Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-16570-9

Fruit ripeness identification using YOLOv8 model

Bingjie Xiao[1] · Minh Nguyen[1] · Wei Qi Yan[1]

Received: 26 February 2023 / Revised: 11 June 2023 / Accepted: 18 August 2023

Abstract
Deep learning-based visual object detection is a fundamental aspect of computer vision. These models not only locate and classify multiple objects within an image, but they also identify bounding boxes. The focus of this paper's research work is to classify fruits as ripe or overripe using digital images. Our proposed model extracts visual features from fruit images and analyzes fruit peel characteristics to predict the fruit's class. We utilize our own datasets to train two "anchor-free" models: YOLOv8 and CenterNet, aiming to produce accurate predictions. The CenterNet network primarily incorporates ResNet-50 and employs the deconvolution module DeConv for feature map upsampling. The final three branches of convolutional neural networks are applied to predict the heatmap. The YOLOv8 model leverages CSP and C2f modules for lightweight processing. After analyzing and comparing the two models, we found that the C2f module of the YOLOv8 model significantly enhances classification results, achieving an impressive accuracy rate of 99.5%.

Keywords YOLOv8 · CenterNet · Visual object detection

1 Introduction

In the field of computer vision, identifying regions of interest (ROI) in digital images is a fundamental task, often taking precedence over other problems, particularly in fruit image classification. Identifying the ROI is crucial for detecting and recognizing visual objects on a screen.

Deep learning models for visual object detection are primarily divided into two categories: one-stage models and two-stage models. Two-stage models generate a preselected box, known as a region proposal (RP), which potentially contains an object to be detected. These models then classify the given samples using convolutional neural networks. In contrast, one-stage object detection models bypass the use of RP, directly extracting visual features to predict the object class and location. Examples of two-stage models include R-CNN, SPPNet, Fast R-CNN, Faster R-CNN [21], etc., while one-stage

✉ Bingjie Xiao
bingjie.xiao@autuni.ac.nz

[1] Auckland University of Technology, Auckland 1010, New Zealand

Published online: 31 August 2023

models include YOLO (You Only Look Once) [31, 36], SSD, CenterNet, etc [13, 20, 22, 41].

The motivation for this article stems from the shortage of human labor in orchards during the harvest season. This critical task requires completion within a very short time frame, thus necessitating machine vision and robots for automated picking.

The primary objective of this study is to train a fruit detection model capable of identifying the location of a fruit within a given image and distinguishing the fruit's class. The classes, labelled based on the visual characteristics of the fruit skin, include: "Ripe Apple", "Overripe Apple", "Ripe Pear", and "Overripe Pear".

Despite architectural differences between one-stage and two-stage object detection models, their training methods remain similar. Visual object detection primarily involves two phases: the training process and the testing process. The main goal of model training is to use an image dataset to derive parameters for the detection network. This training dataset includes annotated information such as the object location and class.

Figure 1 illustrates how the CenterNet model is trained based on the provided training data and how it produces prediction results. The rectangular box indicates the location of the visual object that we manually marked, along with the corresponding visual object classes. Thus, both the images and the annotation tags serve as inputs for the CenterNet model training. Once the training process concludes, the model outputs the predicted results.

The contributions of this article:

(1) In this paper, we utilize the YOLOv8 model for fruit detection. Additionally, we employ a transfer learning model, achieving an impressive accuracy rate of 99.5% and precisely classifying the ripeness of various fruits.

(2) We have created our own dataset, incorporating factors such as fruit occlusion, overlapping, and mixed datasets with various classes.

In the first part of this article, we introduce the background and objectives of the experiment. In the second part, we introduce past research work for visual object detection based on YOLO model and CenterNet. In the third part, we will introduce YOLOv8 model and the CenterNet model in detail. In the fourth part, we analyze the experimental results. Finally, we summarize our contributions of this paper and envision future work.

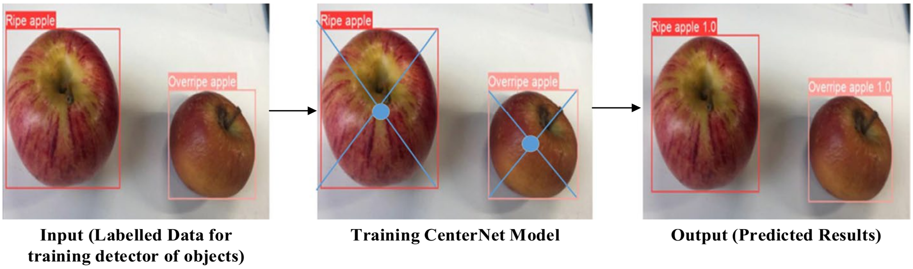


Fig. 1 The progress of object detection (one-stage model)

2 Literature review

2.1 Fruit detectio n

Fruit detection from digital videos is based on computer vision and deep learning methods at present [39, 41], which requires the use of computers and digital cameras instead of manual operations [5, 7, 10, 11, 17, 28]. There are a lot of methods for fruit object detection [2, 4, 15, 18, 33]. The fruit detection essentially needs both classification and localization by providing the class labels and bounding box coordinates of the targets [8, 14, 16, 26, 27]. Visual object detection is to use YOLOv8 model or CenterNet model and dataset composed of ground truths, so that the model can extract visual features of the fruits, and then output the predicted results.

The two-stage models can output better recognition results, but the one-stage models are able to achieve faster detection [37]. Faster R-CNN is a typical two-stage model. The two-stage model is a proposal-based method, which needs to use selective search to generate a region proposal, and conduct object classification and bounding box regression. Faster R-CNN has high accuracy, but with slow speed. In our previous experiments, Faster R-CNN model combined with ResNet-50 model achieved a precision 93%, while YOLOv3 model combined with Darknet model achieved a precision 99.96% [34]. Wan and Goudos [29] improved the convolutional layer and pooling layer of Faster R-CNN network to increase the speed of visual object detection and obtained a mean average precision 86.41%. At the same time, it achieved 84.89% precision with YOLOv3 model. RGB colors were utilized as visual feature. Sa, et. al. [23] explored multiple modalities, which inspired us on how to extract the features of visual objects with multiple bits. Sa, et al. conducted transfer learning based on Faster R-CNN and achieved the precision and recall 0.807 to 0.838 for detecting sweet peppers, respectively.

