

Article

Rapid Identification and Accurate Localization of Walnut Trunks Based on TIoU-YOLOv8n-Pruned

Chenchen Ye ¹, Yan Xu ^{1,*} , Jianping Zhou ^{1,2}, Chengcheng Li ¹ , Fubao Fang ¹  and Zhengyang Jin ¹

¹ College of Mechanical Engineering, Xinjiang University, Urumqi 830017, China

² Xinjiang Institute of Engineering, Urumqi 830063, China

* Correspondence: xuyan2018@xju.edu.cn

Abstract

Visual perception has become a prerequisite for the operation of automated walnut vibration harvesting robots under complex orchard conditions. This study proposes an effective trunk target detection algorithm, TIoU-YOLOv8n-Pruned, based on YOLOv8n. First, to enhance the accuracy of walnut trunk prediction boxes matching true boxes at high overlap levels, a TIoU loss function is introduced. Second, to mitigate vibration effect caused by the PTO axis of the tractor, vibration trajectory fitting and coordinate correction are performed by capturing multiple images per second. To meet the frame rate requirements for coordinate correction, channel pruning removes 55% of the model's non-essential channels. Experimental results show that the number of parameters and GFLOPs of TIoU-YOLOv8-Pruned are 950,000 and 3.6 GFLOPs, respectively, while its accuracy and mAP@0.5:0.95 reach 94.1% and 57.2%, outperforming YOLOv5n, YOLOv8n, YOLOv11n, FasterNet-YOLOv8n, Ghost-YOLOv8n, ShuffleNetv2-YOLOv8n, MobileNetV3-YOLOv8n, EfficientNet-YOLOv8n, GhostNetV3-YOLOv8n and MobileNetV4-YOLOv8n. After trajectory fitting and coordinate correction, it significantly reduces vibration-induced errors and enhances localization accuracy. Overall, the TIoU-YOLOv8n-Pruned model demonstrates applicability for trunk identification and localization in walnut orchard mechanical shaking harvesting, offering theoretical guidance for developing automated shaking harvesting equipment.

Keywords: walnut harvesting; image recognition; loss functions; model pruning; coordinate correction



Academic Editors: Weizhen Liang and Jingqiu Chen

Received: 2 October 2025

Revised: 13 November 2025

Accepted: 19 November 2025

Published: 21 November 2025

Citation: Ye, C.; Xu, Y.; Zhou, J.; Li, C.; Fang, F.; Jin, Z. Rapid Identification and Accurate Localization of Walnut Trunks Based on TIoU-YOLOv8n-Pruned. *Agriculture* **2025**, *15*, 2405. <https://doi.org/10.3390/agriculture15232405>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Walnuts are a type of dry fruit with significant economic value and nutritional benefits highly favored by consumers [1–3]. In recent years, the walnut industry has experienced sustained growth in China, particularly in the Xinjiang region. According to 2022 statistical data, China's walnut production was close to 4.69 million tons in 2019, nearly 4.8 million tons in 2020, and went up to 5.4 million tons in 2021 [4]. However, walnuts are traditionally harvested by hand, with the harvesting process accounting for approximately 50% of the labor in the entire fruit production chain [5]. In recent years, vibratory harvesting has gained significant attention in the field of agricultural technology [6–11]. The principle behind this equipment is to manually push a clamping device to grip the tree trunk, while the excitation force generated by the rotating eccentric block is transmitted to the trunk, causing the tree to vibrate and separating the fruit from the stalk. The trunk gripping process is

time-consuming, and the use of a robotic arm capable of automatically recognizing and gripping the trunk could solve this problem.

In order to realize the fast and precise grasping of walnut trunks by shaking vibration equipment, early-stage identification and positioning are crucial. Fang et al. used a distance-adaptive Euclidean clustering method combined with cylindrical fitting and composite criteria screening to identify tree trunks [12]. Juman et al. used color space combination preprocessing to remove portions of the background and combined Viola-Jones detectors with depth information for trunk localization [13]. Shen et al. used a RealSense depth camera to capture color images and depth data of an orchard and detected the tree trunks using superpixel segmentation of the S-channel in the HSV color space, as well as trunk width feature detection in the depth images [14]. However, problems such as complex lighting in walnut orchards and severe shading by weeds and leaves made machine vision difficult to apply in these orchards. With the development of deep learning in recent years, object Detection methods based on convolutional neural networks have shown great advantages and are widely used in the field of image recognition [15].

Currently, most of the existing research on tree trunk detection is used for map building and navigation. Pinto de Aguiar et al. demonstrated the feasibility of using deep learning to detect grapevine trunks by running seven different deep learning networks for grapevine trunk detection on a Google USB accelerator with an NVIDIA Jetson Nano [16]. Da Silva et al. compared the effectiveness of various object detection models for trunk recognition in forests, among which the YOLO (You Only Look Once) series models achieved better recognition results [17,18]. Xu used the YOLOv5 object detection algorithm to recognize walnut tree trunks [19]. Zhang et al. improved YOLOv5 for orchard trunk detection by using the SIOU (Smoothed Intersection over Union) loss [20]. Wang et al. used the EIOU (Extended Intersection over Union) loss to improve YOLOv7 for recognizing oleaginous trunks [21]. Brown et al. used Masked2Former to enhance the YOLOv8s-seg model for segmenting tree trunk images [22].

In order to deploy object detection models on edge computing devices, many researchers have focused on improving model lightweighting. Zhu et al. replaced the standard convolution in Neck of YOLOv7 with the GSConv module, reducing the model's parameters by 28.3% and GFLOPs (Giga Floating-Point Operations) by 21.4% [23]. Fan et al. used the ShuffleNetV2 lightweight network to replace the backbone of YOLOv5 for weed detection, reducing the single-image inference time by 18 ms [24]. Liu et al. introduced the YOLOv8p model with a smaller network by adjusting the scaling factor of YOLOv8 and replacing the C2F module with the PDW FasterNet module, resulting in a reduction in the model's parameters and GFLOPs to 96.34% and 97.69%, respectively, of the original model [25]. Zhao et al. replaced the backbone of YOLOv4 with MobileNetV3, improving the frame rate of detection by 1.76 times [26].

