Engineering Applications of Artificial Intelligence 142 (2025) 109923

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Estimation of the orientation of potatoes and detection bud eye position using potato orientation detection you only look once with fast and accurate features for the movement strategy of intelligent cutting robots

Jie Huang [a], Xiangyou Wang [a,*], Chengqian Jin [b], Fernando Auat Cheein [c,d], Xinyu Yang [a]

[a] School of Agricultural Engineering and Food Science, Shandong University of Technology, Zibo, 255000, China
[b] Nanjing Institute of Agricultural Mechanization, Ministry of Agriculture and Rural Affairs, Nanjing, 210014, China
[c] Department of Engineering, Harper Adams University, England, UK
[d] Department of Electronic Engineering, Advanced Center for Electrical and Electronic Engineering (AC3E), Federico Santa Maria Technical University, Valparaiso, Chile

ARTICLE INFO

Keywords:
Potato
Rotated object detection
Deep learning
You-only-look-once-v8
Lightweight
Cutting robot

ABSTRACT

The accurate detection of potato orientation and bud eye positions is critical for guiding the end-effector of intelligent cutting robots. This study introduced Potato Orientation Detection You Only Look Once (POD-YOLO), a novel lightweight model based on YOLOv8n, designed for fast and precise detection of potato orientation and bud eye locations. Key innovations include replacing the Cross Stage Partial Dark Network (CSPDarkNet) with the Cross Stage Partial and Dual Partial Network (CSPDPNet) to reduce parameter count and improve detection accuracy. Additionally, the "no more strided convolutions or pooling" approach replaced downsampling modules in the backbone and neck, enhancing detection of small targets and low-resolution images. The regression loss function was further optimized by substituting Kalman Filtering Intersection over Union (KFIoU) for improved rotated bounding box performance. Experimental results showed that POD-YOLO achieved a mean Average Precision (mAP) of 97.2%, with a precision of 95.2%, recall of 94.0%, and detection time of 9.01 ms. With only 1.75 million parameters, POD-YOLO was lightweight and efficient, meeting real-time requirements. This research offers a robust and effective solution for automated potato orientation and bud eye detection, laying the groundwork for advanced agricultural automation.

1. Introduction

Potatoes are the fourth-most important food crop after corn, wheat, and rice (Johnson and Auat Cheein, 2023). As of 2022, according to statistics from the International Food and Agriculture Organization, the planting area of potatoes worldwide is approximately 1.7×10^7 ha, and the total output is approximately 3.75×10^8 tons. The potato planting area of Asian countries accounts for 43.9% of the world's planting area (UN Food and Agriculture Organization, 2023). However, the cutting of potatoes before sowing is still mostly done by hand, which is characterized by problems such as a low degree of mechanization, high labor intensity, and high labor costs.

To address these challenges, countries such as China and the United States, as well as some European nations, have begun to develop intelligent and automated cutting equipment to handle the busy sowing season (Milestone, 2024; Peterson, 2024; Wang et al., 2020). The cutting principle of the intelligent potato-cutting robot is to first identify the eye positions and then calculate the cutting angle based on these positions (Yang et al., 2023). However, when calculating the cutting angle, considering only the bud eye positions does not allow for real-time tracking of the potato's position and orientation, making it difficult to meet the cutting requirements. This study focuses primarily on the orientation of potatoes and the identification of bud eye positions, laying a solid foundation for the next step of calculating the cutting angle in potato-cutting robots.

The traditional image-based method for potato bud eye detection mainly relies on the information of color features (Wu et al., 2020), shape features (Bargoti and Underwood, 2017), and texture features (Sengupta and Lee, 2014) to complete the detection task. Based on the use of multispectral images, Yang et al. (2023) combined the supervised multi-threshold segmentation model and the Canny edge detector to obtain a segmentation mask and complete the detection of potato bud

* Corresponding author.
E-mail address: wxy@sdut.edu.cn (X. Wang).

https://doi.org/10.1016/j.engappai.2024.109923
Received 14 August 2024; Received in revised form 21 November 2024; Accepted 20 December 2024
Available online 24 December 2024

eyes, and achieved an average detection accuracy of 89.2%. Li et al. (2018) proposed a potato bud eye recognition method based on three-dimensional (3D) geometric features of color saturation, and the bud eye recognition accuracy reached 91.4%. While these research methods can identify the position of the potato bud relatively accurately, their application scenarios are singular and their robustness is poor; thus, they cannot complete the detection task in complex environments.

In recent years, deep learning technology has been increasingly used in agriculture (Ariza-Sentís et al., 2024; Koirala et al., 2019). For instance, Paul et al. (2024) adopted various You Only Look Once (YOLO) algorithms for pepper detection, peduncle detection, and counting/-tracking detection. Their results showed that the YOLOv8s model achieved the highest accuracy for the pepper detection task. Prasetyo et al. (2022) proposed the YOLOv4-tiny lightweight object detector for the detection of fish body parts, the accuracy of which is improved via the enhancement and balancing of feature diversity and the addition of extra branch detectors. By introducing DenseNet, SPP blocks, and an improved PANet to the YOLOv4 framework, Roy and Bhaduri (2022) proposed Dense-YOLOv4, an improved real-time target detection framework based on the YOLOv4 algorithm, for the detection of mangoes in complex scenes. Cardellicchio et al. (2023) developed a method based on the YOLOv5 object detection algorithm to identify tomatoes, flowers, and nodes, either independently or collectively. Mirhaji et al. (2021) created image data of orange trees under different lighting conditions, applied YOLOv2, YOLOv3, and YOLOv4 to count and detect fruits in citrus orchards, and identified YOLOv4 as the best detection model. Zhou et al., (2024) proposed correlation filters with adaptive modality weights and cross-modality learning capabilities to perform multimodal tracking tasks. The method demonstrates excellent tracking performance across several benchmark datasets and is capable of overcoming challenges such as background clutter and partial occlusion. Zhou et al., 2023 proposed a blind image quality assessment (BIQA) method that uses self-attention and recurrent neural networks (RNNs). This method can simultaneously consider both local and global influences on image quality perception. Zhou et al., 2023 proposed a joint architecture with user perception and an efficient transformer dedicated to no-reference (NR) image quality assessment (IQA) for 360-degree images. This method can learn both global and local features in 360-degree images and predict their quality score. Kaur et al. (2023) proposed a deep ensemble learning model (DELM) for autonomous plant disease identification. Experimental results show that the model, which integrates VGG16, InceptionV3, and GoogleNet, achieves higher accuracy. Samant et al. (2023) used deep learning techniques to predict whether potato leaves are diseased, comparing the prediction results of ANN and CNN algorithms. The results show that the CNN algorithm yields the best prediction performance. Trivedi et al., n.d. used a "deep convolutional neural network (DCNN)" based encoder-decoder architecture for semantic segmentation of leaf lesions. This method shows significant improvements compared to present crop disease classification systems. In addition, a series of similar object detection studies based on deep learning has been carried out (Ganesan and Chinnappan, 2022; Huang et al., 2023; Jiang et al., 2024; Magalhães et al., 2021; Marset et al., 2021; Onoufriou et al., 2023). Object detection algorithms based on deep learning can obtain the characteristic information of objects in complex environments to detect their positions.

However, most of the previously mentioned object detection methods use a horizontal frame to obtain the specific position of a certain object, but cannot detect the specific orientation of the object. In real applications, there exist many scenarios in which the angle information of objects must be obtained; these include text scenes (Liu et al., 2018), retail scenes (Pan et al., 2020), 3D scenes (Wang et al., 2021), and aerial image detection (Yang et al., 2020; Yi et al., 2020). Zhao et al. (2022) constructed a directional wheat cob detection algorithm by using the CSL loss function for angle classification, adding the CIoU loss function to optimize the fixed loss, and improving the YOLOv5 model; the average accuracy of this method was found to be 90.5%. Song et al. (2022) generated a corn cob position bounding box based on the Oriented R-CNN model, and the correct rate of position estimation was found to be 88.56%. Zhou et al., (2024) proposed a comprehensive framework based on multi-object oriented detection specifically designed for the detection and analysis of rod-like crops. The proposed YOLO-OBB model predicts oriented bounding boxes, with a mAP@0.5 of 90.3%.

The aforementioned research methods have achieved notable results in specific domains. However, challenges remain in accurately recognizing bud eyes and determining the orientation of potatoes for intelligent cutting robots, particularly in orientation determination. To address these issues, this study proposes a rotational object detection model, Potato Orientation Detection You Only Look Once (POD-YOLO), based on YOLOv8n, which enables fast and accurate recognition of potato orientation and bud eye positions. The model overcomes the limitations of existing methods in simultaneously detecting bud eye positions and potato orientation. The main contributions of this study are as follows.

1) A novel lightweight model, POD-YOLO, was proposed to detect the potato orientation and bud eye position in different states.
2) For low-resolution and small objects, the method of "no more strided convolutions or pooling" was used to replace the downsampling module to improve the detection accuracy of the model.
3) The Kalman Filtering Intersection-over-Union (KFIoU) was introduced as the bounding box regression loss to improve the angle regression quality of the bounding box.

The rest of this paper was organized as follows. Section 2 described the structure of the intelligent cutting robot, introduced the data collection and processing methods, and presented the POD-YOLO algorithm developed in this study. Section 3 detailed the hardware configuration of the experiment, the network training parameters, and provided a comprehensive presentation of the experimental results. In Section 4, the proposed methods were discussed, and finally, the article was concluded in Section 5.

2. Materials and methods

This section described the structural composition and operating principles of the intelligent cutting robot, followed by an introduction to potatoes and bud eye collection methods, data processing procedures, and the basic architecture of YOLOv8. Finally, it detailed the network structure and design principles of POD-YOLO.

2.1. Hardware system

A custom intelligent potato-cutting robot based on a Delta robot was utilized in this study to detect the orientation and bud eye position of potatoes. Fig. 1(a) shows the front view of the 3D model of the potato-cutting robot. The cutting robot consists of three main components: the visual detection module (Region A), the cutting action execution module (Region B), and the conveyor module (Region C).

During operation, as shown in the isometric view in Fig. 1(b), both the top and bottom cameras of the visual detection module simultaneously capture information about the positions of the buds and the orientation of the potatoes. The conveyor transports the potatoes to the cutting area, and the Delta robot performs the cutting actions.

In the cutting process, the positions of the buds and the orientation of the potatoes are identified by the cameras, and the cutting angle is calculated. The calculation results are sent to the Delta robot controller, which adjusts the cutting tool to the specified angle. The Delta robot then executes the cutting action, completing the intelligent cutting of the potatoes. Fig. 2 shows a physical image of the potato-cutting robot, which is used to collect data on the potatoes and their bud eye positions.

