INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING
DOI: 10.46300/9106.2022.16.29 Volume 16, 2022

Walnut Ripeness Detection Based on Coupling Information and Lightweight YOLOv4

Kaixuan Cui 1,2, Shuchai Su 3,4, Jiawei Cai 1,2, Fengjun Chen 1,2*
1. School of Technology, Beijing Forestry University, Beijing, 100083, China
2. Beijing Laboratory of Urban and Rural Ecological Environment, Beijing, 100083, China
3. Key Laboratory for Silviculture and Conservation of Ministry of Education, Beijing, 100083, China
4. National Energy Research&Development Center for Non-food Biomass, Beijing, 100083, China

Received: July 6, 2021. Revised: December 15, 2021. Accepted: January 8, 2022. Published: January 10, 2022.

***Abstract*—To realize rapid and accurate ripeness detection for walnut on mobile terminals such as mobile phones, we propose a method based on coupling information and lightweight YOLOv4. First, we collected 50 walnuts at each ripeness (Unripe, Mid-ripe, Ripe, Over-ripe) to determine the kernel oil content. Pearson correlation analysis and one-way analysis of variance (ANOVA) prove that the division of walnut ripeness reflects the change in kernel oil content. It is feasible to estimate the kernel oil content by detecting the ripeness of walnut. Next, we achieve ripeness detection based on lightweight YOLOv4. We adopt MobileNetV3 as the backbone feature extractor and adopt depthwise separable convolution to replace the traditional convolution. We design a parallel convolution structure with depthwise convolution stacking (PCSDCS) to reduce parameters and improve feature extraction ability. To enhance the model's detection ability for walnuts in the growth-intensive areas, we design a Gaussian Soft DIoU non-maximum suppression (GSDIoU-NMS) algorithm. The dataset used for model optimization contains 3600 images, of which 2880 images in the training set, 320 images in the validation set, and 400 images in the test set. We adopt a multi-training strategy based on dynamic learning rate and transfer learning to get training weights. The lightweight YOLOv4 model achieves 94.05%, 90.72%, 88.30%, 76.92 FPS, and 38.14 MB in mean average precision, precision, recall, average detection speed, and weight capacity, respectively. Compared with the Faster R-CNN model, EfficientDet-D1 model, YOLOv3 model, and YOLOv4 model, the lightweight YOLOv4 model improves 8.77%, 4.84%, 5.43%, and 0.06% in mean average precision, 74.60 FPS, 55.60 FPS, 38.83 FPS, and 46.63 FPS in detection speed, respectively. And the lightweight YOLOv4 is 84.4% smaller than the original YOLOv4 model in terms of weight capacity. This paper provides a theoretical reference for the rapid ripeness detection of walnut and exploration for the model's lightweight.**

***Keywords*—Ripeness, Coupling information, Lightweight, YOLOv4**

I. Introduction

THE ripeness is a vital basis for guiding the harvesting of walnuts and a significant factor affecting the oil content of kernels. Accurately judging the ripeness of walnuts is conducive to scientifically determining the timing of harvesting and improving the oil content of kernels to boost the economic benefits of the walnut oil production industry [1]. Traditional ripeness detection for walnuts relies on farmers' experience and is subjective to personal factors, making it impossible to guarantee the detection accuracy of the ripeness. An automatic, fast, and accurate ripeness detection method for walnuts has become an urgent need for the walnut planting industry and walnut oil producers.

At present, non-destructive detection of fruit ripeness is divided into non-visual and visual means. Non-visual means are methods using electrical [2], vibrational frequency [3]-[6], optical [7], and chemical properties [8]-[11] of the fruit. The methods do not damage the internal tissues of the fruit and achieve highly accurate ripeness detection. However, they rely on complex and expensive detecting equipment for operation in the laboratory, making it difficult to achieve timely detection in the natural environment.

With computer technology and image processing technology development, fruit ripeness detection based on computer vision has become a research hotspot. Using the traditional computer vision method to detect the fruit ripeness needs designing artificial operator to extract the key features of fruit, such as color, texture, and edge. Then the key features are input into machine learning models such as support vector machine [12], decision tree [13][14], random forest [15], artificial neural network [16][17], and partial least squares [18]-[20] for ripeness detection. The essential feature extraction of these methods is complex, the real-time performance is poor, and the accuracy is low in the natural environment. By inputting large-scale image data and iterative training, deep learning can extract the key features of the target independently, which has strong adaptability and robustness [21][22]. At present, fruit ripeness detection based on deep learning and computer vision mainly uses segmentation [23][24] and object detection methods [25]-[28]. Xue et al. proposed an improved FCN-8s based segmentation method to achieve the segmentation of Lingwu long jujubes of different ripeness [23]. The improved network used a multiscale feature extraction module to extract features from different sizes of objects. The experimental results on Lingwu long jujubes dataset showed that the intersection over union, mean intersection over union, precision accuracy, recall rate, and F1 score were 93.50%, 96.41%, 98.44%, 97.86%, and 98.15%, respectively. The network parameters of the improved FCN-8s were 5.37 million, and the segmentation speed was 16.20 frames/s. Huang et al. used the fuzzy Mask R-CNN model to identify tomato ripeness automatically [24]. For the detection of 100 tomato images, the fuzzy Mask R-CNN achieved an accuracy of 98.00%. The ripeness classification of tomatoes achieved overall weighted

precision and recall rates of 0.9614 and 0.9591, respectively. S. Parvathi et al. used the Faster R-CNN model with the ResNet-50 network to detect ripeness stages of coconuts in natural backgrounds [25]. The method achieved a mean average precision of 89.40% and an average recognition speed of 3.124 frames/s. Chen et al. achieved detecting citrus in an orchard environment using improved YOLOv4 [26]. The network adopted the canopy algorithm and the K-means++ algorithm for improvement. The experimental results showed that the improved YOLOv4 detector works excellent for detecting different growth periods of citrus in the natural environment, with an average increase in accuracy of 3.15%. The average detection time of this model is 16.7 frames/s at 1920×1080 pixels. A. Kuznetsova et al. used the YOLOv3 algorithm for apple detection in a fruit-harvesting robot, providing an average apple detection time of 19 ms with a share of objects being mistaken for apples at 7.8% and a share of unrecognized apples at 9.2% [27]. Liu et al. proposed an improved YOLOv3 to identify the ripeness of strawberries [28]. The model adopted Gamma transform image enhancement to improve the detection ability. The results showed that the improved YOLOv3 algorithm provided the mean average precision of 87.51% and detection speed of 58.1 FPS. It should be emphasized that there is no research on the detection method of walnut ripeness according to the current literature.

