Fruit detection and picking keypoint localization with state space model and geometric compensation for strawberry harvesting robots

Dewei Kong[1], Yangjun Deng[1], Xinghui Zhu[1], Jintao Hao[1], Xianwen Yang[1], Xiangchao Zhou[2], Bo Qiao[1], Yiming Chen[1*]

(1. *College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China*;
2. *Department of Computing, The Hong Kong Polytechnic University, Hong Kong 999077, China*)

Abstract: Strawberry harvesting robots are vital for efficient harvesting and labor cost reduction. However, accurate strawberry detection and picking keypoint localization remain challenging due to dense fruit clusters, variable illumination, leaf and stem occlusions, and limited onboard computational resources. This study proposed STRAW-MAMBA, a lightweight state space model (SSM)-based network for accurate strawberry detection and keypoint localization in unconstrained environments. Specifically, the C2f-MAMBA block was developed to enhance global and multi-scale feature extraction for densely clustered or partly obscured strawberries. It redesigned the bottleneck layer of C2f as the Straw-Vim module, which integrates Hidden State Mixer-based State Space Duality (HSM-SSD) for global feature extraction and multi-scale convolutional attention (MSCA) for multi-scale feature extraction. Meanwhile, to preserve more fruit edge feature details, the large-stride convolution was optimized as a Haar wavelet downsampling module, thereby addressing illumination-induced edge blurring of strawberries while improving computational efficiency. In addition, the FasterNet block was superseded to reduce the model size further. Finally, a novel five-point geometric compensation method was proposed to address occlusions, reduce unpickable fruits, and minimize crop waste. Experiments show STRAW-MAMBA achieves 85.5% precision, 84.8% recall, and 89.7% mAP@0.5 for detection, and 88.7% precision, 77.5% recall, and 84.9% mAP@0.5 for keypoint localization, with 93.5 fps and only 1.9 M parameters. Ablation studies confirm each module's effectiveness, demonstrating suitability for greenhouse strawberry harvesting robots. This study can provide an efficient and robust visual perception framework, significantly advancing the practical deployment of automated harvesting systems in precision agriculture.

Keywords: strawberry detection, picking keypoint localization, STRAW-MAMBA, state space model, lightweight model
DOI: 10.25165/j.ijabe.20261904.10528

Citation: Kong D W, Deng Y J, Zhu X H, Hao J T, Yang X W, Zhou X C, et al. Fruit detection and picking keypoint localization with state space model and geometric compensation for strawberry harvesting robots. Int J Agric & Biol Eng, 2026; 19(4): 203–213.

1 Introduction

Strawberries are perennial herbaceous plants of the Rosaceae family and the *Fragaria genus*, widely cultivated worldwide[1]. In strawberry production, traditional manual harvesting methods are labor-intensive and time-consuming, making them unsuitable for modern large-scale production requirements. With the increasing application of artificial intelligence technology, research on fruit harvesting robots aimed at improving harvesting efficiency has become a key focus in the field of smart agriculture[2]. Developing strawberry harvesting robots holds significant importance for reducing manual labor, enhancing harvesting efficiency, and lowering production costs.

Unconstrained greenhouse environments pose challenges for robotic harvesting, including dense fruit clustering, variable illumination, and occlusions from leaves and stems, which hinder accurate detection and picking keypoint localization[3], while the fruit's delicate texture heightens the risk of mechanical damage[4], and high model complexity raises costs and limits deployment[5].

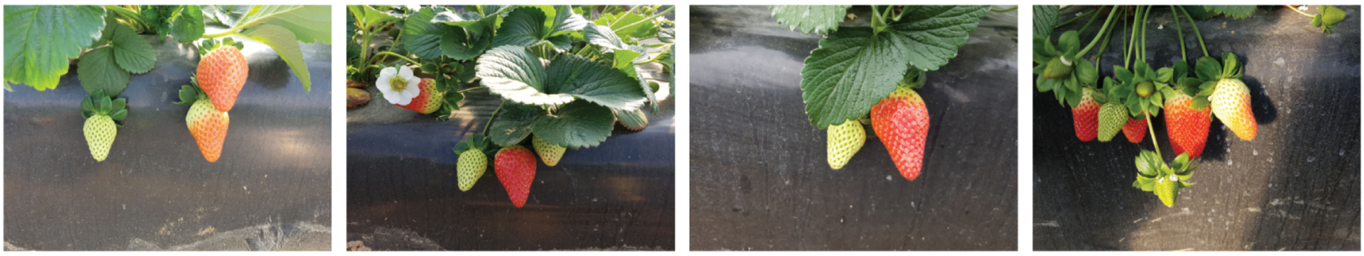
Figure 1 shows typical strawberry images captured by robots under greenhouse environment disturbances. Therefore, developing a lightweight, accurate model for strawberry detection and picking keypoint localization is crucial for improving harvesting quality and operational efficiency in robotic strawberry harvesting.

Early keypoint localization methods for fruit harvesting primarily relied on combining image processing with geometric compensation. For example, Hayashi et al. utilized a stereo vision system alongside HSV thresholding and connected region analysis to identify stem candidates and determine cutting points[3]. Huang et al.[6] employed OHTA color space segmentation and stem direction estimation to locate picking points. However, these traditional vision-based methods rely heavily on handcrafted features, exhibiting limited robustness and struggling to adapt to the complex, unstructured environments of real-world orchards.

Received date: 2026-02-20 **Accepted date:** 2026-07-01

Biographies: Dewei Kong, MS candidate, research interest: computer vision, smart agriculture, Email: davidkong539@stu.hunau.edu.cn; **Yangjun Deng**, PhD, Associate Professor, research interest: remote sensing image processing, smart agriculture, Email: dengyangjun@hunau.edu.cn; **Xinghui Zhu**, PhD, Professor, research interest: distributed systems and advanced database technology, Email: zhuxh@hunau.edu.cn; **Jintao Hao**, MS candidate, research interest: neural networks, smart agriculture, Email: haojt0104@stu.hunau.edu.cn; **Xianwen Yang**, MS, research interest: deep learning, image processing, Email: yxwwwen@stu.hunau.edu.cn; **Xiangchao Zhou**, MS, Research Assistant, research interest: knowledge graphs and large language models, Email: xiangchao.zhou@polyu.edu.hk; **Bo Qiao**, PhD, Associate Professor, research interest: agricultural knowledge graphs, smart agriculture, Email: qiaobo@hunau.edu.cn.

***Corresponding author: Yiming Chen**, PhD, Associate Professor, research interest: computer vision, smart agriculture. College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China. Tel: +86-13397497821, Email: chenym@hunau.edu.cn.



a. Strong light b. Low-light c. Strawberry keypoint is partially occluded d. Dense clusters of strawberries

Figure 1 Images from the strawberry dataset with environment disturbances

With the advent of deep learning, CNN-based multi-task frameworks have been widely adopted for simultaneous fruit detection and keypoint localization from RGB images. Various improved architectures have been customized for harvesting specific crops, such as combining YOLOv3 and Mask R-CNN for strawberries[7], or modifying YOLO variants for wolfberries[8], grapes[9], litchis[10], and coconuts[11] under complex conditions. While these data-driven models demonstrate strong robustness and adaptability, their inherent reliance on local convolution operations restricts their ability to model long-range dependencies, which ultimately limits localization accuracy in dense and highly occluded environments.

Recent methods increasingly combine neural networks with geometric calculations, where deep learning models first detect or segment the fruit, followed by geometric derivation of picking keypoints. For instance, several studies have utilized Mask R-CNN or YOLO variants to identify fruit axes and segment stems for cutting point extraction[12-14]. Notably, Ma et al.[15] proposed STRAW-YOLO, an enhanced YOLOv8-Pose model for lightweight strawberry localization, which employs a three-point geometric assumption to handle occlusions. However, the final localization in these approaches still relies heavily on simple, rigid geometric assumptions, which fundamentally limit their applicability, robustness, and overall effectiveness in complex scenarios.