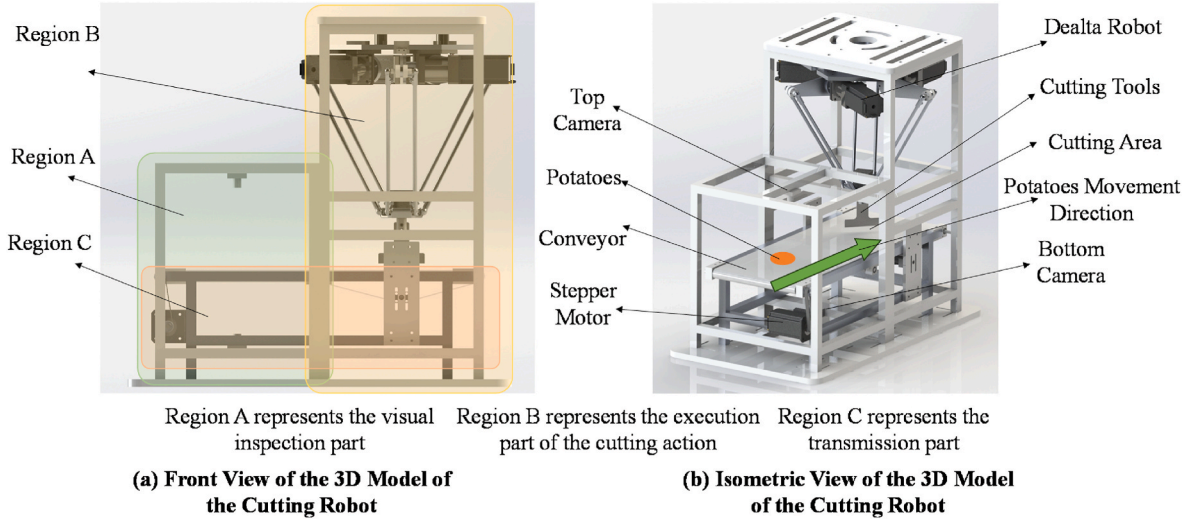

Fig. 1. 3D Model of the potato-cutting robot.

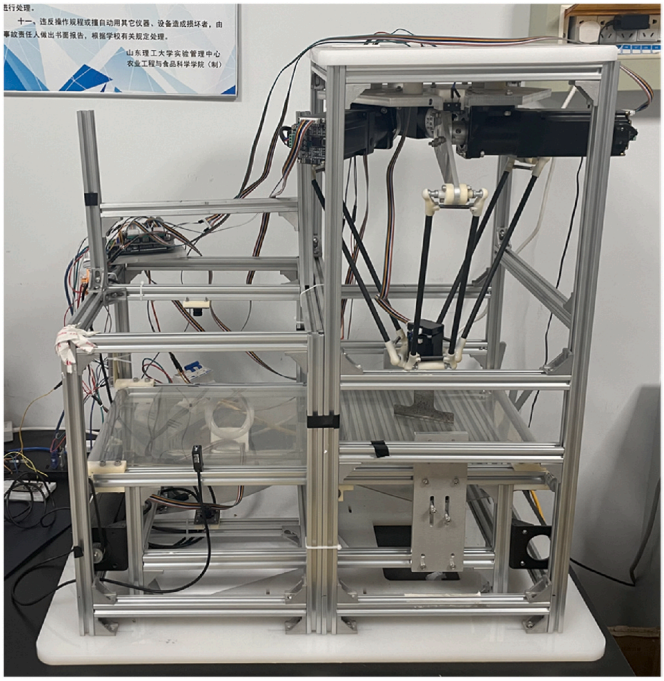

Fig. 2. physical image of the potato-cutting robot.

The primary focus of this research lies in the acquisition of the orientation and bud eye position of potatoes, thus laying the groundwork for subsequent studies on decision-making algorithms for potato-cutting.

2.2. Sample and image collection

In the experiment, a WH-L2140.K214L camera with a resolution of 1920 × 1080 pixels and a capture rate of 60 frames per second (FPS) was used. The Zhongshu No. 2 potato variety was used, and 300 potatoes with eyes were selected for the study. During image acquisition, to obtain information from different angles, each potato was placed on the experimental platform in the order of 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° for data collection, as shown in Fig. 3. The potatoes were photographed under natural lighting conditions. For each potato, simultaneous captures were taken by the top and bottom cameras on the experimental platform, thus acquiring two images containing the angle information and eye position. Sixteen images were obtained for each potato. This process was repeated in sequence, resulting in a total of 4800 potato images, each capturing different positional information.

2.3. Data processing

After acquiring 4800 raw images, the images were annotated using the "labelme" tool, and the annotations were saved in JSON format with polygon shapes. The annotations identified two categories, namely potatoes and buds, which were respectively labeled as "potato" and "bud". Given the relatively fixed detection environment, no additional data augmentation was necessary to enhance the generalization ability of the model. Therefore, the labeled potato images were randomly divided into training (3840 images), and test (960 images) sets at the ratio of 8:2 for utilization in subsequent model training and testing tasks.

During manual annotation, the precise location of the four corners of a rotated rectangle is challenging. As a result, a cross-marking method was employed to enhance the annotation efficiency. The specific technique is illustrated in Fig. 4. Firstly, one potato orientation was selected to serve as the reference line for the first diagonal. Then, the two points that were the closest to the background of the target object on both sides of this baseline were selected, as indicated by Point3 and Point4 in Fig. 4 (a), forming a closed cross. Subsequently, based on the principle of the perpendicular distance from a point to a line, convert the cross label into a rotated rectangular box as shown in Fig. 4(b). For reference, please see the code at: https://github.com/DDGRCF/YOLOX_OBB.

2.4. YOLOv8 structure

YOLOv8 is a representative algorithm in single-stage object detection, and has achieved good results on public datasets, such as the ImangeNet-1K, COCO, and DOTAv1 life scene and aerial photography datasets (Jocher et al., 2023). Compared to other object detection algorithms, such as YOLOv3 (Redmon and Farhadi, 2018), YOLOv5 (Jocher, 2020), and YOLOv9 (Wang et al., 2024), YOLOv8 features a wide range of application scenarios, a strong open-source foundation, and frequent maintenance, and can be applied to tasks such as object classification, object detection, instance segmentation, keypoint detection, object tracking, and rotated object detection.

In terms of the network architecture, YOLOv8 retains its original structural characteristics and mainly consists of three components: the backbone for feature extraction, the neck for enhancing feature information, and the detection head for obtaining object categories and

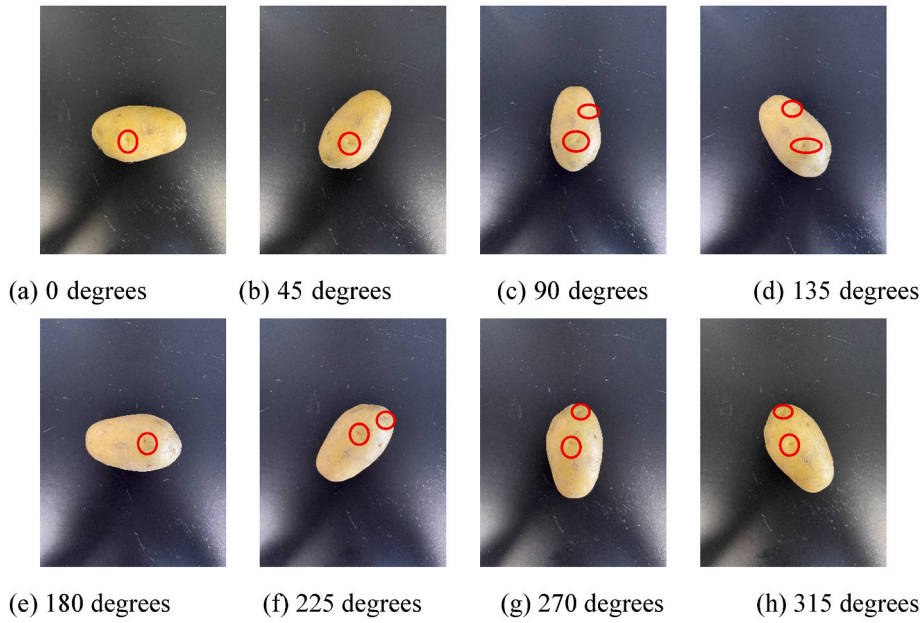

Fig. 3. Potato placement style. The red circle mark in the picture is the bud eye of the potatoes. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

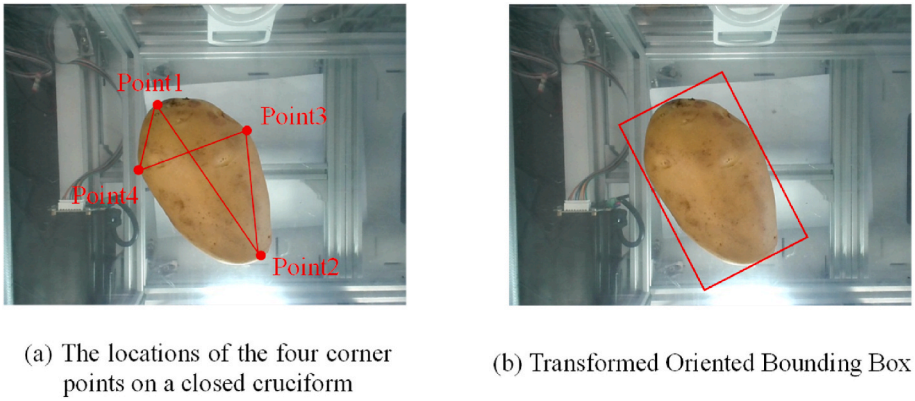

Fig. 4. Cross-marking label for rotating rectangle.

bounding boxes.

In the detection head portion, the design principle of YOLOv8 is altered as compared to its predecessors via the use of a decoupled head to separately calculate the losses for bounding boxes and categories. It separately extracts target location and category information, and learns the loss values of the model through different network branches before finally merging the information. The structure of this approach is indicated as "Decoupled" in Fig. 8. This design effectively reduces the parameter count and computational complexity of the model, thus enhancing its generalizability and robustness. Furthermore, YOLOv8 adopts an anchor-free philosophy, and directly learns the shapes of various bounding boxes. During inference, it does not rely on clustering, but instead fits the object size based on learned bounding box distances or keypoint positions. This method allows the network to better express object shapes and generalize without depending on prior knowledge of the data; thus, it particularly demonstrates significant improvement for moving objects, objects of inconsistent sizes, and targets of anomalous scales.

Additionally, YOLOv8 has been used to construct a variety of detection models for different application scenarios, namely YOLOv8n, YOLOv8s, YOLOv8m, and YOLOv8l, where n, s, m, and l represent the number of parameters of the model; n indicates the least number of parameters, s indicates the second-least number of parameters, m indicates a medium number of parameters, and l indicates the most computationally intensive model.