FCN segmentation model, Faster R-CNN, and YOLO object detection model have achieved satisfying results in fruit ripeness detection. The segmentation model represented by FCN and the two-stage object detection model represented by Faster R-CNN have strong generalization ability. However, they consume more computer resources and take a long time to detect, so it is difficult to adapt to mobile terminals [29]. An additional note is that a real-time segmentation method exists. But the method is mainly applied to halftone images [30]. However, images taken by mobile terminals such as mobile phones are RGB images. In this paper, the YOLOv4 model is selected in the walnut ripeness detection task. The model has excellent advantages in detection accuracy and detection speed. Above all things, the model has a particular structure, making it easy to capture the subtle changes of walnut peel during the ripening process. By aggregating the top-down semantic features of the FPN layer and the bottom-up positioning features of the PAN layer, the particular structure has an outstanding effect in capturing the subtle changeable features [31]. However, the backbone feature extractor of the YOLOv4 model is CSPDarkNet53, which has a complex structure and a large number of parameters. The original feature fusion network uses the traditional alternating convolution for feature extraction and fusion, resulting in network parameters redundancy and reducing the detection speed. Therefore, we had improved the model by making it lighter. The lightweight improvement would reduce the detection performance of the model, especially for the target dense area. And walnuts grow in clusters, leading to mutual shelter between the fruits. Therefore, the problem of missing detection is serious. So we proposed a new non-maximum suppression algorithm to improve this phenomenon.

II. Materials & Methods

A. Experimental sample collection

According to the criteria used by industry experts to classify the ripeness of walnuts, this paper classifies the ripeness of walnuts into the following four grades: Unripe: the peel color is dark green or green. Mid-ripe: the skin turns yellow-green or yellowish. Ripe: the peel is yellow-green or yellowish, and the top of the fruit is split and separated from the hull. Over-ripe: the sides of the peel are cracked and open, and the nuts are exposed.

The walnut variety selected for this experiment is "Liaoning No. 1", and the samples were collected at the Dashan Xingang walnut plantation in Mentougou District, Beijing, China (115°58′01″E， 40°01′29″N， 818 m above sea level). Fifty walnuts of each ripeness were picked in September-October 2020, and 200 fruits were collected. Figure 1 shows the sample of four ripeness grades.

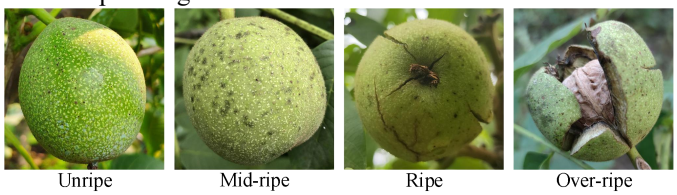


Fig. 1 Sample of walnuts with different ripeness

B. Image acquisition

The images were taken with an iPhone XR smartphone. Images were taken on 12 September, 19 September, 26 September, and 3 October 2020. The shooting distance was less than 1.5 m, and the shooting angle was a random multi-angle. Each period was photographed with 300 shots, giving a total of 3600 images of walnut fruit. The image size is 3240×4032 pixels, and the image format is jpg.

C. Dataset construction

Three thousand six hundred images were obtained, of which 2880 images were in the training set, 320 images in the validation set, and 400 images in the test set. All images in the dataset are captured by mobile phones without other processing methods. The ratio of the number of images in the training set and the validation set is 9:1. That follows the common strategy of data set division, which is conducive to the optimization of the model. In the test set, there were 100 walnut images at each ripeness grade. It maintains the balance of data and is conducive to obtaining accurate detection results. The image was annotated using the LableImg script according to the four ripeness grades of walnuts. Unripe, Mid-ripe, Ripe, and Over-ripe walnuts were labeled Maturity-1, Maturity-2, Maturity-3, and Maturity-4, respectively. The format of the markup file is XML, and the composition of the dataset is Pascal VOC.

Fig. 2 Walnuts labeled by LabelImg script

III. Coupling Relationship Analysis

Walnut ripeness detection aims to improve the kernel oil extraction rate and determine the best harvest time. Firstly, we need to analyze whether the ripeness divided by experts can reflect the change in kernel oil content. We figured out the coupling relationship between the ripeness and the oil content of walnut.

The oil content of 200 samples of kernels was determined by the Soxhlet extraction method according to the Chinese National Standard for Fat Determination GB 5009.6-2016. The statistical and one-way analysis of variance (ANOVA) results for oil content is shown in Table 1.

Tab. 1 Oil content statistics and one-way analysis of variance

ripeness	Maximum /%	Minimum /%	Mean /%	Standard Deviation	Significance Index	F value	P value
Unripe	64.08	46.54	55.41	0.078	0.823	22.009	0.000015
Mid-ripe	73.46	70.12	72.25	0.015	0.314		
Ripe	74.69	74.16	74.41	0.002	0.758		
Over-ripe	69.97	67.73	68.97	0.008	0.887		

As can be seen from the table above, the range of oil content between the ripeness classes is clearly defined from Unripe to Ripe, with no overlap between the values and a steady increase in oil content. The oil content of the kernel decreases from Ripe to Over-ripe. Variance satisfies the chi-squared condition. Shapiro-Wilk was used to test the normality of the data within the group, the significance indexes were all greater than 0.05, and the data satisfied the normal distribution. The P-value in the table was 0.000015<0.01, indicating that the oil content of the samples at different ripeness was very significantly distinctive. In addition, we analyzed the correlation between ripeness and oil content using Pearson correlation. We found that the correlation coefficient between ripeness and oil content was 0.832 from Unripe to Ripe, indicating a significant positive correlation. The correlation coefficient between ripeness and oil content from Ripe to Over-ripe was -0.981, demonstrating a significant negative correlation. Therefore, the walnut ripeness divided by experts can reflect the change of kernel oil content. The corresponding relationship between the walnut ripeness and the range of kernel oil content is shown in Table 1. Further, post-hoc multiple tests were conducted using the least significant difference (LSD) method for the different ripeness grades, and the results are shown in Table 2.