Existing research primarily focuses on extracting navigation paths in field settings, where low overlap between prediction boxes and walnut tree trunks is acceptable. Mainstream lightweight backbone networks that modify the YOLO algorithm result in significantly reduced detection accuracy and are therefore unsuitable for the task at hand. Therefore, based on the growth characteristics of walnut tree trunks, this study designed a TIOU (Trunk Intersection over Union) loss function on top of the CIOU (Complete Intersection over Union) loss function to increase the generation of high-overlap prediction boxes for walnut tree trunks. By pruning the YOLOv8n model, the number of parameters and computational load were reduced. The harvesting device is powered by the tractor's PTO (Power Take Off) shaft, and its vibration during the recognition process affects the localization of the trunk grasping point. To mitigate the effect of vibration, this paper fits vibration trajectories to correct the localization coordinates based on coordinates rec-

ognized within one second. The sampling theorem states that a continuous signal can be reconstructed from its discrete samples if the sampling frequency is more than twice the highest frequency of the signal [27]. Since the tractor's PTO shaft speed is 540 rpm (corresponding to a vibration frequency of 9 Hz), the visual algorithm must detect image frames at a rate exceeding 18 frames per second. Existing research indicates that during vibration harvesting, greater transmission distances result in increased kinetic energy attenuation [28,29]. Therefore, vibration grip points should be positioned as high as possible on the trunk. In actual harvesting operations, walnut tree trunks exhibit two growth patterns: vertical and slanted. To address this, this study sets thresholds based on the aspect ratio of the trunk detection frame. Samples exceeding the threshold are classified as slender trunks, with the gripping point positioned at the upper quarter of the detection frame. Samples below the threshold were classified as obliquely growing trunks. For such trunks, the upper half of the detection frame typically deviated from the actual trunk position, while the frame's center point remained on the trunk. Therefore, the clamping point was set at the frame's center point. This study randomly sampled the diameter and height of 160 walnut tree trunks in the field. When the aspect ratio exceeded 1.8, the trunks generally exhibited vertical growth patterns. To maintain high robustness during actual harvesting, the threshold was set to 2.0 in this study. When the aspect ratio exceeds 2.0, the capture point is located at the top quarter of the detection box. When the aspect ratio is below 2.0, the capture point is centered at the detection box's midpoint. The research content of this study is as follows:

- (1) A walnut trunk dataset was created.
- (2) The loss function of TIoU was designed to increase the generation of prediction frames for the YOLOv8 model for walnut trunks with high overlap.
- (3) The YOLOv8 model was pruned, and the effects of different pruning rates on model performance were compared to determine the optimal pruning rate applicable to this study.
- (4) Corrected trunk coordinates by fitting vibration trajectories generated by towing equipment during identification, improving the positioning accuracy of vibration capture points on the trunk.

Trunk identification and grasping point localization are critical for achieving automated vibration harvesting. To address this challenge, this paper establishes the TIoU-Yolov8n-Pruned detection model, providing a technical reference for upgrading the vibration-based harvesting industry for non-fresh-consumption fruit trees.

2. Materials and Methods

2.1. Walnut Shaking Vibration Harvesting Platform

2.1.1. Hardware Platform

In this study, a tractor-driven shaking vibration harvester was designed for the walnut cultivation model in Xinjiang. As shown in Figure 1, an Intel RealSense Depth Camera D455 is installed to perceive walnut trunk information. The walnut trunk recognition process is as follows: the camera collects data on the walnut trunk, identifies the trunk using a vision algorithm, and locates the clamping point. Then, the clamping point information is transmitted to the host computer, which controls the tandem robotic arm to position and secure the clamping device on the trunk. Finally, the eccentric block in the excitation device rotates to shake the walnut tree for harvesting.

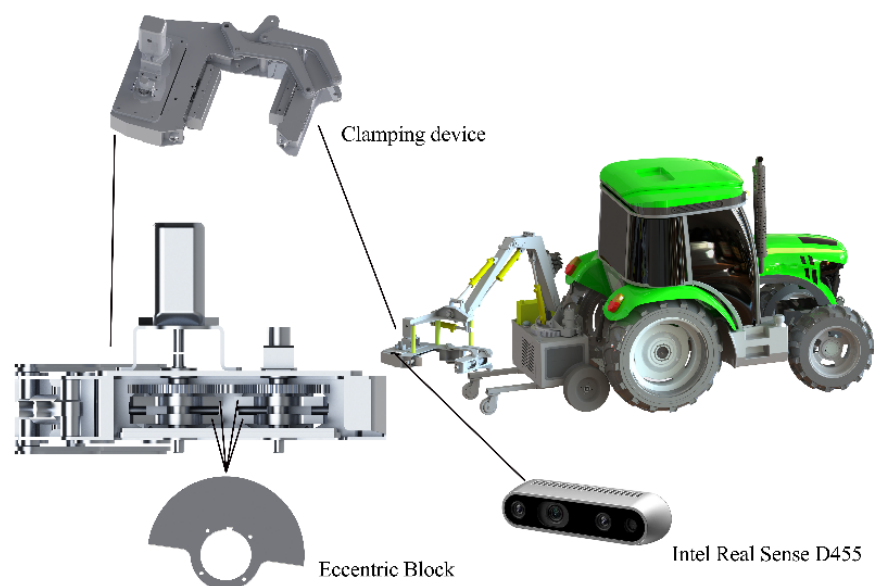


Figure 1. Shaking Harvester.