According to the preceding discussion, YOLOv8 features rich application scenarios, a simple structure, enhanced capability in representing irregular objects, and a wide range of model selectivity. YOLOv8n, which has a smaller number of parameters, was selected as the baseline model in this study, based on which the POD-YOLO detection model was designed to achieve the rapid, accurate, and lightweight acquisition of the orientation and bud eye positions of potatoes.

2.5. POD-YOLO

This section presented the detailed structure of the proposed POD-YOLO model. The model included a lightweight backbone network for efficient computation. It incorporated the Cross Stage Partial and Dual Partial Network (CSPDPNet) for robust feature extraction. Moreover, the Space-to-Depth Convolution (SPD-Conv) module was integrated to enhance the detection accuracy of small targets. Specific modifications to the YOLOv8n model were also described to further optimize performance. Additionally, it discussed the optimization of the Probabilistic Intersection-over-Union (ProbIoU) loss function to the KFIoU loss

function.

2.5.1. Cross Stage Partial and Dual Partial Network (CSPDPNet) structure

The Cross Stage Partial Dark Network (CSPDarkNet) in YOLOv8 consists of two ConvModules and n DarknetBottleneck modules, as shown in Fig. 5(a). In the Figure, the DarknetBottleneck module consists of two ConvModules, which are standard convolution modules with a large amount of calculation. The total number of Floating point Operations (FLOPs) executed in a standard convolution layer FL_{SC} is defined as

$$FL_{SC} = W \times H \times C^2 \times K^2, \tag{1}$$

where $W \times H$ is the size of the output feature map, C is the number of channels of the input and output feature maps, and K is the convolution kernel size.

Therefore, a lightweight convolution module, referred to as the CSPDPNet module, is proposed to replace the ConvModule in Darknet-Bottleneck. Its structure is shown in Fig. 5(b).

The CSPDPNet module is composed of the Partial Convolution (PConv) module from the FasterNet lightweight Convolutional Neural Network (CNN) (as shown in Fig. 6(b)) and the Dual Convolution (DualConv) design concept of lightweight CNN (Chen et al., 2023; Zhong et al., 2023). Its structure, as shown in Fig. 5(b), maintains the original structure of CSPDarkNet, but the DarknetBottleneck is changed to the PConv module. Moreover, drawing on the design idea of Dual-Conv, the two PConv modules are connected in series, which is repeated n times.

Without a loss of generality, assuming that the input and output have the same number of channels, then the FLOPs number of the PConv module is defined as

$$FL_{PC} = H \times W \times K^2 \times Cp^2, \tag{2}$$

where Cp is the number of input and output channels of the PConv module. By comparing the standard and PConv convolution modules, the following can be concluded:

$$R_{PC/SC} = \frac{FL_{PC}}{FL_{SC}} = \frac{C^2}{Cp^2}. \tag{3}$$

If Cp takes the value of 1/4 of C for the convolution operation, then the FLOPs number of the standard convolution is 16 times that of the PConv convolution.

This design reduces the number of parameters of the model, and, on the other hand, it can enhance the learning ability and generalization performance of the model via dual paths, thereby improving the deep learning efficiency and accuracy of the network while retaining spatial information.

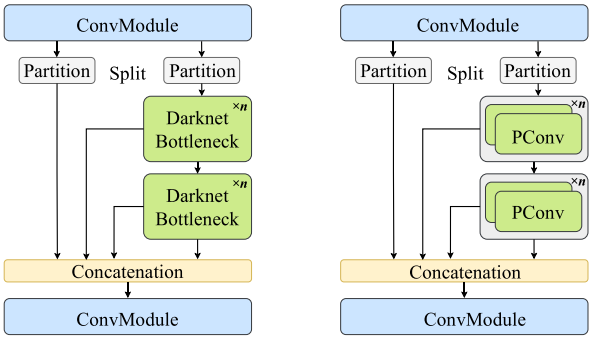

Fig. 5. CSPDarkNet and CSPDPNet module structures.

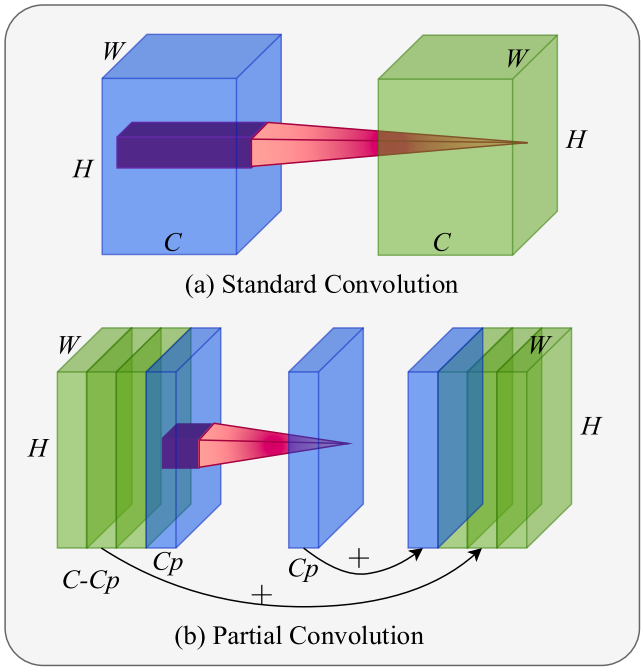

Fig. 6. ConvModules and PConv Module structure.

2.5.2. SPD-conv module

Most of the data collected on mobile devices are low-resolution images of 640 × 640 pixels. However, the bud eye is a small target object, and a too-low resolution will affect the expressive power of the image. In YOLOv8, strided convolution or pooling layers are used to downsample feature maps, which results in the serious loss of fine-grained information and weaker feature information.

For this reason, the downsampling modules of the backbone and neck in YOLOv8n are replaced. The Space-to-Depth and non-strided Convolution (SPD-Conv) modules were introduced and the method of "no more strided convolutions or pooling" was adopted to replace the downsampling module, aiming to better extract small objects and capture semantic information from low-resolution images (Sunkara and Luo, 2022). Fig. 7 shows the structure of SPD-Conv.

From the figure, it can be observed that for the input feature map X, a submap $f_{x,y}$ is composed of all elements $X(i, j)$, where both $i + x$ and $j + y$ are divisible by *scale*. Consequently, each submap can undergo downsampling by the factor *scale*. Fig. 7(a), (b), and 7(c) illustrate this process when $scale = 2$, resulting in four submaps $f_{0,0}, f_{1,0}, f_{0,1}, f_{1,1}$, each with dimensions $\left(\frac{S}{2}, \frac{S}{2}, C_1\right)$, as depicted in Fig. 7(b). The downsampled sub-feature maps are then concatenated along the channel dimension, yielding a feature map X' with spatial dimensions reduced by a factor of *scale* and channel dimensions increased by $scale^2$. In essence, the Space-to-Depth (SPD) operation transforms the feature map $X(S, S, C_1)$ into an intermediate feature map $X'\left(\frac{S}{scale}, \frac{S}{scale}, scale^2 C_1\right)$, as depicted in Fig. 7(c).

Subsequent to the SPD-Conv feature transformation layer, a non-strided convolutional layer with C_2 filters, where $C_2 < scale^2 C_1$, further maps $X'\left(\frac{S}{scale}, \frac{S}{scale}, scale^2 C_1\right)$ to $X''\left(\frac{S}{scale}, \frac{S}{scale}, C_2\right)$, as depicted in Fig. 7(d).

In standard downsampling with a stride greater than 1, there is a risk of discriminative information loss. Although it seemingly also projects the feature map $X(S, S, C_1) \rightarrow X''\left(\frac{S}{scale}, \frac{S}{scale}, C_2\right)$, no such intermediate representation X' is involved in that process.

In summary, the downsampling module constructed via SPD-Conv

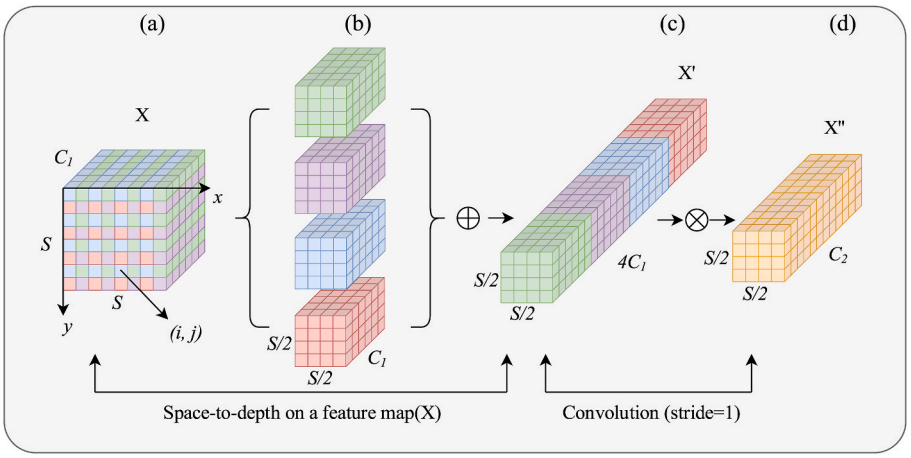

Fig. 7. SPD-Conv downsampling module structure.

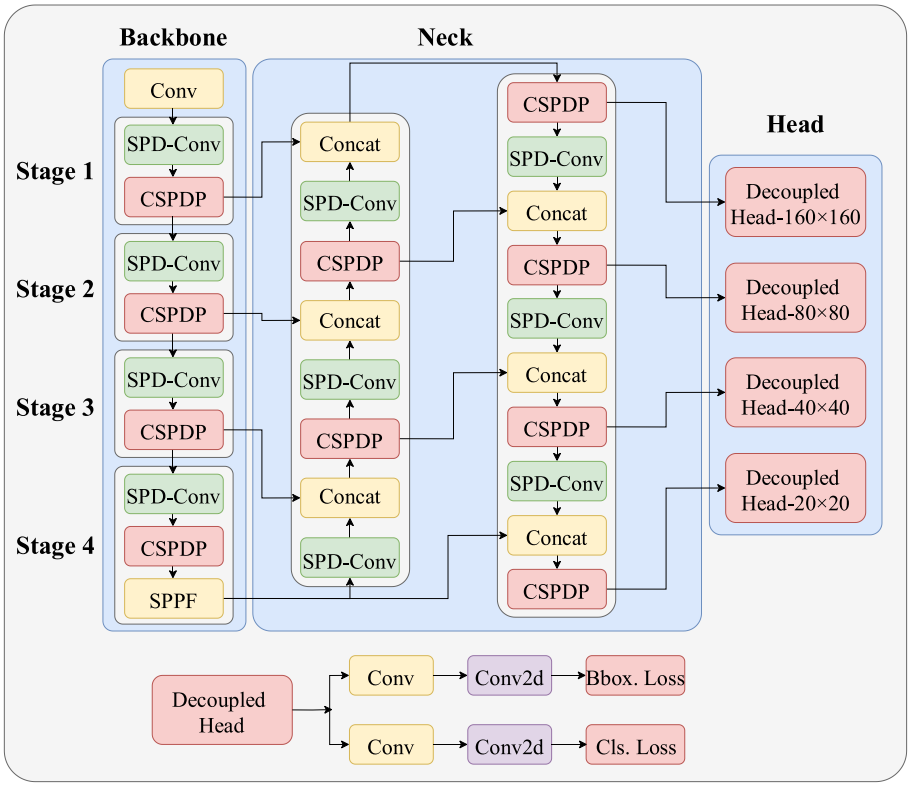

Fig. 8. Improved rotated object detection model.

effectively performs dimensionality reduction without losing learnable information, thereby serving as a substitute for traditional strided convolutions and pooling operations commonly adopted in existing network architectures. This novel approach demonstrates improved feature extraction performance for low-resolution images and small target objects.