Tab. 2 Multiple tests between different ripeness

Comparison between different ripeness grades		P value
Unripe	Mid-ripe	0.000017
	Ripe	0.000002
	Over-ripe	0.000089
Mid-ripe	Ripeness	0.400758
	Over-ripe	0.210182
Ripe	Over-ripe	0.036495

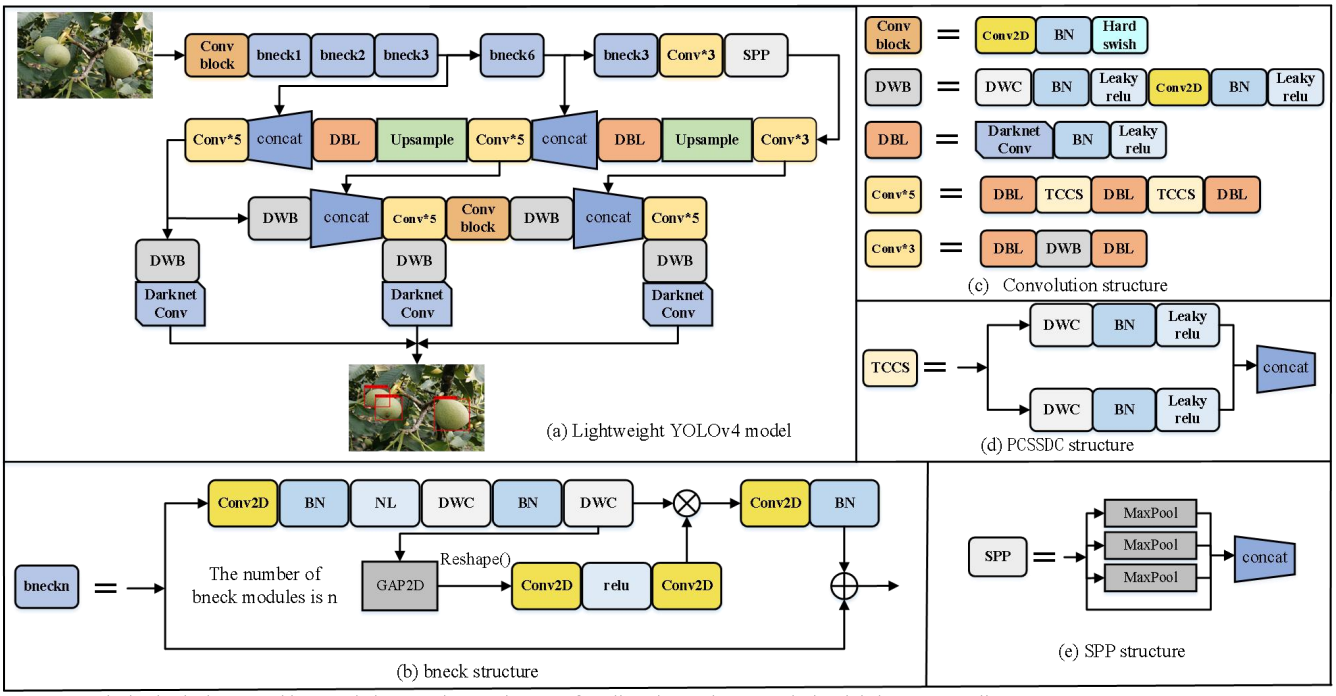
As can be seen from Table 2, the kernel oil content at Unripe was very significantly different from that at the other ripeness grades. The kernel oil content at Mid-ripe was distinctively different from that at Unripe. The kernel oil content at Ripe was significantly different from that at Unripe and Over-ripe. The difference in oil content from Mid-ripe to Ripe was not significant, indicating that the increase in kernel oil content slowed down at this stage. The oil content peaked at Ripe and decreased significantly at Over-ripe. Therefore, Ripe is the most suitable harvesting period. Harvesting and extracting walnut oil during this period can yield the most significant economic benefits.

In summary, we can adopt the division of walnut ripeness by experts to estimate the oil content of kernels. The next step is to detect the ripeness of walnuts automatically and quickly.

IV. Lightweight YOLOv4 Model

The weight capacity of the original YOLOv4 is large, the detection speed is slow, which leads to difficulty in deploying on the mobile terminal. Walnuts grow in clusters, and the lightweight improvement will reduce the detection ability of the model to dense areas [32], resulting in missing detection. Therefore, we had made the following improvements to the original YOLOv4: we chose the MobileNetV3 network as the feature extractor; we designed a parallel convolution structure with depthwise convolution stacking (PCSDCS) and adopted the depthwise separable convolution to replace the traditional convolution. We followed the above strategies to reduce the model's weight capacity and improve the model's detection speed. To minimize the missing detection, we designed a Gaussian Soft DIoU non-maximum suppression (GSDIoU-NMS) algorithm.

The lightweight YOLOv4 model is shown in Figure 3.



Note: DWC is the depthwise separable convolution; NL denotes the type of nonlinearity used; GAP2D is the global average pooling.

Fig. 3 Lightweight YOLOv4 model

A. Backbone Feature Extractor

MobileNetV3 is a lightweight feature extractor. The network uses the bneck structure [33], which combines the characteristics with the depthwise separable convolution [34] and the linear bottleneck & inverse residual [35]. The depthwise separable convolution uses k×k depthwise convolution for feature extraction and then fixes the channels using 1×1 pointwise convolutions. Adjusting the feature map's channel from N to M, the depthwise separable convolution is only $\frac{1}{M}+\frac{1}{k^2}$ of conventional convolution in terms of parameters [36]. The linear bottleneck & inverted residual structure contains two parts [35]: the backbone part uses 1×1 kernels to ascend the feature map's dimensions, deep separable convolutions for feature extraction, and 1×1 kernels for reducing dimensions. The residual edge part superimposes the input and output of the bneck structure. In addition, MobileNetV3 introduces the SENet attention mechanism to improve the ability for feature extraction and uses h-swish functions to enhance the model's optimized performance [33].

MobileNetV3 is designed especially for mobile terminals. The input image size of the network is 416×416 pixels, and its main structure is shown in Table 3.