FDR model [12] was developed to deal with the problems that fruit identification. The FDR model can better overcome the complexity caused by the overlapping of fruits under a specific dataset. At the same time, the baseline of the convolutional neural network was improved based on classification and recognition, the model can reduce the impact of background noises and resolutions based on the given dataset. This method achieved an accuracy rate 97.83%.

In the fruit detection methods, Transformer models can also achieve satisfactory detection results. Sun, et al. solved the impact of the complex environment of real orchards on detection through the focal bottleneck changer module [24]. The focus changer block took a focus changer layer and embedded it into the original bottleneck architecture through replacing the spatial convolution layer with a focus changer layer. The focus changer block is a solution for the similarity between the green apple peel in an orchard environment and the green background environment of the leaves in an orchard. In the experiment, window-based focal multi-head self-attention is embedded into the focal transformer layer, which can filter the noise of the orchard environment background and enhance the local features of green apples. Sun's work attained 34.2% accuracy based on the Pascal VOC dataset.

Swin Transformer model was treated as a basis, combined with Mask R-CNN to solve the impact of the natural environment on detection [35]. The model can effectively identify the size and types of tomatoes and achieved an accuracy rate 89.4%. DenseNet-169, ResNet-50v2 and Vision Transformer models were developed to detect plant diseases and achieved an accuracy rate 99.88% [1]. The loss function as sparse categorical cross entropy can achieve multi-classification problems.

2.2 CanterNet model

CenterNet model is characterized by key point detection. Key point detection means that the entire detected object is modeled as a point. As shown in Fig. 1, the dot is the center point of the bounding box. The CenterNet model is to locate the center point of the detected target. The structure of Residual Network (ResNet) can enable the learning ability to increase with the increase of network depth. The CenterNet model combined with the ResNet network is usually employed to achieve visual object detection [9].

ResNet [38] was embedded into the backbone of CenterNet and attained an accuracy rate 78.6%. The loss function of ResNet can assist CenterNet to better complete the target detection task. CenterNet can utilize a grid of feature maps at the center of the object in current 2D object detection. However, in practical applications, 3D object detection is prone to ambiguity in the size and direction of the detected object, resulting in misjudgment of global information. 3D-CenterNet processes the parameters of the bounding box so that it can accurately estimate the position of center point and aid the model to identify the local features of the target [30].

2.3 YOLO models

YOLO models take advantage of the entire given image as the input [25, 40, 42], and directly regress the position and the bounding box at the output layer [3, 6]. YOLO models segment the input image into s × s grids, predict the boundary of each grid, and analyze whether each border is the position and confidence of the detected object.

Liu et al. [19] probed the method of detecting pineapples based on YOLOv3 model. In the work, Darknet53 was introduced as the backbone of YOLOv3 model. DenseNet was added to the backbone of the Darknet model to enhance the representation ability of feature maps. The improved YOLOv3 model can complete the disparity calculation between the ROI and other regions, achieved an average precision 97.55%.

Wang, et al. [32] took use of YOLOv5 model to probe apple stems for product packaging automation. YOLOv5 is use of the training method of transfer learning to obtain better detection performance. By comparing the number of detection heads and feature map size, layer pruning, and channel pruning to optimize YOLO-v5s, the complexity of the model was further reduced, the model parameters and weight volume were reduced about 71%, the mAP was only allieved by 1.57%. The optimized algorithm achieved 93.89% accuracy in stem/calyx detection of a variety of apples.

3 Methodology

3.1 One-stage and two-stage models

Visual object detection algorithms are grouped into two cagtegories: One-stage and two-stage. Two-stage algorithms usually generate region proposals, and classify each candidate box. The two-stage algorithm requires multiple detection and classification processes, so the algorithm is relatively slow.

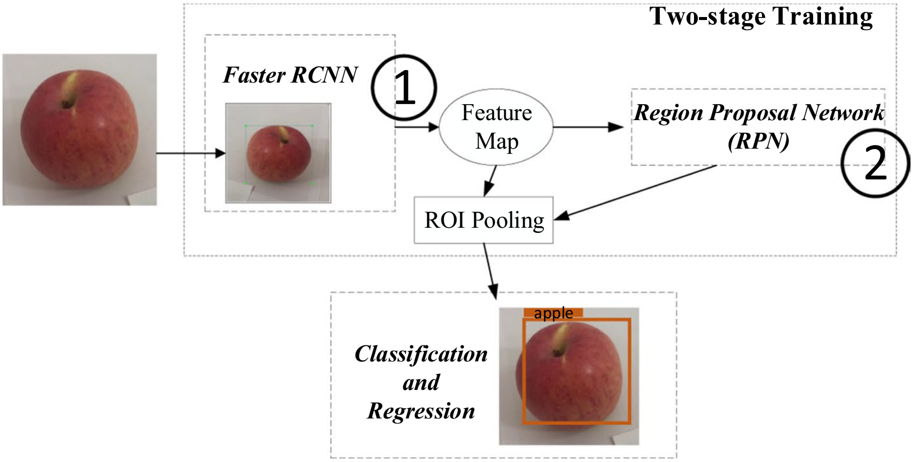


Fig. 2 The sample of two-stage model

In Fig. 2, the anchor box is a sliding window that traverses the image which obtains feature map. In the two-stage algorithm, c_1, c_2, c_3, c_4 represents ripe apple, overripe apple, ripe pear, and overripe pear, respectively. (x, y, w, h) shows the position of one of the corners of the bounding box. T in Eq. (1) indicates an anchor box in two-stage model.

$$T = (x, y, w, h, c_1, c_2, c_3, c_4) \tag{1}$$

Similar to the one-stage model in Fig. 4, the predictive model may produce multiple bounding boxes, the boxes are represented by (A, B, C, D). The two-stage model calculates the bounding boxes that regress to the ground truth iteratively. The regression equation is,

$$R_1(A_1, B_1, C_1, D_1) \to R_i(A_i, B_i, C_i, D_i) \to \cdots \to R_{groundtruth}(A_1, B_1, C_1, D_1) \tag{2}$$

The one-stage object detection algorithm usually sends the images to the network model once and can generate all the bounding boxes, so it is fast and very suitable for real-time detection. Thus, whether the model can achieve rapid object detection is also within the scope of our evaluations. Both CenterNet and YOLOv8 models are typical one-stage algorithms.