2.5.3. *Improved POD-YOLO model*

The improved POD-YOLO model mainly exhibits changes in the CSPDarkNet module and downsampling ConvModule in the backbone network. Additionally, a small target detection head is added. Specifically, CSPDarkNet is changed to the CSPDPNet module, and ConvModule is changed to the SPD-Conv module, as shown in Fig. 8. When the model performs forward reasoning, it first downsamples and expands the channel number of the image with a size of 640 × 640 × 3, and then uses four sets of stage layers composed of SPD-Conv and CSPDPNet to complete the downsampling of the image. Second, the Spatial Pyramid Pooling-Fast (SPPF) layer is used to further obtain richer semantic information. The feature information in Stage1, Stage2, Stage3, and Stage4 is then extracted and connected to the neck network of YOLOv8, and a Feature Pyramid Network (FPN) module is constructed to extract feature information about different receptive fields. The SPD-Conv and CSPDPNet modules are then used again to obtain small target objects and reduce the amount of model calculation. Finally, the neck is divided into four feature layers of different scales to respectively complete the detection tasks for large, medium, small, and extra small objects.

2.5.4. *Angle regression loss function*

In rotated object detection, the perspective of the object is often from above, vertically overlooking the state. The shape and position of objects

often contain different angle information. At this time, the use of traditional horizontal detection frames cannot meet the detection needs. In January 2024, YOLOv8 OBB (Jocher et al., 2023) included the addition of an angle regression loss function on the basis of horizontal object detection to achieve rotated object detection. The angle regression loss function is the Probabilistic Intersection-over-Union (ProbIoU), the calculation formula of which is as follows (Murrugarra-Llerena et al., 2021):

$$\mathscr{L}_1(p,q) = H_D(p,q) = \sqrt{1 - B_c(p,q)}, \tag{4}$$

where $B_C = e^{-B_D}$, considering that $p \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $q \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ are Gaussian distributions with

$$\boldsymbol{\mu}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} a_1 & c_1 \\ c_1 & b_1 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} a_2 & c_2 \\ c_2 & b_2 \end{bmatrix}, \tag{5}$$

we can obtain a closed-form expression for B_D given by

$$B_D = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \ln\left(\frac{\det \Sigma}{\sqrt{\det \Sigma_1 \det \Sigma_2}}\right), \tag{6}$$

where, $\Sigma = \frac{1}{2}(\Sigma_1 + \Sigma_2)$, $\Sigma = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top$ and $\mu = (x,y)^\top$.

Here, $\mathbf{R}$ denotes the rotation matrix and $\boldsymbol{\Lambda}$ denotes the diagonal matrix of eigenvalues. For a 2D object $\mathscr{B}_{2d}(x,y,w,h,\theta)$,

$$\mathbf{R_{2d}} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \boldsymbol{\Lambda_{2d}} = \begin{pmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{pmatrix}. \tag{7}$$

ProbIoU uses a Gaussian distribution to fuzzily represent the target area, which is equivalent to using an ellipse to represent the bounding box. It is scale-invariant and will not cause the loss value to change as the scale changes. Furthermore, it no longer requires the definition of the bounding boxes; however, for a square target frame, there is no way to provide more accurate angle information.

Therefore, in this study, the Kalman Filtering Intersection-over-Union (KFIoU) loss function (Yang et al., 2023) is used to replace the ProbIoU loss function. While KFIoU also uses the Gaussian distribution principle, the Kalman filter principle is used to calculate the Gaussian distribution probability, and the center point loss function is introduced to calculate the center distance between the two bounding boxes to improve the training accuracy of the model.

The specific implementation process is shown in Fig. 9. The two rotated rectangular boxes are first converted into Gaussian probability distribution regions. The center point loss is then introduced, and the center position of the two Gaussian distributions is calculated. This is followed by the calculation of the intersection region of the two Gaussian distributions, and the size of the angular regression loss function is then evaluated. Finally, the three Gaussian distributions are inverted into a rotated rectangular box, and the approximate SkewIoU is calculated.

When using KFIoU to calculate the rotated bounding box loss function, the area of the corresponding rotation box is first calculated based on the covariance (Yang et al., 2023):

$$\mathscr{V}_{\mathscr{B}}(\boldsymbol{\Sigma}) = 2^n \sqrt{\prod eig(\boldsymbol{\Sigma})} = 2^n \cdot \left|\boldsymbol{\Sigma}^{\frac{1}{2}}\right| = 2^n \cdot |\boldsymbol{\Sigma}|^{\frac{1}{2}}, \tag{8}$$

where n denotes the dimension information, here $n = 2$.

The key to the calculation of SkewIoU is to obtain the area of the intersection area. If the intersection area is also approximated as a Gaussian distribution, its area can be calculated by Eq. (8). Moreover, the intersection areas $\mathcal{N}_{\mathbf{x}}(\mu_1, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}_{\mathbf{x}}(\mu_2, \boldsymbol{\Sigma}_2)$ of the two Gaussian distributions can be obtained by the product of the current two Gaussian distributions, as follows (Liao et al., 2023; Tian et al., 2019):

$$\alpha \mathcal{N}_{\mathbf{x}}(\mu, \boldsymbol{\Sigma}) = \mathcal{N}_{\mathbf{x}}(\mu_1, \boldsymbol{\Sigma}_1) \mathcal{N}_{\mathbf{x}}(\mu_2, \boldsymbol{\Sigma}_2), \tag{9}$$

where $\alpha = \mathcal{N}_{\mu_1}(\mu_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$, $\mu = \mu_1 + \mathbf{K}(\mu_2 - \mu_1)$, and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 - \mathbf{K}\boldsymbol{\Sigma}_1$, among which $\mathbf{K}$ is the Kalman gain $\mathbf{K} = \boldsymbol{\Sigma}_1(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}$.

Because the intersected Gaussian distribution is not a standard Gaussian distribution, there is a coefficient in front of it, and this coefficient is related to the center distance between the two boxes. Therefore, center point loss is introduced to make the two Gaussian distributions concentric. Consequently, the coefficient is approximately constant, and there is no need to consider calculations.

The resulting KFIoU loss function can be derived from the following equation:

$$\text{KFIoU} = \frac{\mathscr{V}_{\mathscr{B}_3}(\boldsymbol{\Sigma})}{\mathscr{V}_{\mathscr{B}_1}(\boldsymbol{\Sigma}_1) + \mathscr{V}_{\mathscr{B}_2}(\boldsymbol{\Sigma}_2) - \mathscr{V}_{\mathscr{B}_3}(\boldsymbol{\Sigma})}, \tag{10}$$

where $\mathscr{V}_{\mathscr{B}_1}(\boldsymbol{\Sigma}_1)$ denotes the Gaussian area of the ground truth bounding box, $\mathscr{V}_{\mathscr{B}_2}(\boldsymbol{\Sigma}_2)$ denotes the Gaussian area of the predicted bounding box, and $\mathscr{V}_{\mathscr{B}_3}(\boldsymbol{\Sigma})$ signifies the area of intersection between $\mathscr{V}_{\mathscr{B}_1}(\boldsymbol{\Sigma}_1)$ and $\mathscr{V}_{\mathscr{B}_2}(\boldsymbol{\Sigma}_2)$, as shown in Fig. 8(d). In n dimensions, $0 \le \text{KFIoU} \le \frac{1}{2^{\frac{n}{n+1}} - 1}$, and when n is equal to 2, $0 \le \text{KFIoU} \le \frac{1}{3}$.

The Smooth L_1 loss is employed for the center point loss function, which is defined as follows (Girshick, 2015):

$$Smooth\, L_1(x) = \begin{cases} 0.5x^2 & \text{if} |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}, \tag{11}$$

where $x = f(x_i) - y_i$ represents the difference between the true and

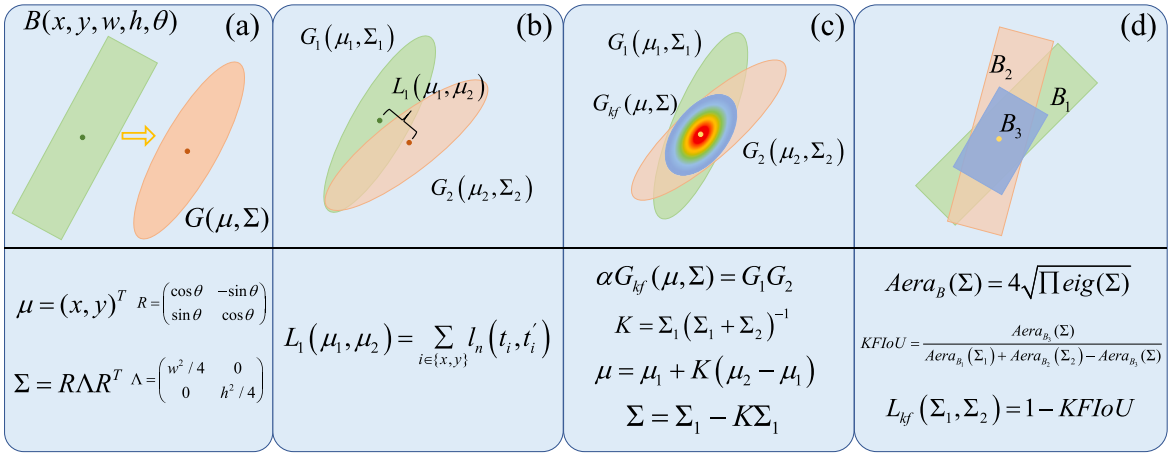

Fig. 9. KFIoU function calculation process.

predicted values.

The regression loss function is set by

$$L_{reg} = L_1 + L_{kf}, \tag{12}$$

where $L_{kf}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2) = e^{1-\mathrm{KFIoU}} - 1$.

3. Results

This section begined with a detailed description of the computational hardware configuration and training parameters of the POD-YOLO model. It then introduced the various improvement modules, followed by an ablation study and an analysis of the experimental results before and after the model's enhancements. Finally, it provided a comparative analysis of various rotated object detection algorithms.