Tab. 3 MobileNetV3 network architecture

Input	*Operator*	*Exp size*	*#out*	*SE*	*NL*	*s*
224×224×3	conv2d	-	16	-	*HS*	2
112×112×16	bneck, 3×3	16	16	-	*RE*	1
112×112×16	bneck, 3×3	64	24	-	*RE*	2
56×56×24	bneck, 3×3	72	24	-	*RE*	1
56×56×24	bneck, 5×5	72	40	√	*RE*	2
28×28×40	bneck, 5×5	120	40	√	*RE*	1
28×28×40	bneck, 5×5	120	40	√	*RE*	1
28×28×40	bneck, 3×3	240	80	-	*HS*	2
14×14×80	bneck, 3×3	200	80	-	*HS*	1
14×14×80	bneck, 3×3	184	80	-	*HS*	1
14×14×80	bneck, 3×3	184	80	-	*HS*	1
14×14×80	bneck, 3×3	480	112	√	*HS*	1
14×14×112	bneck, 3×3	672	112	√	*HS*	1
14×14×112	bneck, 5×5	672	160	√	*HS*	2
7×7×160	bneck, 5×5	960	160	√	*HS*	1
7×7×160	bneck, 5×5	960	160	√	*HS*	1
7×7×160	conv2d, 1×1	-	960	-	*HS*	1
7×7×960	pool, 7×7	-	-	-	-	1
1×1×960	conv2d 1×1, *NBN*	-	1280	-	*HS*	1
1×1×1280	conv2d 1×1, *NBN*	-	k	-	-	1

Among them, ***Input*** denotes the shape of the feature map. ***Operator*** denotes the type of the feature extraction structure. ***exp size*** denotes the channels after feature map ascending dimension. ***#out*** denotes the channels of the output feature map. ***SE*** denotes whether there is a Squeeze-And-Excite in that block. ***NL*** denotes the type of nonlinearity used. Here, *HS* denotes h-swish function and *RE* denotes ReLU function. *NBN* denotes no batch normalization. ***s*** denotes stride. In addition, we can see that the bneck structure is an important part of the MobileNetV3 network. Its structure is as follows:

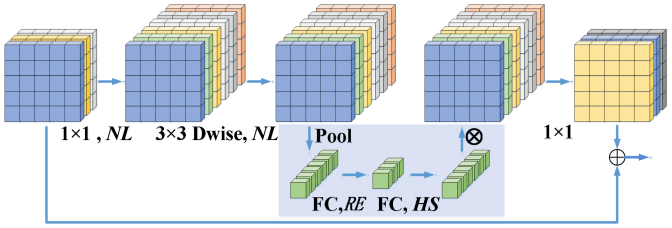


Fig. 4 Bneck structure

The bneck structure first uses the backbone part for feature extraction, and then the original feature map is added to the feature map output from the backbone part. In the above figure, 1×1 stands for 1×1 conventional convolution, 3×3 Dwise stands for 3×3 depthwise convolution, Pool stands for global average pooling, and FC stands for fully connected. *RE* is the

activation function, and its expression is $\mathrm{ReLU} = \max(0, x)$. *HS* stands for h-swish activation function, and its expression is $\text{h-swish}[x] = x\frac{\mathrm{ReLU6}(x+3)}{6}$.

We selected the output feature maps of the 6th, 12th, and 15th bneck structures for the construction of the feature fusion network. We deleted the structure after the 15th bneck structure in the MobileNetV3 network because this part applies to the classification algorithm. When the shape of the input image is 416×416×3, the shape of the output feature maps are 52×52×40，26×26×112, and 13×13×160, respectively.

B. Improved feature fusion networks

Regarding the improvements to feature fusion networks, we replaced the traditional convolution with the depthwise separable convolution and designed a structure of PCSDCS.

In the depthwise separable convolution, the depthwise convolution applies a single filter to each input channel for feature extraction. The pointwise convolution then uses a 1×1 convolution to combine the outputs of the depthwise convolution. The depthwise separable convolution is as follows.

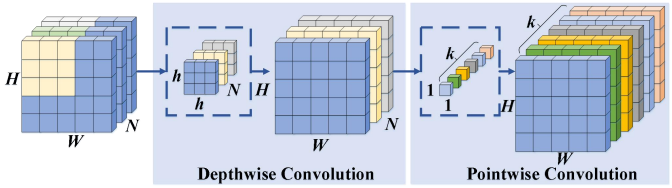


Fig. 5 Depthwise separable convolution

The original YOLOv4 feature fusion network uses five successive 1×1 and 3×3 alternating convolutions (hereafter referred to as alternating convolutions) to extract and fuse features from feature maps stacked at different scales. So the alternating convolutions cause redundancy of parameters and slow down the detection speed. We designed a parallel convolution structure with depthwise convolution stacking (PCSDCS) to solve the above problems. When the feature map input shape is H×W×N, the PCSDCS uses two sets of different depthwise convolution to extract the feature for each input channel, and the feature map output shape is H×W×N. Finally, the feature map output shape by stacking is H×W×2N. If the feature map shape of the input convolution layer is H × W × N and the output is H × W × 2N, the number of parameters required for conventional convolution is $a = 2k^2N^2$, the number of parameters required for depthwise separable convolution is $b = k^2N + 2N^2$, and the number of parameters used for the PCSDCS is $c = 2k^2N$. Here, k is much smaller than N. The ratio of the number of parameters required for the PCSDCS to the number of parameters required for conventional convolution and depthwise separable convolution, respectively, is：

$$\frac{c}{a} = \frac{2k^2N}{2k^2N^2} = \frac{1}{N} < 1 \tag{1}$$

$$\frac{c}{b} = \frac{2k^2N}{k^2N + 2N^2} = \frac{2k}{k + 2N} < 1 \tag{2}$$

The PCSDCS completes the feature extraction, and the 1×1 convolution in the original alternate convolution completes the feature fusion reorganization. With this structure, the detection precision of the model did not drop significantly, but the detection speed was improved, and the model's weight capacity was further reduced. The computational schematic of the PCSDCS is as follows.