3.2 CenterNet & ResNet-50

CenterNet object detection is based on bounding box of the identified object as a center point, and returns to other object attributes based on this center point. As shown in Fig. 3, CenterNet is an end-to-end one-stage object detection model. In Fig. 3, the CenterNet prediction module contains three branches, namely the prediction of heatmap of the center point, the prediction of offset, and the prediction of object size. The heatmap contains C channels, and each channel contains a class. The blue shaded part in Fig. 4 indicates the center point of the target region.

As shown in Fig. 4, if the bounding box is accurate, the probability of blue center point that can be detected will be high. If the bounding box in the orange area is inaccurate, the

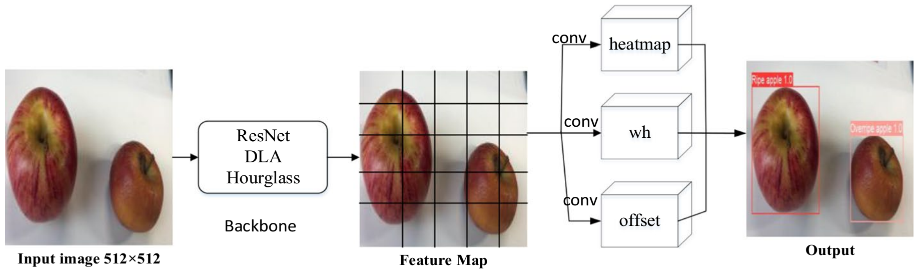


Fig. 3 The flowchart of CenterNet model

Fig. 4 CenterNet bounding box

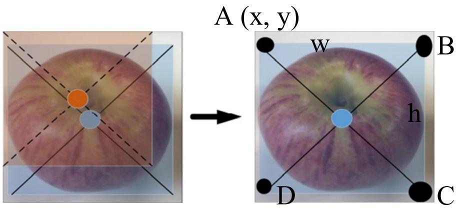
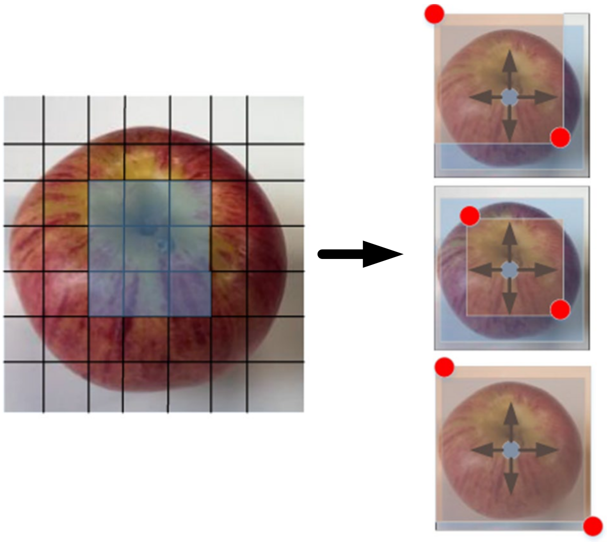


Fig. 5 Heatmap ground truth



probability of the detected orange center point is low. Therefore, the bounding box marked by the upper left and lower right corner points defines a central area, and CenterNet model detects the center point in the central area of each box in Fig. 3. In Fig. 4, the blue center points and boxes with high probability will be retained, while the orange center points and boxes will be deleted.

CenterNet algorithm is implemented by using heatmap. While the CanterNet network predicts the center point, which presents a Gaussian distribution. In Fig. 5, we are use of a grid to model the center point. The blue box is ground truth, and the orange box is the predicted box. There are three situations between the prediction box and the real ground truth box, the ground truth box and the prediction box overlap, the ground truth box contains the prediction box, the prediction box includes the ground truth box.

The loss function of the entire CenterNet consists of three branches of prediction modules. L_{dat} represents the loss function. L_k shows the loss heatmap center point. L_{off} indicates the loss of object center point offset. L_{size} displays the loss of length and width. The prediction loss function is presented as following,

$$\mathrm{L_{dat}} = \mathrm{L_k} + \lambda_{\mathrm{size}}\mathrm{L_{size}} + \lambda_{\mathrm{off}}\mathrm{L_{off}}(\lambda_{\mathrm{size}} = 0.1, \lambda_{\mathrm{off}} = 1) \tag{3}$$

where $\mathrm{Y_{xyc}}$ indicates the ground truth value. $\overset{\vee}{Y}_{xyc}$ is the ground truth value.

$$L_K = \frac{-1}{N}\sum\nolimits_{xyc}\begin{cases}(1-\overset{\vee}{Y}_{xyc})^{\alpha}\log(\overset{\vee}{Y}_{xyc}), if\ Y_{xyc}=1\\(1-Y_{xyc})^{\beta}\log(1-\overset{\vee}{Y}_{xyc}),\ otherwise\end{cases} \tag{4}$$

The heatmap loss function in Eq. (4) is improved based on the basis of focal loss, where α and β are two hyperparameters to balance difficult and easy samples, N represents the number of key points.

The resolution of feature map output by CenterNet network is a quarter of the original input image, which will bring a large error. Therefore, the offset center point loss function in Eq. (5) takes use of L_1 loss to calculate the offset loss of the positive sample block, where $\widehat{O}_{\widetilde{p}}$ represents the offset value predicted by the network, p shows the coordinates of the center point of the image, R displays the scaling factor of the heatmap, and $\widetilde{p}$ indicates the approximate integer coordinates of the center point after scaling. Each pixel on the output feature map corresponds to a 4×4 region of the original image.

$$\mathrm{L_{off}} = \frac{1}{\mathrm{N}}\sum\nolimits_{\mathrm{p}}\left|\widehat{\mathrm{O}}_{\widetilde{\mathrm{p}}} - (\frac{\mathrm{p}}{\mathrm{R}} - \widetilde{\mathrm{p}})\right| \tag{5}$$

$$\mathrm{L_{size}} = \frac{1}{\mathrm{N}}\sum\nolimits_{\mathrm{k}=1}^{\mathrm{N}}\left|\widehat{\mathrm{S}}_{\mathrm{pk}} - \mathrm{s_k}\right| \tag{6}$$

where the length and width loss function are shown in Eq. (6), where N represents the number of key points, s_k shows the real size of the target, $\widehat{S}_{pk}$ indicates the predicted size, and the whole process is calculated by using L_1 loss function.