3.1. Experimental setup

The computing resources used in this study were obtained from the online server (AutoDL) of the Chinese Academy of Sciences AutoDL Technology Co., Ltd. The processor CPU model was an AMD EPYC7642 48-core processor. The running memory capacity was 80 GB, the solid-state drive (SDD) capacity was 50 GB, and the number of cores was 24. The graphics card (GPU) model was an NVIDIA GeForce RTX4090, the video memory was 24 GB, and the system environment was Ubuntu 20.04. The Python 3.8 programming language was used, as was the PyTorch 1.13 deep learning framework, and the parallel computing operator of CUDA 11.3 was adopted.

The network model parameter settings were as follows. The stand-alone, single-card mode was used, and the official YOLOv8n-obb pre-training model was adopted. The input size of 640 × 640 × 3 was used, the number of samples in each batch of images was 16, the number of workers was 8, the training optimizer was adaptive moment estimation (Adam), the number of training epochs was 300, and the learning rate was decreased using cosine annealing (cos). To prevent the model from overfitting, the weight decay was set to 0, the initial learning rate was 0.001, the weight decay coefficient was 0.0005, the momentum factor was 0.937, and the mosaic and mixup data enhancement factors were set to 1.0 and 0.5, respectively.

3.2. Experimental results of the improved model

This section presented the experimental results of different improved modules in the POD-YOLO model, including results from various lightweight model modules, different downsampling modules, and different angle regression loss functions.

3.2.1. Experimental results of different lightweight feature extraction modules

In this study, an improved YOLOv8n rotated object detection model, POD-YOLO, was constructed. In the proposed model, the backbone network CSPDarkNet was replaced with CSPDPNet, which integrated the concepts of PConv and DualConv. The downsampling module ConvModule was also changed to the SPD-Conv module. Additionally, a detection head was introduced from stage1 of the backbone network.

This experiment was based on the POD-YOLO model, for which different lightweight backbone networks, such as MobileNetV2, MobileNetV3, GhostNetV2, ShuffleNetV2, and FasterBlock, were used to replace the feature extraction module in the backbone network. The experimental results were analyzed and compared to investigate the feature extraction performance of the improved rotated object detection model. The training parameters were used to train the altered network models one by one according to the experimental setup described in Section 3.1.

The experimental results for the YOLOv8n rotated object detection model with four detection heads, along with each of the improved rotated object detection models, are presented in Table 1. When the CSPDPNet module was used as the backbone feature extraction network of YOLOv8n, the number of parameters was 2.50 million, the FLOPs number was 11.60 G, the mean average precision (mAP) was 96.50%, the precision (P) was 93.30%, the recall (R) was 93.70 %, and the network forward inference time was 9.01 ms.

Compared with the backbone CSPDarkNet module, the number of parameters was reduced by 20.8% and the FLOPs number was reduced by 11.2%, while the model detection accuracy and forward inference speed remained basically the same.

Compared with other lightweight feature extraction networks, namely MobileNetV2, MobileNetV3, GhostNetV2, and ShuffleNetV2, the mAP value of the CSPDPNet module was respectively increased by 0.20%, 0.10%, 0.70%, and 0.40%. And the forward reasoning time was respectively reduced by 6.54, 5.73, 13.45, and 3.90 ms. Compared with FasterBlock, the forward reasoning time and detection accuracy were basically the same.

The experimental results show that when CSPDPNet was used as the feature extraction network of YOLOv8n, the model demonstrated significant advantages in the number of model parameters and the FLOPs number. The detection accuracy and detection time were on the same level as when CSPDarkNet was used as the feature extraction network, and, compared with other lightweight backbone networks, CSPDPNet contributed to a faster forward inference speed and higher detection accuracy.

3.2.2. Experimental results of different downsampling modules

The downsampling module in the YOLOv8n rotated object detection model uses a standard convolution to change the convolution step size to achieve downsampling, which may result in the loss of feature information during the downsampling process. In this experiment, the detection performance of various downsampling modules, such as HWD, CGNet, ADown, and SPD-Conv, was tested on low-resolution images and small target objects.

The results exhibited in Table 2 reveal that when using the SPD-Conv module as the downsampling operator for the YOLOv8n model, the mAP value was 96.80%, the precision was 93.80%, the recall was 94.50%, AP_{bud} was 94.20%, and AP_{potato} was 99.50%. Compared to the standard convolutional downsampling module, the mAP value of the model was 0.1% higher and the bud eye prediction accuracy was 0.3% higher. Compared with the HWD, CGNet, and ADown modules, the mAP value was respectively improved by 0.7%, 1.3%, and 0.4%, AP_{bud} was respectively improved by 0.7%, 2.7%, and 0.8%, and AP_{potato} was basically the same. It is evident that the use of the SPD-Conv module to detect the orientation and bud eye position of potatoes displays a greater advantage. This is partly because the resolution of the input image itself is low. Furthermore, the bud eyes are small target objects, and the feature information of the target can be obtained at a finer granularity by using the SPD-Conv module. Fig. 10 presents the heat map visualization of the 15th layer of the POD-YOLO model. From the figure, it can be determined that the model was more focused on the bud eye position after the addition of the SPD-Conv module. Fig. 11 displays the results of different downsampling modules for the detection of the potato orientation and bud eye position. The use of SPD-Conv as the downsampling

Table 1
Experimental results of different lightweight feature extraction modules.

Module	mAP (%)	P (%)	R (%)	Latency (ms)	Params (M)	FLOPs (G)
CSPDarkNet	96.90	95.50	91.80	9.84	3.02	12.90
MobileNetV2	96.30	93.20	93.40	15.55	3.76	14.80
MobileNetV3	96.40	93.70	93.30	14.74	5.66	14.20
GhostNetV2	95.80	94.10	91.10	22.46	6.34	13.40
ShuffleNetV2	96.10	92.40	93.50	12.91	2.80	12.20
FasterBlock	96.30	94.00	93.40	10.32	2.65	11.90
CSPDPNet	96.50	93.30	93.70	9.01	2.50	11.60

Table 2
Experimental results of different downsampling modules.

Model	mAP (%)	P (%)	R (%)	AP_{bud}	AP_{potato}
ConvModule	96.70	93.80	95.00	93.90	99.40
HWD	96.10	94.00	93.60	93.50	99.50
CGNet	95.50	92.70	91.80	91.50	99.50
ADwon	96.40	92.70	99.90	93.40	99.50
SPD-Conv	96.80	93.80	94.50	94.20	99.50

module resulted in the detection of more information about the bud eyes, especially those at the edge position, such as the position marked by the red arrow in the figure graphic.

3.2.3. Experimental results of different angular regression loss functions

For an object whose shape is close to a square, the existing angular regression loss function cannot adequately fit the shape of the object. Thus, in this experiment, a variety of different angular regression loss functions were used to test and analyze the prediction accuracy and angular bounding box regression of the proposed POD-YOLO rotated object detection model.

The experimental results listed in Table 3 reveal that when using the KFIoU angular regression loss function, the highest values of 97.2% and 79.7% were respectively achieved for $mAP_{0.5}$ and $mAP_{0.75}$, thus displaying respective improvements of 0.4% and 2.6% as compared to ProbIoU.

Compared to the other loss functions, namely GWD and KLD, $mAP_{0.5}$ exhibited respective improvements of 4.3% and 5.3%. Fig. 12 shows the results of using different angular regression loss functions to predict the potato orientation and bud eye position. The KFIoU angular regression loss function was found to have a better regression effect when used to predict bud eyes at the edge position.

It is evident that using the KFIoU angular loss function achieved some advantages in terms of the detection accuracy and prediction effect. Thus, KFIoU was chosen as the angular regression loss function for POD-YOLO.

After analysis, the possible reason for this result is that KFIoU uses the centroid distance loss function, which combines the advantages of the distance loss function and the Gaussian probability distribution. This makes it easier for the model to learn the difference between the bounding box loss functions, and ultimately improves the overall prediction accuracy of the model.

3.3. Experimental results before and after improvement

The original YOLOv8n framework employs CSPDarkNet as its backbone, standard convolution modules for downsampling, and ProbIoU as the angle regression loss function. In the improved POD-YOLO, CSPDPNet was used as the backbone, reducing model parameters while maintaining high detection accuracy, as detailed in Table 1. The standard convolution modules in the downsampling process were replaced with SPD-Conv, improving the detection of small targets, such as potato buds, and enhancing performance on low-resolution images. KFIoU replaced ProbIoU as the angle regression loss function, improving bounding box quality and overall detection precision.

The visual comparison between YOLOv8n and POD-YOLO for potato bud detection and potatoes orientation were illustrated in Figs. 13 and 14. YOLOv8n showed frequent missed detections of potato buds, whereas POD-YOLO accurately detected most bud eye locations. For orientation detection, YOLOv8n struggled with angle regression (e.g., Fig. 12(d)), while POD-YOLO better aligned bounding boxes and provided more precise angle predictions.

In Table 4, POD-YOLO achieved a significant reduction in model parameters, approximately halving those of YOLOv8n. Despite the

Table 3
Comparison of regression loss function results from different angles.

Loss Function	$mAP_{0.5}$ (%)	$mAP_{0.75}$ (%)	P (%)	R (%)
ProbIoU	96.80	76.80	94.90	93.40
GWD	92.90	72.40	87.70	90.90
KLD	91.90	71.40	87.30	87.90
KFIoU	97.20	79.40	95.20	94.00

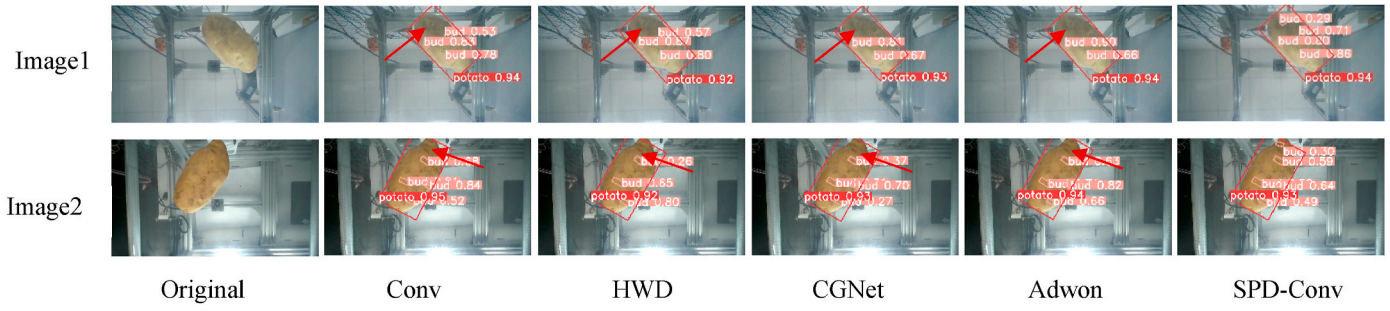

Fig. 10. Heatmap visualization results of different downsampling modules.