Traditional CNNs struggle to capture long-range spatial relationships among strawberry stems and leaves due to limited receptive fields, reducing detection and keypoint localization accuracy. To address this, this study proposes STRAW-MAMBA, a novel deep neural network based on YOLOv8-Pose. STRAW-MAMBA integrates the Mamba time-series state space model for global dependency modeling while incorporating lightweight modifications for efficiency. The primary contributions of this study are as follows:

1) A novel module, C2f-MAMBA, for global and multi-scale feature extraction was developed to improve the accuracy of strawberry detection and picking keypoint localization. To the best of our knowledge, it is the first time that a state space model has been applied to the strawberry picking task.

2) The Haar wavelet downsampling technology is introduced into STRAW-MAMBA to redesign the large-stride convolution module for feature extraction and downsampling, effectively preserving more feature details and improving the accuracy of picking keypoint localization.

3) Lightweight improvements are applied to STRAW-MAMBA. Two C2f modules are integrated with FasterNet blocks, and the Haar wavelet downsampling is more efficient than standard large-stride convolutions, reducing deployment costs for harvesting robots.

4) A novel five-point geometric compensation method for occlusion is proposed to address occlusions, reduce unpickable fruits, and minimize crop waste.

2 Materials and methods

2.1 Overview of the method pipeline

As illustrated in Figure 2, the proposed framework involves three stages: data sensing, model training and compensation, and qualitative evaluation. First, strawberry images were collected and processed through data cleaning, annotation, dataset partitioning, and training set augmentation. Next, the proposed STRAW-MAMBA network was trained to detect strawberries and localize picking keypoints. To handle occlusion, a geometric compensation method is applied: if keypoints are completely visible, the predicted coordinates are directly used; if the keypoints are partly occluded, geometric compensation is performed to estimate their cutting point. Finally, a qualitative evaluation was conducted to comprehensively assess the robustness of the model in detecting bounding boxes and localizing picking keypoints under complex real-world environments.

2.2 Data collection and cleaning

From March 2024 to April 2025, experimental images were collected at Xiaomao Strawberry Greenhouse (28.146914°N, 113.126233°E) in Yuhua District, Changsha City, Hunan Province, China, which features a subtropical humid climate. The greenhouse uses a plastic-film facility cultivation system.

To capture lighting variations in greenhouse environments, 3621 RGB-D image pairs at a resolution of 1008×756 pixels were collected. During acquisition, the camera was angled at approximately 45° to minimize obstruction from branches and leaves, ensuring highly reliable data.

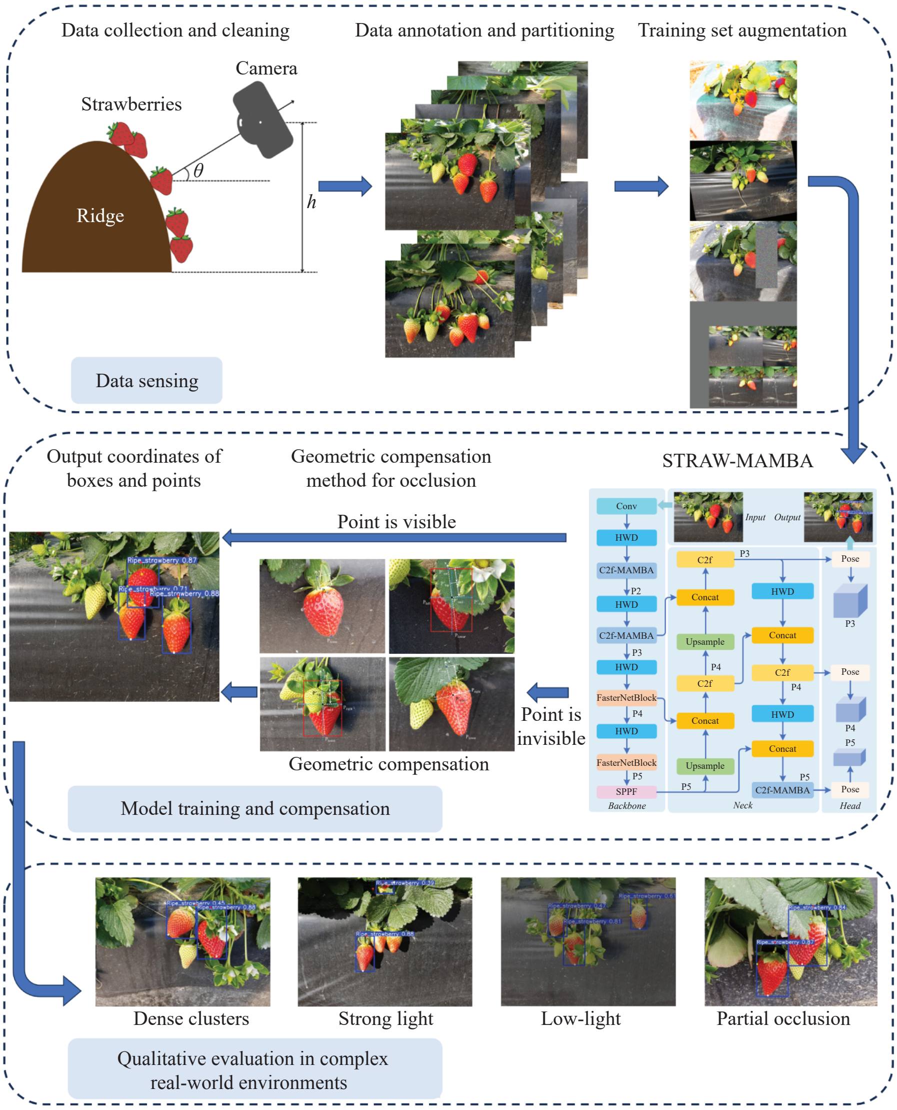
2.3 Data annotation and partitioning

A dataset of 3600 strawberry images under varying weather and lighting conditions was compiled and annotated using Labelme. Ma et al. proposed STRAW-YOLO with three keypoints for picking: cutting, top, and lower points[15]. However, this method can only estimate keypoints when two or more are visible, potentially causing missed detections. To address occlusions and reduce unpickable instances, this study proposes a five-point annotation method. The detailed definitions, spatial locations, and corresponding numerical IDs of the five picking keypoints are listed in Table 1.

Table 1 Definitions of the five picking keypoints

No.	Keypoint name	Description
1	Cutting point (KP1)	Around 8 mm above the strawberry top along the stem.
2	Top point (KP2)	Connection of the fruit to the stem.
3	Left grasping point (KP3)	Leftmost point of the fruit.
4	Right grasping point (KP4)	Rightmost point of the fruit.
5	Lower point (KP5)	Bottom of the fruit.

Bounding boxes, including the strawberry and cutting point, were annotated to assist keypoint localization. The three-point and



Note: The framework consists of three main stages: 1) data sensing, 2) model training alongside a geometric compensation strategy for handling occlusion, and 3) qualitative evaluation of bounding boxes and picking keypoints in complex real-world environments.

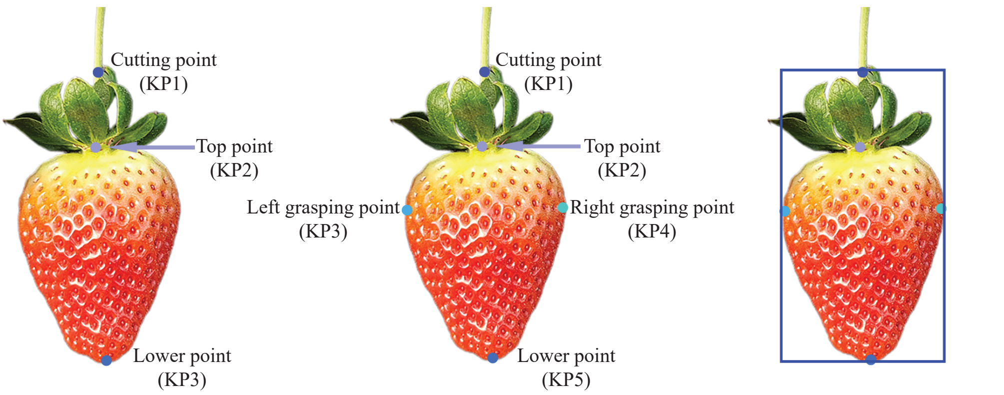
Figure 2 Overview of the proposed strawberry harvesting pipeline

five-point annotation methods are shown in Figure 3a and Figure 3b, respectively, and the bounding box annotation is shown in Figure 3c.

