Journal of Food Measurement and Characterization (2023) 17:5062–5080
https://doi.org/10.1007/s11694-023-02020-5

ORIGINAL PAPER

DGCC-Fruit: a lightweight fine-grained fruit recognition network

Yuan Ma[1] · Dongfeng Liu[2] · Huijun Yang[1,3,4]

Received: 28 January 2023 / Accepted: 15 June 2023 / Published online: 26 June 2023

Abstract
In recent years, image recognition technology based on deep learning has become a research hotspot in smart agriculture. Aiming at the problem of dataset insufficient in fine-grained fruit object detection, a class mixed fine-grained fruit image object detection dataset *ZFruit* is constructed covering clean, natural and complex backgrounds. At the same time, in view of the current fruit image detection and recognition algorithm with high complexity, large parameters, and difficulty in high precision and lightweight detection of fine-grained fruits in different environments, this paper proposes a lightweight fruit recognition network model *DGCC-Fruit* based on YOLOv5. Firstly, a GC-based low-cost feature extraction network is proposed by integrating the GhostBottleneck module and the coordinate attention mechanism (CA), which enhances the fine-grained feature extraction capability; secondly, a new feature fusion network is constructed by introducing CARAFE content-aware upsampling operator to make full use of deep semantic information to improve the detection performance of fine-grained fruit images; finally, the model is further optimized by the knowledge distillation strategy. Taking the smallest-scale model as an example, the experimental results on the self-made dataset *ZFruit* and the public dataset *VOC2007* show that our $DGCC_n$-Fruit network has better performance than the original YOLOv5n (*ZFruit*: + 2.1% mAP@.5, + 1.9% mAP@.5:.95; *VOC2007*: + 5.4% mAP@.5, + 5.5% mAP@.5:.95), with a reduction of about 14% in the parameters and 11% in the model size.

Keywords Fruit detection dataset · Lightweight network · Fine-grained fruit recognition · YOLOv5

Introduction

Fruit, as one of major agricultural products in the world, is loved by people because of its rich nutritional value, and sweet and sour taste. Since 1990, the global production and output value of major fruits have shown a relatively obvious growth trend. At present, fruit production in most developing countries is dominated by small farmers. Compared with the large-scale and specialized processing processes of the fruit industry in developed countries, the automation level of picking and post-harvest processing is low, and the fruit products lack market competitiveness. China is the largest fruit producer and consumer in the world [1]. However, due to the complex growing environment and the high similarity between different fruits, the robot's recognition and positioning accuracy is low, which affects the efficiency of fruit picking and post-harvest processing [2, 3]. As a result, its export trade volume is only half of the world average.

Fruit detection and recognition is a very critical part in smart agriculture. Many researchers have launched related studies, including traditional and deep learning methods. The traditional methods [4–6] need to manually design features according to the different detected fruits, which has a cumbersome process and poor adaptability to fruit color, illumination change and occlusion under complex conditions. The deep learning methods use CNNs to automatically extract features with more robust and accurate performance, and have become a research hotspot. According to different design ideas, it can be divided into the following two

Yuan Ma and Dongfeng Liu have contributed equally to this work.

✉ Huijun Yang
yhj740225@163.com

1 College of Information Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China

2 Shenzhen Agricultural Science and Technology Promotion Center, Nanshan 518000, Guangdong, China

3 Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling 712100, Shaanxi, China

4 Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, Shaanxi, China

categories: candidate regions-based, such as R-CNN [7], SPPNet [8], Fast-RCNN [9] and Faster-RCNN [10]; regression-based, such as YOLO [11], SSD [12] and RetinaNet [13]. Bargoti et al. [14] proposed a fruit detection system based on Faster-RCNN by studying transfer learning and data enhancement, which can better detect three types of fruits in orchard environment. Borianne et al. [15] also used Faster-RCNN to explore the effect of "detection and recognition" fruits under some special heterogeneous conditions, with limitations in detecting mango fruits and recognizing varieties simultaneously. The methods based on candidate regions have greatly improved the accuracy of fruit detection tasks, but there are still limitations in detecting fruits and recognizing varieties simultaneously, and the staged training causes additional time overhead, so the detection speed has some gaps from real-time.

The subsequently proposed regression-based methods do not have a candidate region stage, and the object location and category information can be obtained by regression directly on the input image through CNNs, which can achieve real-time detection speed and gradually become the mainstream, among which the most representative is the YOLO series. In addition, since large CNNs are difficult to achieve efficient mobile deployment, researchers are paying attention to the lightweight research of algorithm models, trying to obtain higher detection accuracy while designing lightweight networks. Zhou et al. [16] used two lightweight networks, MobileNetV2 and InceptionV3, to develope a KiwiDetector APP based on SSD, and used 8-bit quantization method to compress the model to improve the detection speed. Tian et al. [17] proposed an improved YOLOv3 model using DenseNet, which can effectively improve the detection of apples under occlusion. Koirala et al. [18] compared six object detection algorithms, and finally constructed a MangoYOLO network based on YOLOv2-tiny and YOLOv3, which performs well in real-time detection of mango fruits. Chen et al. [19] proposed an improved YOLOv4 method by introducing ResNet to avoid gradient disappearance, and using the Mish loss function and Mosaic method to enhance small object recognition, which outperforms Faster R-CNN, YOLOv3 and the original YOLOv4 for rapid detection of citrus species and locations. Yang et al. [20] proposed a BCo-YOLOv5 network based on YOLOv5s and BCAM attention mechanism, which realized effective detection of citrus, apples and grapes in orchards. Wang et al. [21] proposed a SM-YOLOv5 detection model by using the lightweight network MobileNetV3-Large as backbone and adding a small object detection layer for object detection of tomato picking robots in plant factories. Li et al. [22] proposed an improved YOLOv5 method for apple recognition in natural environments by using a depthwise separable convolution to achieve lightweight and adopting a visual attention mechanism to solve non-attentional preferences and parameter redundancy. Zhang et al. [23] proposed a YoloV5-Gap method by modifying Conv layer to Focus layer, changing C3 structure layer to better extract global feature information, increasing network jump connection, and dynamically controlling the degree of nonlinearity using an adaptive activation function, which outperforms the YOLOv4, YOLOv5 and YOLOv7 for fast and accurate of grape detection in an orchard environment. Lai et al. [24] proposed an improved YOLOv7 method by adopting SimAM attention mechanism to improve feature extraction ability, improving maximum pool convolution structure to reduce downsampling feature loss, and using Soft-NMS algorithm to improve detection effect when blocked or overlapped, which realized efficient detection of pineapple in complex field environments.

To sum up, many researchers have solved some problems in the field of fruit detection and recognition, but the current research mainly focuses on some specific categories of fruits, and there are few studies on subcategories. However, fine-grained fruit detection and recognition technology has broad application prospects in orchard smart management, smart catering, smart retail and other fields. The only few fine-grained fruit detection studies have problems such as low accuracy and few dataset categories, mainly due to the following two reasons: first, due to the complex growing environment, the great similarity between different varieties of fruits, and the high variability of fruit appearance owing to lighting, occlusion and other reasons, the network is required to capture subtle discriminative local features; second, the labeling of detection datasets is more cumbersome with expert knowledge than classification datasets, so the existing research on fine-grained fruits mainly focuses on classification datasets, and lacks relevant detection datasets. To this end, in this paper, a fine-grained fruit image dataset *ZFruit* containing different scenes is constructed, on which a more efficient and lightweight fine-grained fruit recognition model *DGCC-Fruit* is designed. Aiming at the defects of high complexity, large parameters, and difficulty in lightweight detection, our *DGCC-Fruit* improves image detection and recognition accuracy, and provides new ideas for the research of deep learning in fruit image recognition. The main contributions of this paper are as follows:

(1) In order to solve the lack of public datasets for fine-grained fruit detection and recognition research, a single-multi-class mixed fine-grained fruit image object detection dataset *ZFruit* with clean, natural and complex backgrounds is constructed.
(2) Aiming at the difficulty of existing networks in extracting discriminative features from similar fine-grained fruits and the large number of parameters in industrialization, a GC-based feature extraction network is proposed to focus on the key features of fine-grained fruit

images to improve the model detection performance in a cost-effective way.

(3) Aiming at the low detection accuracy of the existing model for fine-grained fruit images, a feature fusion network based on CARAFE is proposed, which makes full use of the deep semantic information to more effectively retain the fine-grained fruit features, and the knowledge distillation strategy is used to improve the model detection accuracy.

Methodology

This paper constructs a fine-grained fruit image object detection dataset *ZFruit*, improves YOLOv5 algorithm, and proposes a new lightweight fruit recognition network to accurately predict fine-grained fruit in different environments. Firstly, a GC-based key feature extraction network is proposed by integrating the GhostBottleneck and CA module to improve the detection performance at a lower cost; secondly, a CARAFE-based feature fusion network is proposed by introducing CARAFE content-aware upsampling operator to better utilize the semantic information of the feature maps and retain the fine-grained fruit features; finally, the knowledge distillation strategy is used to further optimize the model. The structure of our model *DGCC-Fruit* is shown in Fig. 1. In order to adapt to different application requirements, the model introduces the Bottleneck series scaling factor depth_multiple in the C3 layer and the channel numbers scaling factor width_multiple to control the network depth and width respectively. Five different scale models are constructed, and the specific configurations are shown in Table 1.

Feature extraction network based on GC

The feature extraction network based on YOLOv5 consists of three modules: CBS, C3 and SPPF. It mainly uses ordinary convolution operations for feature extraction, lacking special attention to important features, making it difficult to extract discriminative features from similar features of fine-grained fruits, and with large parameters. GhostBottleneck module can reduce redundant feature computation in feature maps by 1×1 convolution and 5×5 depthwise convolution, and CA attention mechanism can focus on fine-grained fruit

Table 1 Configurations of different scale models

Model scale	x	l	m	s	n
depth_multiple	1.33	1.00	0.67	0.33	0.33
width_multiple	1.25	1.00	0.75	0.50	0.25