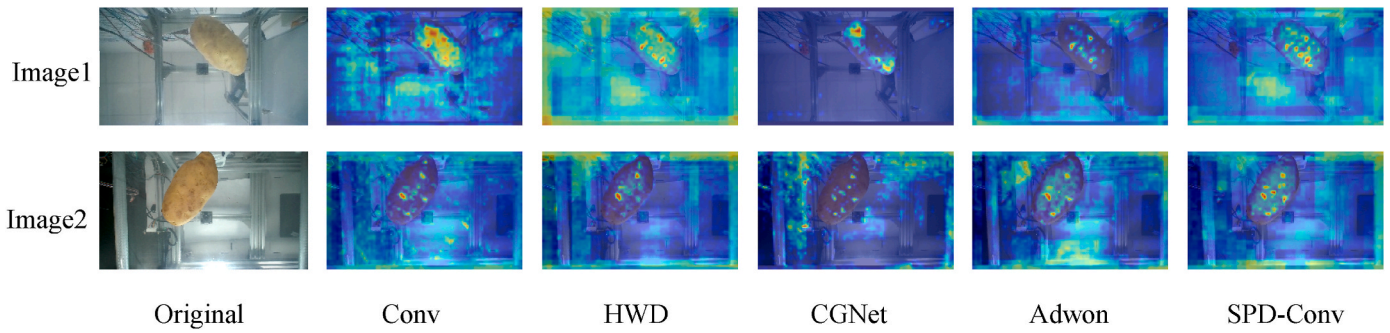

Fig. 11. The results of detecting potato orientation and bud eye using different downsampling.

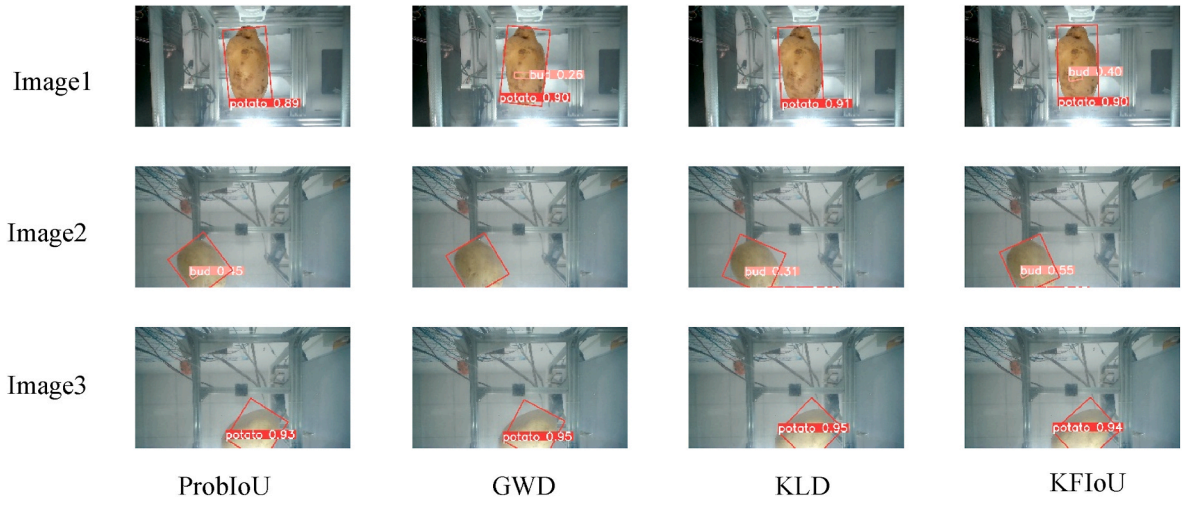

Fig. 12. Results of detecting potato orientation and bud eye position using different angle regression loss.

Fig. 13. YOLOv8n visualization detection results.

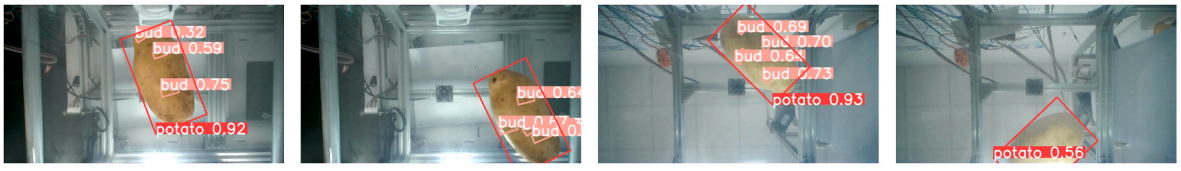

Fig. 14. POD-YOLO visualization detection results.

Table 4
Test results of the model before and after improvement.

Model	mAP (%)	AP_{bud} (%)	AP_{potato} (%)	Params (M)	Latency (ms)
YOLOv8n	96.80	93.90	99.40	3.08	8.10
POD-YOLO	97.20	95.00	99.40	1.75	9.01

reduced complexity, POD-YOLO increased potato bud detection accuracy by 1.1% and improved the overall mAP by 0.4%, demonstrating superior performance in both efficiency and detection quality.

3.4. Ablation experiment

In this experiment, based on the YOLOv8n rotated object detection model, the improved modules were added one by one to test the overall performance of the model in different states. The evaluation focused on the mAP value, the latency, and the number of model parameters.

Table 5 reveals that when no module was added, the mAP value of YOLOv8n was 96.8%, the forward reasoning time was 8.1 ms, and the number of model parameters was 3.08 million. After adding the

Table 5
POD-YOLO model ablation experimental results.

Model	mAP (%)	Latency (ms)	Params (M)
YOLOv8n	96.80	8.10	3.08
YOLOv8n + CSPDP + Conv + ProbIoU	96.70	10.12	2.11
YOLOv8n + CSPDP + SPD-Conv + ProbIoU	96.80	10.04	1.75
YOLOv8n + CSPDP + SPD-Conv + KFIoU	97.20	9.01	1.75

CSPDPNet module, the number of parameters of the improved YOLOv8n model decreased significantly to 31.4% of the original number of parameters, but the model detection accuracy also decreased to 96.70%. On this basis, the SPD-Conv module was added; this module is specially designed for small target objects and low-resolution images. The mAP value of the model was increased to 96.80%, and the number of parameters also decreased slightly. Finally, after changing the original ProbIoU regression loss function to KFIoU, the mAP value of the model increased again by 0.4%, the other parameters were unchanged.

These results demonstrate that the proposed POD-YOLO rotated object detection model using CSPDPNet, SPD-Conv, and KFIoU has

strong prediction ability, which provides a basis for the study of edge devices and embedded devices.

3.5. Experimental results of different rotated object detection algorithms

In this experiment, the potato test set was used as the image data for the evaluation of the overall performance of the models. The training parameters were set according to those described in Section 2.1, the pre-trained models were set according to the files provided in MMRotate (MMRotate Contributors, 2022), and the training data were the same for each model. To validate the model performance, the test set data were subjected to forward inference using the FasteRCNN, RetinaNet, S^2A-Net, R^3Det, RoI Transformer, and POD-YOLO rotated object detection models. The number of parameters of each detection model, the forward inference time, the overall detection accuracy of the model (mAP value), and the bud and potato AP values were investigated.

From the results reported in Table 6, it can be seen that the improved YOLOv8n rotated detection model (POD-YOLO) achieved an overall mAP value of 97.2%, thus exhibiting respective improvements of 39.3%, 35.2%, and 47.4%, 47.4%, and 47.4% as compared to FasterCNN, RoI Transformer, RetinaNet, S^2A-Net, and R^3Det. In terms of the number of model parameters, the improved rotating frame detection model achieved a significant advantage over the other five rotated object detection models, and had only 1.75 million parameters. In terms of the detection time, the forward inference speed of POD-YOLO was 9.01 ms, a significant advantage over the other models.

In addition, to more clearly observe the detection results, the five aforementioned rotated object detection algorithms were selected as the research objects, and five images predicted by each algorithm were randomly selected for analysis. The visualization results are shown in Fig. 15. The POD-YOLO model correctly detected all the potato and bud eye information in the pictures, and the regression bounding box accurately fit the potato orientation. When using the FasterCNN, RetinaNet, S^2A-Net, R^3Det, and RoI Transformer algorithms to detect the potato orientation and bud eye locations, most of the bounding boxes could not accurately fit the potato orientation, with the exception of RetinaNet. When detecting the potato bud eye position information, only FasterCNN and RoI Transformer could detect a portion of the bud eye positions, while the other three models performed poorly, and relatively serious leakage occurred.

In summary, the results of the evaluation indicators show that POD-YOLO has strong detection ability, and can adapt to the different angles and positions of the potato in the image. Simultaneously, it can meet the needs of real-time detection.

4. Discussion

This paper proposes a method for potatoes orientation and bud eye detection, filling the technical gap in existing methods that fail to simultaneously detect both bud eye positions and potatoes orientation. The proposed POD-YOLO model improves detection accuracy in three key aspects: the design of a lightweight backbone network, modification of the downsampling module, and the introduction of a bounding box loss function.

First, the CSPDPNet module was designed to reduce the number of parameters in the backbone network while capturing rich feature information. The CSPDPNet module was a modification of the CSPDarkNet module, where the Darknet Bottleneck module in CSPDarkNet was replaced by two PConv modules. This design was inspired by the PConv concept used in DualConv and FasterNet. Compared to MobileNetV2, MobileNetV3, GhostNetV2, and ShuffleNetV2, the CSPDPNet model showed an improvement in mAP (mean average precision) and a significant reduction in the number of parameters. The SPD-Conv downsampling module effectively captured low-resolution and small target object information, which was particularly beneficial for detecting bud eyes, a small target. Using SPD-Conv to extract bud eye features enables more accurate and efficient capture of these features. The rotated bounding box regression loss function KFIoU leveraged the Kalman filter method to obtain the distribution probability of bounding boxes, calculated the overlap between the ground truth and predicted boxes, and updated the model's training accuracy.

The potatoes orientation and bud eye detection model constructed using this approach can accurately and rapidly detect the positions of bud eyes and the orientation of potatoes. This provides technical support for automated potato-cutting robots and offers solutions for advancing agricultural automation.

The cutting principle of the intelligent potato-cutting robot is to first identify the eye positions and then calculate the cutting angle based on the eye positions (Yang et al., 2023). However, when calculating the cutting angle, considering only the bud eye positions does not allow for real-time tracking of the potato's orientation and position, making it difficult to meet the cutting requirements. Therefore, both the orientation of the potatoes and the bud eye positions must be considered to accurately calculate the cutting angle. This study primarily focuses on the orientation of potatoes and the positions of bud eye, laying a solid foundation for the next step of calculating the cutting angle in the potato-cutting robot.