Annotation results are saved in JSON format, including strawberry ripeness, bounding box center coordinates, width and height, and picking keypoint coordinates with visibility. The dataset is split into training, validation, and test sets in an 8:1:1 ratio, comprising 2880, 360, and 360 images, respectively.

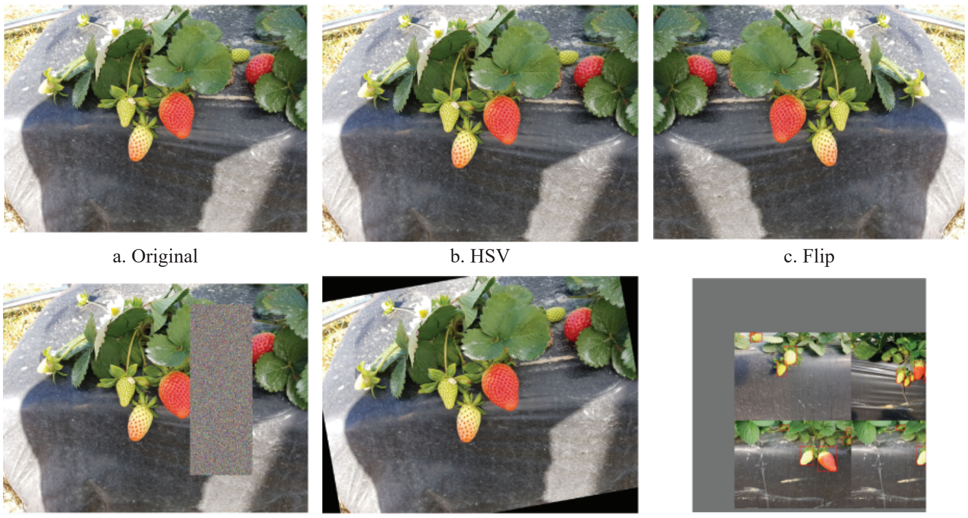
2.4 Training set augmentation

To expand the training dataset, multiple image augmentation strategies were employed. Dynamic augmentations, including HSV adjustments[16], geometric transformations, Mosaic[16], random erasing[17], and RandAugment[18] were applied probabilistically. These training augmentations are shown in Figure 4.



a. Three-point annotation[15]
b. Five-point annotation proposed in this study, adding left and right grasping points
c. Strawberry bounding box based on the five-point annotation

Figure 3 Comparison of strawberry annotation methods



a. Original b. HSV c. Flip

d. Random erasing e. Rotate f. Mosaic

Figure 4 Image augmentation methods

Augmentations were applied dynamically and probabilistically, yielding a total of approximately 14 400 images during training, which was sufficient for the task.

2.5 STRAW-MAMBA model

To overcome challenges in complex greenhouse environments such as fruit clustering, leaf occlusion, and variable lighting while maintaining low computational costs for robotic deployment, this study proposes the STRAW-MAMBA model. Inspired by YOLOv8-Pose, this lightweight architecture integrates multi-scale feature extraction to achieve highly accurate fruit detection and efficient keypoint localization.

As illustrated in Figure 5, the proposed architecture features three core enhancements. First, the C2f bottleneck is reconstructed into C2f-MAMBA blocks using a novel Straw-Vim unit, which combines the HSM-SSD state-space mechanism with multi-scale convolutional attention (MSCA) to capture long-range dependencies and multi-scale spatial features. Second, Haar wavelet downsampling replaces conventional large-stride convolutions in the backbone and neck to preserve critical structural and edge details. Finally, FasterNet blocks are integrated into the deeper backbone stages to minimize computational complexity while maintaining strong feature representation.

2.6 C2f-MAMBA block

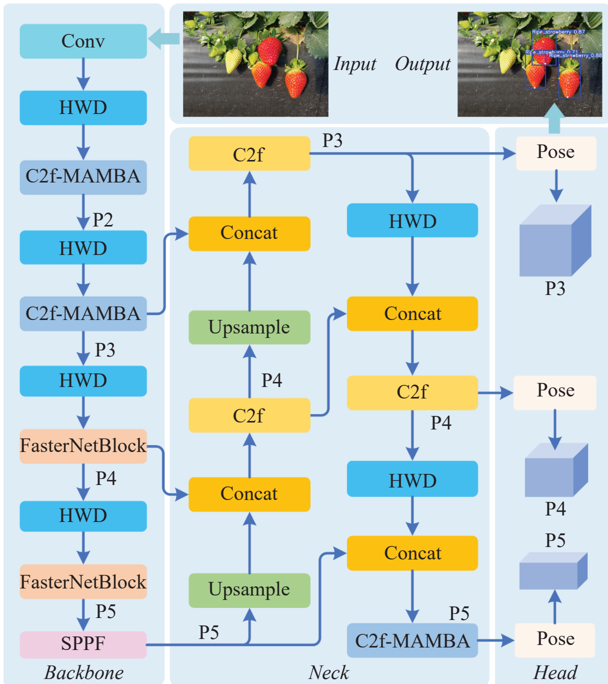
To overcome the limitations of traditional C2f modules in handling dense clustering and foliage occlusion, this study proposes the C2f-MAMBA module (Figure 6). By replacing the standard bottleneck with a novel Straw-Vim module, which integrates a Mamba-based HSM-SSD and an MSCA mechanism, C2f-MAMBA jointly captures global long-range dependencies and fine-grained local patterns. This enhanced multi-scale spatial perception significantly improves the model’s robustness and discriminative capability against complex fruit structures and severe occlusions.

2.6.1 Global feature extraction with state space model

SSM originates from linear time-invariant system theory and maps input sequences to output sequences. Mamba[19] constructs linear time-varying discrete state space models via linear transformations and zero-order hold. Mamba2 further modifies the output computation into a masked linear transformation, termed State Space Dual (SSD), where the mask only accesses past information, known as causal duality[20]. Since image processing requires global context, Non-Causal SSD (NC-SSD) was introduced to remove causal constraints[21,22]. To reduce the computational cost of gating and output projection in NC-SSD, Hidden State Mixer (HSM) was proposed, which performs channel mixing directly on the linear projections and gated projections of hidden states h_{in}[23]. The output formulation of HSM-SSD is given in Equation (1), and its structure is shown in Figure 6.

$$x_{out}=(Ch\odot\sigma(x_{in}W_z))W_{out}\approx C((h\odot\sigma(h_{in}W_z))W_{out})=Cf(h) \quad (1)$$

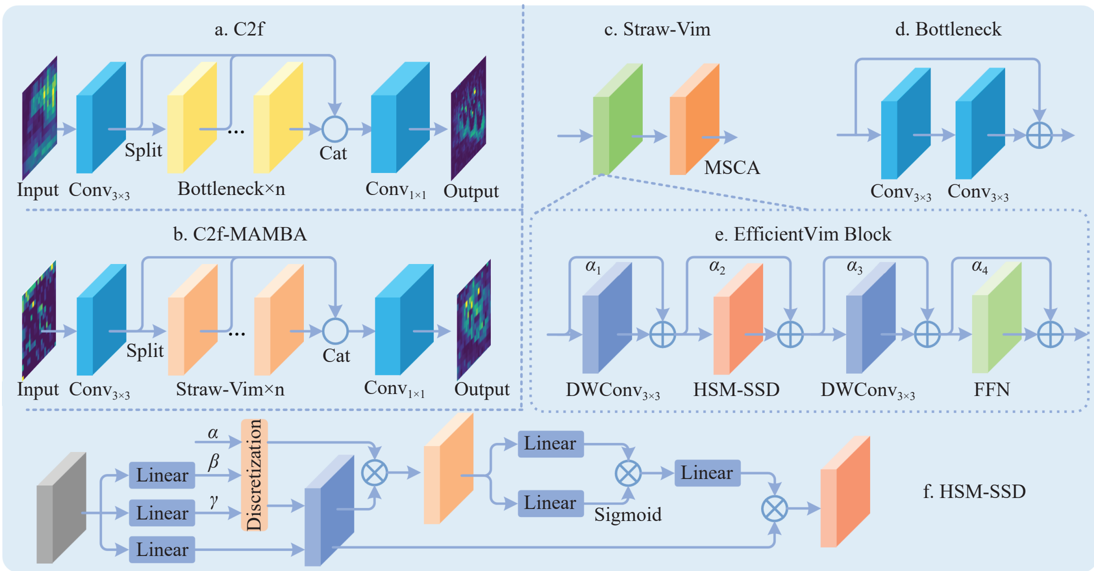
where, C represents the state-space output projection parameter; h and h_{in} denote the continuous hidden state and the input hidden state, respectively; W_z and W_{out} are the learnable weight matrices for the gating mechanism and the final output projection; σ represents the non-linear activation function; and $f(h)$ denotes the