2.1.2. Overall Process

As shown in Figure 2, the overall process of the field walnut identification and localization algorithm model consists of data, recognition, and localization stages. In the data stage, field walnut trunk data are collected and labeled as training samples. In the recognition stage, the pruned YOLOv8 model is used as the main model for detecting walnut trunks, and the loss is switched to the TlOv8 loss to improve recognition performance, especially for those with large aspect ratio differences. Walnut trunks can grow either vertically or obliquely. However, in both cases, the center of the minimum bounding rectangle remains on the trunk. Therefore, in the localization stage, the center coordinates of the predicted bounding box are taken as the output, which aids in the subsequent grasping of the walnut trunks by the tandem robotic arm.

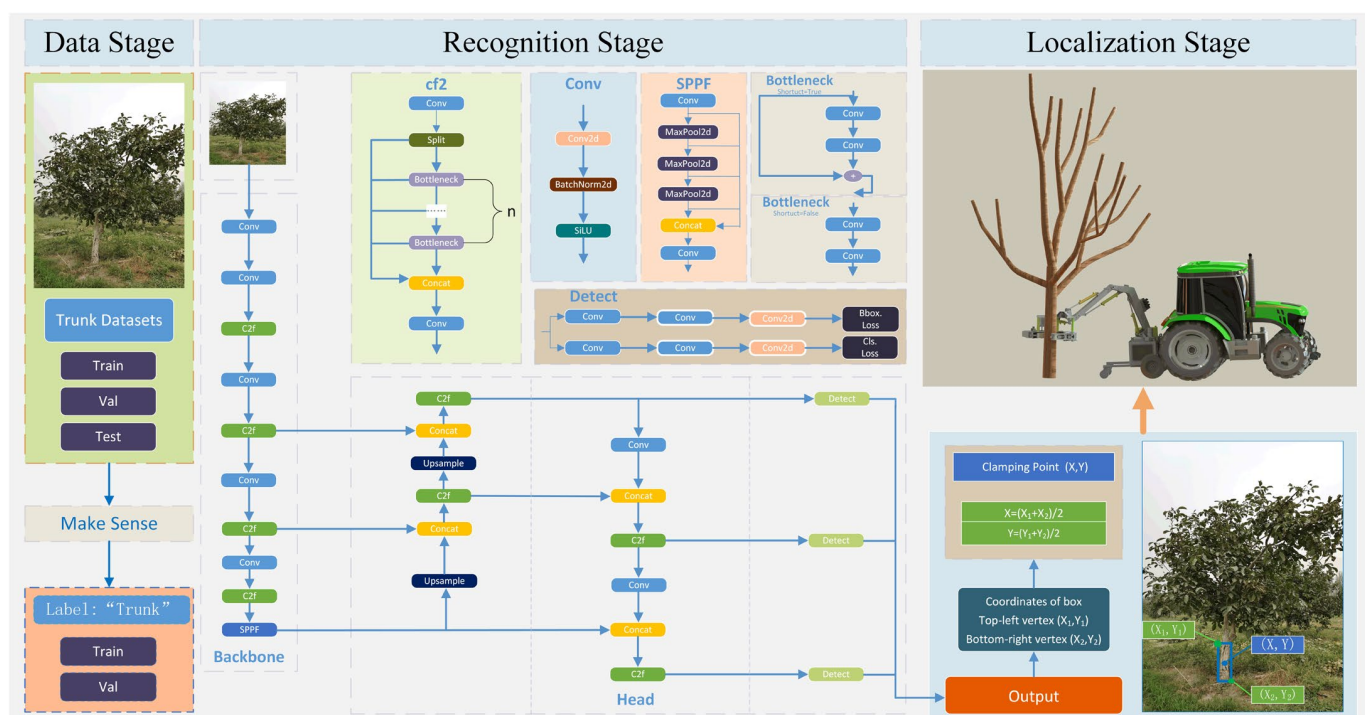


Figure 2. Overall Process Diagram.

2.2. Field Walnut Tree Trunk Sample Data

The data for the experiment were collected from the walnut germplasm resource nursery in Yecheng County, Kashgar Region, Xinjiang, where walnut trees are planted by local farmers. The nursery is located in Luohe Township, Yecheng County, at coordinates 77.6° E, 37.8° N; the satellite map is shown in Figure 3.

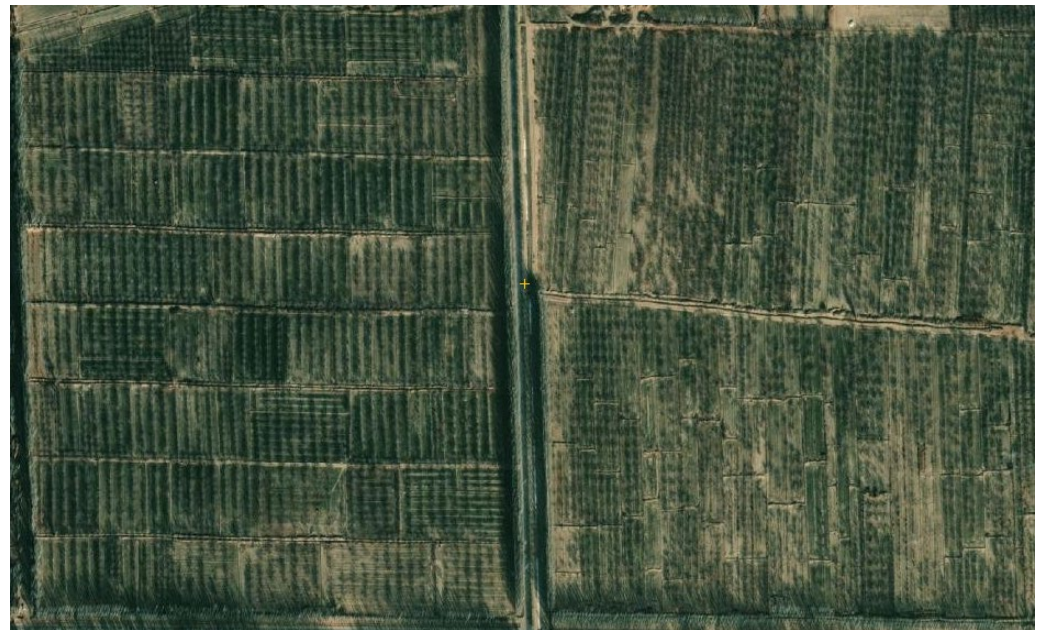


Figure 3. Satellite imagery of the data collection site.