Currently, object detection algorithms based on CNNs are widely applied in agriculture, and achieve real-time and accurate detection in complex environments. They have gained widespread recognition, such as in the detection of kiwifruit (Suo et al., 2021), passion fruit (Tu et al., 2020), green peppers (Li et al., 2021), dragon fruit (Nan et al., 2023), and tomatoes (Zeng et al., 2023). However, there has been limited research on the detection of the direction and bud eye position of potatoes. The POD-YOLO algorithm proposed in this study successfully performs bud eye position detection and potatoes orientation, yielding satisfactory results. However, there are several areas for improvement in future research. (1) When potatoes are at the edge of the image, the regression effect of the bounding box is poor, sometimes failing to fit the potato boundaries well. This results in incorrect regression angles. A possible reason for this is that the bounding boxes lack complete potato feature information. Thus, object features are lost during forward propagation, causing the model to misjudge and provide incorrect regression results. (2) The difficulty in bud eye recognition may stem from an insufficient number of bud eye samples, which prevents the model from effectively learning relevant features. Future studies could consider incorporating self-attention mechanisms to enhance the model's focus on bud eye information. (3) For bounding boxes similar to

Table 6
Evaluation results of different rotated object detection models.

Model	Backbone	mAP (%)	Weight (MB)	AP (%)		Params (M)	Detection Speed (FPS)	Latency (ms)
				bud	potato			
FasterRCNN	R-50	57.90	330.30	16.80	40.20	41.12	11.5	87.2
RoI Trans.	R-50	62.00	441.60	24.90	99.00	55.03	9.6	104.3
RetinaNet	R-50	49.80	290.50	0	99.50	36.15	13.3	75.1
S^2A-Net	R-50	49.80	309.60	9.40	90.20	38.54	11.7	85.7
R^3Det	R-50	49.80	334.00	0.10	99.40	41.60	9.7	103.3
POD-YOLO	CSPDP	97.20	3.89	95.00	99.40	1.75	110.9	9.01

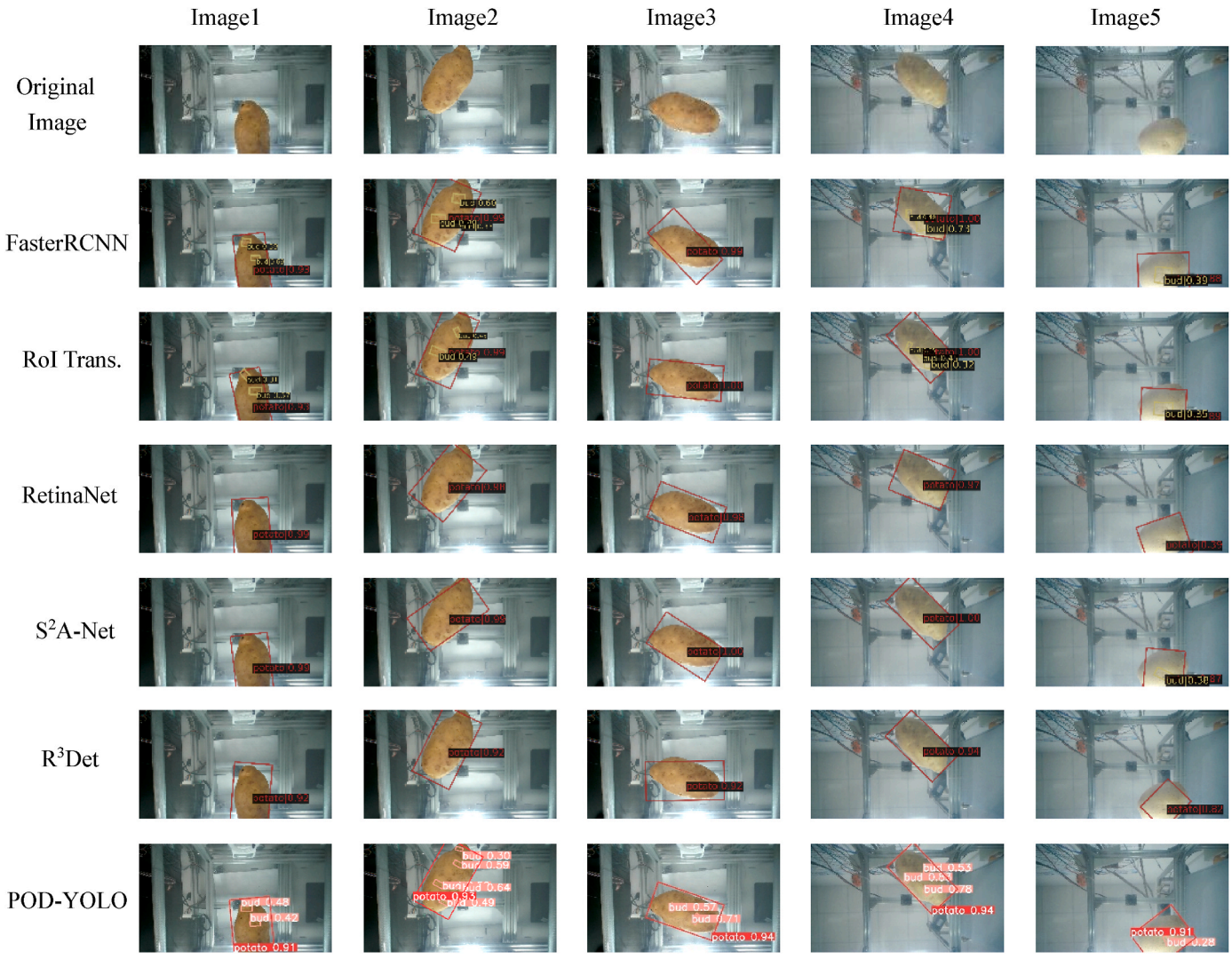

Fig. 15. Results of different rotated object detection algorithms.

a square the angle regression loss function still cannot completely avoid the generation of a certain degree of estimation error during the calculation process. This causes the bounding box to not align precisely with the object edges, and the center point of the bounding box deviates from the object center.

5. Conclusions

This study proposed a novel lightweight model, POD-YOLO, for detecting potato orientation and bud eye locations. The model effectively captures the orientation and bud eye positions of potatoes under various conditions. CSPDarkNet in the backbone network was replaced with CSPDPNet to reduce the number of model parameters. The SPD-Conv downsampling module was introduced to replace the convolutional modules in both the backbone and neck, enhancing the detection capability for small objects like bud eyes and low-resolution images. The angle regression loss function, ProbIoU, was replaced with KFIoU to improve the regression quality of rotated bounding boxes and the overall detection performance of the model. The POD-YOLO model achieved a mAP of 97.2%, a detection time of 9.01 ms, and a parameter count of 175 million, meeting the requirements for real-time, accurate, and lightweight detection in intelligent cutting robots. In future research, we will focus on improving the orientation recognition of occluded potatoes and enhancing the detection accuracy of bud eyes, providing technical support for the practical application of the potato-cutting robot and offering solutions for the development of agricultural automation.

CRediT authorship contribution statement

Jie Huang: Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Xiangyou Wang:** Writing – review & editing, Resources, Project administration, Funding acquisition. **Chengqian Jin:** Visualization, Software, Formal analysis. **Fernando Auat Cheein:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Xinyu Yang:** Validation, Investigation, Formal analysis.

Declaration of competing interest

We would like to submit the enclosed manuscript entitled “Estimation of the orientation of potatoes and detection bud eye position using Potato Orientation Detection You Only Look Once with fast and accurate features for the movement strategy of intelligent cutting robots”, which we wish to be considered for publication in “Engineering Applications of Artificial Intelligence”. The work was carried out by Jie Huang, Xiangyou Wang*, Chengqian Jin, Fernando Auat Cheein, Xinyu Yang. No conflict of interest exists in the submission of this manuscript,and the

manuscript is approved by all authors for publication. I would like to declare on behalf of my co-authors that the work described was original research that has not been published previously,and not under consideration for publication elsewhere, in whole or in part. All the authors listed have approved the manuscript that is enclosed.

Acknowledgments

The authors gratefully acknowledge the Key Laboratory of Modern Agricultural Equipment, Ministry of Agriculture and Rural Affairs , P. R. China (No. HT20230528) for supporting this research.

Data availability

Data will be made available on request.

References

Ariza-Sentís, M., Vélez, S., Martínez-Peña, R., Baja, H., Valente, J., 2024. Object detection and tracking in Precision Farming: a systematic review. Comput. Electron. Agric. 219, 108757. https://doi.org/10.1016/j.compag.2024.108757.

Bargoti, S., Underwood, J.P., 2017. Image segmentation for fruit detection and yield estimation in apple orchards. J. Field Robot. 34, 1039–1060. https://doi.org/10.1002/rob.21699.

Cardellicchio, A., Solimani, F., Dimauro, G., Petrozza, A., Summerer, S., Cellini, F., Renó, V., 2023. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. Comput. Electron. Agric. 207, 107757. https://doi.org/10.1016/j.compag.2023.107757.

Chen, J., Kao, S., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H.G., 2023. Run, don't walk: chasing higher FLOPS for faster neural networks. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Vancouver, BC, Canada, pp. 12021–12031. https://doi.org/10.1109/CVPR52729.2023.01157.

Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, Santiago, Chile. https://doi.org/10.1109/ICCV.2015.169.

Ganesan, G., Chinnappan, J., 2022. Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: an optimized model. J. Field Robot. 39, 1087–1111. https://doi.org/10.1002/rob.22089.

Huang, J., Wang, X., Wu, H., Liu, S., Yang, X., Liu, W., 2023. Detecting potato seed bud eye using lightweight convolutional neural network (CNN). Trans. Chin. Soc. Agric. Eng. 39, 172–182. https://doi.org/10.11975/j.issn.1002-6819.202303035.

Johnson, C.M., Auat Cheein, F., 2023. Machinery for potato harvesting: a state-of-the-art review. Front. Plant Sci. 14, 1156734. https://doi.org/10.3389/fpls.2023.1156734.

Jiang, H., Gilbert Murengami, B., Jiang, L., Chen, C., Johnson, C., Auat Cheein, F., Fountas, S., Li, R., Fu, L., 2024. Automated segmentation of individual leafy potato stems after canopy consolidation using YOLOv8x with spatial and spectral features for UAV-based dense crop identification. Comput. Electron. Agric. 219, 108795. https://doi.org/10.1016/j.compag.2024.108795.

Jocher, G., 2020. YOLOv5 by ultralytics. https://doi.org/10.5281/zenodo.3908559.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO. https://github.com/ultralytics/ultralytics.

Kaur, P., Singh, M.P., Mishra, A.M., Shankar, A., Singh, P., Diwakar, M., Nayak, S.R., 2023. DELM: deep ensemble learning model for multiclass classification of super-resolution leaf disease images. Turk. J. Agric. For. 47, 727–745. https://doi.org/10.55730/1300-011X.3123.