CanterNet removes the non-maximum suppression module, which enables the algorithm to achieve faster processing speed and higher detection accuracy. We observe in Fig. 3 that CenterNet's backbone is use of a residual network to solve the problem of gradient explosion and gradient disappearance. ResNet includes deformable convolution, increases upsampling, and reduces the number of channels. The ResNet model can increase the size of output feature map and reduce the amount of calculation.

3.3 YOLOv8 model

YOLOv8 is an improvement on the previous version of YOLO, which further improves the performance, makes the model fast, accurate and easy to use. The backbone of YOLOv8 model continues the CSP module of YOLOv5. As shown in Fig. 6, the C2f module is employed to extract visual features. In YOLOv8, we delete the CBS 1×1

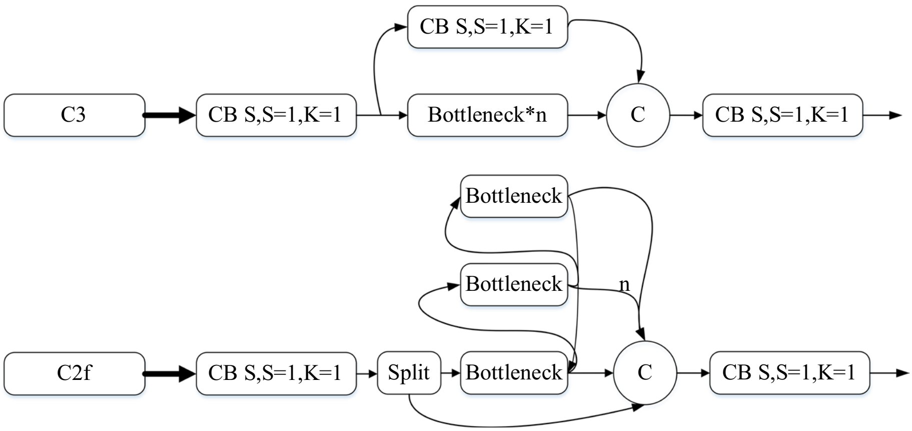


Fig. 6 C2f block module in YOLOv8 model

convolution structure in the PAN-FPN up-sampling stage in YOLOv5, and also replaces the C3 module with the C2f module. Decoupled-head in YOLOv8 is use of two convolutions for classification and regression respectively, and takes advantage of the idea of DFL at the same time.

The most important update of YOLOv8 model is to adopt the anchor-free method and use task alignment learning to align classification (or *cls*) and regression (or *reg*) tasks. Normally aligned anchors should be able to be accurately positioned. YOLOv8 model is use of a new anchor alignment metric. The anchor alignment metric is obtained by multiplying *cls* score and the IOU between the predicted frame and the ground truth real frame. The alignment metric is integrated in the sample allocation and loss function to dynamically optimize the prediction of each anchor. YOLOv8 model takes use of VFL loss as classification loss and DFL loss + CIOU loss as classification loss.

$$\mathrm{VFL(p,q)} = \begin{cases} -q(q\mathrm{log(p)} + (1-\mathrm{q})\mathrm{log}(1-\mathrm{p})), \mathrm{q} > 0 \\ -\alpha \mathrm{p}^{\gamma}\mathrm{log}(1-\mathrm{p}), q = 0 \end{cases} \tag{7}$$

where *VFL* indicates an asymmetric weighting operation based on the imbalance between positive and negative samples, both FL and QFL are symmetrical. As shown in Eq. (7), p is the label, q is the value calculated by using *norm_align_metric* if the positive sample is taken, and $p=0$ if the negative sample is taken. *Norm_align_metric* weighting for highlighting master samples.

DFL (Distribution Focal Loss) changes the single value of coordinate regression to output $n+1$ values, each value represents the probability of the corresponding regression distance, and the integral is calculated to obtain the final regression distance. DFL can make the network focus on the target y faster nearby values, increasing their probability.

$$\mathrm{DFL(S_i, S_{i+1})} = -((\mathrm{y_{i+1}} - \mathrm{y})\mathrm{log(S_i)} + (\mathrm{y} - \mathrm{y_i})\mathrm{log(S_{i+1})}) \tag{8}$$

The meaning of DFL is to optimize the probability of the two positions which is the closest one to the label y, one left and one right, in the form of cross entropy, so that the network can focus on the distribution of adjacent area of the target position faster.

Table 1 Dataset Description

Datasets		Number of apple images	Number of apple labels	Number of pear images	Number of pear labels	Image Size
I	Ripe	144	552	—	—	224×224
	Overripe	111	452	—	—	
	Unripe	92	409	—	—	
II	Ripe	1,325	5,416	—	—	
	Overripe	1,083	4,475	—	—	
	Unripe	1,040	3,976	—	—	
III	Ripe	4,149	12,564	—	—	
	Overripe	3,713	10,812	—	—	
IV	Ripe	200	200	200	99	640×640
	Overripe	200	200	200	101	

Table 2 Training Parameters

Parameter	Value
optimizer	SGD
batch	2
mask_ratio	4
box	7.5
cls	0.5
weight_decay	0.0005