To meet the sampling requirements, images were captured between 17 July and 2 August 2024, under both sunny and cloudy conditions, to meet the harvesting requirements. The equipment used was a Sony α6500 camera with a Sony 16–55 mm f/2.8G lens, and the image resolution was 6000 × 4000 pixels in JPEG format. For each tree, a single photograph was taken with the camera lens positioned 1.3–1.7 m above ground level and kept horizontal. Of the total images, 732 were taken on sunny days and 418 on cloudy days. Due to complex field conditions, images were affected by weed and leaf occlusions during mechanized harvesting, in addition to the standard visibility of tree trunks, as depicted in Figure 4. Specifically, “Weed Occlusion” refers to partial trunk coverage by weeds, with the upper trunk remaining visible, accounting for approximately 3.1% of the images. “Leaf Occlusion” indicates minor canopy obstruction, with the lower trunk still visible, representing about 59.1%. “Normal Trunk” visibility was observed in approximately 37.8% of the cases. Additionally, there is considerable variation in trunk morphology. In the dataset, about 26.0% of the trunks are of normal form, about 24.2% are short and thick, about 26.0% are thin and elongated, and about 23.8% are slanted. Manual annotation was performed using MakeSense software, where walnut tree trunks were annotated, and the coordinates of the four corner points of the bounding boxes were recorded. The dataset was divided into training and validation sets in an 8:2 ratio, with an additional 230 images collected for the test set.



Figure 4. Walnut trunks. (a) Normal trunk; (b) Short and thick trunk; (c) Thin and elongated trunk; (d) Slanted trunk; (e) Leaf occlusion; (f) Weed occlusion; (g) Sunny; (h) Cloudy.

2.3. Algorithmic Improvements

2.3.1. Overview of YOLOv8 Model

YOLOv8 is an open-source, deep learning-based target detection algorithm developed by Ultralytics. It inherits the advantages of YOLOv5 and improves upon them. The model structure consists of three parts: Backbone, Neck, and Head, as shown in Figure 2. The Backbone is responsible for feature extraction, consisting of convolutional layers. The C2f module, as the basic unit of the Backbone, has fewer parameters and superior feature extraction capability compared to the C3 module of YOLOv5. The Neck is responsible for multi-scale feature fusion, enhancing feature representation by combining feature maps from different stages of the Backbone. The Head is responsible for target detection and classification. It consists of three Detect modules. Each Detect module includes a Detect head and a Classification head. Unlike YOLOv5's anchor-based predefined bounding box generation, the anchor-free YOLOv8 eliminates the need for predefined boxes of varying sizes and aspect ratios. By directly predicting object location and size, it reduces dependence on specific aspect ratios and box dimensions, making it more suitable for objects like walnut tree trunks with highly variable bounding rectangle aspect ratios. Therefore, this study builds upon the YOLOv8 model by designing and refining its architecture to adapt to the task of walnut tree trunk recognition.

2.3.2. TIoU Loss

The YOLOv8 network predicts the border coordinate loss using the CIoU Loss (Complete Intersection over Union) [30], which is calculated as shown in the following equation:

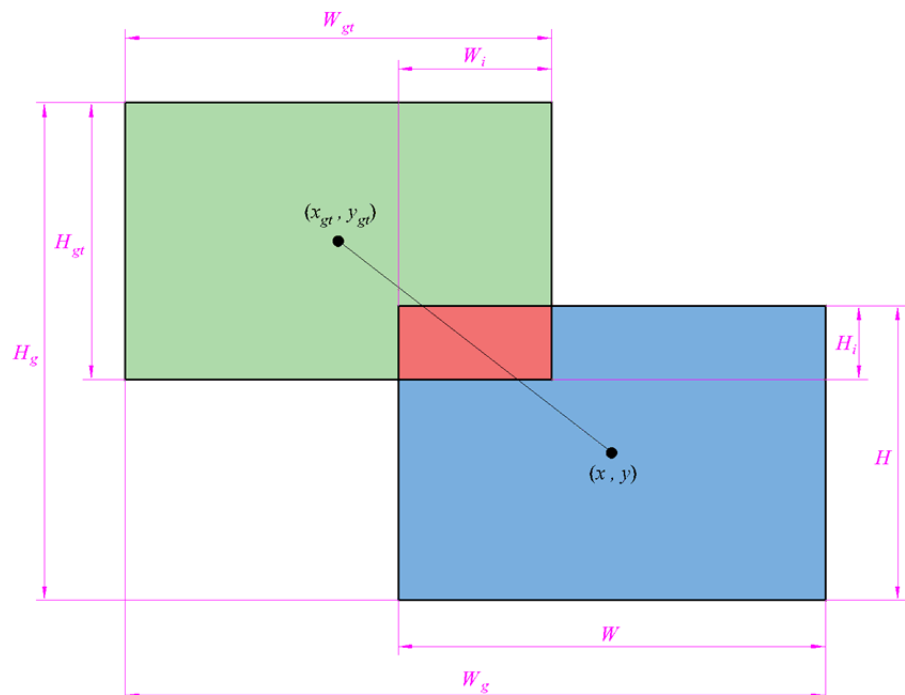
$$L_{CIoU} = L_{IoU} + \frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)} + \alpha \nu \quad (1)$$

$$\alpha = \frac{\nu}{L_{IoU} + \nu} \quad (2)$$

$$\nu = \frac{4}{\pi^2} \left(\tan^{-1} \frac{W}{H} - \tan^{-1} \frac{W_{gt}}{H_{gt}} \right) \quad (3)$$

$$L_{IoU} = 1 - \frac{W_i H_i}{WH + W_{gt} H_{gt} + W_i H_i} \quad (4)$$

The parameters in the expression are shown in Figure 5. α is the weight coefficient, used to balance the parameters, and ν is the aspect ratio metric, used to measure aspect ratio consistency. Although the CIoU loss considers the overlap area, center distance, and aspect ratio, the penalty term in CIoU becomes 0 when the predicted boxes and ground truth boxes have a linear relationship in terms of aspect ratio during regression. In such cases, both high-quality and low-quality anchor boxes can negatively affect the regression loss [31]. Additionally, since this study only employed walnut tree trunks as the detection category, the trunk aspect ratio exhibits significant and irregular variations. Consequently, using aspect ratio as a penalty term during model training hinders model convergence.