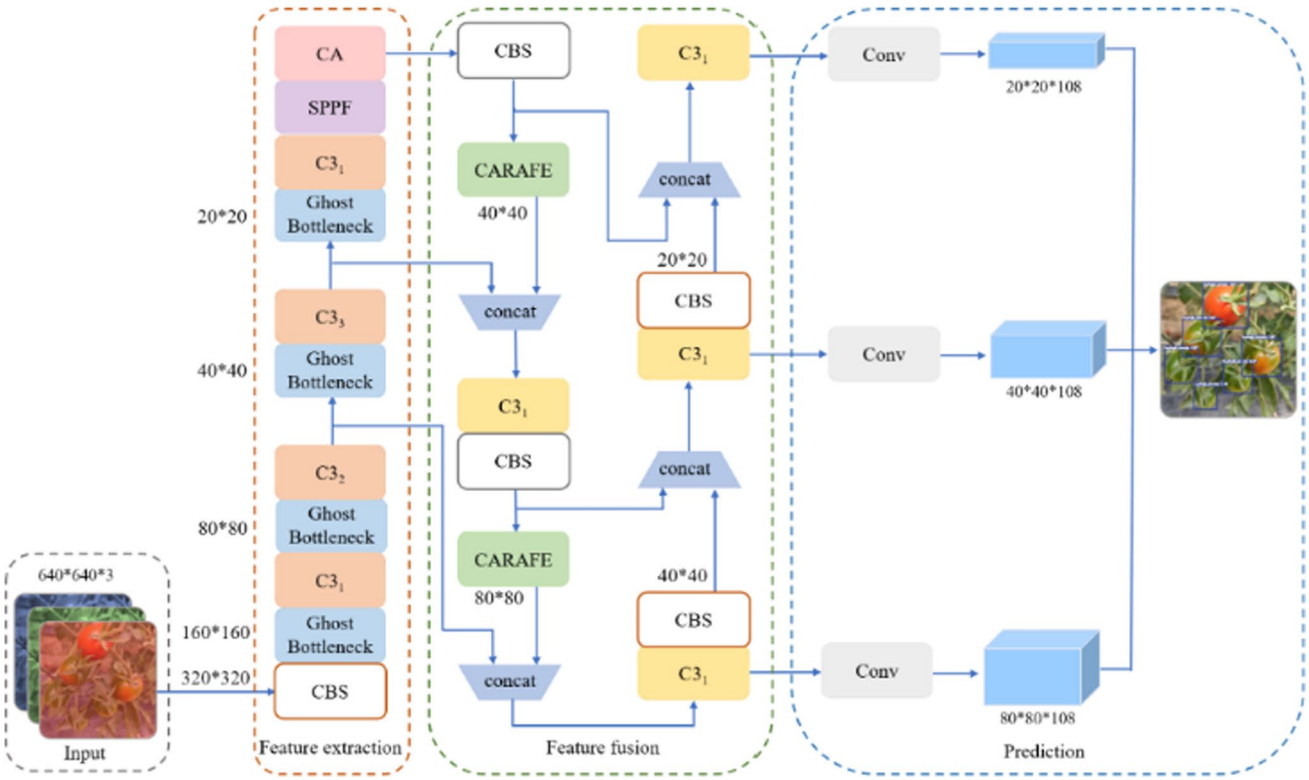

Fig. 1 DGCC-Fruit network structure

key features with high weight, so as to obtain more effective location and category information. Therefore, this paper optimizes the network in a low-cost way by embedding the GC module (GhostBottleneck + CA) to improve the detection performance for fine-grained fruit. The final network structure is shown in Fig. 2.

CBS module: CBS is a standard convolution module, including the convolution (Conv) layer, batch normalization (BN) layer, and SiLU activation function. Among them, Conv(kernel = 6, stride = 2, padding = 2) is used in the CBS of the first layer of the network to replace the Focus layer in the previous version to facilitate model export and improve computational efficiency; the BN layer is used to speed up the training and convergence of the network, preventing gradient disappearance and overfitting; SiLU activation function can be regarded as a smoother ReLU activation function to increase the nonlinear expression ability of the model.

C3 module: C3 is the main module for learning residual features, including three CBS modules and multiple Bottleneck modules. This module can reduce the repetition of gradient information in the optimization process of CNNs, and improve the feature extraction ability of the network while reducing the amount of calculation.

SPPF module: SPPF is a fast spatial pyramid pooling module, which consists of two CBS modules and three concatenated max-pooling layers with the same kernel size. This module increases the receptive field of the network, integrates local features with global features, and enriches the expressive ability of feature maps, which is conducive to the detection of objects with large differences in size. It has lower calculation and faster running speed than SPP.

GhostBottleneck module: Similar to the basic residual block in ResNet, it mainly consists of two stacked GhostConv and depthwise convolution (DWConv), with the structure shown in Fig. 3. Compared with the ordinary convolution downsampling operation, this module can greatly reduce the parameters while ensuring the network recognition effect.

GhostConv is mainly derived from Ghost Module in the lightweight network GhostNet [25]. Due to the large redundant features in the feature map, Ghost Module applies a series of cheap linear transformations to obtain similar features in the original feature map, and generates feature maps with fewer parameters to reduce the computational cost of CNNs. The GhostConv in the GhostBottleneck module is a Ghost Module with a 5×5 linear kernel. It is mainly divided into two steps. First, a set of 1×1 convolutions are used to reduce the number of channels and generate some inherent feature maps to avoid the high parameters caused by the subsequent high number of channels. Then, a 5×5 DWConv

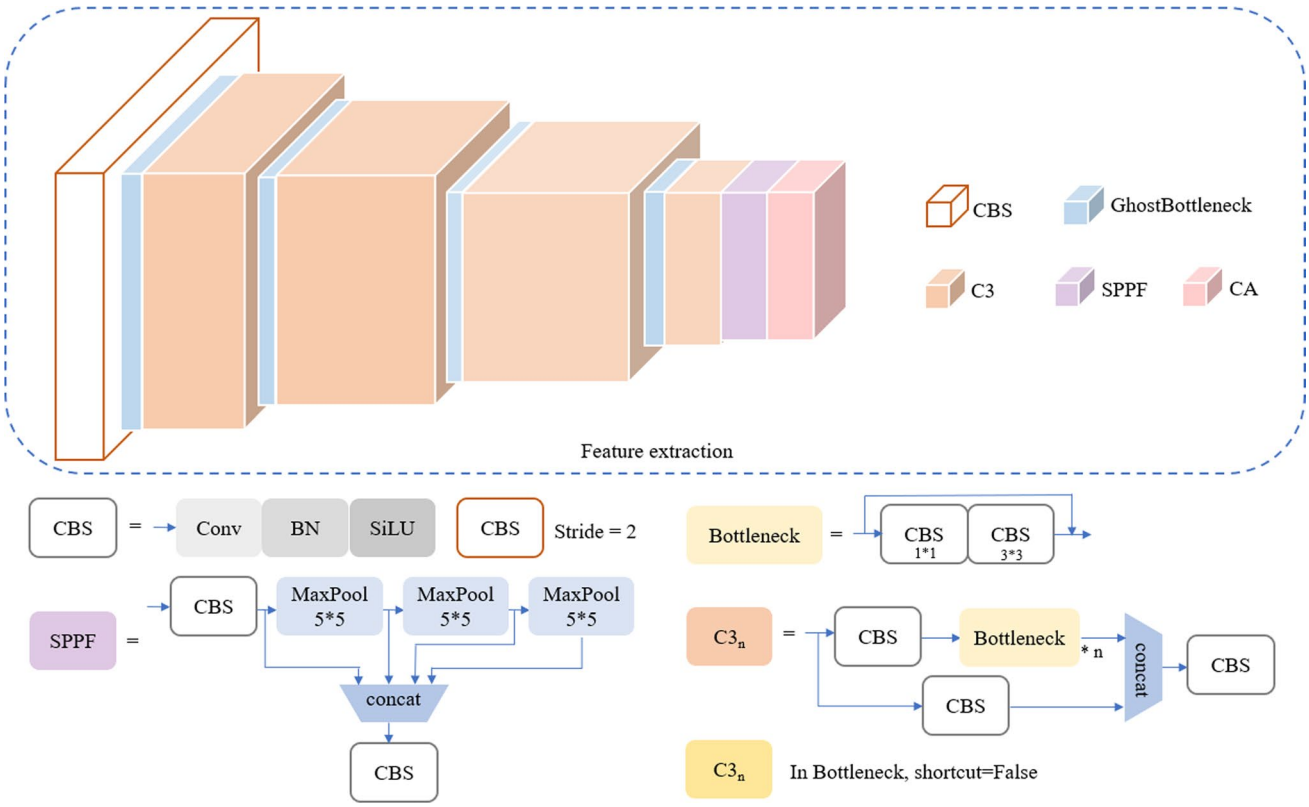

Fig. 2 Feature extraction network based on GC

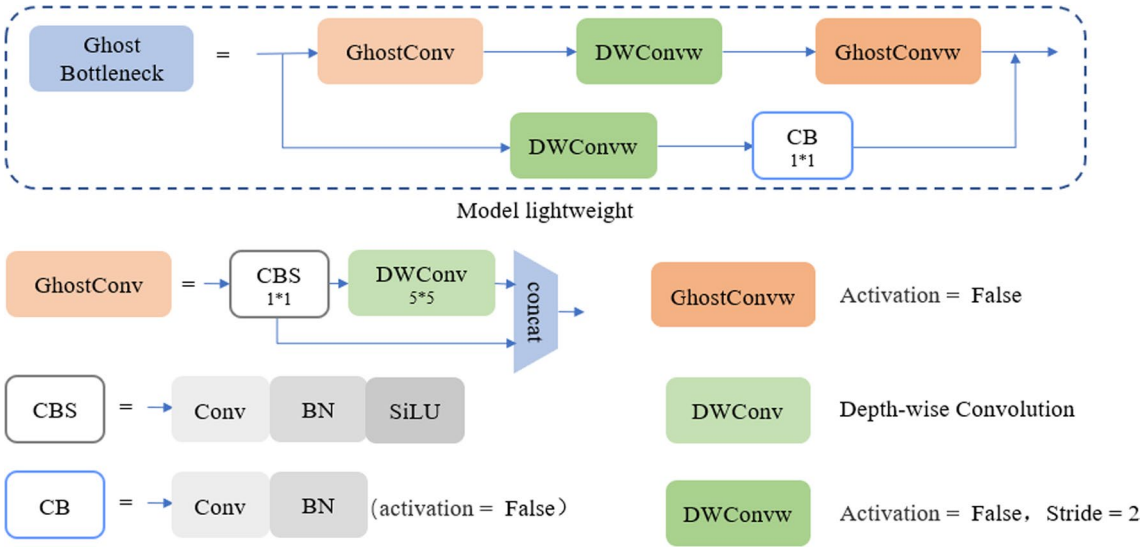

Fig. 3 GhostBottleneck module

is used to enhance the number of features and channels by using a series of simple linear operations, so as to restore the number of channels for final stitching.

Different from ordinary convolution, DWconv performs convolution operation on each channel of the input layer independently. The number of output feature maps is the same as the number of input channels with lower parameters and computational costs. The downsampling operation in the GhostBottleneck module is implemented by a DWconv with stride 2, which lacks interaction on the channel. Therefore, after the DWconv of the lower branch, a 1×1 convolution is used to exchange channel information and change the number of channels to match the upper branch. Drawing on the idea of MobileNetV2 [26], this module only uses the SiLU activation function in the first GhostConv to avoid information loss caused by nonlinear activation functions. It was also pointed out in Xception [27] that it is better not to use the activation function after DWconv.

CA module: CA [28] is a lightweight and efficient attention mechanism, which mainly consists of two average pooling layers and three convolutional layers, with the structure shown in Fig. 4. The module can simultaneously capture cross-channel and direction-aware information to more accurately locate and recognize the region of interest, suppress background interference, and enhance the feature extraction ability of fine-grained fruit in different environments.