Note: The proposed network integrates three targeted architectural enhancements: 1) Mamba-based C2f-MAMBA modules to strengthen global contextual modeling; 2) Haar wavelet downsampling to preserve high-frequency edge details; and 3) FasterNet blocks to construct a computationally efficient, lightweight backbone for edge-device deployment. Additionally, P2, P3, P4, and P5 denote the multi-scale feature maps extracted at different stages of the network, which are used for subsequent feature fusion and prediction.

Figure 5 Architecture of STRAW-MAMBA

combined transformation function applied to the hidden states. Unlike transformers with quadratic overhead, visual SSMs[22] capture global features with linear computational complexity $O(N)$. Accordingly, the Straw-Vim module incorporates HSM-SSD to achieve efficient, long-range feature extraction for strawberry detection.



Note: a. Proposed C2f-MAMBA module b. Original C2f module in YOLOv8-Pose c. Proposed Straw-Vim module in C2f-MAMBA d. Standard Bottleneck structure in C2f e. Mamba-based HSM-SSD module embedded in Straw-Vim f. Structure of HSM-SSD in EfficientVim Block

Figure 6 Structures of the proposed modules

2.6.2 Multi-scale feature extraction

The HSM-SSD module performs well in capturing long-range dependencies of sequences. However, it is not as effective at capturing the features of strawberries of different sizes. To enhance multi-scale information, the MSCA module is integrated into Straw-Vim. By concatenating the outputs of parallel convolutional branches with varying receptive fields, it generates a channel attention tensor used to dynamically re-weight the original input[24]. The network structure of MSCA is shown in Figure 7. All calculations are formulated as follows:

$$Out = Conv_{1\times1}\left(\sum_{i=0}^{3} Scale_i(DWConv(F))\right) \otimes F \tag{2}$$

where, F is a tensor with shape $H\times W\times C$; it is first operated by Depthwise Convolution (DWConv) with kernel size 5×5 and becomes another feature tensor with shape $H\times W\times C$. Secondly, a group of parallel convolutions is used to capture multi-scale context features; they are residual connections and convolutions $scale_1$, $scale_2$, $scale_3$ with kernel sizes 7, 11, 21, respectively. To reduce the number of parameters and computational complexity, the convolution operations in each branch are approximated by combining vertical and horizontal strip convolution. For example, to simulate a standard two-dimensional convolution with a kernel size of 7×7, only a 7×1 convolution and a 1×7 convolution need to be combined, which can help extract strip features too. The outputs of the four branches are merged via element-wise addition, resulting in a tensor that maintains the shape $H\times W\times C$. Finally, this tensor is operated with 1×1 convolutions; the output tensor with shape $H\times W\times C$ is used to weight the initial input feature F.

2.7 Edge feature enhancement module

While downsampling reduces computational costs, conventional methods such as convolution and pooling often discard critical spatial details like fruit edges and stems, and introduce high parameter overhead that hinders edge deployment. As shown in Figure 8, comprising a lossless feature encoding block and a feature representation learning block, the Haar wavelet downsampling module utilizes the compact, orthonormal Haar wavelet transform[25] to reduce spatial resolution without information loss. This explicitly preserves essential structural details while significantly lowering computational overhead for lightweight deployment. The Haar wavelet downsampling is formulated as Equations (3) and (4).

$$\phi(x) = \begin{cases} 1, & 0 \le x < 1 \\ 0, & x < 0 \text{ or } x \ge 1 \end{cases} \tag{3}$$

$$\psi(x) = \phi(2x) - \phi(2x-1) \tag{4}$$

where, $\phi(x)$ is a basic scaling function and $\psi(x)$ is the mother wavelet function.

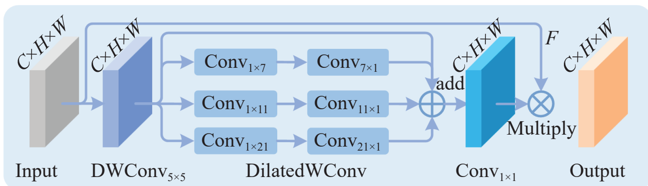


Figure 7 The structure of MSCA

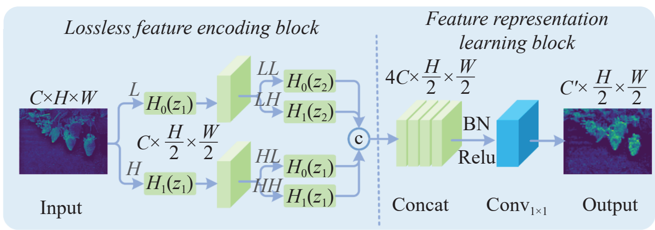


Figure 8 The structure of the edge feature enhancement module

Through Haar wavelet decomposition, a 2D signal, such as a strawberry image, is separated into four subbands: one approximate

(LL) and three detail components (horizontal LH, vertical HL, and diagonal HH). As illustrated in Figure 8, each subband achieves a halved spatial resolution of $\frac{H}{2}\times\frac{W}{2}$. This lossless spatial downsampling is subsequently paired with a Conv-BN-ReLU block to refine features and adjust channels. To maximize spatial detail preservation with minimal parameter overhead, STRAW-MAMBA replaces all conventional downsampling layers, excluding the first with this Haar wavelet downsampling module, effectively enhancing the critical edge features of strawberry fruits and stems.

2.8 Model lightweight enhancement

To facilitate rapid inference on edge devices, STRAW-MAMBA requires a highly efficient architecture. Because standard C2f modules in the YOLOv8-Pose backbone incur significant computational overhead, this study replaces the third and fourth C2f stages with lightweight FasterNet blocks[26]. The structures of the FasterNet block and its underlying partial convolution (PConv) are illustrated in Figure 9.

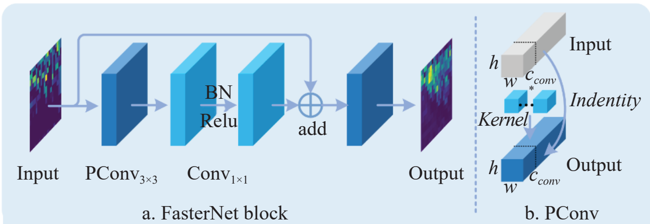


Figure 9 The structure of FasterNet block and PConv

As shown in Figure 9, PConv is the core of FasterNet block. It only performs convolution operations on partial channels of the input feature map tensor to reduce computational complexity. Based on the splitting factor λ, the input tensor $\boldsymbol{X}$ is split channel-wise into a processed segment $\boldsymbol{X}_{conv}$ ($C\times\lambda$ channels) and an unprocessed segment $\boldsymbol{X}_{unconv}$. Convolution is performed only on $\boldsymbol{X}_{conv}$ to yield $\boldsymbol{X}'_{conv}\in\mathrm{R}^{C_{conv}\times H\times W}$, which is subsequently concatenated with $\boldsymbol{X}_{unconv}$ to produce the final output $\boldsymbol{X}'=[\boldsymbol{X}'_{conv},\boldsymbol{X}_{unconv}]$. PConv reduces both the channels per filter and the total filter count to $\frac{1}{\lambda}$, effectively decreasing the overall parameter count to $\frac{1}{\lambda^2}$ of the original. By combining this efficient PConv with group convolution, the FasterNet block drastically lowers computational overhead while preserving robust feature representation[26].