Note: $w, h, (x, y)$ denote the width, height and center coordinates of the prediction box, respectively; $w_{gt}, h_{gt}, (x_{gt}, y_{gt})$ denote the width, height and center coordinates of the ground truth box, respectively; W_i, H_i denote the intersection width and height, respectively; W_g, H_g denote the width and height of the minimum enclosing rectangle that bounds both the predicted and ground truth boxes, respectively.

Figure 5. Ground truth box and predicted box.

In walnut shaking harvesting, when the robotic arm autonomously clamps the trunk, the lateral position accuracy of the clamping point needs to be higher than the longitudinal

position accuracy. Additionally, the aspect ratio of the trunk's bounding box varies significantly and irregularly. Therefore, this paper proposes a TIoU (Trunk Intersection over Union) loss function, which is calculated using the following formula:

$$L_{TIoU} = L_{IoU} + \frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)} + \frac{(W_{gt} - W_i)^2 + (H - H_i)^2}{W_g^2} \quad (5)$$

While retaining the overlap area and center distance concepts from the original CIoU loss function, the aspect ratio penalty term is removed to balance the impact of significant changes in trunk bounding box aspect ratios on model regression. Simultaneously, a horizontal overlap penalty term between predicted and ground truth boxes is introduced. This term imposes a smaller penalty as the horizontal overlap between the predicted and ground truth boxes increases at their left and right edges. This promotes lateral convergence of the predicted box center point, enabling it to move more efficiently toward the ground truth center point during iteration and thereby improving the lateral positioning accuracy of the predicted bounding box. The ultimate goal is to enhance the robotic arm's accuracy in autonomously clamping the walnut tree trunk.

2.3.3. Model Pruning

Although the TIoU-optimized YOLOv8n model improves the accuracy of walnut trunk clamping point detection, it remains relatively complex and has poor real-time performance during deployment. When running on the NVIDIA Jetson Xavier NX, the frame rate is approximately 12 FPS, which does not meet the required 18 FPS threshold. To address this, the study employs model pruning to reduce complexity and improve the detection frame rate in field operations, with minimal impact on detection accuracy [32].

In this study, we use the channel pruning method to optimize the YOLOv8n network, which consists of three main processes: sparsity training, channel pruning and model fine-tuning [33]. The principle of the channel pruning algorithm is shown in Figure 6. The YOLOv8 network architecture employs a Batch Normalization (BN) layer after each convolutional operation. By applying sparse training to the BN layers, the importance of each convolutional channel can be distinguished. Channels contributing less to model learning can be pruned, thereby reducing model parameters and accelerating computation while only minimally impacting model performance [34].

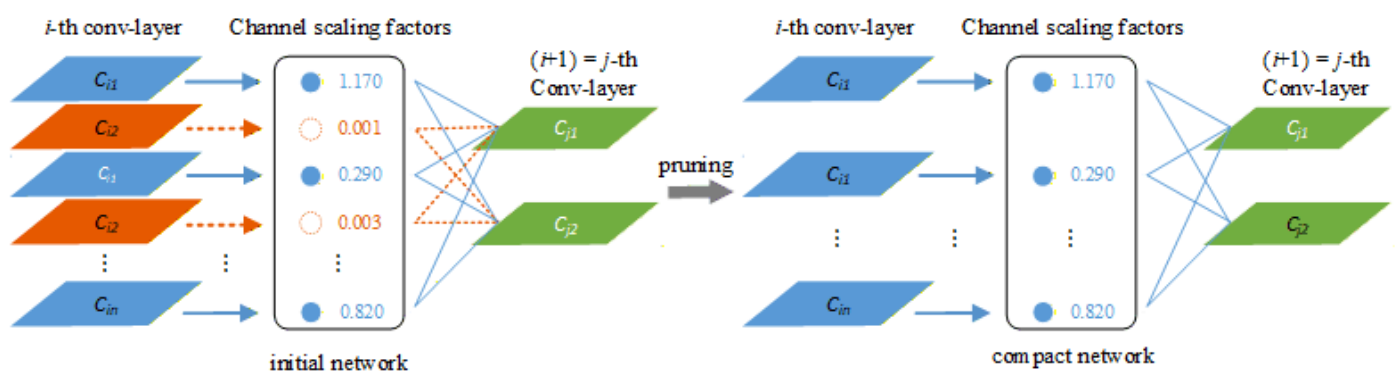


Figure 6. Channel pruning algorithm principle.

(1) Sparsity training

The hyperparameters γ and β are introduced to the BN layer, and the channel data are normalized using both scaling and shifting, computed as follows:

$$Z_{out} = \gamma \hat{Z} + \beta \quad (6)$$

$$\hat{Z} = \frac{Z_{in} - \mu_B}{\sqrt{\sigma_B^2 + \delta}} \quad (7)$$

$$\mu_B = \frac{\sum_{i=1}^n x_i}{n} \quad (8)$$

$$\sigma_B^2 = \frac{\sum_{i=1}^n (x_i - \mu_B)^2}{n} \quad (9)$$

where Z_{in} and Z_{out} denote the inputs and outputs of the BN layer, $\hat{Z}$ denotes the normalized batch-normalized values of the inputs of the BN layer, σ_B^2 , μ_B represent the variance and mean of the batch data, n is the number of batch samples, x_i is the i -th input value in the batch, and δ is a small positive constant.