In the fine-grained fruit task, it is difficult to achieve correct localization and recognition due to the high similarity of different categories of fruits, and different angles and background interference of the same category. The attention mechanism makes the network selectively pay attention to key features by focusing on basic features and suppressing unnecessary features, which can effectively improve the network representation ability. Currently, the SE [29] attention mechanism commonly used in lightweight networks only considers internal channel information and ignores the importance of positional information. And the CBAM [30] attention mechanism introduces local positional information by global pooling on channels, which cannot obtain the long-range dependent information. The CA attention mechanism solves the above problems by embedding positional information into the channel attention, so that the network can obtain

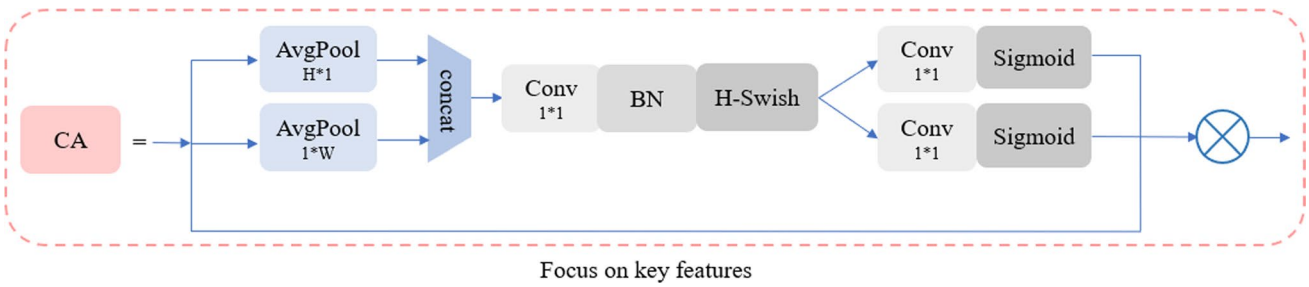

Fig. 4 CA module

long-range dependent information while avoiding excessive computation. Therefore, this paper integrates the CA module in the feature extraction network to make the network focus on the distinctive and important features of the fine-grained fruit to more accurately locate and recognize.

First, in order to avoid the loss of positional information caused by the traditional two-dimensional global pooling, for each channel x_c of the input X with dimension (C, H, W), according to formula (1), the two-dimensional global pooling is decomposed into a pair of one-dimensional feature encodings by the pooling kernels of (H, 1) and (1, W), and the output of the c-th channel with height h and width w is shown in formula (2) and (3). They aggregate features along two spatial directions to generate a pair of direction-aware feature maps that allow the attention module to capture long-range dependencies along one spatial direction and preserve precise positional information along the other.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{1}$$

$$z_c^h(h) = \frac{1}{W} \sum_{0}^{W-1} x_c(h,i) \tag{2}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0}^{H-1} x_c(j,w) \tag{3}$$

Then the pair of one-dimensional feature encodings is spliced along the spatial dimension, and transformed through the 1×1 convolution transformation function F_1, as shown in formula (4). Among them, δ is the H-Swish activation function, f is the intermediate feature map encoded by spatial information in the horizontal and vertical directions, and r is the reduction rate of the number of channels to reduce the complexity and computational overhead of the model.

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right), f \in R^{\frac{C}{r} \times (H+W)} \tag{4}$$

Then f is decomposed into two independent tensors f^h and f^w along the spatial dimension, and f^h and f^w are transformed into tensors with the same number of channels through the 1×1 convolution transformation functions F_h and F_w respectively, as shown in formula (5) (6). Among them, σ is the sigmoid activation function, and g^h and g^w are the attention weights of the input feature map on the height and width.

$$g^h = \sigma\left(F_h\left(f^h\right)\right), f^h \in \mathrm{R}^{\frac{C}{r} \times H} \tag{5}$$

$$g^w = \sigma\left(F_w\left(f^w\right)\right), f^w \in \mathrm{R}^{\frac{C}{r} \times W} \tag{6}$$

Finally, a feature map with attention weights on the height and width directions is obtained by multiplicative weighting calculation on the original feature map, that is, the final output $y_c(i,j)$ of the CA attention mechanism, as shown in formula (7).

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{7}$$

Feature fusion network based on CARAFE

The feature fusion network based on YOLOv5 adopts the structure of FPN and PAN, where FPN conveys strong semantic features from top to bottom, and PAN conveys strong localization features from bottom to top. The Fusion of the extracted semantic and localization features enables the network to obtain richer feature information. Since the current network cannot fully utilize the multi-scale spatial features of fruit, the upsampling in the model cannot be adaptively generated, resulting in poor performance on fine-grained fruit detection. The CARAFE [31] operator can generate upsampling kernel adaptively according to input by operations such as convolution, PixelShuffle and feature reassembly, which is both lightweight and efficient. Therefore, in this paper, the content-aware upsampling operator CARAFE is introduced in the feature fusion stage to combine with the GC-based feature extraction network to better utilize the multi-scale spatial information of fine-grained fruit, and adaptively generate upsampling kernels based on different fine-grained fruit input features to improve detection accuracy. The final network structure is shown in Fig. 5.

The purpose of upsampling is to expand the image resolution. At present, two methods of interpolation and transposed convolution are commonly used for upsampling. The interpolation method directly predicts unknown pixels based on known pixels, which only considers the sub-pixel neighborhood and cannot obtain sufficient semantic information. The transposed convolution allows the network to automatically learn the weights of the convolution kernel by introducing parameters to better restore the image resolution. However, the application of the same upsampling kernel on the entire image limits its responsiveness to local changes and brings a lot of parameters. The CARAFE upsampling operator is introduced for adaptively obtaining high-quality upsampling. It can not only make full use of the deep network to extract the semantic information of fine-grained fruit feature maps, but also aggregate feature information in a larger receptive field with fewer parameters and calculations, with the structure shown in Fig. 6.

CARAFE consists of two key modules: kernel prediction and content-aware reassembly. The former is responsible for the prediction of the upsampling kernel. First, the channel of the fruit image input feature map X (H, W, C) is compressed

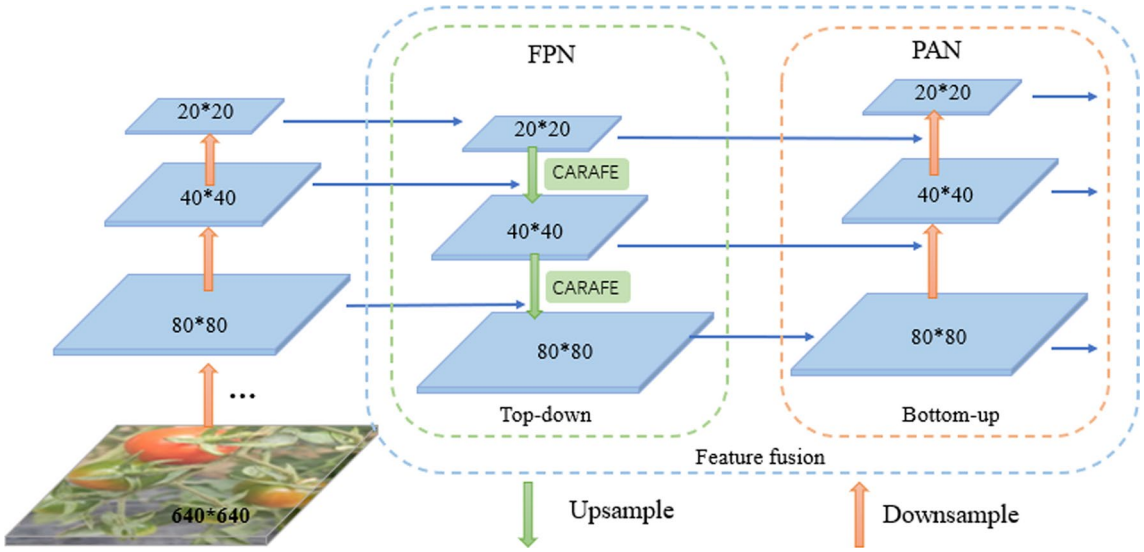

Fig. 5 Feature fusion network based on CARAFE

into C_m through 1×1 convolution to reduce the amount of calculation; then the feature map obtained is encoded by $k_{encoder} \times k_{encoder}$ convolution to obtain a feature map of size $(H, W, \sigma^2 k_{up}^2)$, where σ is the multiple of upsampling and k_{up} is the size of the upsampling kernel; then the feature map is expanded in the spatial dimension according to the channel dimension to obtain a new feature map of size $(\sigma H, \sigma W, k_{up}^2)$; finally, the softmax operation is used for normalization, so that the sum of the predicted upsampling kernel weights is 1. The latter is responsible for the reassembly of features. It mainly obtains the upsampling features of the corresponding pixels through multiplicative weighting calculation. For any pixel X_l' on the output feature map X' of the fruit image, it can be obtained by the weighted summation of the k_{up} neighborhood pixels of X_l in the input feature map X and the upsampling kernel W_l' in the prediction module. By introducing the CARAFE content-aware upsampling operator, the feature fusion network can better integrate multi-scale fine-grained fruit features and promote the transmission of contextual semantic information to enhance the ability of feature expression.

Model optimization based on knowledge distillation

Knowledge distillation was first proposed and applied to classification tasks [32]. It adopts the Teacher-Student learning strategy to achieve model compression, uses the soft labels trained by the strong teacher model to assist the student model training, and transfers the dark knowledge in the complex model to the simple model to improve

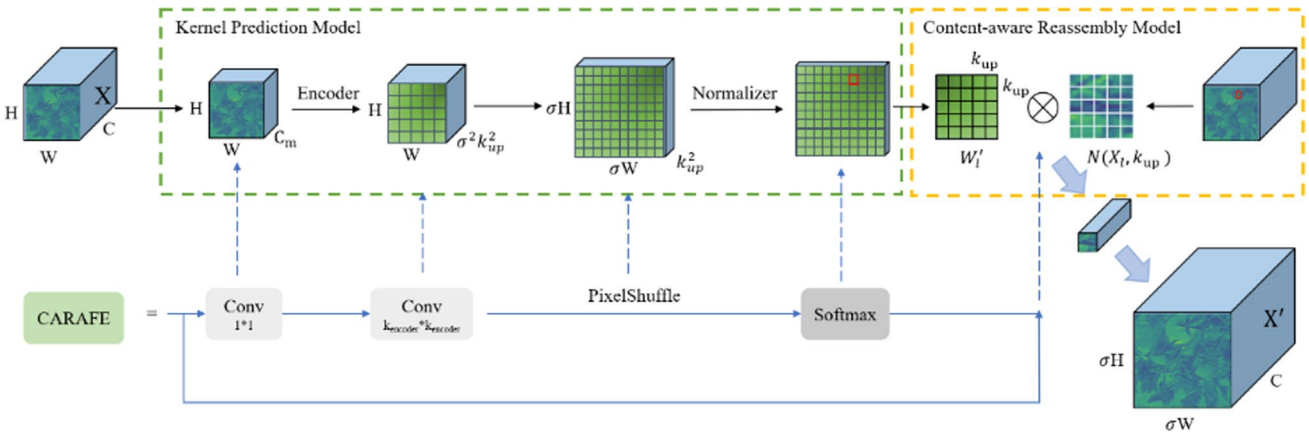

Fig. 6 The network structure of CARAFE

performance. In model compression, pruning and quantization can reduce the model accuracy, while knowledge distillation has been proved to be an effective solution for model compression, which can improve detection accuracy without increasing model size. Therefore, we optimize the model through the method and transfer the knowledge of the largest fine-grained fruit recognition model to the small-scale models, so the models have better detection performance with lightweight.