4 Our results

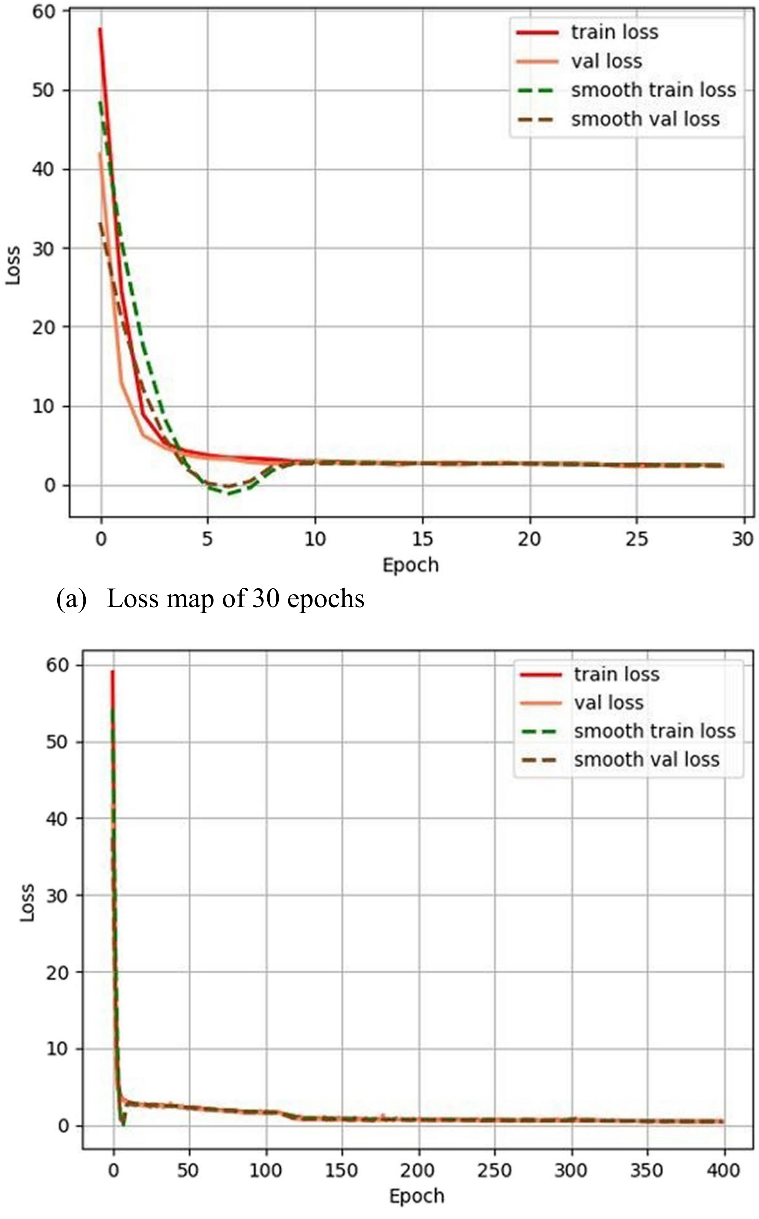
4.1 Experimental settings and evaluation method

In this project, pyTorch is adopted as an experimental platform. We made use of mobile phones to create four datasets in Table 1, with a total of 4,000 images and 20,000 labels. The three groups of data are sorted according to the size and the quantity of images. We found when we are use of all the datasets in model training, too many visual features cause redundancy and generate overload of the model. So, we manually discarded the data with inconspicuous features in the training, and finally we used two thousand samples. We

Table 3 The results of precisions by training CenterNet model

Model	Weights	Freeze Epoch	Epoch	AP0.5	AP@0.5:0.95	Average inference time(millisecond)
CenterNet	CenterNet _ResNet-50	20	30	0.138	0.122	10
		30	50	0.138	0.125	10
		50	100	0.135	0.126	10
		100	200	0.913	0.854	10
		200	300	0.933	0.889	10
		300	400	0.960	0.909	10
		400	500	0.928	0.881	10

classify the ripeness of fruits according to the degree of fruit peel. In Fig. 3, the smooth peel is from ripe apple, and the wrinkled peel is from overripe apple. The dataset is labeled with software Labelimg. The image on the left side of Fig. 1 is the dataset we labelled. We manually marked the location and class of the apples with a bounding box in red.



(a) Loss map of 30 epochs

(b) Loss map of 400 epochs

Fig. 7 Loss map with various epochs (**a**) 30 epochs (**b**) 400 epochs

According to the characteristics of one-stage models, the training images can be input in any size, and then the algorithm resizes the image to a size 640×640.

Table 2 shows the parameter settings of this experiment. With regard to supervised learning, initially we set a larger learning rate, and then decreased the learning rate as the number of iterations increases, we set the learning rate to 0.01. All the data is input into the network during training, and the gradients are calculated. Due to the huge difference in different gradient values, it is difficult to use a global learning rate. Therefore, we set the batch value to 2 in order to avoid memory explosion. We take use of precision as an indicator for evaluating the model in Eq. (9).