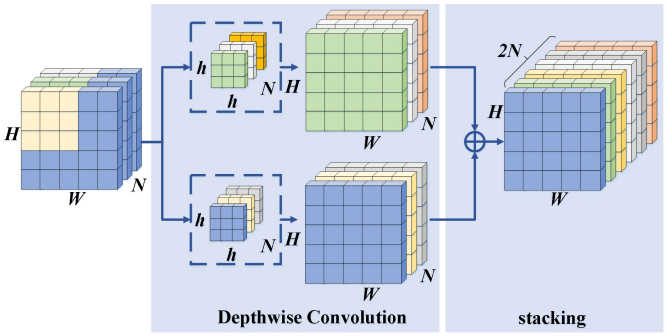


Fig. 6 parallel convolution structure with depthwise convolution stacking (PCSDCS)

C. Non-maximal suppression method

The non-maximum suppression algorithm was used to select the candidate box that belongs to the same category with the highest score within a specific region, while other boxes will be eliminated. Because walnuts grow intensively, and the lightweight improvement will reduce the detection ability of the model to dense areas, it is easy to miss detection targets when detecting for ripeness. To solving this problem, the DIoU-NMS method used in the original YOLOv4 is modified to a GSDIoU-NMS in this paper.

The DIoU-NMS method used in YOLOv4 considers both the value of Intersection over Union (IoU) and the distance between the centers of the two candidate boxes to accelerate loss convergence, which is given by:

$$\begin{cases} S_i = \begin{cases} S_i & IoU - R_{DIoU}(M, b_i) < N_t \\ 0 & IoU - R_{DIoU}(M, b_i) \geq N_t \end{cases} \\ R_{DIoU} = \dfrac{\rho^2(b, b^{gt})}{c^2} \end{cases} \tag{3}$$

Where *Si* - the score of candidate box *i*

M - the candidate box with the highest score

b_i - the candidate box used for comparison in the current category

IoU - the Intersection over Union calculated by candidate box *M* and *bi*

R_{DIoU} - the *DIoU* loss function

N_t - the confidence level of the hyperparameter setting

b, b^{gt} - coordinates of the central pixel points of the two prediction boxes

c - the diagonal pixel lengths of the outer bounding boxes of the two prediction boxes

ρ - Euclidean distance

The GSDIoU-NMS method is constructed based on the

Soft-NMS method [37]. When $IoU - R_{DIoU}(M, b_i) \geq N_t$, the score of candidate box b_i is no longer directly set to 0 but is multiplied by a penalty factor, which is calculated as:

$$S_i = S_i e^{-\frac{IoU(M,b_i)^2}{\sigma}} \quad (4)$$

Where σ - width parameter, taken as 0.5 in this paper.

Since the GSDIoU-NMS method is not applied to all candidate boxes, this step hardly reduces the runtime of the detector.

V. Model Training and Evaluation Indexes

A. Model training

The model training hardware device and environment configuration is Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz processor, 16GB RAM, 8GB NVIDIA GeForce RTX 2080 graphics card, 500GB SSD, Windows 10 system, Tensorflow 1.13.2, and Keras 2.1.5 deep learning frameworks. To obtaining training weights, we used a multi-training strategy based on dynamic learning rate and transfer learning.

First, the Adam optimizer was used to make the loss converge quickly, which facilitates the search for the globally optimal solution region quickly: Freeze Epoch is 100 with the initial learning rate of 0.001, the batch size of 32; Unfreeze Epoch is 100 with the learning rate of 0.0001, the batch size of 8; Every 20 Epoch intervals, cut the learning rate to 1/10 of the original.

Next, based on the weight of Adam optimizer training, the SGD optimizer was used to obtain the globally optimal solution: Freeze Epoch is 50 with the initial learning rate of 0.001, the batch size of 32; Unfreeze Epoch is 50 with the learning rate of 0.0001, the batch size of 8; Every 10 Epoch intervals, cut the learning rate to 1/10 of the original.

B. Evaluation indexes

To evaluating the lightweight YOLOv4 model, it is necessary to consider its detection accuracy, speed, and weight capacity. The weight capacity is measured according to the amount of device memory occupied by the model. The model's detection accuracy is measured by the precision, the recall, the average precision (AP), and the mean average precision (mAP). The average detection speed (ADS) measures the detection speed. The calculation formulas are:

$$P = \frac{T_P}{T_P + F_P} \quad (5)$$

$$R = \frac{T_p}{T_p + F_N} \quad (6)$$

$$AP = \int_0^1 P \cdot R \, dR \quad (7)$$

$$mAP = \frac{1}{m} \sum_1^m AP \quad (8)$$

$$ADS = \frac{N}{t_N} \quad (9)$$

Where T_P - the case that the model predicts a positive sample and is a positive sample as well

F_P - the case that the model predicts a positive sample but is a negative sample

F_N - the case that the model predicts a negative sample but is a positive sample

m - the number of categories detected by the model

N - the number of images detected by the model

t_N - the total time taken by the model to detect N images

VI. Results & Analysis

A. Effectiveness of improvement strategy

To prove the effectiveness of the improved method, we designed ablation experiments. We tested the following networks by controlling variables: (1) YOLOv4; (2) YOLOv4 + MobileNetV1; (3)YOLOv4 + MobileNetV2; (4) YOLOv4 + MobileNetV3; (5) YOLOv4+improved feature fusion networks; (6) YOLOv4+GSDIoU-NMS; (7) Lightweight YOLOv4.

We trained the networks with the training set containing 2880 images. And we used the test set containing 400 images to output the detection results. The detection results of the ablation experiments are shown in Table 4.