Our proposed *DGCC-Fruit* network is a single-stage network architecture based on regression, without background filtering, so it needs to directly process candidate regions containing large backgrounds. In the process of knowledge distillation, if numerous backgrounds are passed to the student network, it will cause the network to continuously regress the coordinates of the backgrounds, and the model will be difficult to converge. In addition, if the teacher network is too complex, the performance of the student network will decline without sufficient learning ability. Therefore, this paper draws on the idea of objectness scaled distillation [33] to solve the category imbalance caused by background candidate regions and the idea of teacher assistant knowledge distillation [34] to solve the low efficiency when the gap between teacher and student networks is too large.

Through the objectness scaled distillation, the objectness output of *DGCC-Fruit* is used to limit the distillation loss. Only the candidate boxes with high objectness values from the teacher network can learn the class probabilities and bounding box coordinates, and then contribute to the final loss function of the student network. The final prediction output of the *DGCC-Fruit* network is the objectness values, class probabilities and bounding box coordinates. The total loss L_{total} can be decomposed into three parts: objectness loss f_{obj}, classification loss f_{cl} and regression loss f_{bb}, as shown in formula (8).

$$L_{total} = f_{obj}\left(o_i^{gt}, \hat{o}_i\right) + f_{cl}\left(p_i^{gt}, \hat{p}_i\right) + f_{bb}\left(b_i^{gt}, \hat{b}_i\right), \tag{8}$$

where $\hat{o}_i, \hat{p}_i, \hat{b}_i$ represent the objectness, class probability and bounding box coordinates of the student network respectively, and $o_i^{gt}, p_i^{gt}, b_i^{gt}$ represent the values derived from the ground truth.

When performing knowledge distillation on the network, the loss function consists of two parts, one is the detection loss generated by the student model and the ground truth, and the other is the distillation loss generated by the student model and the teacher model. After distillation, the objectness loss is f_{obj}^{Comb}, the classification loss is f_{cl}^{Comb}, the regression loss is f_{bb}^{Comb}, and the total loss function is $L_{DGCC-Fruit}$, as shown in formulas (9)–(12).

$$f_{obj}^{Comb}\left(o_i^{gt}, \hat{o}_i, o_i^{T}\right) = \underbrace{f_{obj}(o_i^{gt}, (\hat{o}_i))}_{\textit{Detection loss}} + \underbrace{\lambda_D \cdot f_{obj}(o_i^{T}, (\hat{o}_i))}_{\textit{Distillation loss}} \tag{9}$$

$$f_{cl}^{Comb}\left(p_i^{gt}, \hat{p}_i, p_i^{T}, \widehat{o_i^{T}}\right) = f_{cl}\left(p_i^{gt}, \hat{p}_i\right) + \widehat{o_i^{T}} \cdot \lambda_D \cdot f_{cl}\left(p_i^{T}, \hat{p}_i\right) \tag{10}$$

$$f_{bb}^{Comb}\left(b_i^{gt}, \hat{b}_i, b_i^{T}, \widehat{o_i^{T}}\right) = f_{bb}(b_i^{gt}, \hat{b}_i) + \widehat{o_i^{T}} \cdot \lambda_D \cdot f_{bb}(b_i^{T}, \hat{b}_i) \tag{11}$$

$$\begin{aligned} L_{DGCC-Fruit} = & f_{obj}^{Comb}\left(o_i^{gt}, \hat{o}_i, o_i^{T}\right) + f_{cl}^{Comb}\left(p_i^{gt}, \hat{p}_i, p_i^{T}, \widehat{o_i^{T}}\right) \\ & + f_{bb}^{Comb}\left(b_i^{gt}, \hat{b}_i, b_i^{T}, \widehat{o_i^{T}}\right) \end{aligned} \tag{12}$$

where o_i^T, p_i^T, b_i^T represent the objectness, class probability and bounding box coordinates of the teacher network respectively; λ_D is the balance coefficient of distillation loss, which is 1 by default; $\widehat{o_i^T}$ is the objectness score of the teacher network, which is used as a weight to suppress the learning of the background box by the student network.

Through teacher assistant knowledge distillation, a better knowledge distillation effect can be achieved by using medium-sized teaching assistant models for multi-step knowledge distillation operations to bridge the gap between teachers and students. When the gap between the teacher and student networks is too large, the student network can learn through the soft labels of the teaching assistant more effectively than the teacher model. In this paper, the improved model without knowledge distillation optimization is named GCC-Fruit, and the corresponding models of different scales are named $GCC_{x\backslash l\backslash m\backslash s\backslash n}$-Fruit. As the largest fruit recognition model, GCC_x-Fruit can achieve the best results in the experiment, but it has large parameters and model size.

Therefore, this study uses GCC_x-Fruit as the teacher model and $DGCC_{l\backslash m\backslash s\backslash n}$-Fruit as the student model. By learning the knowledge of GCC_x-Fruit, the small-scale fruit models have better detection performance. When $DGCC_l$-Fruit is a student model, since it is only twice as different from GCC_x-Fruit, the teacher assistant model is not used to assist distillation. However, the difference between the subsequent model and GCC_x-Fruit is too large, the previous model is needed to assist distillation. For example, when $DGCC_m$-Fruit is a student model, $DGCC_l$-Fruit is used as a teacher assistant model to assist distillation; when $DGCC_s$-Fruit is a student model, $DGCC_l$-Fruit and $DGCC_m$-Fruit are used as teacher assistant models to assist distillation, and so on. In this experiment, the student model is trained from the pretrained model to reduce the overhead of distillation training, as shown in Fig. 7.

Experiments

Datasets

The self-made dataset in this paper was collected at the kiwi experimental station of Northwest Agriculture and Forestry University in Meixian County, Shaanxi Province, Kengzi experimental base of Shenzhen Agricultural Science and Technology Promotion Center, Nansha headquarters of Guangzhou Academy of Agricultural Sciences and other bases. Firstly, the fruit varieties were identified by professionals, and then the fruit images were collected by ordinary cameras. After the collection, these data was labeled by LabelImg software with fruits marked by the minimum circumscribed rectangle, and annotation files in txt format were generated, which contains the fruit categories, x and y coordinates of the center point and their width w and height h relative to the image. The dataset *ZFruit* produced in this paper contains five fruit categories and thirty-one fine-grained sub-categories fruits (ten kinds of kiwifruit, twelve kinds of tomatoes, four kinds of watermelons, three kinds of pomelos, and two kinds of navel oranges), covering single-class and multi-class mixed fruit image data in clean, natural and complex backgrounds, with a total of 41,799 images, including 154,569 fruit objects, and are divided into training set and test set according to the ratio of 7:3. On this basis, the dataset is augmented during training through homogeneous enhancements such as HSV, rotation, translation, scaling, and flipping transformation, as well as heterogeneous enhancements such as Mosaic, Mixup, and Copy-Paste. Part of the dataset is shown in Fig. 8, an example of data annotation is shown in Fig. 9, the detailed information of the dataset is shown in Fig. 10(a), and the data enhancement effect is shown in Fig. 11. In addition, this study also selected the object detection public dataset Pascal VOC 2007 to test the generality of the model. The dataset covers 20 categories with a total of 9,963 images, including 5,011 training images and 4,952 testing images, containing 30,638 sample objects, and the detailed information is shown in Fig. 10(b).

Experimental setting

The experimental environment is shown in Table 2. During the experiment, the stochastic gradient descent method (SGD) is used to optimize the training network, and the Warmup and cosine annealing algorithms are used to dynamically adjust the learning rate. The parameter settings of the training part are shown in Table 3.

Evaluation metric

In order to comprehensively and objectively evaluate the performance of the *DGCC-Fruit* on fine-grained fruit recognition, four indicators including Average Precision (AP), mean Average Precision (mAP), parameter quantity and model size are selected for model evaluation. The specific calculation formulas are shown in formulas (13)–(16).

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$AP = \int_0^1 PdR \tag{15}$$

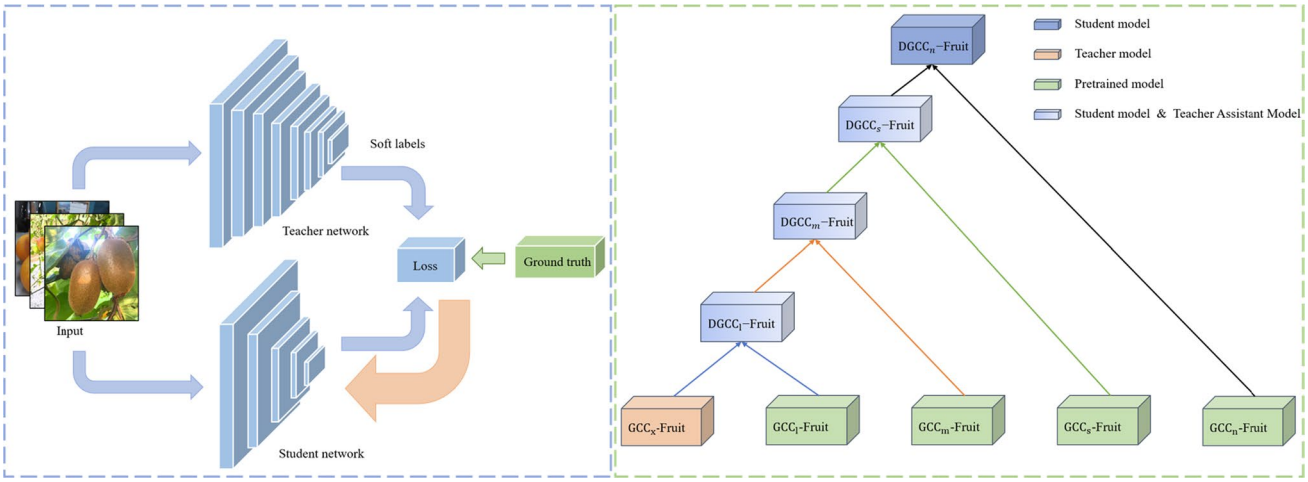

Fig. 7 Experimental process of knowledge distillation

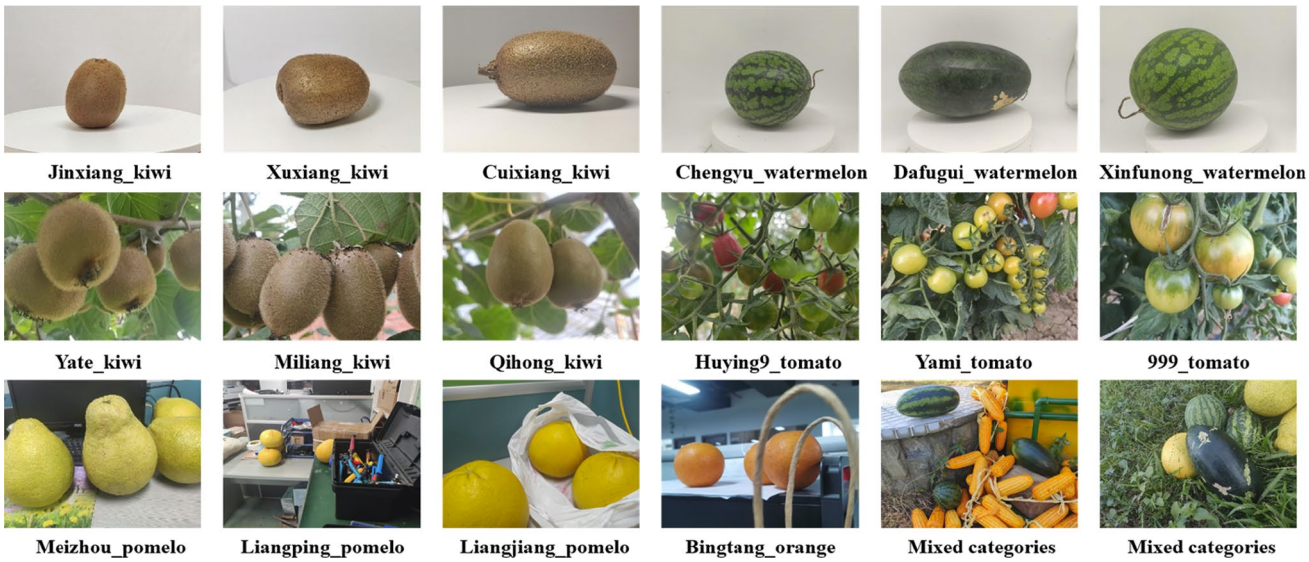

Fig. 8 Part of the dataset

Fig. 9 Example of data annotation

$$mAP = \frac{1}{n}\sum_{i=0}^{n} AP_i \tag{16}$$

Among them, TP refers to the number of objects that the fruit is in the image and is correctly detected; FP refers to the number of objects that the fruit is not in the image but is incorrectly detected; FN refers to the number of objects that the fruit is in the image but is incorrectly detected. AP is the area under the Precision–Recall (P–R) curve, n is the total number of categories of detected objects, i is the number of the current category, and mAP is the mean value of AP for all categories. In this paper, mAP@0.5 means mAP when