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative} \tag{9}$$

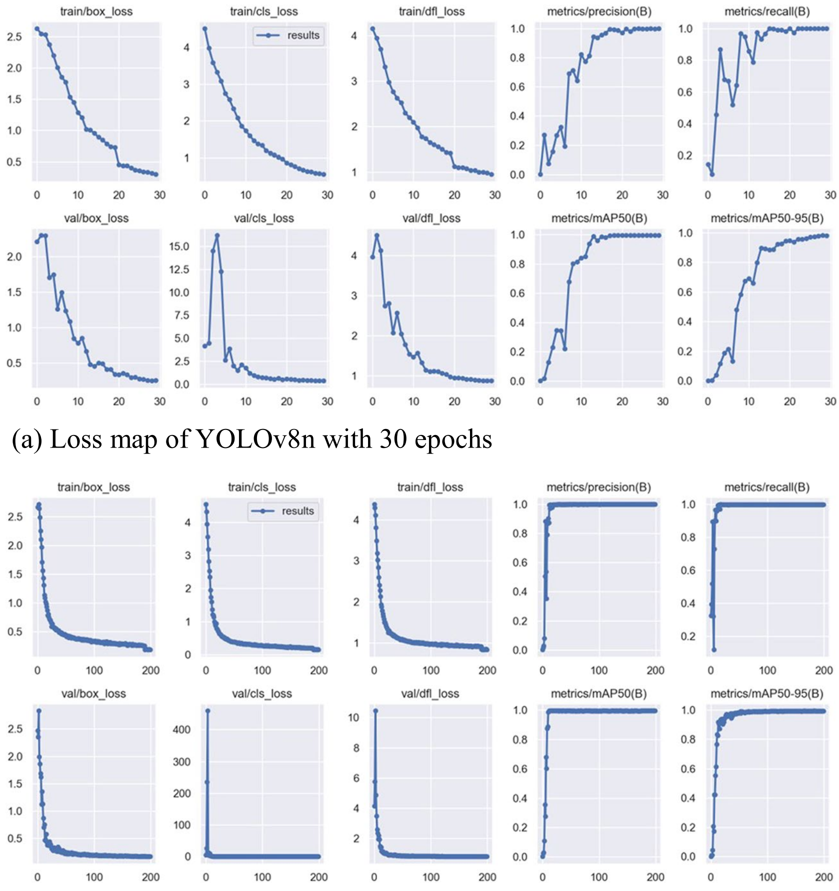
As shown in Fig. 4, if the prediction bounding box and the real bounding box IOU (Intersection over Union) are greater than or equal to 0.5, it is considered a positive sample. AP measures the detection of a class, mAP is the detection of multiple classes. In AP0.5, the confidence threshold IoU is set to 0.5, and only the preselected boxes with $IoU > 0.5$ are calculated. mAP@0.5:0.95 represents the average mAP on different IoU thresholds (0.5–0.95, step size 0.05).

4.2 Results and analysis

Both CenterNet model and YOLOv8 model are basic anchor-free. For the CenterNet model, we chose ResNet-50 as the backbone. For YOLOv8 model, we made use of YOLO8n, YOLOv8m, and YOLOv8x with three weights from small to large. We compare the precision values obtained by different models and epochs in the same experimental

Table 4 The results of average precisions by training YOLOv8 model

Model	Weights	Epoch	mAP0.5	mAP@0.5:0.95	Average inference time(millisecond)
YOLOv8	YOLOv8n	10	0.995	0.793	2.8
		20	0.994	0.958	3.4
		30	0.995	0.982	3.1
		50	0.995	0.990	3.6
		100	0.995	0.993	2.9
	YOLOv8m	10	0.982	0.831	6.6
		20	0.995	0.956	6.6
		30	0.995	0.985	6.7
		50	0.995	0.991	6.6
		100	0.995	0.993	6.6
		200	0.995	0.994	6.5
	YOLOv8x	10	0.947	0.815	16.7
		20	0.995	0.962	17.3
		30	0.995	0.986	16.9
		50	0.995	0.992	17.2
		100	0.995	0.993	17.2
		200	0.995	0.993	17.3



(a) Loss map of YOLOv8n with 30 epochs

(b) Loss map of YOLOv8n with 200 epochs

Fig. 8 Loss map of YOLOv8n model (**a**) 30 epochs (**b**) 200 epochs

environment. At the same time, based on the real-time detection requirements of the experiment, the average inference time of the detection is also employed to evaluate the quality of the model.

While we are training the model, we split samples into a training set and a validation set. The training set and verification set are divided according to the ratio 9:1, then the loss value calculated by the training model will be divided into the overall loss of the training set and the val loss of the test set. From Table 3, we observe that when the number of iterations is too small, the model cannot learn the characteristics of the fruit. In Fig. 7 (b), if the loss decreases, *val_loss* decreases, which indicates that the training is normal and the model is in the optimal state. In Fig. 7 (a), the loss is stable and *val_loss* is stable, which indicate that the learning process encounters a bottleneck, the training parameters are not set properly, and the model is in the worst case. During the training process of CenterNet, the backbone is frozen, and the feature extraction network does not change, so more training can help jump out of the local optimal solution.

The anchor-free structure (AFS) of YOLOv8 adopts task alignment learning dynamic matching, and introduces distribution focal loss (DFL) combined with CIoU loss as the function of the regression branch, which makes the classification and regression tasks have a high consistency. In the data enhancement part of training, turning off mosaic enhancement in the last 10 epochs is conducive to the stability of model convergence. In Table 4, increasing the number of training epochs from 50 to 100 makes the model training more adequate. YOLOv8 achieves lightweight and fast detection.

In Table 4, the larger the pre-training weight, the more time the model needs for average inference time. If the threshold is set as 0.5, the model outputs a satisfactory result. If AP@0.5:0.95, the average precision fluctuates. Although the precision results are all higher than 80%, we observe from Fig. 8 that if the number of iterations is too small, the convergence result of the model is not good.

In Table 4, larger pre-training weights lead to longer average inference time. In Tables 5, 6, and 7, the large weight processing takes longer time on average for each picture and does not generate better average precision. On the contrary, the large weight also brings the problem of training overfitting and wastes resources. During the training process, we chose a smaller learning rate to ensure that the model can better find the optimal point. However, if too many iterations are assigned, the model will still have the problem of not being able to be trained.

In Table 8, we are use of the idea of ablation experiments to analyze the training results of the YOLOv8 model and CenterNet model. As an anchor free model, both

Table 5 The results of precisions by training YOLOv8n model

Model	Weights	Epoch	Class	AP0.5	AP@0.5:0.95
YOLOv8	YOLOv8n	10	Ripe apple	0.994	0.800
			Overripe apple	0.995	0.864
			Ripe pear	0.985	0.808
			Overripe pear	0.848	0.679
		20	Ripe apple	0.994	0.973
			Overripe apple	0.995	0.942
			Ripe pear	0.995	0.972
			Overripe pear	0.992	0.944
		30	Ripe apple	0.995	0.981
			Overripe apple	0.995	0.973
			Ripe pear	0995	0.993
			Overripe pear	0.995	0.981
		50	Ripe apple	0.994	0.990
			Overripe apple	0.995	0.984
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.991
		100	Ripe apple	0.995	0.991
			Overripe apple	0.995	0.990
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.994

the YOLOv8 model and the CenterNet model can complete fruit ripeness recognition. Although the CenterNet model freezes the backbone part during training time, which is equivalent to accomplish a transfer learning, the precision of CenterNet training does not have much advantage over the YOLOv8 model. While training for two hundred iterations, the CenterNet model requires more inference time than YOLOv8m. If the CenterNet model saves a lot of resources during training process, the C2f module of YOLOv8 model also reduces the weights of the proposed model. But if the training parameters are the same, the lightweight model YOLOv8n can complete the detection with a faster response speed in 100 iterations. CenterNet has not been able to extract the feature of the fruits at 100 iterations.