Tab. 4 Detection results of ablation experiments

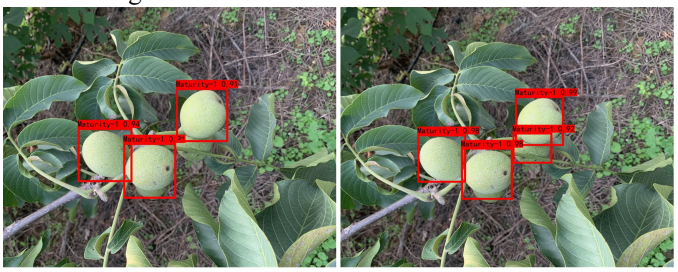
Method	Mean average precision /%	Precision /%	Recall /%	Average detection speed /FPS	Weight capacity /MB
YOLOv4	93.99	91.57	89.51	30.29	245.01
YOLOv4 + MobileNetV1	90.90	88.22	82.15	56.34	155.31
YOLOv4 + MobileNetV2	93.36	89.32	86.84	47.12	148.30
YOLOv4 + MobileNetV3	94.32	91.03	87.83	54.12	152.04
YOLOv4+improved feature fusion network	92.88	91.22	90.02	34.58	131.10
YOLOv4+GSDIoU-NMS	94.58	91.93	89.88	30.26	245.01
Lightweight YOLOv4	94.05	90.72	88.30	76.92	38.14

As shown in Table 4, MobileNet series networks effectively reduce the model's weight capacity and improve the model's detection speed. Experiments showed that the average detection speed and the weight capacity of MobileNetV3 are slightly different from those of MobileNetV1 and MobileNetV2. But it has a significant advantage in the mean average precision, precision, and recall. Therefore, the MobileNetV3 network was selected as the backbone feature extractor for lightweight YOLOv4.

It also can be seen from Table 4 that the improved feature fusion network reduces the weight capacity and improves the average detection speed. Moreover, the recall increased by 0.51%, the precision decreased by 0.35%, and the average precision decreased by 1.11%. In the lightweight improvement, the precision and the average precision of the model are not significantly declining. On the contrary, the recall rate of the model improved. It is due to the PCSDCS structure can aggregate the multi-scale features of objects and improve the detection ability of the model to target dense areas. This structure can make up for the problem of reducing the recall

caused by the lightweight improvement of the model.

About the method of YOLOv4+GSDIoU-NMS, we focus on the impact of GSDIoU-NMS on the recall (The missing detection rate is equal to 1 minus the recall). The GSDIoU-NMS improves 0.37% over DIoU-NMS in terms of recall. Furthermore, due to the reduction of the missing detection rate, the mean average precision is improved by 0.59%. This method enhances the screening ability of target dense areas. The comparison of different NMS algorithms is shown in Figure 7.



(a) YOLOV4+DIoU-NMS (b) YOLOv4+GSDIoU-NMS

Fig. 7 Comparison of different NMS algorithms

Combining three improved methods, we proposed the lightweight YOLOv4. The lightweight YOLOv4 is slightly higher than the original YOLOv4 by 0.06% in mean average precision. The precision and recall of lightweight YOLOv4 are slightly lower than those of the original network, caused by significantly reducing parameters. Even so, these two indexes are still very close to the original model. It is worth noting that the average detection speed and weight capacity are important evaluation indexes of the improved model, which are related to whether the model can be deployed on the mobile terminal. The average detection speed is 2.54 times that of the original, and the weight capacity is reduced by 84.4%. Therefore, the proposed improved method effectively reduces the model's weight capacity and enhances its detection speed without reducing the mean average precision, making it possible to deploy mobile terminals.

B. Experimental results

We used the lightweight YOLOv4 model to detect walnuts at each ripeness grade. We used the test set containing 400 images to output the test results. The performance of the lightweight YOLOv4 in terms of average precision, precision, and recall is shown in Table 5.

Tab. 5 Each ripeness test results

ripeness	Mean average precision /%	Precision /%	Recall /%
Unripe	93.74	88.46	87.79
Mid-ripe	92.24	89.68	84.33
Ripe	93.26	86.23	89.47
Over-ripe	96.96	98.50	91.61

Table 5 demonstrates that the lightweight YOLOv4 model's mean average precision, precision, and recall are the highest when walnuts are in the Over-ripe. Those indexes achieve 96.96%, 98.50%, and 91.61%, respectively. When walnuts are in Mid-ripe, the detection model's mean average precision and recall are the lowest, 92.24% and 84.33%, respectively. When walnuts are in Ripe, the detection model's precision is the lowest with 86.23%.

We analyzed the reasons for this result. Compared with other ripeness, the characteristics of walnuts are more distinctive in the Over-ripe. The walnut peel cracked, and the nut was exposed at this ripeness grade, so the model is easy to detect walnuts with this ripeness. The Mid-ripe and the Ripe are in the transitional ripeness stage. Compared with adjacent ripeness stages, there is no significant difference in walnut appearance in this period. The lightweight YOLOv4 model is prone to incorrectly detecting Ripe and Mid-ripe as Over-ripe and Unripe. Moreover, the detection model is sensitive to significant changes of characteristics, but it is easy to make mistakes for the subtle changes of the peel. So the mean average precision, precision, and recall hit the highest when walnuts are in the Over-ripe. Since Ripe is the optimum harvest time, we need to pay more attention to the test results under this ripeness. At Ripe, the mean average precision, precision, and recall of lightweight YOLOv4 are 93.26%, 86.23%, and 89.47% respectively. The three indexes are greater than 85%, meeting the requirements of ripeness detection accuracy. Therefore, the method we proposed enables the accurate detection of walnut ripeness.

The detection images output by the lightweight YOLOv4 model is shown in Figure 8.

Unripe Mid-ripe Ripe Over-ripe

Fig. 8 Walnut ripeness detection results

C. Comparative analysis

To further test the lightweight YOLOv4 model, the lightweight YOLOv4 model was compared with the Faster R-CNN+ResNet50 model, the EfficientDet-D1 model, the YOLOv3 model, and the YOLOv4 model on the test set. We adopted the mean average precision, the precision, the recall, the average detection speed, and the weight capacity as the evaluation indexes. The detection results are shown in Table 6.