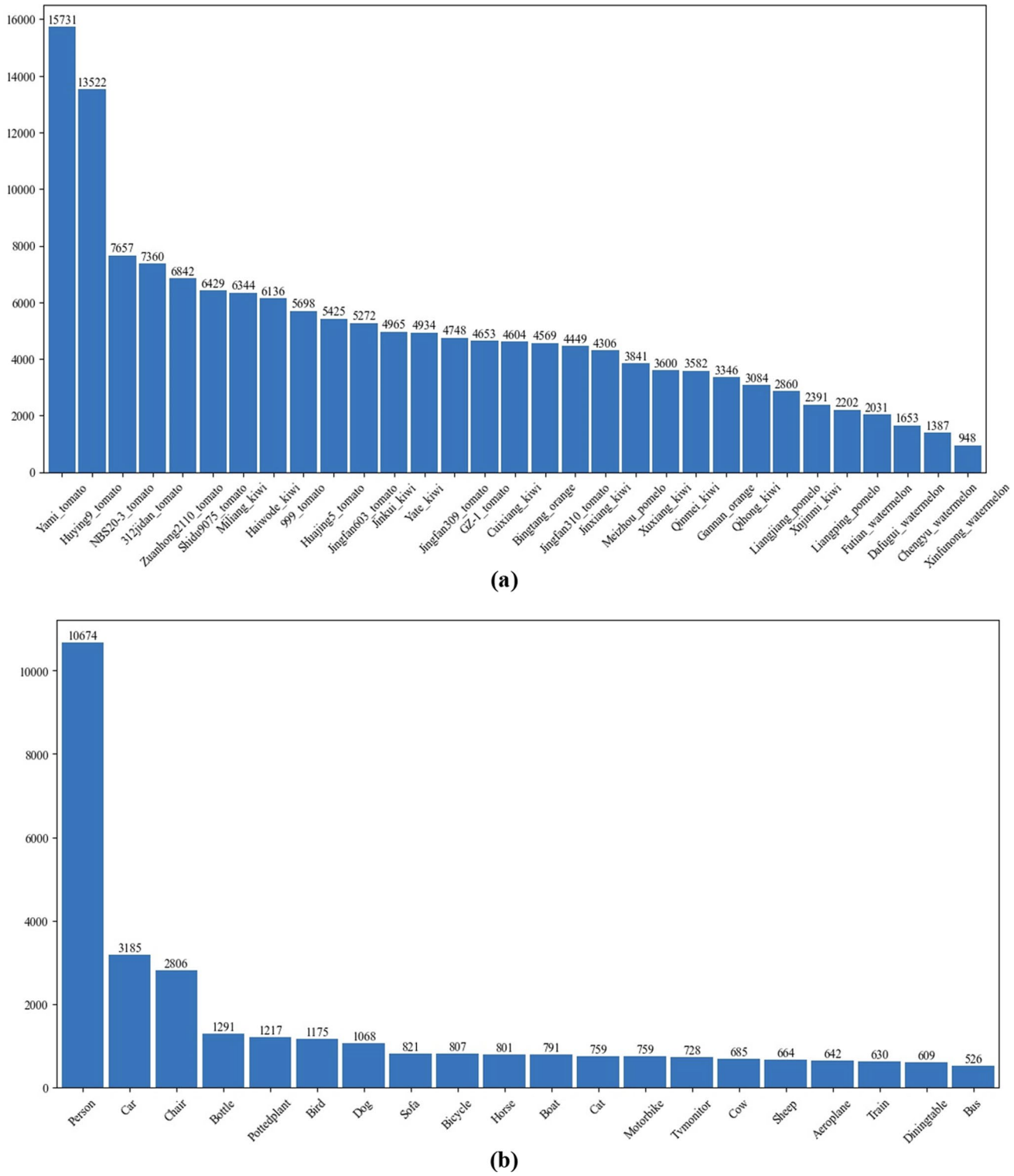

Fig. 10 Ranking of the distribution of the number of samples in the ZFruit (**a**) and VOC2007 (**b**) dataset

the IoU is 0.5, and mAP@0.5:0.95 represents the average mAP over different IoU thresholds (from 0.5 to 0.95 in steps of 0.05). The same is true for AP@0.5 and AP@0.5:0.95. IoU is the degree of overlap between the model prediction box and the ground truth, that is, intersection over union.

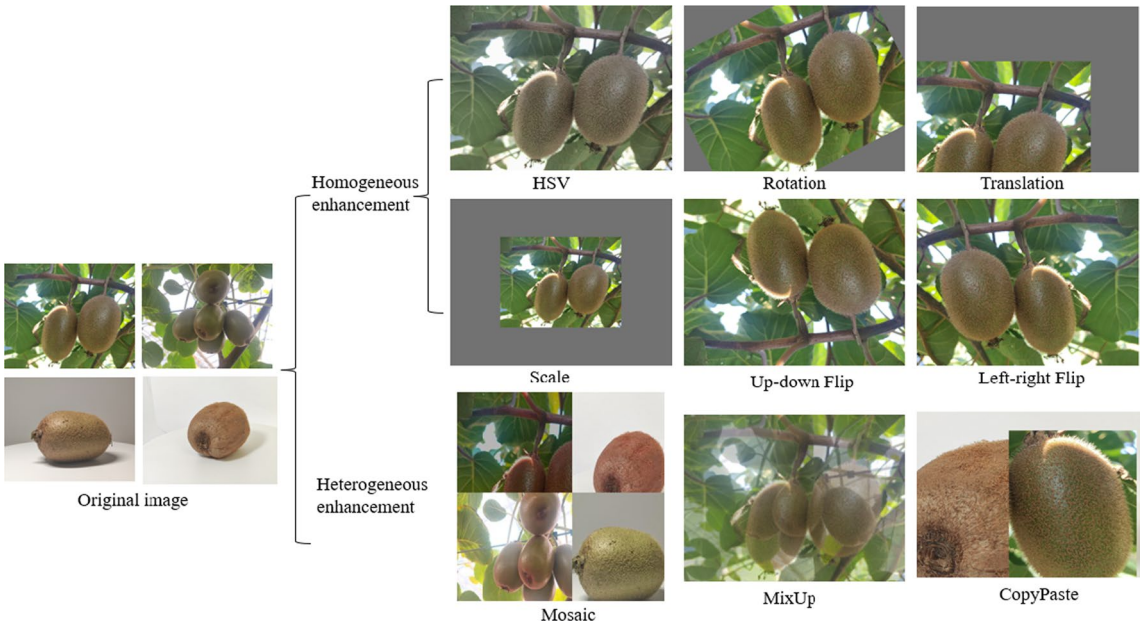

Fig. 11 Data enhancement effect

Table 2 Configuration of the experimental environment

Experimental environment	Configuration information
OS	Ubuntu 16.04 LTS
CPU	Intel XeonSilver 4210 CPU @ 2.20 GHz
GPU	GeForce RTX 2080 Ti 11 GB*4
Memory	125 GB
ML lib	Pytorch
Programming language	Python3.8
GPU acceleration env	CUDA10.1 and cuDNN7.6.5

Table 3 The setting of training parameters

Name	Setting information
Input size of the image	640×640
Initial learning rate	0.01
Momentum	0.937
Weight decay coefficient	0.005
Batch	16
Epoch	300 (ZFruit); 600 (VOC)

Experimental results

In order to rigorously verify the advantages of the proposed model, this paper firstly conducted the comparison of proposed method, including the performance comparison data analysis, visualization comparison of the detection accuracy of each category of fine-grained fruit and the detection results, and then conducted the detection performance data analysis and visualization comparison of different specification models of the benchmark network and mainstream models on both the homemade dataset *ZFruit* and the public dataset *PASCAL VOC 2007*, which proved the effectiveness of the proposed model.

Comparison of proposed method

To test the effectiveness of the proposed model, taking YOLOv5n as an example, a comparative experiment of the improved method proposed in this paper was conducted on the self-made dataset *ZFruit*, and the results are shown in Table 4, and the bolded part indicates the performance of the final proposed model $DGCC_n$-Fruit.

It can be seen from the experimental results that compared with the original model, the feature extraction integrating CA module can increase mAP@0.5 and mAP@0.5:0.95 by 0.9% and 0.5% respectively, under the condition that the number of parameters and model size are increased by 0.6% and 0.3% respectively. Then, the introduction of the CARAFE operator for feature fusion can increase mAP@0.5 and mAP@0.5:0.95 by 1.5% and 1.3% respectively, when the number of parameters and model size are increased by about 2.7%. It is obvious that both methods bring a slight increase in the number of parameters and model size while effectively improving the detection accuracy of the model for fine-grained fruit images. Therefore, the GhostBottleneck module is combined to maintain a lightweight network, which can increase mAP@0.5 and mAP@0.5:0.95 by 1.8% and 1.6% respectively when the number of parameters and model size are reduced by about 14% and 11% respectively.

Table 4 Comparison of different methods

CA	CARAFE	Ghost-Bottle-neck	Distill	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	Weights (MB)
				89.5	79.3	1.80	3.78
✓				90.4	79.8	1.81	3.79
✓	✓			91.0	80.6	1.85	3.88
✓	✓	✓		91.3	80.9	1.55	3.37
✓	✓	✓	✓	**91.6**	**81.2**	**1.55**	**3.37**

On this basis, the final model DGCC$_n$-Fruit is obtained by knowledge distillation strategy, which can increase mAP@0.5 and mAP@0.5:0.95 by 2.1% and 1.9% respectively, with the same variation in the number of parameters and model size as the former. The experimental results show that DGCC$_n$-Fruit can effectively improve the detection performance of fine-grained fruits while greatly reducing the number of network parameters and model size, which verifies the effectiveness of the model.