2.9 Geometric compensation method for occlusion

The cutting point is located approximately 8 mm above the strawberry apex. In complex field environments, dense clustered fruits frequently occlude these points, hindering robotic harvesting. To address this challenge, a five-point geometric compensation method was developed to infer invisible cutting points from visible fruit parts, effectively reducing the number of unpickable strawberries. Table 2 lists four scenarios of keypoint occlusion, and the keypoint estimation strategies for the four scenarios of strawberry picking keypoint occlusion are shown in Figure 10.

Scenario 1: Keypoint 1 (cutting point) is invisible.

When the cutting point is invisible, the model draws a ray from the lower point $P_{lower}=(x_{lower},y_{lower})$ upward to $P_{top}=(x_{top},y_{top})$, extends the ray 20 pixels in the direction of P_{top}, and the position of the invisible cutting point is determined. The updated coordinates of the cutting point P'_{cut} can be calculated using Equation (5):

$$P'_{cut}=P_{top}+20\cdot\boldsymbol{u} \tag{5}$$

where the direction vector of the ray is $\boldsymbol{v}=(x_{top}-x_{lower}, y_{top}-y_{lower})$, normalized to $\boldsymbol{u}=\frac{\boldsymbol{v}}{|\boldsymbol{v}|}$.

Scenario 2: Keypoint 1 and keypoint 2 (cutting point and top point) are invisible.

Assuming the strawberry is symmetrical about the axis connecting the upper and lower vertices, and the distance from the top point to both grasping points is 20 pixels, the cutting point can be updated geometrically. Here, the upper and lower points form an isosceles triangle with the grasping points. Let P_{mid} be the midpoint of the line connecting the left and right grasping points. The updated cutting point P'_{cut} can be calculated using Equation (6):

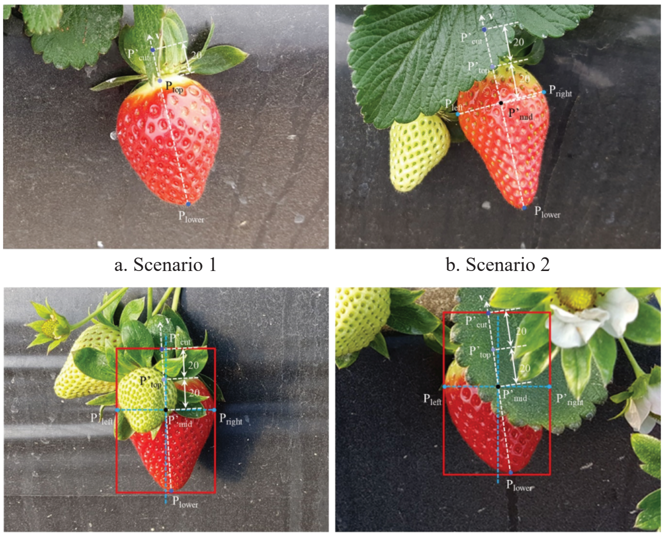
$$P'_{cut}=\frac{P_{left}+P_{right}}{2}+40\cdot\boldsymbol{u} \tag{6}$$

where, $\boldsymbol{u}$ is the unit direction vector from the lower point P_{lower} to P_{mid}.

Table 2 Definition of strawberry occlusion scenarios based on keypoint visibility

Scenario number	1	2	3	4	5	Scenario description
1	×	√	√	√	√	Strawberry stalk fully or partially covered, fruit visible.
2	×	×	√	√	√	Stalk fully covered, upper corner of fruit obscured.
3	×	×	×	√	√	Approximately half of the fruit surface is covered.
	×	×	√	×	√	
4	×	×	×	×	√	Most or entire fruit is covered.

Note: Columns 2-6 correspond to keypoints 1-5: cutting point, top point, left grasping point, right grasping point, and lower point. Visible keypoints are marked with √, and invisible keypoints with ×.



a. Scenario 1 b. Scenario 2

c. Scenario 3(a) d. Scenario 3(b)

Figure 10 Strawberry cutting point estimation strategy under different occlusion conditions

Scenario 3: Keypoints 1, 2, 3, or keypoints 1, 2, 4 (cutting point, top point, and one grasping point) are invisible.

Scenario 3a occurs when the cutting, top, and left grasping points are invisible. Under the assumptions of Scenario 2, and further assuming the center of the detection bounding box (P_{center}) is the strawberry center, the right grasping point, top point, and cutting point can be calculated. Let $\boldsymbol{u}$ be the unit direction vector from P_{lower} to P_{center}. Geometrically, the right grasping point is symmetric to the left grasping point with respect to $\boldsymbol{u}$, which can be calculated using Equation (7). Once the right grasping point is obtained, Scenario 3a converts to Scenario 2 to further update the top and cutting points. Scenario 3b, where the cutting, top, and right grasping points are invisible, is calculated similarly.

$$P'_{right} = 2 < P_{left}, \boldsymbol{u} > \boldsymbol{u} - P_{left} \tag{7}$$

where the inner product $\langle P_{left}, \boldsymbol{u} \rangle$ represents the projection length of P_{left} along the direction of $\boldsymbol{u}$, making $\langle P_{left}, \boldsymbol{u} \rangle \boldsymbol{u}$ the projection point.

Scenario 4: Keypoints 1, 2, 3, and 4 (cutting point, top point, left grasping point, and right grasping point) are all invisible.

If all four keypoints are obstructed, leaving only the lower point visible, insufficient geometric information prevents reliable calculation of the remaining keypoints. In this case, the strawberry is considered unpickable to avoid misoperations.

Compared to Ma et al.[15], which defines only three picking keypoints, the proposed five-point annotation method reduces the number of unpickable strawberries. A strawberry is regarded as unpickable only when the cutting, top, and both grasping points are invisible, substantially improving harvesting coverage.

3 Results and analysis

3.1 Experimental setting

All experiments were conducted on the same server. After testing multiple training parameter combinations, key parameters were selected to balance network performance and resource efficiency. Each experiment was repeated five times, and the results were averaged. The hardware, software, and training parameters are listed in Table 3.

Table 3 Experimental environment and hyperparameter settings

Environment	Value	Training parameter	Value
CPU	Intel® Xeon® Processors	Image size	640×640
RAM	20 G	Epochs	300
GPU	NVIDIA® T4 GPU×8	Batch size	16
System	Ubuntu 20.04	Initial learning rate	0.01
Python	3.10.5	Optimizer	SGD
Pytorch	2.0.0	Momentum	0.937
CUDA	11.7	Patience	50

3.2 Evaluation metrics

Model performance is evaluated across two sub-tasks: bounding box detection and picking keypoint localization.

For fruit detection, the evaluation relies on Precision (P), Recall (R), mean Average Precision (mAP), parameter count, GFLOPs, and FPS. Given the strict picking windows and high economic cost of missed detections, maximizing recall is prioritized for fruit detection to ensure harvestable fruits are not left behind.

For keypoint localization, the metrics include Object Keypoint Similarity (OKS)[27], $P(pose)$, $R(pose)$, and $mAP(pose)$. Standard metrics such as P, R, and mAP for both sub-tasks share the foundational definitions based on true positives, false positives, and false negatives. The OKS is calculated as Equation (8):

$$\mathrm{OKS} = \frac{\sum_{i=1}^{5} \exp\left(-\frac{d_i^2}{2s_k^2\sigma_i^2}\right) \cdot \delta(v_i > 0)}{\sum_{i} \delta(v_i > 0)} \tag{8}$$

where, exp represents the exponential function; d_i is the Euclidean distance between the predicted and ground-truth positions of the i-th keypoint, measured in pixels; v_i denotes the visibility flag of the i-th keypoint; s_k is the scale factor of the target bounding box; σ_i is the normalization constant (annotation standard deviation) for the i-th type of keypoint; $\delta(\cdot)$ acts as an indicator function, which equals 1 if the condition is satisfied and 0 otherwise; n refers to the total number of strawberry keypoints. In this study, $n = 5$; and i represents the keypoint index ($i = 1,2,3,4,5$).