Tab. 6 Comparison of detection results of different methods

Method	Mean average precision /%	Precision /%	Recall /%	Average detection speed /FPS	Weight capacity /MB
Faster R-CNN+ResNet50	85.28	86.25	75.91	2.32	108.66
EfficientDet-D1	89.21	86.40	86.43	21.32	27.08
YOLOv3	88.62	86.86	77.67	38.09	235.49
YOLOv4	93.99	91.57	89.51	30.29	245.01
Lightweight YOLOv4+DIoU_NMS	93.54	90.37	86.87	76.96	38.14
Lightweight YOLOv4	94.05	90.72	88.30	76.92	38.14

The mean average precision, precision, recall, average detection speed, and the weight capacity of the lightweight YOLOv4 model on the test set were: 94.05%, 90.72%, 88.30%, 76.92 FPS, and 38.14 MB, respectively, which superior to the Faster R-CNN model and the YOLOv3 model for the same indexes. Compared with Faster RCNN and YOLOv3, the mean average precision, precision, and recall of the lightweight YOLOv4 model are significantly improved, indicating that the ripeness detection effect of this model is better. The average detection speed of the lightweight YOLOv4 model is 33 times that of Faster RCNN and 2 times that of YOLOv3, indicating that the detection efficiency of this model is higher. The weight capacity of the lightweight YOLOv4 model is about one-third of Faster RCNN and one-fifth of YOLOv3, indicating that the model occupies less device memory. It is very beneficial to the deployment of the model in mobile terminals. It is worth noting that the weight capacity of the EfficientDet-D1 model is 27.08 MB that is 11.06 MB smaller than the lightweight YOLOv4. However, other evaluation indexes of the EfficientDet-D1 model are worse than lightweight YOLOv4. Weight capacity is one of the criteria for evaluating the advantages and disadvantages of the model. The lightweight YOLOv4 occupies 38.14MB of memory and also can be easily deployed for mobile terminals. For other indicators, lightweight YOLOv4 is better. Therefore, Compared with the EfficientDet-D1 model, lightweight YOLOv4 is more suitable for deployment in mobile terminals for ripeness detection. Compared with the original YOLOv4 model, the mean average precision of lightweight YOLOv4 is improved by 0.06%, the detection speed is increased by 2.54 times, and the weight capacity is reduced by 84.4%. However, the precision and recall of lightweight YOLOv4 are lower than those of the YOLOv4 model. It is a general consequence of the model for lightweight improvement [32]. The lightweight improvement will reduce the detection ability of the model to the target dense area. We replaced the GSDIoU-NMS algorithm of the lightweight YOLOv4 model with the original DIoU-NMS algorithm. It is found that if the non-maximum suppression is not improved, the precision and recall of the lightweight model will be lower. Therefore, the GSDIoU-NMS algorithm proposed by us is meaningful. In summary, it shows that the lightweight YOLOv4 model realizes the weight capacity reduction and the detection speed increase while ensuring detection precision.

VII. Conclusion

In this paper, we analyzed the coupling relationship between walnut ripeness and kernel oil content. The coupling relationship proved that the oil content of walnut kernel was significantly different at each ripeness grade. We can estimate the range of oil content in the walnut kernel by detecting the ripeness of walnut. We proposed a lightweight YOLOv4 model to detect walnut ripeness at different growth stages rapidly. The conclusions are as following:

The lightweight YOLOv4 model reduces the number of parameters, improves the detection speed, and mitigates the detection performance of the model decreases in walnut growth-intensive areas. In terms of model structure, we selected MobileNetV3 as the backbone feature extractor for the lightweight YOLOv4 model, designed a structure of PCSDCS, and used the depthwise separable convolution to replace the conventional convolutional. We developed the GSDIoU-NMS method to enhance the model's ability to detect the ripeness of walnuts growing in intensive areas. We used multiple training strategies based on dynamic learning rates and transfer learning to optimize the model in terms of training.

The lightweight YOLOv4 model achieved 94.05%, 90.72%, 88.30%, 76.92 FPS, and 38.14 MB in the mean average precision, precision, recall, average detection speed, and weight capacity, respectively. Compared to the original YOLOv4, the lightweight YOLOv4 is 0.06% better in mean average precision, 2.5 times faster in average detection speed, and 84% smaller in weight capacity. Therefore, the model can meet the detection requirements and mobile terminal deployment requirements for the different ripeness grades of walnuts.

We make a practical discussion on the lightweight improvement of the object detection algorithm. We used the lightweight YOLOv4 model to detect the ripeness of walnut and judge the oil content of walnut kernel according to the coupling relationship, so as to scientifically determine the harvest time and improve the oil extraction rate of the walnut kernel. However, it should be emphasized that the walnut variety for the experiment we selected is Liaoning No. 1. Therefore, it may be necessary to limit the variety of walnut when applying the lightweight YOLOv4 model. In the next stage of research, we will study the ripeness detection of more varieties of walnut.

References

[1] Y. Li, S. Ma, Y. Wang, et al., "The dynamics of fat, protein and sugar metabolism during walnut (Juglans regia L.) fruit development," *AFRICAN JOURNAL OF BIOTECHNOLOGY*, vol. 11, no. 5, pp. 1267–1276, Jan. 2012.

[2] N. A. Aliteh, K. Minakata, K. Tashiro, et al., "Fruit Battery Method for Oil Palm Fruit Ripeness Sensor and Comparison with Computer Vision Method," *Sensors*, vol. 20, no. 3, pp. 637–650, Jan. 2020.

[3] R. Sinambela, T. Mandang, I. Subrata, and W. Hermawan, "Application of an inductive sensor system for identifying ripeness and forecasting harvest time of oil palm," *Scientia Horticulturae*, vol. 265, pp. 109231, Jan. 2020.

[4] N. Misron, N. A. Aliteh, N. H. Harun, et al., "Relative Estimation of Water Content for Flat-Type Inductive-Based Oil Palm Fruit Maturity Sensor," *Sensors*, vol. 17, no. 1, pp. 52–61, Dec. 2016.

[5] S. Landahl and L. A. Terry, "Non-destructive discrimination of avocado fruit ripeness using laser Doppler vibrometry," *Biosystems Engineering*, vol. 194, pp. 251–260, Apr. 2020.