The comparison of the detection accuracy of each category of fruit before and after the model improvement is shown in Fig. 12. The improved model has a better detection effect on 31 categories of fruit objects, and the detection accuracy of almost all categories has been improved. The experimental results further show the effectiveness of the improved strategy in this paper.

In order to further verify the advantages of the proposed algorithm, fruits in different scenarios were selected to test,

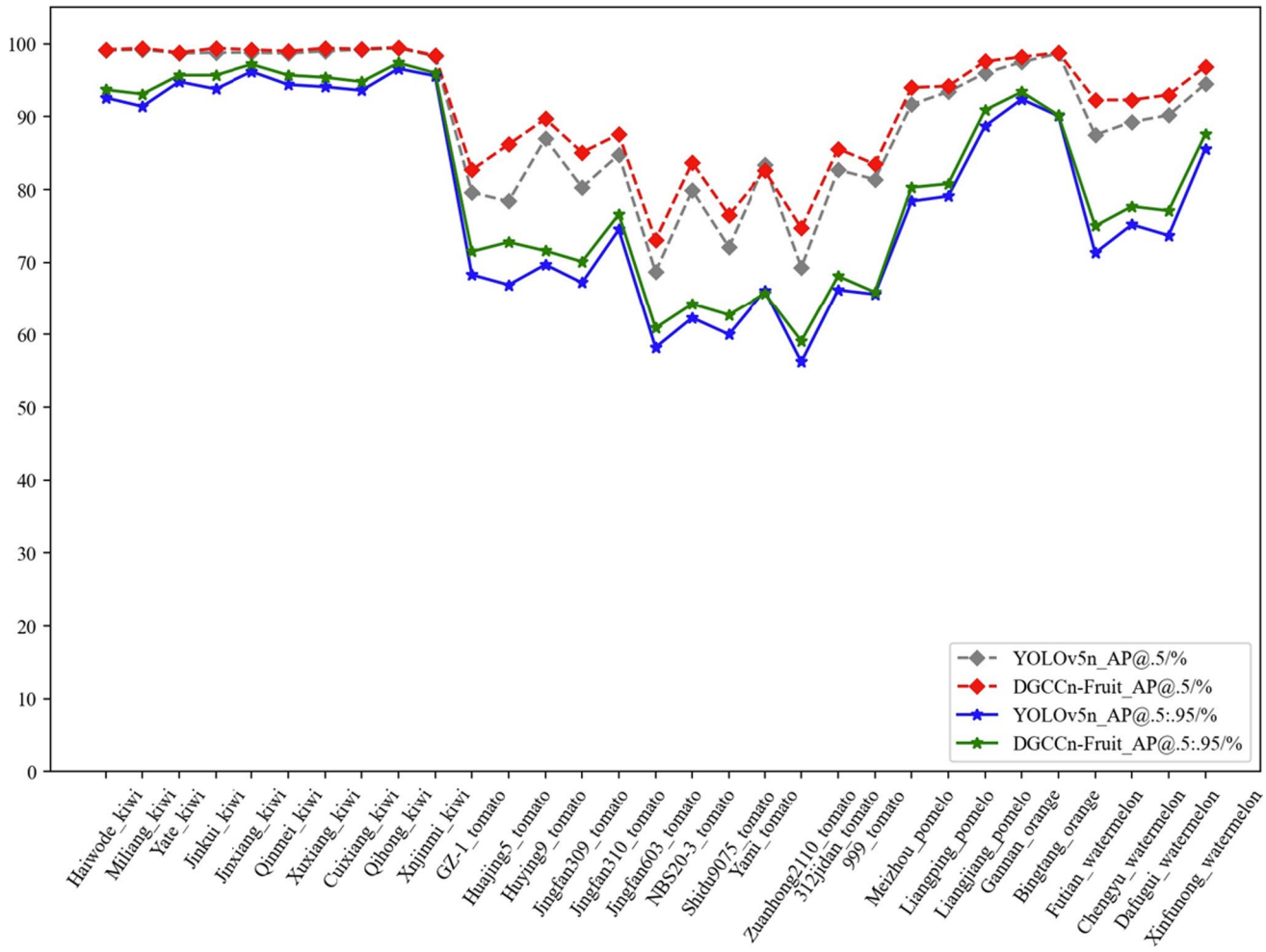

Fig. 12 Comparison of detection accuracy of each category of fruit before and after improvement

including poor lighting conditions, multi-scale objects, close overlapping or occlusion of objects, and mixed categories. We compared the model detection effects before and after improvement, and the experimental results are shown in Fig. 13.

As can be seen from Fig. 13, when the fruits are in poor light conditions, the algorithm models before and after the improvement can correctly predict the position and category, but the detection accuracy of our algorithm is higher; when there are multi-scale fruit objects, due to the great similarity between fine-grained fruit varieties, YOLOv5n has a false detection between Huajing5 tomato and 312 egg tomato, but our algorithm can correctly recognize; when the fruit objects are closely overlapped, YOLOv5n has missed the detection of small objects at the overlap, our algorithm can predict better; when the fruit objects are occluded, the detection boxes of YOLOv5n are significantly larger than the ground truth, but our algorithm is more accurate in positioning; when there are multiple fruit objects mixed, both algorithms can predict different fruit objects, but DGCC$_n$-Fruit has higher detection accuracy. It shows that by integrating the GC module for feature extraction, introducing the CARAFE operator for feature fusion, and using the knowledge distillation for model optimization, our algorithm better improves the localization and recognition accuracy of the model for fine-grained fruits in different scenarios.

Proposed method versus other models

In order to objectively evaluate the algorithm in this paper, we conducted the improvement of YOLOv5 models of different scales and compared them with other mainstream algorithms (two-stage Faster-RCNN, single-stage SSD, YOLOv4 and the latest U version of YOLOv3, YOLOX series and YOLOv7-tiny) on the self-made fine-grained fruit dataset *ZFruit*. The experimental results are shown in Table 5. For clearer presentation, the italicized part of the table indicates the performance of the improved model optimized without knowledge distillation, and the bolded part indicates the performance of the final proposed model. The visual comparison is shown in Fig. 14.

According to the experimental results, our improved strategy has good performance in the detection and recognition of fine-grained fruit images for YOLOv5 models of different scales. Compared with the original model, it greatly reduces the number of network parameters and model volume while improving the average detection accuracy mAP@0.5 and mAP@0.5:0.95. According to the comparison between the *DGCC-Fruit* series models and other mainstream algorithms, the proposed model has better detection accuracy with smaller network parameters and model volume. Among them, DGCC$_n$-Fruit has the smallest parameter quantity and model volume, but its accuracy far exceeds that of advanced network models such as Faster-RCNN, SSD, YOLOX-nano, YOLOX-tiny, YOLOv4-tiny, YOLOv3-tiny, YOLOX-s and YOLOv7-tiny. In terms of model size, DGCC$_n$-Fruit is only 3.37 MB, which is about 1/33 of Faster-RCNN, 1/60 of SSD,

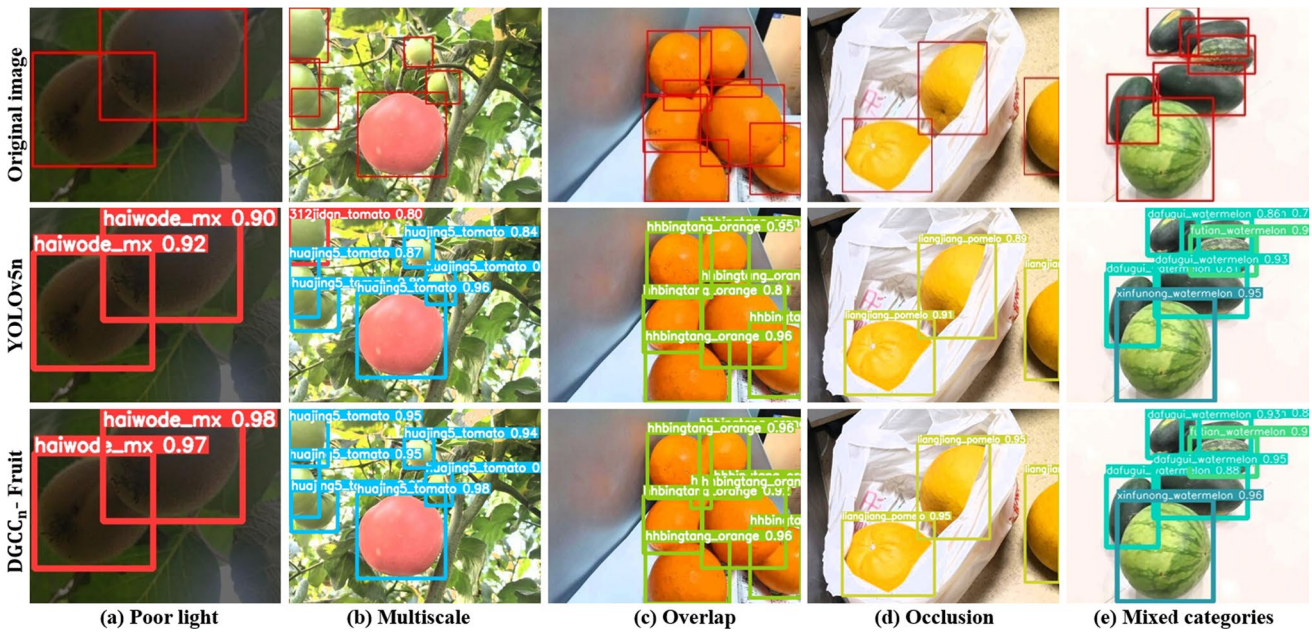

Fig. 13 Comparison of model detection results before and after improvement

Table 5 Comparison results with other models

Model (backbone)	Input	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	Weights (MB)
YOLOv5x (CSPDarknet)	640	93.7	85.8	86.38	165.48
GCC_x-Fruit	*640*	*94.3*	*86.6*	*79.44*	*152.34*
YOLOv5l	640	93.4	85.7	46.30	88.85
GCC_l-Fruit	*640*	*93.8*	*86.0*	*41.85*	*80.52*
$DGCC_l$-Fruit	**640**	**95.1**	**87.2**	**41.85**	**80.52**
YOLOv5m	640	92.9	84.7	20.99	40.48
GCC_m-Fruit	*640*	*93.9*	*85.8*	*18.51*	*35.87*
$DGCC_m$-Fruit	**640**	**94.9**	**86.6**	**18.51**	**35.87**
YOLOv5s	640	92.4	83.1	7.10	13.90
GCC_s-Fruit	*640*	*93.0*	*83.8*	*6.02*	*11.93*
$DGCC_s$-Fruit	**640**	**93.6**	**84.3**	**6.02**	**11.93**
YOLOv5n	640	89.5	79.3	1.80	3.78
GCC_n-Fruit	*640*	*91.3*	*80.9*	*1.55*	*3.37*
$DGCC_n$-Fruit	**640**	**91.6**	**81.2**	**1.55**	**3.37**
YOLOv4 (CSPDarknet)	640	92.8	84.4	64.10	245.00
YOLOv3 (Darknet53)	640	92.3	83.7	61.69	118.11
YOLOv7-tiny	640	91.3	80.9	6.09	11.86
YOLOX-s (CSPDarknet)	640	82.6	74.7	8.95	68.58
YOLOv3-tiny	640	85.2	72.9	8.74	16.76
YOLOv4-tiny	640	81.1	71.3	5.94	22.74
YOLOX-tiny	416	78.0	68.9	5.04	38.77
YOLOX-nano	416	73.2	62.5	0.90	7.29
SSD (VGG16)	300	77.2	61.7	27.76	211.81
Faster-RCNN (Resnet50)	640	80.0	60.0	41.22	109.30

1/2 of YOLOX-nano, 1/11 of YOLOX-tiny, 1/7 of YOLOv4-tiny, 1/5 of YOLOv3-tiny, 1/20 of YOLOX-s and 1/3 of YOLOv7-tiny. When we compare the larger-scale *DGCC-Fruit*, we can see that compared with the YOLOv3, $DGCC_s$-Fruit can reduce the number of parameters by about 90%, while mAP@0.5 and mAP@0.5:0.95 are increased by 1.3% and 0.6% respectively; compared with YOLOv4, $DGCC_m$-Fruit can reduce the number of parameters by about 71%, while mAP@0.5 and mAP@0.5:0.95 are increased by 2.1% and 2.2% respectively. It shows that compared with the original model and the current mainstream detection algorithm models, our *DGCC-Fruit* not only has better detection performance but also has smaller parameters and model volume, which greatly reduces the storage cost and is more suitable for small embedded terminals.