Sparsity in γ is enforced by incorporating an L1 regularization penalty term into the loss function. The combined objective for sparse training is formulated as

$$L = \sum_{(x,y)} l(f(x,W),y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma) \quad (10)$$

where x is the input value, W is the weight, y is the expected output, and λ is the sparsity regularization term.

(2) Channel pruning model fine-tuning

After training the model with sparsity, sort all scaling factor γ values in the BN layer by absolute magnitude from largest to smallest. Set a pruning threshold proportionally based on this sorting, and prune channels with values below the threshold [35]. During pruning, to avoid dimensionality calculations, exclude Split, Concat, and Add-related layers from the pruning process [36].

(3) Model fine-tuning

After channel pruning, the model retains the original parameters, but accuracy is significantly reduced due to structural changes. The model must be fine-tuned to recover performance loss caused by pruning, allowing it to adapt to the new pruned structure.

2.4. Spatial Localization of the Clamping Point

(1) Spatial 3D Positioning

The pyrealsense2 library for the Intel RealSense D455 camera was used to obtain pixel position information of the target area. During this process, when the depth camera captured images, an object detection algorithm was employed to locate the grasping point on the walnut tree trunk within the image. The camera's depth sensor then acquired the 3D point cloud of that point, enabling the calculation of the distance from the pixel point to the camera. This allowed the derivation of the grasping point's 3D coordinate data within the camera coordinate system.

(2) Trajectory Fitting and Coordinate Correction

To minimize the impact of the tractor PTO axis vibration during the identification process, this study captures multiple predicted point coordinates by the camera within 1 s and applies the least squares method to fit the amplitude and phase of the vibration, thereby correcting the coordinates of the identified points. The equipment's vibration follows sinusoidal motion at 9 Hz, and its displacement can be expressed as:

$$x(t) = A \sin(2\pi \times 9t + \varphi) \quad (11)$$

where A is the amplitude, φ is the phase, and t is time.

The amplitude A and phase φ are obtained by solving the following optimization problem using the least squares method to fit the sinusoidal function:

$$\min_{A,\varphi} \sum_{i=1}^{20} (x_i - A \sin(2\pi \times 9t + \varphi))^2 \quad (12)$$

The coordinates of each sampling point are then corrected:

$$x_{ti} = x_i - A \sin(2\pi \times 9t_i + \varphi) \quad (13)$$

The coordinates of the corrected sampling points are averaged to get the final grab point coordinates:

$$x_t = \frac{1}{n} \sum_{i=1}^{20} x_{ti} \quad (14)$$

2.5. Experimental Setup

The TIoU loss function formula was implemented in the metrics.py file to train the YOLOv8n model. Upon completion of training, the model was sparsely trained using the last.pt weights file with varying sparsity rates to determine the optimal sparsity rate. Subsequently, pruning was performed at various rates to identify the optimal pruning rate. After pruning, 100 rounds of fine-tuning were performed. The improved vision model was then deployed onto the vision development board, where a depth camera was connected to localize the walnut trunk for recognition. Finally, the coordinates of the recognized grasp points were compared with data obtained from manual measurements.

2.5.1. Training Platform and Parameters

The hardware configuration of the training platform includes an i9-12900H CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3060 Laptop GPU with 6 GB of video memory. The software environment consists of the 64-bit Windows 11 operating system, Python 3.11.5, PyTorch 2.2.1, and CUDA 12.1. The training dataset is the walnut trunk dataset established in the previous section. The input image size is 640×640 pixels, with 8 threads, a batch size of 16, and 100 iterations. The Adam optimizer is used with an initial learning rate of 0.1.

2.5.2. Evaluation Criteria

To evaluate the performance of the walnut trunk recognition model, we use the following evaluation metrics: accuracy (P), recall (R), frame rate (FPS), number of parameters, computational volume (GFLOPs), and mAP@0.5:0.95. The performance is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

$$Recall = TP / (TP + FN) \quad (16)$$

$$FPS = 1000 / (T_{pre-process} + T_{inference} + T_{NMS}) \quad (17)$$

$$IOU = \frac{A \cap B}{A \cup B} \quad (18)$$

$$mAP@0.5 : 0.95 = \frac{1}{N} \sum_{i=1}^N \sum_{IOU=0.5}^{0.95, \Delta IOU=0.05} \int_0^1 P_i R_i R \quad (19)$$

where TP denotes a positive sample and was correctly predicted, FP denotes a positive sample but was incorrectly predicted, FN denotes a negative sample but was incorrectly predicted, $T_{pre-process}$ denotes the processing time, $T_{inference}$ denotes the inference time, T_{NMS} denotes non-maximal suppression time, A denotes the predicted bounding box area,

B denotes the ground truth bounding box area, IOU denotes the intersection over union, i denotes the index of the i -th class of objects, N denotes the total number of recognized object classes, and $N = 1$ since only walnut trunks are recognized in this experiment.

3. Results

3.1. Comparison of Recognition Algorithms

To validate the detection performance of walnut trunks using the YOLO algorithm, we trained the YOLOv5n, YOLOv8n, and YOLOv11n models, with the experimental results shown in Table 1. In terms of precision, recall, and average precision, YOLOv5n performs the worst, while YOLOv8n slightly outperforms YOLOv11n. This is because YOLOv5, which uses an anchor-based method, struggles to detect objects like tree trunks that exhibit significant variation in aspect ratio, compared to YOLOv8 and YOLOv11, which employ an anchor-free method. Regarding the number of model parameters and GFLOPs, YOLOv11n outperforms YOLOv8n. This is due to YOLOv11, which is based on YOLOv8, replacing the C2f module with the more lightweight C3k2 module, reducing the model size at the cost of a slight reduction in detection accuracy.

Table 1. Comparison of YOLO algorithms.