[6] N. Arai, M. Miyake, K. Yamamoto, and I. Kajiwara, "Soft mango firmness assessment based on rayleigh waves generated by a laser-induced plasma shock wave technique," *Foods*, vol. 10, no. 2, pp. 323–338, Jan. 2021.
[7] E. N. Obledo-Vázquez, and J. Cervantes-Martínez, "Laser-induced fluorescence spectral analysis of papaya fruits at different stages of ripening," *Applied optics*, vol. 56, no. 6, pp. 1753–1756, Feb. 2017.
[8] N. Aghilinategh, M. J. Dalvand, and A. Anvar, "Detection of ripeness grades of berries using an electronic nose," *Food Science & Nutrition*, vol. 8, no. 9, pp. 4919–4928, Jun. 2020.
[9] Q. Zhao, Z. Duan, Z. Yuan, et al., "High performance ethylene sensor based on palladium-loaded tin oxide:Application in fruit quality detection," *Chinese Chemical Letters*, vol. 31, no. 8, pp. 2045–2049, May. 2020.
[10] M. Baietto and A. D. Wilson, "Electronic-nose applications for fruit identification, ripeness and quality grading," *Sensors*, vol. 15, no. 1, pp. 899–931, Jan. 2015.
[11] L. Y. Chen, C. C. Wu, T. I. Chou, S. W. Chiu, and K. T. Tang, "Development of a Dual MOS electronic nose/camera system for improving fruit ripeness classification," *Sensors*, vol. 18, no. 10, pp. 3256–3266, Sep. 2018.
[12] K. R. Borba, F. Oldoni, T. Monaretto, L. A. Colnago, and M. D. Ferreira, "Selection of industrial tomatoes using TD-NMR data and computational classification methods," *Microchemical Journal*, vol. 164, no. 4, pp. 106048, Feb. 2021.
[13] N. Goel and P. Sehgal, "Fuzzy classification of pre-harvest tomatoes for ripeness estimation – An approach based on automatic rule learning using decision tree," *Applied Soft Computing*, vol. 36, pp. 45–56, Jul. 2015.
[14] R. Hamza and M. Chtourou, "Design of fuzzy inference system for apple ripeness estimation using gradient method," *IET Image Processing*, vol. 14, no. 3, pp. 561–569, Feb. 2020.
[15] L. F. Santos Pereira, S. Barbon, N. A. Valous, and D. F. Barbin, "Predicting the ripening of papaya fruit with digital imaging and random forests," *Computers and Electronics in Agriculture*, vol. 145, pp. 76–82, Dec. 2017.
[16] I. H. Kao, Y. W. Hsu, Y. Z. Yang, et al., "Determination of Lycopersicon maturity using convolutional autoencoders," *Scientia Horticulturae*, vol. 256, pp. 108538, Jun. 2019.
[17] F. M. A. Mazen and A. A. Nashat, "Ripeness classification of bananas using an artificial neural network," *Arabian Journal for Science and Engineering*, vol. 44, no. 8, pp. 6901–6910, Jan. 2019.
[18] P. Rungpichayapichet, B. Mahayothee, M. Nagle, P. Khuwijitjaru, and J. Müller, "Robust NIRS models for non-destructive prediction of postharvest fruit ripeness and quality in mango," *Postharvest Biology and Technology*, vol. 111, pp. 31–40, Jan. 2016.
[19] Y. Y. Pu, D. W. Sun, M. Buccheri, et al., "Ripeness classification of bananito fruit (Musa acuminata, AA): a comparison study of visible spectroscopy and hyperspectral imaging," *Food Analytical Methods*, vol. 12, no. 8, pp. 1693–1704, May. 2019.
[20] S. Munera, J. M. Amigo, J. Blasco, et al., "Ripeness monitoring of two cultivars of nectarine using VIS-NIR hyperspectral reflectance imaging," *Journal of Food Engineering*, vol. 214, no. 8, pp. 29–39, Dec. 2017.
[21] X. Bai, X. Wang, X. L. Liu, et al., "Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments," *Pattern Recognition*, vol. 120, pp. 108102, Dec. 2021.
[22] S. Muni Rathnam, G. Siva Koteswara Rao, "A Novel Deep Learning Architecture for Image Hiding," *WSEAS Transactions on Signal Processing*, vol. 16, pp. 206-210, Feb. 2020.
[23] J. Xue, Y. Wang, A. Qu, et al., "Image segmentation method for Lingwu long jujubes based on improved FCN-8s," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 37, no. 5, pp. 191–197, Mar. 2021.
[24] Y. P. Huang, T. H. Wang, and H. Basanta, "Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness," *IEEE Access*, vol. 8, pp. 207672–207682, Nov. 2020.
[25] S. Parvathi and S. T. Selvi, "Detection of maturity stages of coconuts in complex background using Faster R-CNN model," *Biosystems Engineering*, vol. 202, pp. 119–132, Jan. 2021.
[26] W. Chen, S. Lu, B. Liu, G. Li, and T. Qian, "Detecting Citrus in Orchard Environment by Using Improved YOLOv4," *Scientific Programming*, vol. 2020, pp. 8859237, Nov. 2020.
[27] A. Kuznetsova, T. Maleva, and V. Soloviev, "Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot," *Agronomy*, vol. 10, no. 7, pp. 1016-1034, Jul. 2020.
[28] X. Liu, C. Cheng, J. Li, et al., "Identification Method of Strawberry Based on Convolutional Neural Network," *Transactions of the Chinese Society for Agricultural Machinery*, vol. 51, no. 2, pp. 237–244, Feb. 2020.
[29] G. Li, Y. Huang, Z. Chen, et al., "Practices and Applications of Convolutional Neural Network-Based Computer Vision Systems in Animal Farming: A Review," *Sensors (Basel, Switzerland)*, vol. 21, pp. 1492–1492, Feb. 2021.
[30] Roumen Kountchev, Roumiana Kountcheva, "Image Segmentation based on Adaptive Mode Quantization and 2D Histograms Analysis," *WSEAS Transactions on Signal Processing*, vol. 15, pp. 121-128, Mar. 2019.
[31] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Apr. 2020.
[32] H. Gao, Y. L. Tian, F. Y. Xu, and S. Zhong, "Survey of Deep Learning Model Compression and Acceleration," *Journal of Software*, vol. 32, no. 1, pp. 68–92, Jun. 2020.
[33] A. Howard, M. Sandler, G. Chu, et al., "Searching for mobilenetv3," *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1314–1324, Nov. 2019.
[34] A. G. Howard, M. Zhu, B. Chen, et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *https://arxiv.org/abs/1704. 04861*, Apr. 2017.
[35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4510–4520, Mar. 2019.
[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *In Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, Jul. 2017.
[37] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS–improving object detection with one line of code," *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5561–5569, Aug. 2017.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Kaixuan Cui contributed to methods design, data analysis and drafted the manuscript. Jiawei Cai polished the manuscript. Fengjun Chen conceived the technical solution. All authors have read and agreed to the published version of the manuscript.

Sources of funding for research presented in a scientific article or scientific article itself

National Key Research and Development Program of China (2019YFD1002401)