In addition, to further verify the effectiveness of the proposed model, Frame Per Second (FPS) is used as a real-time evaluation to compare with the existing model, and the comparison results are shown in Fig. 15. $DGCC_n$-Fruit compared to Faster-RCNN, SSD, YOLOX-nano, YOLOX-tiny, YOLOv4-tiny, YOLOX-s and Yolov7-tiny, and $DGCC_m$-Fruit compared to YOLOv4, they still have certain advantages in inference speed, but $DGCC_n$-Fruit compared to YOLOv3-tiny and $DGCC_s$-Fruit compared to YOLOv3, they are slightly lower in inference speed. Overall, the proposed model meets the requirements of real-time applications.

Combined with the above comparison, it can be found that $DGCC_{n\backslash s\backslash m\backslash l}$-Fruit models of different scales improve the detection accuracy by gradually increasing the depth and width of the network, but also with the increase of computing cost, memory space and reasoning time. $DGCC_n$-Fruit model is the smallest and the detection accuracy is relatively low, but it is fast and suitable for the application scenarios with limited computing power and storage space or high requirements for detection speed. $DGCC_l$-Fruit model has the highest detection accuracy, but it is slow and suitable for the scenarios with large computing power and storage space or high requirement for accuracy. $DGCC_{s\backslash m}$-Fruit achieves a balance between speed and accuracy, and is suitable for applications with high requirements on both speed and accuracy.

Generality comparison

In order to test the generality of the proposed model and its ability to deal with other complex scenarios, we conducted the improvement of YOLOv5 models of different scales and compared them with other mainstream algorithms on the

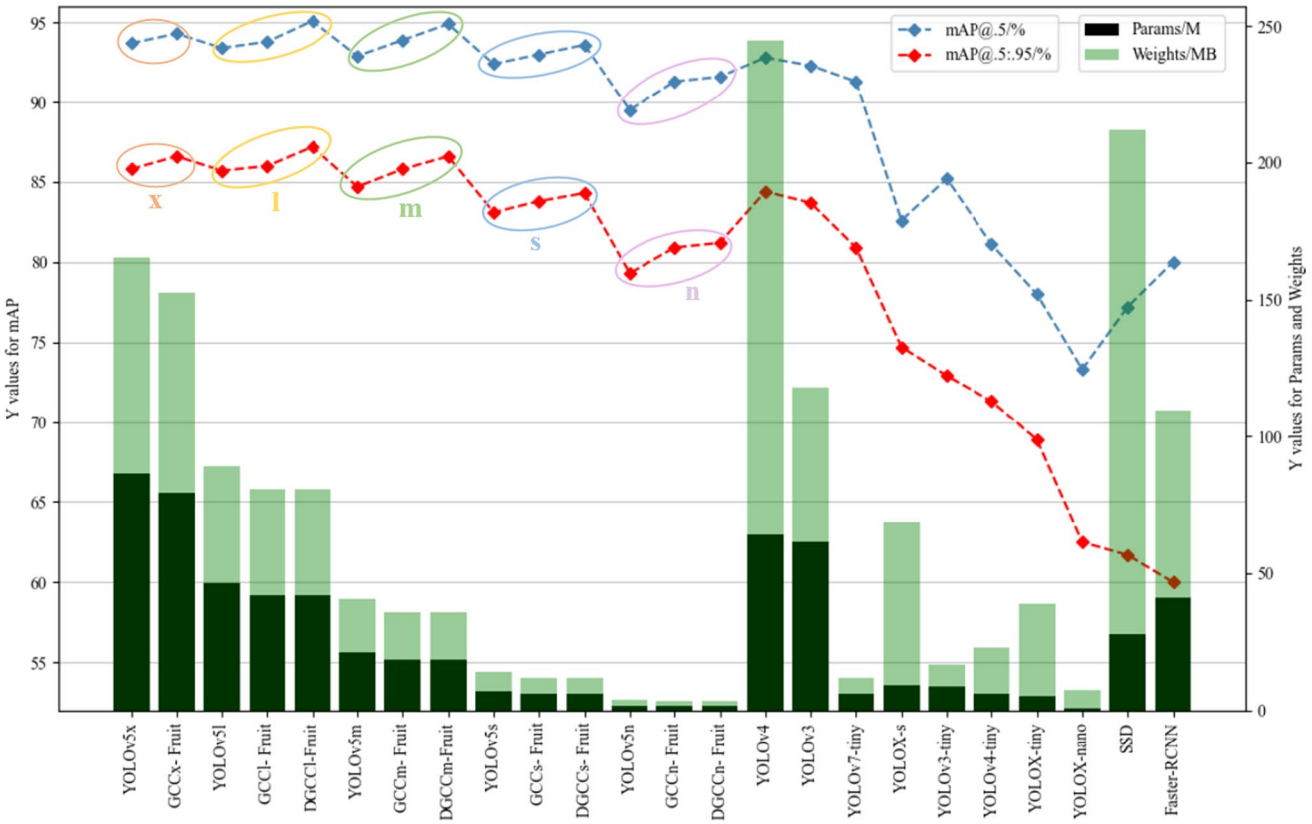

Fig. 14 Comparison of experimental results

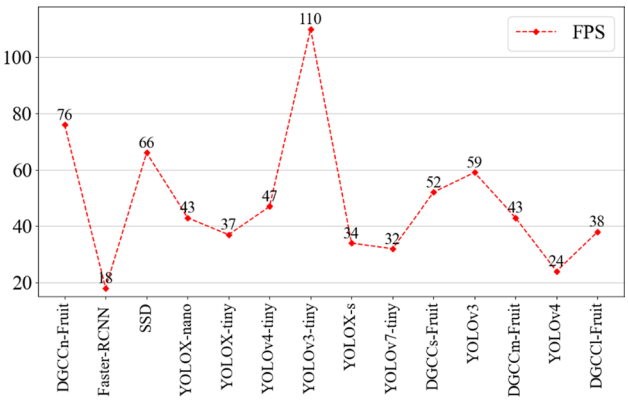

Fig. 15 Comparison of inference speed

object detection public dataset Pascal VOC 2007. The experimental results are shown in Table 6, and the italicized and bolded parts of the table indicate the same meaning as the previous table. The visual comparison is shown in Fig. 16.

According to the experimental results, the improvement strategy proposed in this paper has a good performance on the object detection public dataset *Pascal VOC 2007* for the improvement of YOLOv5 models of different scales and its comparison with other mainstream algorithms, which greatly reduces the number of network parameters and model volume while improving the average detection accuracy mAP@0.5 and mAP@0.5:0.95. Among them, *DGCC$_n$-Fruit* has the smallest parameter quantity and model volume. Compared with the original model, the number of parameters is reduced by about 15%, and the model volume is reduced by about 11%. At the same time, mAP@0.5 and mAP@0.5:0.95 are increased by 5.4% and 5.5% respectively. The detection accuracy far exceeds that of advanced network models such as YOLOX-tiny, YOLOX-nano, YOLOv4-tiny, and YOLOv3-tiny. When we compare the larger-scale *DGCC-Fruit*, we can see that DGCC$_s$-Fruit compared with YOLOv7-tiny, YOLOX-s, Faster-RCNN and SSD, and DGCC$_m$-Fruit compared with YOLOv4 and YOLOv3, both of which have better detection accuracy when the network parameters and model volume are lower. The experimental results show that compared with the original algorithm model and the current mainstream detection algorithm models, the *DGCC-Fruit* model proposed in this paper also has better performance and advantages on public dataset *Pascal VOC 2007*, and has better generality.

Table 6 Experimental results of generality comparison

Model (backbone)	Input	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	Weights (MB)
YOLOv5x (CSPDarknet)	640	76.8	54.3	86.30	165.28
GCC_x-Fruit	*640*	*77.4*	*54.9*	*79.36*	*152.15*
YOLOv5l	640	74.9	51.8	46.21	88.70
GCC_l-Fruit	*640*	*75.3*	*52.1*	*41.79*	*80.37*
$DGCC_l$-Fruit	**640**	**78.5**	**55.1**	**41.79**	**80.37**
YOLOv5m	640	72.6	48.6	20.93	40.36
GCC_m-Fruit	*640*	*73.4*	*49.1*	*18.47*	*35.76*
$DGCC_m$-Fruit	**640**	**77.2**	**53.3**	**18.47**	**35.76**
YOLOv5s	640	67.5	41.0	7.06	13.81
GCC_s-Fruit	*640*	*70.1*	*44.2*	*5.99*	*11.85*
$DGCC_s$-Fruit	**640**	**73.8**	**48.4**	**5.99**	**11.85**
YOLOv5n	640	62.7	35.3	1.79	3.72
GCC_n-Fruit	*640*	*64.2*	*37.7*	*1.53*	*3.31*
$DGCC_n$-Fruit	**640**	**68.1**	**40.8**	**1.53**	**3.31**
YOLOv4 (CSPDarknet)	640	74.3	51.7	64.04	244.77
YOLOv3 (Darknet53)	640	74.5	49.0	61.63	117.94
YOLOv7-tiny	640	70.6	44.4	6.06	11.79
YOLOX-s (CSPDarknet)	640	70.3	44.1	8.95	68.58
Faster-RCNN (Resnet50)	640	72.4	38.4	41.22	108.87
SSD (VGG16)	300	69.2	38.3	27.76	200.09
YOLOX-tiny	416	61.4	35.1	5.04	38.75
YOLOX-nano	416	52.4	29.9	0.90	7.27
YOLOv4-tiny	640	53.6	25.9	5.92	22.64
YOLOv3-tiny	640	48.2	21.4	8.71	16.70

Discussion

In this paper, YOLOv5 is improved in three models of feature extraction, feature fusion, and model optimization to make it more suitable for application in fine-grained fruit detection and recognition. In first model, to enhance the ability in extracting fine-grained fruit features and reducing background interference, we integrated the CA attention mechanism in feature extraction stage to improve the discrimination of similar features. Through the decomposition and aggregation of input feature map in spatial directions of width and height, the position information is embedded into the channel attention to capture the cross-channel and direction-aware information at the same time, which can help to locate and recognize key areas more accurately in different environments. Then in feature fusion, compared with the original interpolation upsampling, the introduced CARAFE upsampling operator can generate the upsampling kernel adaptively based on the input fine-grained features to make full use of the deep network to extract the semantic information of fine-grained fruit features, and then the multiplicative weighting calculation is used to better integrate the multi-scale fine-grained fruit features and enhance the characterization ability. Last, in model optimization, aiming at the increasing parameters caused by the previous two methods, GhostBottleneck module is introduced by using 1×1 convolution, depthwise convolution, and a series of cheap linear transformations to obtain similar features at a small cost, which greatly reduces the parameters number, model size of CNNs and maintain lightweight. To further improve the detection accuracy while maintaining lightweight, knowledge distillation strategy is adopted to obtain knowledge from complex models, and effectively improve the detection performance without increasing the model size.