Unlike fruit detection, keypoint localization strictly prioritizes precision over recall. False positives in localization can cause the robotic arm to misoperate, damaging the plant or fruit and leading to severe economic loss. Conversely, missed keypoints can often be corrected during subsequent robotic movements at a much lower cost.

3.3 Ablation studies

3.3.1 Module performance

To validate the effectiveness of the proposed modules in STRAW-MAMBA (C2f-MAMBA, FasterNet block, and Haar wavelet downsampling), ablation studies were conducted against the YOLOv8-Pose baseline. Performance was evaluated on mature strawberries for both fruit detection (Box) and keypoint localization (Pose) using precision (P), recall (R), mAP@0.5, and parameter count (Params). Ablation study results are presented in Table 4 and Table 5, respectively.

Table 4 Ablation studies for picking keypoint localization

Baseline	C2f-MAMBA	FasterNet block	HWD	Pose			
				P/%	R/%	mAP@0.5/%	Params (M)
YOLOv8-Pose	×	×	×	83.3	**77.7**	82.6	3.1
YOLOv8-Pose	√	×	×	88.2	76.4	84.3	2.7
YOLOv8-Pose	×	√	×	86.3	75.8	83.7	2.3
YOLOv8-Pose	×	×	√	84.6	**77.7**	83.1	2.3
YOLOv8-Pose	√	√	×	88.5	76.8	84.6	2.3
YOLOv8-Pose	√	×	√	87.6	76.6	84.5	2.7
YOLOv8-Pose	×	√	√	86.4	75.2	83.9	2.2
YOLOv8-Pose	√	√	√	**88.7**	77.5	**84.9**	**1.9**

Table 5 Ablation studies for fruit detection

Baseline	C2f-MAMBA	FasterNet block	HWD	Box			
				P/%	R/%	mAP@0.5/%	Params/M
YOLOv8-Pose	×	×	×	83.3	78.6	87.6	3.1
YOLOv8-Pose	√	×	×	83.6	78.8	88.6	2.7
YOLOv8-Pose	×	√	×	**87.7**	76.4	88.1	2.3
YOLOv8-Pose	×	×	√	83.4	79.5	87.7	2.3
YOLOv8-Pose	√	√	×	83.9	83.9	88.3	2.3
YOLOv8-Pose	√	×	√	86.7	81.4	88.6	2.7
YOLOv8-Pose	×	√	√	84.9	80.2	89.3	2.2
YOLOv8-Pose	√	√	√	85.5	**84.8**	**89.7**	**1.9**

As detailed in Tables 5 and 6, while individual modules and their partial combinations consistently improve upon the baseline, the fully integrated STRAW-MAMBA achieves the optimal balance of accuracy and efficiency. For keypoint localization (Table 5), the proposed model increases P by 5.4% and mAP@0.5 by 2.3%. Similarly, for fruit detection (Table 6), the full model outperforms the baseline by boosting P by 2.2%, R by 6.2%, and mAP@0.5 by 2.1%.

Overall, every module which includes the integration of C2f-MAMBA for local-global feature capture, FasterNet for computational efficiency, and Haar wavelet downsampling for detail preservation significantly reduces model size and computational load while enhancing detection capabilities, making it highly suitable for resource-constrained agricultural robots.

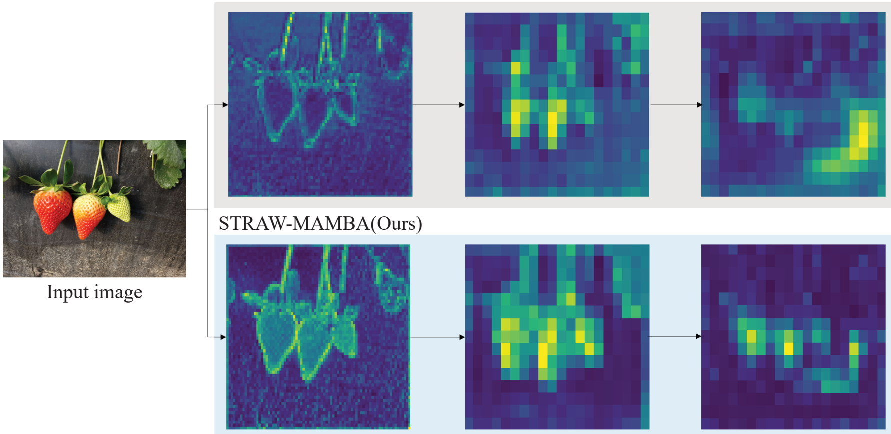
3.3.2 Module feature maps

As shown in Figure 11, to visually validate the proposed modules, compare feature maps between the YOLOv8-Pose baseline and STRAW-MAMBA at three key layers. At Layer 4, C2f-

MAMBA produces highly concentrated activations along fruit edges and textures, overcoming the scattered and weakly discriminative responses of the baseline C2f. At Layer 8, the FasterNet block effectively refines deep features by strengthening responses at color transitions and boundaries, whereas the baseline exhibits uniform, unfocused activations. Finally, at Layer 17, Haar wavelet downsampling yields denser, target-focused activations that preserve crucial spatial details, unlike the baseline convolution, which activates irrelevant regions. Overall, STRAW-MAMBA consistently demonstrates superior feature concentration and target relevance, significantly enhancing the robust localization of strawberry fruits in deeper layers.

Table 6 Comparative results of STRAW-MAMBA and keypoint localization models

Models	Pose					
	P/%	*R*/%	mAP@0.5/%	Params (M)	FLOPs (G)	FPS
YOLOv5n-Pose	75.2	**81.2**	78.8	7.2	16.5	60.2
YOLOv7-tiny-Pose	78.2	76.8	78.7	6.0	13.5	65.4
YOLOv8-Pose	83.3	77.7	82.6	3.1	8.3	87.7
YOLOv11-Pose	82.7	75.1	83.3	2.6	6.6	**96.2**
YOLOv12-Pose	82.9	71.4	82.5	2.6	6.6	88.5
STRAW-YOLO[15]	85.3	73.3	83.3	15.3	27.8	42.2
STRAW-MAMBA (Proposed in this study)	**88.7**	77.5	**84.9**	**1.9**	**5.7**	93.5

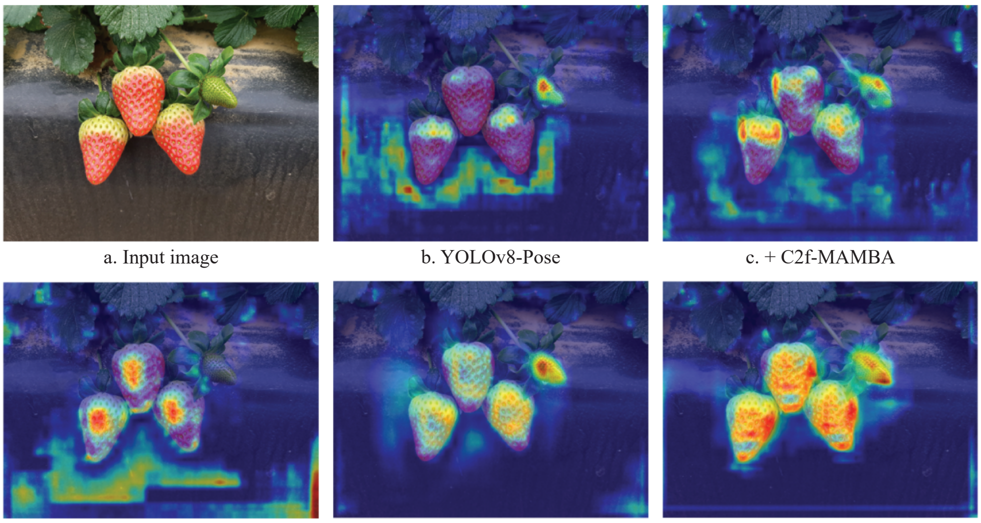


a. Baseline-C2f and C2f-MAMBA at 4th layer b. FasterNet block and baseline-C2f at 8th layer c. Haar wavelet downsampling and baseline-Conv at 17th layer

Figure 11 Comparison of feature maps before and after improvement

3.3.3 Module heat maps

As shown in Figure 12, comparison of heatmaps evaluates the proposed modules through heat map comparisons. The YOLOv8-Pose baseline exhibits sparse, background-biased attention with weak edge activation, indicating limited global context and edge perception. Introducing C2f-MAMBA expands attention continuity across the fruit, reflecting improved global modeling. Incorporating the FasterNet block further refines this by uniformly highlighting surface textures and clearer contours. Finally, Haar wavelet downsampling significantly reinforces boundary activations without compromising global features. Ultimately, the integrated STRAW-MAMBA model achieves a highly balanced attention distribution, demonstrating superior, uniform focus on fruit interiors, edges, and textures compared to the baseline.



a. Input image b. YOLOv8-Pose c. + C2f-MAMBA
d. + FasterNet block e. + HWD f. STRAW-MAMBA(ours)

Figure 12 Comparison of heatmaps between different modules

3.4 Comparative experiments

STRAW-MAMBA is comprehensively compared with five mainstream YOLO-Pose models—YOLOv5n-Pose, YOLOv7-tiny-Pose, YOLOv8-Pose, YOLOv11-Pose, and YOLOv12-Pose—and the current SOTA model STRAW-YOLO[15]. All experiments are evaluated only on mature strawberries that need to be picked.