Koirala, A., Walsh, K., Wang, Z., McCarthy, C., 2019. Deep learning - method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162, 219–234. https://doi.org/10.1016/j.compag.2019.04.017.

Liao, H., Zhou, Z., Liu, N., Zhang, Y., Xu, G., Wang, Z., Mumtaz, S., 2023. Cloud-Edge-Device collaborative reliable and communication-efficient digital twin for low-carbon electrical equipment management. IEEE Trans. Ind. Inf. 19, 1715–1724. https://doi.org/10.1109/TII.2022.3194840.

Li, X., Pan, J., Xie, F., Zeng, J., Li, Q., Huang, X., Liu, D., Wang, X., 2021. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. Comput. Electron. Agric. 191, 106503. https://doi.org/10.1016/j.compag.2021.106503.

Li, Y., Li, T., Niu, Z., Wu, Y., Zhang, Z., Hou, J., 2018. Potato bud eyes recognition based on three-dimensional geometric features of color saturation. Trans. Chin. Soc. Agric. Eng. 158–164. https://doi.org/10.11975/j.issn.1002-6819.2018.24.019.

Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J., 2018. FOTS: fast oriented text spotting with a unified network. arXiv.org. https://doi.org/10.48550/arXiv.1801.01671v1.

Magalhães, S., Castro, L., Moreira, G., Santos, F., Cunha, M., Dias, J., Moreira, A., 2021. Evaluating the single-shot MultiBox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse. Sensors 21, 3569. https://doi.org/10.3390/s21103569.

Marset, W.V., P'erez, D.S., D'iaz, C.A., Bromberg, F., 2021. Towards practical 2D grapevine bud detection with fully convolutional networks. Comput. Electron. Agric. 182, 105947. https://doi.org/10.1016/j.compag.2020.105947.

Milestone, 2024. Milestone. https://milestone-equipment.com/.

Mirhaji, H., Soleymani, M., Asakereh, A., Mehdizadeh, S.A., 2021. Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. Comput. Electron. Agric. 191, 106533. https://doi.org/10.1016/j.compag.2021.106533.

MMRotate Contributors, 2022. OpenMMLab rotated object detection toolbox and benchmark. https://github.com/open-mmlab/mmrotate.

Murrugarra-Llerena, J., Zeni, L.F. de A., Kirsten, L.N., Jung, C., 2021. Probabilistic intersection-over-union for training and evaluation of oriented object detectors. IEEE Trans. Image Process. 33, 671–681. https://doi.org/10.1109/TIP.2023.3348697.

Nan, Y., Zhang, H., Zeng, Y., Zheng, J., Ge, Y., 2023. Intelligent detection of Multi-Class pitaya fruits in target picking row based on WGB-YOLO network. Comput. Electron. Agric. 208, 107780. https://doi.org/10.1016/j.compag.2023.107780.

Onoufriou, G., Hanheide, M., Leontidis, G., 2023. Premonition Net, a multi-timeline transformer network architecture towards strawberry tabletop yield forecasting. Comput. Electron. Agric. 208, 107784. https://doi.org/10.1016/j.compag.2023.107784.

Pan, X., Ren, Y., Sheng, K., Dong, W., Yuan, H., Guo, X., Ma, C., Xu, C., 2020. Dynamic refinement network for oriented and densely packed object detection. arXiv.org. https://doi.org/10.48550/arXiv.2005.09973.

Paul, A., Machavaram, R., Ambuj, Kumar, D., Nagar, H., 2024. Smart solutions for capsicum Harvesting: unleashing the power of YOLO for Detection, Segmentation, growth stage Classification, Counting, and real-time mobile identification. Comput. Electron. Agric. 219, 108832. https://doi.org/10.1016/j.compag.2024.108832.

Peterson, E., 2024. All star manufacturing & design LLC. http://www.allstarmfgllc.com/.

Prasetyo, E., Suciati, N., Fatichah, C., 2022. Yolov4-tiny with wing convolution layer for detecting fish body part. Comput. Electron. Agric. 198, 107023. https://doi.org/10.1016/j.compag.2022.107023.

Roy, A., Bhaduri, J., 2022. Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4. Comput. Electron. Agric. 193, 106694. https://doi.org/10.1016/j.compag.2022.106694.

Redmon, J., Farhadi, A., 2018. YOLOv3: an incremental improvement. arXiv.org. https://doi.org/10.48550/arXiv.1804.02767.

Samant, D., Dhawan, R., Mishra, A.K., Bora, V., Diwakar, M., Singh, P., 2023. Potato leaf disease detection using deep learning. In: 2023 IEEE World Conference on Applied Intelligence and Computing (AIC). Presented at the 2023 IEEE World Conference on Applied Intelligence and Computing (AIC), pp. 752–757. https://doi.org/10.1109/AIC57670.2023.10263960. IEEE, Sonbhadra, India.

Sengupta, S., Lee, W.S., 2014. Identification and determination of the number of immature green citrus fruit in a canopy under different ambient light conditions. Biosyst. Eng. 117, 51–61. https://doi.org/10.1016/j.biosystemseng.2013.07.007.

Song, C., Zhang, F., Li, J., Zhang, J., 2022. Precise maize detasseling base on oriented object detection for tassels. Comput. Electron. Agric. 202, 107382. https://doi.org/10.1016/j.compag.2022.107382.

Sunkara, R., Luo, T., 2022. No more strided convolutions or pooling: a new CNN building block for low-resolution images and small objects. arXiv.org. https://doi.org/10.48550/arXiv.2208.03641.

Suo, R., Gao, F., Zhou, Z., Fu, L., Song, Z., Dhupia, J., Li, R., Cui, Y., 2021. Improved multi-classes kiwifruit detection in orchard to avoid collisions during robotic picking. Comput. Electron. Agric. 182, 106052. https://doi.org/10.1016/j.compag.2021.106052.

Tian, Q., Lin, Y., Guo, X., Wen, J., Fang, Y., Rodriguez, J., Mumtaz, S., 2019. New security mechanisms of high-reliability IoT communication based on radio frequency fingerprint. IEEE Internet Things J. 6, 7980–7987. https://doi.org/10.1109/JIOT.2019.2913627.

Trivedi, P., Narayan, Y., Ravi, V., Kumar, P., Kaur, P., Tabianan, K., Singh, P., Diwakar, M., n.d. Plant leaf disease detection and classification using segmentation encoder techniques. Open Agric. J. 2024, 18: e18743315321139. https://doi.org/10.2174/0118743315321139240627092707.

Tu, S., Pang, J., Liu, H., Nan, Z., Chen, Y., Zheng, C., Wan, H., Xue, Y., 2020. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precis. Agric. 21, 1072–1091. https://doi.org/10.1007/s11119-020-09709-3.

UN Food & Agriculture Organization, 2023. Production of potatoes by the world. https://www.fao.org/faostat/en/#data. (Accessed 11 October 2023).

Wang, C.-Y., Yeh, I.-H., Liao, H.-Y.M., 2024. YOLOv9: learning what you want to learn using programmable gradient information. arXiv.org. https://doi.org/10.48550/arXiv.2402.13616.

Wang, T., Zhu, X., Pang, J., Lin, D., 2021. FCOS3D: fully convolutional one-stage monocular 3D object detection. arXiv.org. https://doi.org/10.48550/arXiv.2104.10956.

Wang, X., Zhu, S., Li, X., Li, T., Wang, L., Hu, Z., 2020. Design and experiment of directional arrangement vertical and horizontal cutting of seed potato cutter. Trans. Chin. Soc. Agric. Mach. 51, 334–345. https://doi.org/10.6041/j.issn.1000-1298.2020.06.036.

Wu, G., Li, B., Zhu, Q., Huang, M., Guo, Y., 2020. Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy. Comput. Electron. Agric. 174, 105475. https://doi.org/10.1016/j.compag.2020.105475.

Yang, X., Yan, J., Feng, Z., He, T., 2020. R^3Det: refined single-stage detector with feature refinement for rotating object. arXiv.org. https://doi.org/10.1609/aaai.v35i4.16426.

Yang, X., Zhou, Y., Zhang, G., Yang, J., Wang, W., Yan, J., Zhang, X., Tian, Q., 2023. The KFIOU loss for rotated object detection. arXiv.org. https://doi.org/10.48550/arXiv.2201.12558.

Yang, Y., Liu, Z., Huang, M., Zhu, Q., Zhao, X., 2023. Automatic detection of multi-type defects on potatoes using multispectral imaging combined with a deep learning model. J. Food Eng. 336, 111213. https://doi.org/10.1016/j.jfoodeng.2022.111213.

Yi, J., Wu, P., Liu, B., Huang, Q., Qu, H., Metaxas, D., 2020. Oriented object detection in aerial images with box boundary-aware vectors. arXiv.org. https://doi.org/10.48550/arXiv.2008.07043.

Zhou, M., Chen, L., Wei, X., Liao, X., Mao, Q., Wang, H., Pu, H., Luo, J., Xiang, T., Fang, B., 2023a. Perception-Oriented U-shaped transformer network for 360-degree no-reference image quality assessment. IEEE Trans 69, 396–405. https://doi.org/10.1109/TBC.2022.3231101 on Broadcast.

Zhou, M., Lan, X., Wei, X., Liao, X., Mao, Q., Li, Y., Wu, C., Xiang, T., Fang, B., 2023b. An end-to-end blind image quality assessment method using a recurrent network and self-attention. IEEE Trans. Broadcast. 69, 369–377. https://doi.org/10.1109/TBC.2022.3215249.

Zhou, M., Zhao, X., Luo, F., Luo, J., Pu, H., Xiang, T., 2024. Robust RGB-T tracking via adaptive modality weight correlation filters and cross-modality learning. ACM Trans. Multimed Comput. Commun. Appl 20, 1–20. https://doi.org/10.1145/3630100.

Zhou, S., Zhong, M., Chai, X., Zhang, N., Zhang, Y., Sun, Q., Sun, T., 2024. Framework of rod-like crops sorting based on multi-object oriented detection and analysis. Comput. Electron. Agric. 216, 108516. https://doi.org/10.1016/j.compag.2023.108516.

Zeng, T., Li, S., Song, Q., Zhong, F., Wei, X., 2023. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. Comput. Electron. Agric. 205, 107625. https://doi.org/10.1016/j.compag.2023.107625.

Zhao, J., Yan, J., Xue, T., Wang, S., Qiu, X., Yao, X., Tian, Y., Zhu, Y., Cao, W., Zhang, X., 2022. A deep learning method for oriented and small wheat spike detection (OSWSDet) in UAV images. Comput. Electron. Agric. 198, 107087. https://doi.org/10.1016/j.compag.2022.107087.

Zhong, J., Chen, J., Mian, A., 2023. DualConv: dual convolutional kernels for lightweight deep neural networks. IEEE Transact. Neural Networks Learn. Syst. 34, 9528–9535. https://doi.org/10.1109/TNNLS.2022.3151138.