5 Discussion

Sun's green apple detection method achieved an accuracy of 34.2%. Wang et al. utilized the transformer model to recognize different sizes and types of tomatoes, reaching a precision of 89.4%. Alzahrani & Alsaade attained a remarkable precision of 99.88% in fruit lesion detection using DenseNet-169. Kim et al. conducted research on detecting tiny objects

Table 6 The results of precisions by training YOLOv8m model

Model	Weights	Epoch	Class	AP0.5	AP@0.5:0.95
YOLOv8	YOLOv8m	10	Ripe apple	0.990	0.841
			Overripe apple	0.995	0.880
			Ripe pear	0.995	0.852
			Overripe pear	0.949	0.753
		20	Ripe apple	0.995	0.961
			Overripe apple	0.995	0.937
			Ripe pear	0.995	0.982
			Overripe pear	0.995	0.945
		30	Ripe apple	0.995	0.948
			Overripe apple	0.995	0.975
			Ripe pear	0.995	0.992
			Overripe pear	0.995	0.988
		50	Ripe apple	0.995	0.992
			Overripe apple	0.995	0.984
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.992
		100	Ripe apple	0.994	0.992
			Overripe apple	0.995	0.992
			Ripe pear	0.995	0.994
			Overripe pear	0.995	0.994
		200	Ripe apple	0.995	0.993
			Overripe apple	0.995	0.992
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.995

Table 7 The results of precisions by training YOLOv8x model

Model	Weights	Epoch	Class	AP0.5	AP@0.5:0.95
YOLOv8	YOLOv8x	10	Ripe apple	0.995	0.840
			Overripe apple	0.995	0.852
			Ripe pear	0.990	0.865
			Overripe pear	0.808	0.703
		20	Ripe apple	0.995	0.978
			Overripe apple	0.995	0.943
			Ripe pear	0.995	0.963
			Overripe pear	0.995	0.965
		30	Ripe apple	0.994	0.989
			Overripe apple	0.995	0.978
			Ripe pear	0.995	0.990
			Overripe pear	0.995	0.989
		50	Ripe apple	0.994	0.991
			Overripe apple	0.995	0.987
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.994
		100	Ripe apple	0.995	0.993
			Overripe apple	0.995	0.991
			Ripe pear	0.995	0.994
			Overripe pear	0.995	0.993
		200	Ripe apple	0.995	0.993
			Overripe apple	0.995	0.990
			Ripe pear	0.995	0.995
			Overripe pear	0.995	0.994

from UAV images. Considering the environmental noise in the experiment, the YOLOv8 model achieved a processing speed of 45.7 fps (frames per second) in the P2 layer. Similar to these experiments, our fruit object detection experiment requires a small number of parameters. If we can guarantee a precision of 99.3%, our YOLOv8 model can control the detection speed to 2.9ms. In practical applications, faster object detection can result in significant time savings.

Table 8 The results of comparison

Model	Weights	Epoch	AP0.5	AP@0.5:0.95	Average inference time(millisecond)
CenterNet	centernet _ResNet-50	100	0.135	0.136	10
		200	0.913	0.854	10
YOLOv8	YOLOv8n	100	0.995	0.993	2.9
	YOLOv8m	200	0.995	0.994	6.5
	YOLOv8x	200	0.995	0.993	17.3

6 Conclusion

From the experimental results, we observe that both YOLOv8 and CenterNet can achieve an accuracy rate of more than 90%. The c2f module of the YOLOv8 model significantly reduces the number of blocks in the largest stage of the backbone network to construct a more lightweight model. Simultaneously, the model decreases the number of output channels in the final stage, which further reduces the number of parameters and computations. In the practical application of fruit detection, the detection speed is of utmost importance. YOLOv8 outperforms in terms of both speed and accuracy, with the lightweight model YOLOv8n requiring only 2.9ms to complete accurate detection.

Although the current model can accurately locate and classify fruits, we must also consider the impact of extreme orchard environments on automatic fruit detection. The effects of severe weather conditions, such as strong winds, heavy rains, or disturbances from birds, represent a research direction that we aim to explore in future experiments.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Declarations

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

References

1. Alzahrani MS, Alsaade FW (2023) Transform and deep learning algorithms for the early detection and recognition of tomato leaf disease. Agronomy 13(5):1184
2. Basri H, Syarif I, Sukaridhoto S (2018) Faster R-CNN implementation method for multi-fruit detection using TensorFlow platform. IEEE International Electronics Symposium on Knowledge Creation and Intelligent Computing, pp. 337–340
3. Egi Y, Hajyzadeh M, Eyceyurt E (2022) Drone-computer communication based tomato generative organ counting model using YOLOv5 and deep-sort. Agriculture 12(9):1290
4. Fu L, Yang Z, Wu F, Zou X, Lin J, Cao Y, Duan J (2022) YOLO-Banana: A lightweight neural network for rapid detection of banana bunches and stalks in the natural environment. Agronomy 12(2):391
5. Gao F, Fang W, Sun X, Wu Z, Zhao G, Li G, ... Zhang Q (2022) A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Comput Electron Agric. 197: 107000
6. Gao J, Dai S, Huang J, Xiao X, Liu L, Wang L, ... Li M (2022) Kiwifruit detection method in orchard via an improved light-weight YOLOv4. Agronomy, 12(9):2081
7. Häni N, Roy P, Isler V (2020) A comparative study of fruit detection and counting methods for yield mapping in apple orchards. J Field Robot 37(2):263–282
8. Huang H, Huang T, Li Z, Lyu S, Hong T (2022) Design of citrus fruit detection system based on mobile platform and edge computer device. Sensors 22(1):59

