mai, H. Zhang, X. Jia, M. Q. H. Meng, Faster R-CNN with classifier fusion for automatic detection of small fruits, *IEEE Trans. Autom. Sci. Eng.*, **17** (2020), 1555–1569. https://doi.org/10.1109/TASE.2020.2964289
9. J. Liu, M. Zhao, X.Guo, A fruit detection algorithm based on r-fcn in natural scene, in *2020 Chinese Control and Decision Conference (CCDC)*, (2020), 487–492. https://doi.org/10.1109/CCDC49329.2020.9163826
10. S. Villon, D. Mouillot, M. Chaumont, E. S. Darling, S. Villéger, A deep learning method for accurate and fast identification of coral reef fishes in underwater images, *Ecolog. Inform.*, **48** (2018), 238–244. https://doi.org/10.7287/peerj.preprints.26818
11. Y. Feng, X. Tao, E. J. Lee, Classification of shellfish recognition based on improved faster r-cnn framework of deep learning, *Math. Probl. Eng.*, **2021** (2021), 1–10. https://doi.org/10.1155/2021/1966848
12. Z. Wang, I. Lee, Y. Tie, J. Cai, L. Qi, Real-world field snail detection and tracking, in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, (2018), 1703–1708. https://doi.org/10.1109/ICARCV.2018.8581271
13. J. R. I. Borreta, J. A. Bautista, A. N. Yumang, Snail Recognition Using YOLO, in *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, (2022), 1–6. https://doi.org/10.1109/IICAIET55139.2022.9936736
14. G. Agorku, S. Hernandez, M. Falquez, S. Poddar, K. Amankwah-Nkyi, Traffic cameras to detect inland waterway barge traffic: An application of machine learning, arXiv preprint arXiv:2401.03070, (2024).
15. Q. H. Cap, A. Fukuda, S. Kagiwada, H. Uga, N. Iwasaki, H. Iyatomi, Towards robust plant disease diagnosis with hard-sample re-mining strategy, *Comput. Electron. Agric.*, **215** (2023), 108375. https://doi.org/10.1016/j.compag.2023.108375
16. M. E. Haque, A. Rahman, I. Junaeid, S. U. Hoque, M. Paul, Rice leaf disease classification and detection using yolov5, (2022), arXiv preprint arXiv:2209.01579.
17. A. Pundhir, D. Verma, P. Kumar, B. Raman, Region extraction-based approach for cigarette usage classification using deep learning, (2021), arXiv:2103.12523.
18. R. Liu, Z. Ren, Application of Yolo on mask detection task, in *2021 IEEE 13th International Conference on Computer Research and Development (ICCRD)*, (2021), 130–136. https://doi.org/10.1109/ICCRD51685.2021.9386366
19. C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2023), 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

20. J. Chen, S. Kao, H. He, W. Zhuo, S. Wen, C. H. Lee, et al., Run, Don't walk: Chasing higher FLOPS for faster neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2023), 12021–12031. https://doi.org/10.1109/CVPR52729.2023.01157
21. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., Mobilenets: Efficient convolutional neural networks for mobile vision applications, (2017), arXiv preprint arXiv:1704.04861.
22. S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Netw.*, **107** (2018), 3–11. https://doi.org/10.1016/j.neunet.2017.12.012
23. N. Ma, X. Zhang, J. Sun, Funnel activation for visual recognition, in *ECCV 2020: 16th European Conference on Computer Vision*, (2020), 351–368. https://doi.org/10.1007/978-3-030-58621-8_21
24. A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in *26th Annual Conference on Neural Information Processing Systems, NIPS 2012*, *Adv. Neural Inf. Proces. Syst.*, **25** (2012), 1097–1105. https://doi.org/10.1145/3065386
25. X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, *J. Mach. Learn. Res.,* **15** (2011), 315–323.
26. Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, X. Wang, Yolo-facev2: A scale and occlusion aware face detector, (2022), arXiv preprint arXiv:2208.02019.
27. F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, (2015), arXiv preprint arXiv:1511.07122.
28. Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, et al., Enhancing geometric factors in model learning and inference for object detection and instance segmentation, *IEEE Trans. Cybern.*, **52** (2021), 8574–8586. https://doi.org/10.1109/TCYB.2021.3095305
29. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in *Proceedings of the 34th AAAI conference on artificial intelligence*, **34** (2020), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999
30. Z. Tong, Y. Chen, Z. Xu, R. Yu, Wise-IoU: Bounding box regression loss with dynamic focusing mechanism, (2023), arXiv preprint arXiv:2301.10051.
31. R. Padilla, S. L. Netto, E. A. B. Da Silva, A survey on performance metrics for object-detection algorithms, in *Proceedings of the 2020 international conference on systems, signals and image processing (IWSSIP)*, (2020), 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130
32. C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, et al., YOLOv6: A single-stage object detection framework for industrial applications, (2022), arXiv preprint arXiv:2209.02976.
33. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, (2021), arXiv preprint arXiv:2107.08430.
34. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, et al., SSD: Single shot multibox detector, in *ECCV 2016: 14th European Conference on Computer Vision*, (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
35. S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern. Anal. Mach. Intell.*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

36. T. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in *Proceedings of the IEEE international conference on computer vision*, (2017), 2980–2988. https://doi.org/10.1109/ICCV.2017.324