In order to adapt to different application requirements and to verify the effectiveness of the proposed method, we conducted experiments on different scale models based on the homemade *ZFruit* and *VOC2007* datasets, and compared them with the current mainstream algorithms. It can be observed that the improvement on the homemade *ZFruit* dataset is smaller than public dataset due to the similarity restriction of fine-grained fruit features, the light model may have a slight degradation on inference speed due to the model branch structure. In addition, the richness of the dataset scenarios also affects the applicability of the model. Overall, the experimental results show that *DGCC-Fruit* can greatly reduce the parameters number and model size while improving feature extraction capability, has better

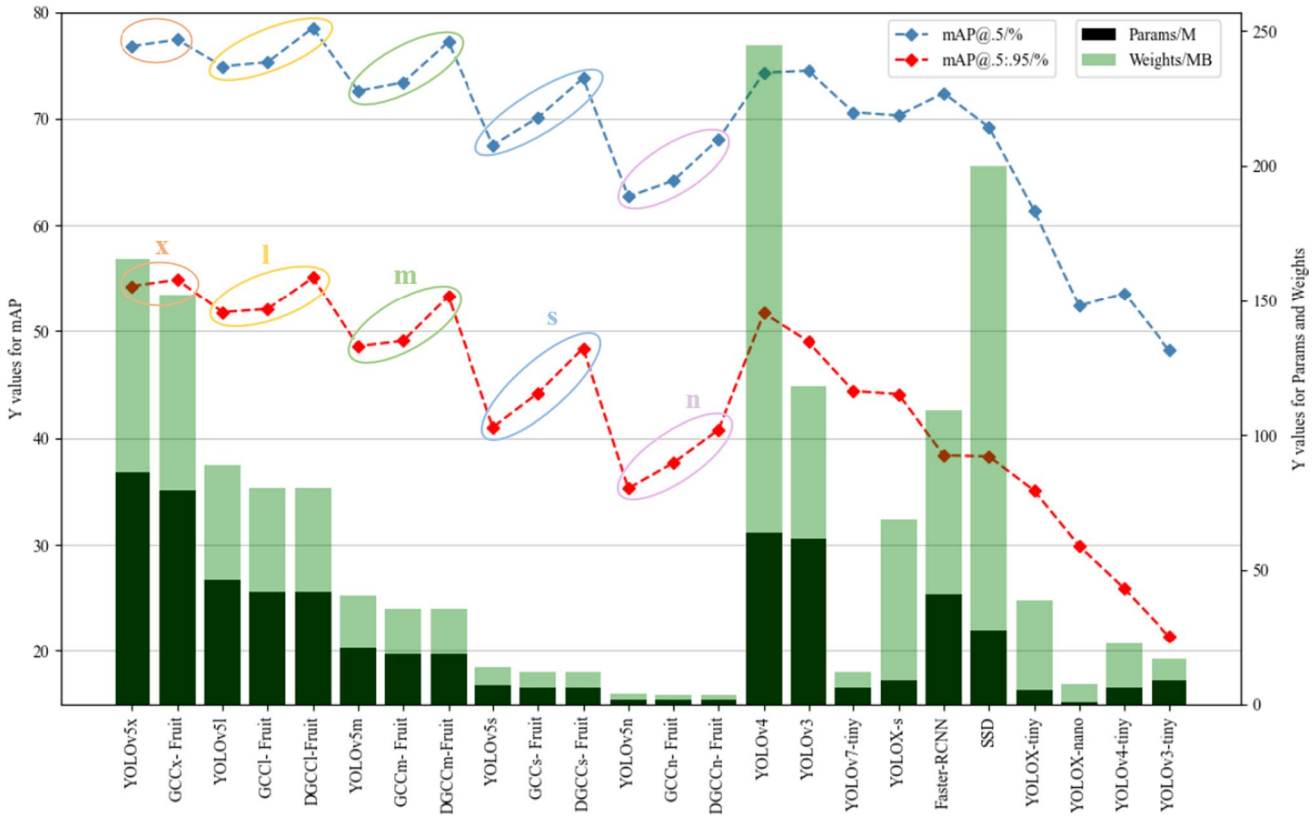

Fig. 16 Experimental results of generality comparison

localization and recognition performance for fine-grained fruits, meet the real-time application with certain generalizability, which can further promote the precise management of smart orchard, intelligent sorting and selling of fruits in supermarket, daily science popularization, identification, and intellectual property protection of seed industry.

Conclusion

By taking the self-made fine-grained fruit dataset *ZFruit* as the main research object, and the YOLOv5 network as the basic network, we construct a lightweight and fine-grained fruit recognition model *DGCC-Fruit*. The experimental results show that *DGCC-Fruit* has significantly reduced model parameters, increased the feature extraction ability, improved the localization and recognition performance of fine-grained fruit objects in different environments, and has better accuracy and robustness. Although our framework sets a new state of the art for fine-grained fruit image detection and recognition with high portability, performs better than the current advanced object detection algorithms on the self-made *ZFruit* and *VOC2007* datasets, due to the small model size, the detection accuracy of $DGCC_n$-Fruit is slightly lower than that of large networks such as YOLOv4 and YOLOv3. In future research, we can add fruit types and sub-categories to expand the dataset to improve the generality of the model. We can also better balance the speed and accuracy by improving the feature fusion network, loss function, pruning, quantization, etc., and deploy our model on real scenarios.

Acknowledgements Research supported by Foundation of Key Research and Development Program of Shaanxi province (2020NY-205, 2021NY-179, 2023-YBNY-229), Undergraduate Training Program for Innovation and entrepreneurship plan (X202110712259, S202110712613).

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. H. Li, A visual recognition and path planning method for intelligent fruit-picking robots. Sci. Program. (2022). https://doi.org/10.1155/2022/1297274

2. Y. Tang, M. Chen, C. Wang et al., Recognition and localization methods for vision-based fruit picking robots: a review. Front. Plant Sci. **11**, 510 (2020)
3. W. Jia, Y. Zhang, J. Lian et al., Apple harvesting robot under information technology: a review. Int. J. Adv. Rob. Syst. **17**(3), 1729881420925310 (2020)
4. S. Nuske, et al., Yield estimation in vineyards by visual grape detection, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, Piscataway, 2011)
5. L. Luo, Y. Tang, X. Zou et al., Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components. Sensors **16**(12), 2098 (2016)
6. G. Lin, Y. Tang, X. Zou et al., Fruit detection in natural environment using partial shape matching and probabilistic Hough transform. Precis. Agric. **21**(1), 160–177 (2020)
7. R. Girshick, J. Donahue, T. Darrell, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
8. K. He, X. Zhang, S. Ren et al., Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015)
9. R. Girshick, Fast r-cnn, in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448
10. S. Ren, K. He, R. Girshick et al., Faster r-cnn: towards real-time object detection with region proposal networks. Adv. Neural. Inf. Process. Syst. **28**, 1–9 (2015)
11. J. Redmon, S. Divvala, R. Girshick, et al., You only look once: unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
12. W. Liu, D. Anguelov, D. Erhan, et al., SSD: single shot multibox detector, in *European Conference on Computer Vision* (Springer, Cham, 2016), pp. 21–37
13. T.Y. Lin, P. Goyal, R. Girshick, et al., Focal loss for dense object detection, in *Proceedings of the IEEE international Conference on Computer Vision*, 2017, pp. 2980–2988.
14. S. Bargoti, J. Underwood, *Deep Fruit Detection in Orchards* (IEEE, Piscataway, 2016)
15. P. Borianne, F. Borne, J. Sarron, et al., Deep mangoes: from fruit detection to cultivar identification in colour images of mango trees (2019). Preprint at arXiv:1909.10939
16. Z. Zhou, Z. Song, L. Fu et al., Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. Comput. Electron. Agric. **179**, 105856 (2020)
17. Y. Tian, G. Yang, Z. Wang et al., Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agric. **157**, 417–426 (2019)
18. A. Koirala, K.B. Walsh, Z. Wang et al., Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO.' Precis. Agric. **20**, 1107–1135 (2019)
19. W. Chen, S. Lu, B. Liu et al., Detecting citrus in orchard environment by using improved YOLOv4. Sci. Program. **2020**(1), 1–13 (2020)
20. R. Yang, Y. Hu, Y. Yao et al., Fruit target detection based on BCo-YOLOv5 model. Mob. Inf. Syst. **2022**, 1–8 (2022)
21. X. Wang, Z. Wu, M. Jia et al., Lightweight SM-YOLOv5 tomato fruit detection algorithm for plant factory. Sensors **23**(6), 3336 (2023)
22. Z. Li, X. Zhang, X. Feng, et al., Detection method of apple based on improved lightweight YOLOv5*, in Cognitive systems and information processing: 6th international conference, ICCSIP 2021*, Suzhou, China, November 20–21, 2021, Revised Selected Papers 6 (Springer, Singapore, 2022), pp. 286–294
23. T. Zhang, F. Wu, M. Wang et al., Grape-bunch identification and location of picking points on occluded fruit axis based on YOLOv5-GAP. Horticulturae **9**(4), 498 (2023)
24. Y. Lai, R. Ma, Y. Chen et al., A pineapple target detection method in a field environment based on improved YOLOv7. Appl. Sci. **13**(4), 2691 (2023)
25. K. Han, Y. Wang, Q. Tian, et al., Ghostnet: more features from cheap operations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589
26. M. Sandler, A. Howard, M. Zhu, et al., MobileNetV2: inverted residuals and linear bottlenecks, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018
27. F. Chollet, Xception: deep learning with depthwise separable convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258
28. Q. Hou, D. Zhou, J. Feng, Coordinate attention for efficient mobile network design, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13713–13722
29. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141
30. S. Woo, J. Park, J.Y. Lee, et al., Cbam: convolutional block attention module, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19
31. J. Wang, K. Chen, R. Xu, et al., Carafe: content-aware reassembly of features, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3007–3016
32. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2(7) (2015). Preprint at arXiv:1503.02531
33. R. Mehta, C. Ozturk, Object detection at 200 frames per second, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018
34. S.I. Mirzadeh, M. Farajtabar, A. Li, et al., Improved knowledge distillation via teacher assistant, in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34(04), 2020, pp. 5191–5198