9. Jaju S, Chandak M (2022) A transfer learning model based on ResNet-50 for flower detection. IEEE International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 307–311
10. Ji W, Pan Y, Xu B, Wang J (2022) A real-time Apple targets detection method for picking robot based on ShufflenetV2-YOLOX. Agriculture 12(6):856
11. Kang H, Chen C (2020) Fast implementation of real-time fruit detection in apple orchards using deep learning. Comput Electron Agric 168:105108
12. Khan R, Debnath R (2019) Multiclass fruit classification using efficient object detection and recognition techniques. Int J Image Graph Signal Process 11(8):1–18
13. Kim JH, Kim N, Won CS (2023) High-speed drone detection based on YOLOv8. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1–5
14. Latha RS, Sreekanth GR, Rajadevi R, Nivetha SK, Kumar KA, Akash V, ... Anbarasu P (2022) Fruits and vegetables recognition using YOLO. IEEE International Conference on Computer Communication and Informatics (ICCCI) pp. 1–6
15. Li G, Fu L, Gao C, Fang W, Zhao G, Shi F, ... Cui Y (2022) Multiclass detection of kiwifruit flower and its distribution identification in orchard based on YOLOv5l and Euclidean distance. Comput Electron Agric, 201: 107342
16. Li K, Zhai L, Pan H, Shi Y, Ding X, Cui Y (2022) Identification of the operating position and orientation of a robotic kiwifruit pollinator. Biosys Eng 222:29–44
17. Li T, Feng Q, Qiu Q, Xie F, Zhao C (2022) Occluded apple fruit detection and localization with a frustum-based point-cloud-processing approach for robotic harvesting. Remote Sensing 14(3):482
18. Liu G, Hou Z, Liu H, Liu J, Zhao W, Li K (2022) TomatoDet: Anchor-free detector for tomato detection. Front Plant Sci 13:942875
19. Liu TH, Nie XN, Wu JM, Zhang D, Liu W, Cheng YF, ... Qi L (2022) Pineapple (Ananas comosus) fruit detection and localization in natural environment based on binocular stereo vision and improved YOLOv3 model. Precis Agric, pp. 1–22
20. Lee J, Hwang KI (2022) YOLO with adaptive frame control for real-time object detection applications. Multimed Tools Appl 81(25):36375–36396
21. Mai X, Zhang H, Meng MQH (2018) Faster R-CNN with classifier fusion for small fruit detection. IEEE International Conference on Robotics and Automation (ICRA), pp. 7166–7172
22. Qi J, Nguyen M, Yan W (2022) Waste classification from digital images using ConvNeXt. Pacific-Rim Symposium on Image and Video Technology, pp.1–13
23. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C (2016) Deepfruits: A fruit detection system using deep neural networks. Sensors 16(8):1222
24. Sun M, Zhao R, Yin X, Xu L, Ruan C, Jia W (2023) FBoT-Net: Focal bottleneck transformer network for small green apple detection. Comput Electron Agric 205:107609
25. Tang Y, Zhou H, Wang H, Zhang Y (2023) Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Expert Syst Appl 211:118573
26. Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z (2019) Apple detection during different growth stages in orchards using the improved YOLOv3 model. Comput Electron Agric 157:417–426
27. Ukwuoma CC, Zhiguang Q, Bin Heyat MB, Ali L, Almaspoor Z, Monday HN (2022) Recent advancements in fruit detection and classification using deep learning techniques. Math Probl Eng 2022:1–29
28. Villacrés J, Viscaino M, Delpiano J, Vougioukas S, Cheein FA (2023) Apple orchard production estimation using deep learning strategies: A comparison of tracking-by-detection algorithms. Comput Electron Agric 204:107513
29. Wan S, Goudos S (2020) Faster R-CNN for multiclass fruit detection using a robotic vision system. Comput Netw 168:107036
30. Wang Q, Chen J, Deng J, Zhang X (2021) 3D-CenterNet: 3D object detection network for point clouds with center estimation priority. Pattern Recogn 115:107884
31. Wang X, Wang S, Cao J, Wang Y (2020) Data-driven based tiny-YOLOv3 method for front vehicle detection inducing SPP-net. IEEE Access 8:110227–110236
32. Wang Z, Jin L, Wang S, Xu H (2022) Apple stem/calyx real-time recognition using YOLOv5 algorithm for fruit automatic loading system. Postharvest Biol Technol 185:111808
33. Xia Y, Nguyen M, Yan WQ (2022) A real-time kiwifruit detection based on improved YOLOv7. IVCNZ, pp. 48–61
34. Xiao B, Nguyen M, YanW (2021) Apple ripeness identification using deep learning. International Symposium on Geometry and Vision (ISGV), pp.53–67
35. Wang C, Yang G, Huang Y, Liu Y, Zhang Y (2023) A transformer-based Mask R-CNN for tomato detection and segmentation. J Intell Fuzzy Syst 44(5):8585–8595

36. Yang R, Hu Y, Yao Y, Gao M, Liu R (2022) Fruit target detection based on BCo-YOLOv5 model. Mobile Information Systems, pp.1–8
37. Yao J, Wang Y, Xiang Y, Yang J, Zhu Y, Li X, ... Gong G (2022) Two-stage detection algorithm for kiwifruit leaf diseases based on deep learning. Plants 11(6):768
38. Zhao K, Yan WQ (2021) Fruit detection from digital images using CenterNet. International Symposium on Geometry and Vision, pp. 313–326
39. Zhang F, Gao J, Zhou H, Zhang J, Zou K, Yuan T (2022) Three-dimensional pose detection method based on key points detection network for tomato bunch. Comput Electron Agric 195:106824
40. Zhang W, Wang J, Liu Y, Chen K, Li H, Duan Y, ... Guo W (2022) Deep-learning-based in-field citrus fruit detection and tracking. Hortic Res 9:uhac003
41. Zeng N, Wu P, Wang Z, Li H, LiuW LX (2022) A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. IEEE Trans Instrum Meas 71:1–14
42. Zhou J, Hu W, Zou A, Zhai S, Liu T, Yang W, Jiang P (2022) Lightweight detection algorithm of kiwifruit based on improved YOLOX-s. Agriculture 12(7):993

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.