Model	Precision	Recall	mAP@0.5:95	Parameters	GFLOPs
YOLOv5n	87.5%	84.1%	39.7%	7.06×10^5	16.5
YOLOv8n	93.8%	91.6%	57.0%	3.01×10^6	8.2
YOLOv11n	92.9%	90.3%	54.9%	2.59×10^5	6.3

3.2. Impact of Loss on Models

To verify the superiority of the TIoU loss function designed in this paper, we compared it with other loss functions, such as GIoU, DIoU, EIoU, SIoU, and WIoU. The comparison results are shown in Table 2. Among these, the YOLOv8n model using the TIoU loss function achieved the highest accuracy and mAP@0.5:0.95, with values of 92.8% and 59.3%, respectively. The higher mAP@0.5:0.95 value demonstrates that the YOLOv8n model with the TIoU loss function better aligns the bounding box of the walnut trunk with the ground truth, thus improving the localization accuracy.

Table 2. Model performance with different loss functions.

Loss	Precision	Recall	mAP@0.5:0.95
GIoU	92.5%	91.0%	55.4%
DIoU	91.6%	88.8%	55.9%
EIoU	93.3%	89.3%	55.3%
SIoU	93.4%	91.8%	57.0%
WIoU	92.4%	94.1%	55.7%
TIoU	93.8%	90.3%	59.3%

A comparison of different loss functions on the test set shows that the bounding boxes predicted using the TIoU loss function fit the walnut trunks more accurately. Additionally, the TIoU loss function improves the recognition accuracy of walnut trunks that grow obliquely and exhibit significant variations in aspect ratio, as shown in Figure 7.

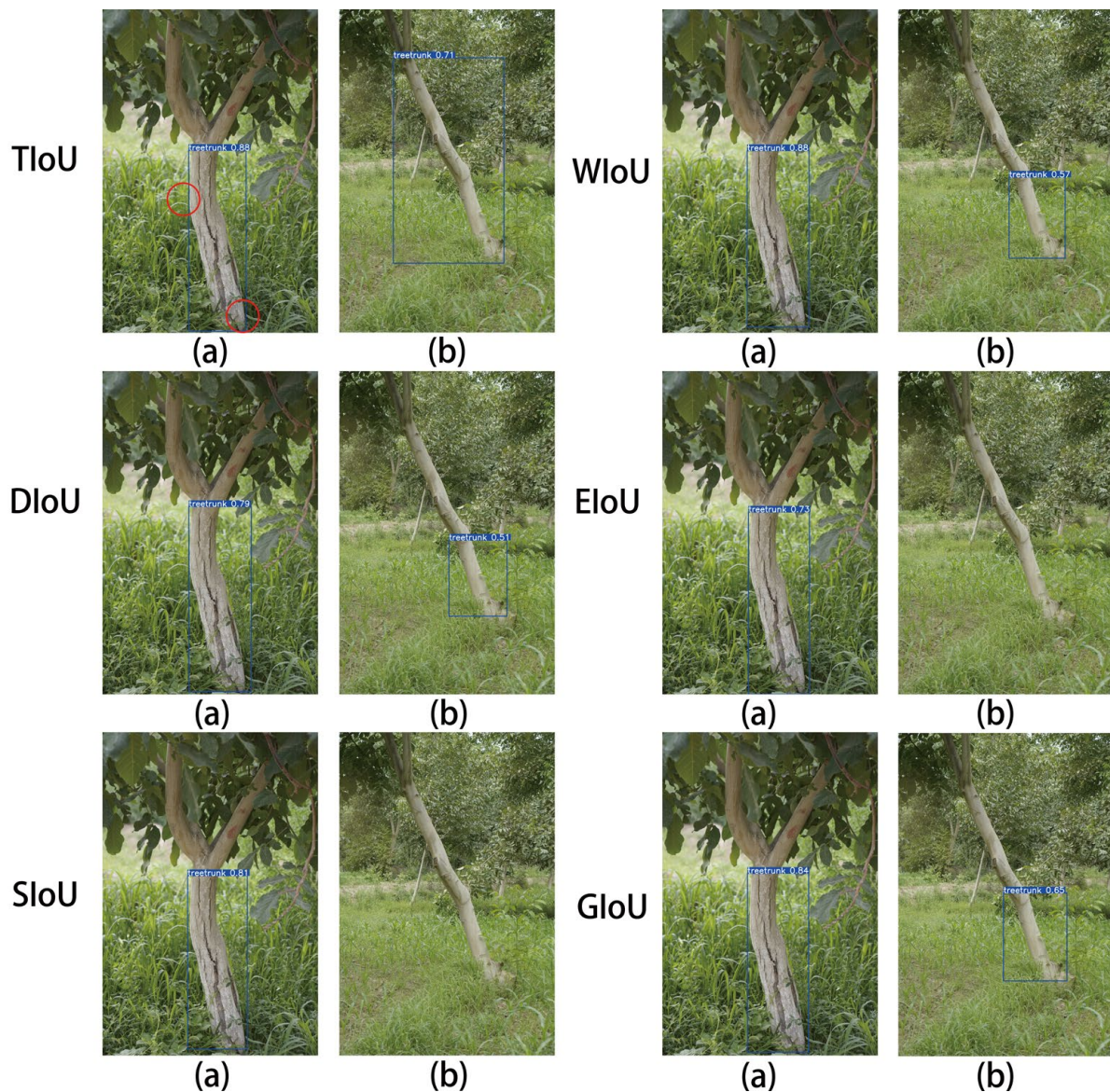


Figure 7. Comparison picture of different loss functions. (a) Partially slanted tree trunk; (b) The slanted tree trunk.

3.3. Sparse Rate Selection

As shown in Figure 8, during sparse training, when the sparsity rate is too small, the model convergence is slow, and the overall distribution follows a normal distribution with a mean of 1. In this case, it is difficult to distinguish the importance of different channels. On the other hand, when the sparsity rate is too large, the model converges too quickly, making it equally difficult to identify the relative importance of the channels. Pre-experiment results indicate that when the number of sparse training rounds is set to 100, a sparsity rate in the range of 0.25 to 0.5 allows for better differentiation of channel importance. Additionally, with a gradient of 0.05 during training, the model begins to converge and stabilize after 70 iterations when the sparsity rate is set to 0.35. Consequently, the sparsity rate for this experiment is selected as 0.35.