As shown in Table 6 and Figure 13, STRAW-MAMBA attains the highest precision of 88.7% and mAP@0.5 of 84.9%, surpassing the strongest YOLOv11-Pose by 6.0 percentage points in precision and 1.6 percentage points in mAP@0.5. Compared with the previous state-of-the-art STRAW-YOLO, our model further improves precision by 3.4 percentage points, recall by 4.2 percentage points, and mAP@0.5 by 1.6 percentage points.

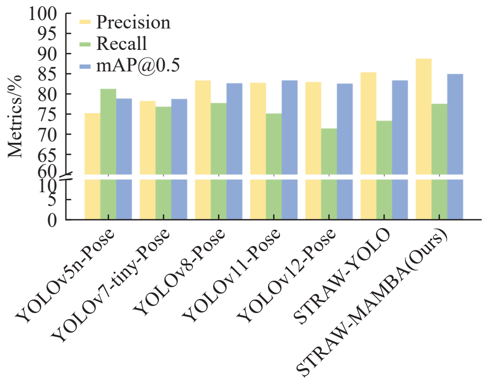


Figure 13 Comparison of keypoint localization performance between STRAW-MAMBA and other models

These results establish STRAW-MAMBA as the new state-of-the-art in keypoint localization for strawberry harvesting, offering leading accuracy and exceptional lightweight characteristics ideally suited for real-time deployment on mobile robotic platforms.

3.5 Model complexity analysis

STRAW-MAMBA is remarkably lightweight, containing only 1.9M parameters and 5.7G FLOPs. It significantly reduces parameters and computational complexity compared to mainstream YOLO-Pose models, up to 73.6% and 65.5% reductions against YOLOv5n-Pose, respectively. Running at 93.5 fps, it surpasses most baselines in speed; substantial accuracy gains fully compensate for its slight 2.8% lag behind YOLOv11-Pose.

Crucially, when compared to the prior SOTA STRAW-YOLO, STRAW-MAMBA reduces parameters and FLOPs by over 85% and 79%, respectively, while more than doubling inference speed from 42.2 to 93.5 fps. This optimal accuracy-efficiency balance enables robust real-time deployment on resource-constrained harvesting platforms like the NVIDIA® Jetson Nano.

3.6 Qualitative evaluation in complex real-world environments

Figure 14 qualitatively compares STRAW-MAMBA with YOLOv8-Pose, YOLOv11-Pose, and STRAW-YOLO across complex greenhouse environments. In dense clusters, baseline models frequently miss targets, whereas STRAW-MAMBA

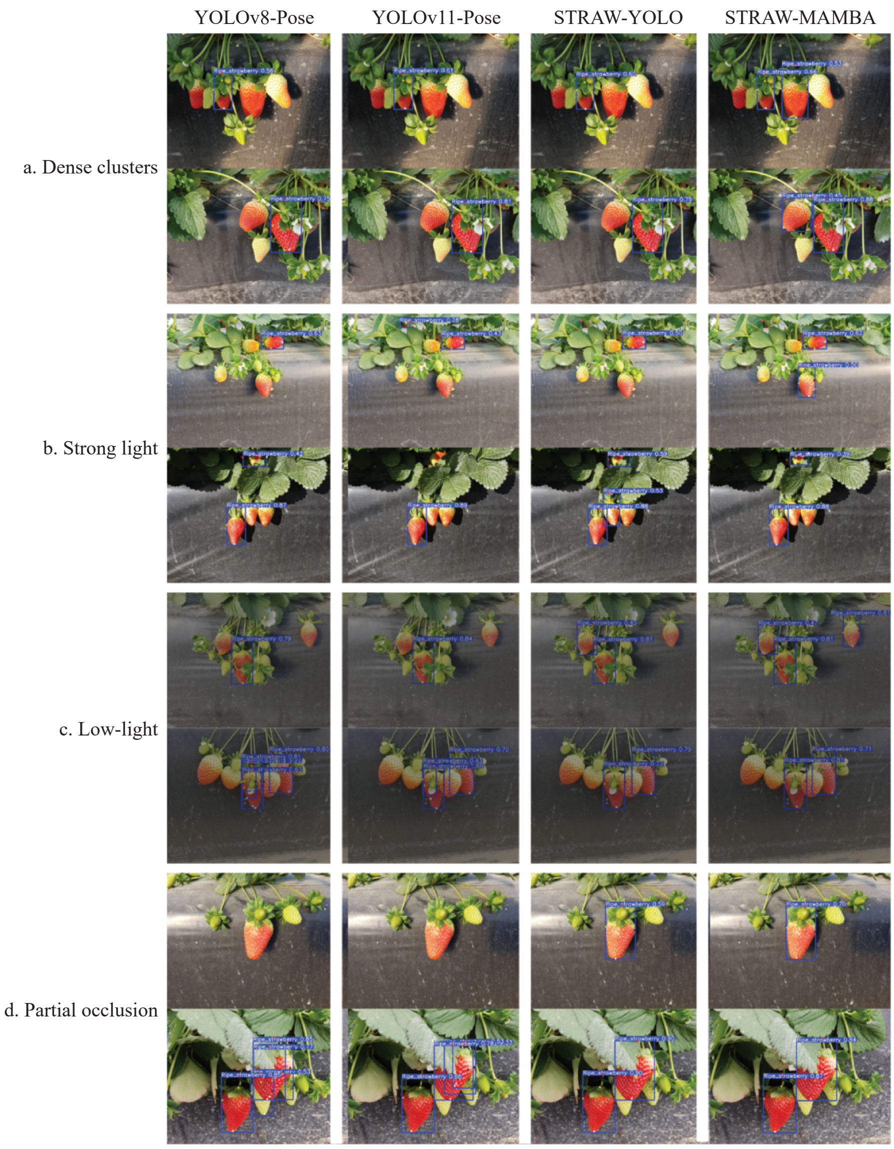


Figure 14 Comparison of detection and picking keypoint localization results between STRAW-MAMBA and mainstream models under complex conditions

leverages C2f-MAMBA's global feature extraction to detect fruits and vines with minimal false negatives accurately. Under extreme illumination (strong sunlight reflections and low-light shadows), the baselines suffer from severe false positives and overlapping detections. Conversely, STRAW-MAMBA maintains robust keypoint localization by preserving crucial edge features through Haar wavelet downsampling. Under partial occlusion, both STRAW-MAMBA and STRAW-YOLO demonstrate stable performance due to their geometric compensation strategies, outperforming the other baselines. Overall, consistent with the quantitative metrics, STRAW-MAMBA exhibits superior robustness and reliability across all challenging real-world scenarios.

3.7 Discussion

This study presents STRAW-MAMBA, a lightweight strawberry detection and keypoint localization model. By integrating the Mamba-based HSM-SSD, a multi-scale convolutional attention module into the C2f-MAMBA module, along with Haar wavelet downsampling and FasterNet blocks, the architecture achieves efficient global feature extraction and fine-detail preservation. Operating at 93.5 fps with only 1.9 M parameters, STRAW-MAMBA consistently delivers superior accuracy at a lower computational cost compared to mainstream baselines and STRAW-YOLO[15]. STRAW-MAMBA delivers comparable or superior accuracy while remaining low-cost.

However, certain limitations remain. First, while the ultra-lightweight design maximizes efficiency, it inherently constrains the capacity to learn highly complex features under extreme dynamic range conditions, such as severe overexposure and deep shadows, occasionally causing fluctuations in localization accuracy and mAP@0.5. Second, the current dataset's limited diversity restricts broader generalization across varying strawberry varieties and growth stages in diverse orchards. Finally, although this study focuses exclusively on 2D image-based localization, the required 3D spatial coordinates and pose angles for physical harvesting can be seamlessly derived in the future by incorporating depth information.

4 Conclusions

This study proposed STRAW-MAMBA, a lightweight and efficient model designed to overcome the limitations of YOLOv8-Pose in global feature representation, detail preservation, and computational complexity. Experimental results demonstrate that STRAW-MAMBA achieves state-of-the-art performance, outperforming mainstream YOLO-Pose variants and the previous SOTA STRAW-YOLO across all primary metrics, while maintaining a highly compact architecture and the fastest inference speed. Furthermore, the proposed five-point geometric annotation scheme significantly reduces the proportion of unpickable strawberries, fundamentally enhancing the adaptability of harvesting robots under severe occlusions.

Future work will proceed in two main directions. First, expanding the dataset to encompass diverse weather conditions, viewing angles, and strawberry varieties. Second, the authors' team plans to deploy the proposed system on NVIDIA Jetson Nano, integrated with a robotic arm to conduct quantitative field picking experiments, thereby validating its practical reliability and cost-effectiveness.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Grant No. 62401203), the Key Research and Development Project of the Science and Technology Plan of Hunan Province, China (2020NK2033), and the Natural Science Foundation Project of Changsha City, Hunan Province, China (kq2402123).

[References]

[1] Giampieri F, Tulipani S, Alvarez-Suarez J M, Quiles J L, Mezzetti B, Battino M. The strawberry: Composition, nutritional quality, and impact on human health. Nutrition, 2012; 28(1): 9–19.

[2] Bac C W, van Henten E J, Hemming J, Edan Y. Harvesting robots for high‐value crops: State‐of‐the‐art review and challenges ahead. Journal of Field Robotics, 2014; 31(6): 888–911.

[3] Hayashi S, Shigematsu K, Yamamoto S, Kobayashi K, Kohno Y, Kamata J, et al. Evaluation of a strawberry-harvesting robot in a field test. Biosystems Engineering, 2010; 105(2): 160–171.

[4] van Henten E J, Hemming J, van Tuijl B A J, Kornet J G, Meuleman J, Bontsema J, et al. An autonomous robot for harvesting cucumbers in greenhouses. Autonomous Robots, 2002; 13(3): 241–258.

[5] Guo J, Yang Z, Karkee M, Jiang Q J, Feng X P, He Y. Technology progress in mechanical harvest of fresh market strawberries. Computers and Electronics in Agriculture, 2024; 226: 109468.

[6] Huang Z L, Wane S, Parsons S. Towards automated strawberry harvesting: Identifying the picking point. In: Towards Autonomous Robotic Systems, Springer, 2017; pp.222–236. doi:10.1007/978-3-319-64107-2_18.

[7] Hu H M, Kaizu Y, Zhang H D, Xu Y W, Imou K, Li M, et al. Recognition and localization of strawberries from 3D binocular cameras for a strawberry picking robot using coupled YOLO/Mask R-CNN. Int J Agric & Biol Eng, 2022; 15(6): 175–179.

[8] Wang J N, Tan D Z, Sui L M, Guo J, Wang R W. Wolfberry recognition and picking-point localization technology in natural environments based on improved Yolov8n-Pose-LBD. Computers and Electronics in Agriculture, 2024; 227: 109551.

[9] Zhang S Y, Hu K, Sha W, Chen Q, Hou Z M, Weng S Z. Efficient one-stage location method for grape picking points in natural scene by combining detection network and point regression. Computers and Electronics in Agriculture, 2025; 230: 109725.

[10] Peng H X, Liang Q J, Zou X J, Wang H J, Xiong J T, Luo Y L, et al. Synchronous detection method for litchi fruits and picking points of a litchi-picking robot based on improved YOLOv8-pose. Int J Agric & Biol Eng, 2025; 18(4): 266–274.

[11] Fu Y X, Zheng H C, Wang Z B, Huang J Y, Fu W. Detection of multi-class coconut clusters for robotic picking under occlusion conditions. Int J Agric & Biol Eng, 2025; 18(1): 267–278.

[12] Yu Y, Zhang K L, Yang L, Zhang D X. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Computers and Electronics in Agriculture, 2019; 163: 104846.

[13] Yu Y, Zhang K L, Liu H, Yang L, Zhang D X. Real-time visual localization of the picking points for a ridge-planting strawberry harvesting robot. IEEE Access, 2020; 8: 116556–116568.

[14] Xie H H, Zhang Z J, Zhang K L, Yang L, Zhang D X, Yu Y. Research on the visual location method for strawberry picking points under complex conditions based on composite models. Journal of Science of Food and Agriculture, 2024; 104(14): 8566–8579.

[15] Ma Z H, Dong N S, Gu J Y, Cheng H C, Meng Z C, Du X Q. STRAW-YOLO: A detection method for strawberry fruits targets and key points. Computers and Electronics in Agriculture, 2025; 230: 109853.

[16] Bochkovskiy A, Wang C-Y, Liao H-Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv, 2020; arXiv: 2004.10934. doi:10.48550/arXiv.2004.10934.

[17] Zhong Z, Zheng L, Kang G L, Li S Z, Yang Y. Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020; pp.13001–13008. doi:10.1609/aaai.v34i07.7000.

[18] Cubuk E D, Zoph B, Shlens J, Le Q V. Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020; pp.702-703. doi:10.5555/3495724.3497141.

[19] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces. arXiv, 2023; arXiv: 2312.00752. doi:10.48550/arXiv.2312.00752.

[20] Dao T, Gu A. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In: Ruslan S, Zico K, Katherine H, Adrian W, Nuria O, Jonathan S, et al., editors. In:

Proceedings of the 41st International Conference on Machine Learning; Proceedings of Machine Learning Research: PMLR, 2024; pp.10041–10071. doi:10.48550/arXiv.2405.21060.

[21] Liu S H, Yu W H, Tan Z X, Wang X C. Linfusion: 1 GPU, 1 minute, 16 k image. arXiv, 2024; arXiv: 2409.02097. doi:10.48550/arXiv.2409.02097.

[22] Zhu L H, Liao B C, Zhang Q, Wang X L, Liu W Y, Wang X G. Vision Mamba: Efficient visual representation learning with bidirectional state space model. In: International Conference on Machine Learning: PMLR, 2024; pp.62429–42. doi:10.48550/arXiv.2401.09417.

[23] Lee S, Choi J, Kim H J. EfficientViM: Efficient vision Mamba with hidden state mixer based state space duality. In: Proceedings of the Computer Vision and Pattern Recognition Conference, 2024; pp.14923–14933. doi:10.48550/arXiv.2411.15241.

[24] Guo M-H, Lu C-Z, Hou Q, Liu Z, Cheng M-M, Hu S-M. Segnext: Rethinking convolutional attention design for semantic segmentation. In: The Thirty-Sixth Annual Conference on Neural Information Processing Systems, 2022; pp.1140–1156. doi:10.48550/arXiv.2209.08575.

[25] Xu G P, Liao W T, Zhang X, Li C, He X W, Wu X L. Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation. Pattern Recognition, 2023; 143: 109819.

[26] Chen J, Kao S-h, He H, Zhuo W, Wen S, Lee C-H, et al. Run, don't walk: chasing higher FLOPS for faster neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023; pp.12021–12031. doi:10.1109/CVPR52729.2023.01157.

[27] Maji D, Nagori S, Mathew M, Poddar D. YOLO-POSE: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022; pp.2637–2646. doi: 10.1109/CVPRW56347.2022.00297.