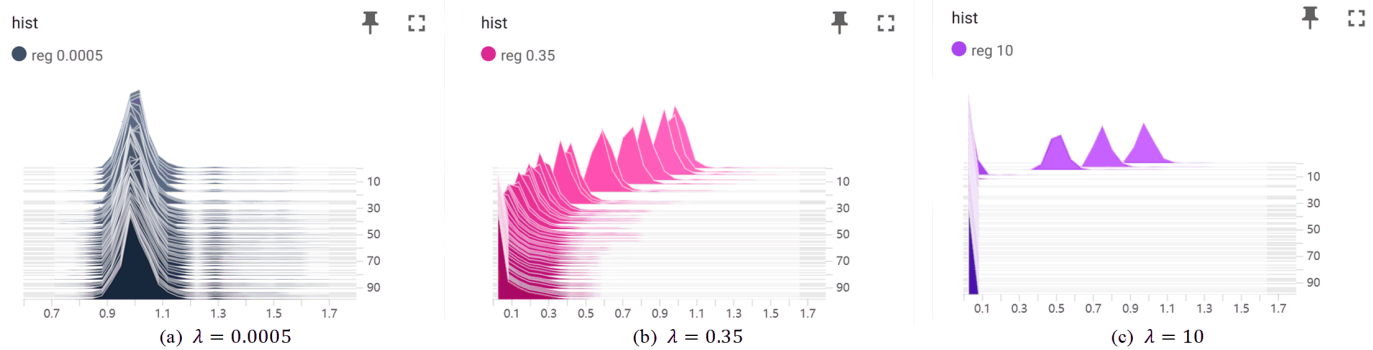


Figure 8. Distribution of weights for different sparsity rate models.

3.4. Impact of Pruning Rate on the Model

As the model pruning ratio increased, the model size decreased, but the detection accuracy decreased accordingly. To balance model accuracy and complexity during compression, experiments were conducted with pruning rates from 5% to 95%, at 5% intervals. The pruning performance of the model is shown in Figure 9.

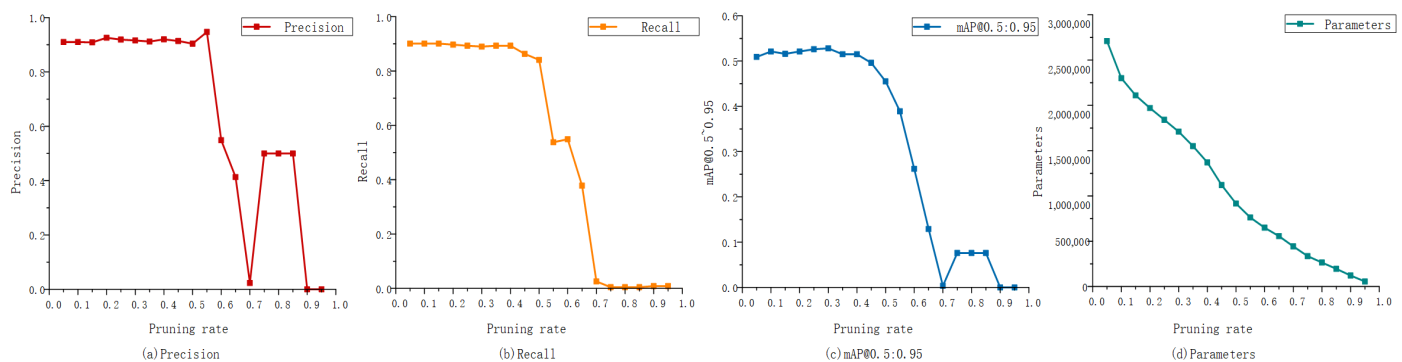


Figure 9. Model performance for different pruning rates.

As shown in Figure 9, as the pruning rate increases, the model precision, mAP@0.5:0.95, and recall remain relatively stable in the initial phase. In the 50–55% pruning rate range, both mAP@0.5:0.95 and recall showed a slight decline, while precision began to increase slightly at 55%, indicating mild underfitting. After pruning, the model's performance could be restored through fine-tuning. However, when the pruning rate exceeded 55%, model performance began to deteriorate rapidly. At a pruning rate of 70%, precision, mAP@0.5:0.95, and recall approached zero, putting the model into an underfitting state, which could not be recovered even with fine-tuning. Further increasing the pruning rate in the 75–85% range resulted in slight improvements in precision and mAP@0.5:0.95. This was attributed to severe underfitting caused by excessive pruning, leading the model to misclassify a large number of negative samples as positive, thereby causing recall to drop to zero. When the pruning rate exceeded 85%, precision, recall, and mAP@0.5:0.95 all dropped to zero, rendering the model ineffective. Considering the potential fluctuations in detection frame rate caused by hardware limitations in actual harvesting operations, it is advisable to prioritize increasing the detection frame rate. Pruning within the 50–55% range, followed by fine-tuning, results in negligible differences in accuracy, recall, and mAP@0.5:0.95. Therefore, the pruning rate was set to 55% in this study, as it effectively compresses the model size with minimal impact on precision, while significantly improving detection speed.

As shown in Figure 10, after pruning 5396 channels from the original model, the number of channels in some convolutional layers decreased significantly, with a maximum

of 230 channels pruned and an average of approximately 36 channels pruned per layer. As a result, the number of model parameters are reduced by 74.8%, and GFLOPs is reduced by 56.1%. After fine-tuning, compared to the original model, precision, recall, and mAP@0.5:0.95 show only a decrease of 2.7%, 4.1%, and 3.6%, respectively, demonstrating the effectiveness of the pruning method.

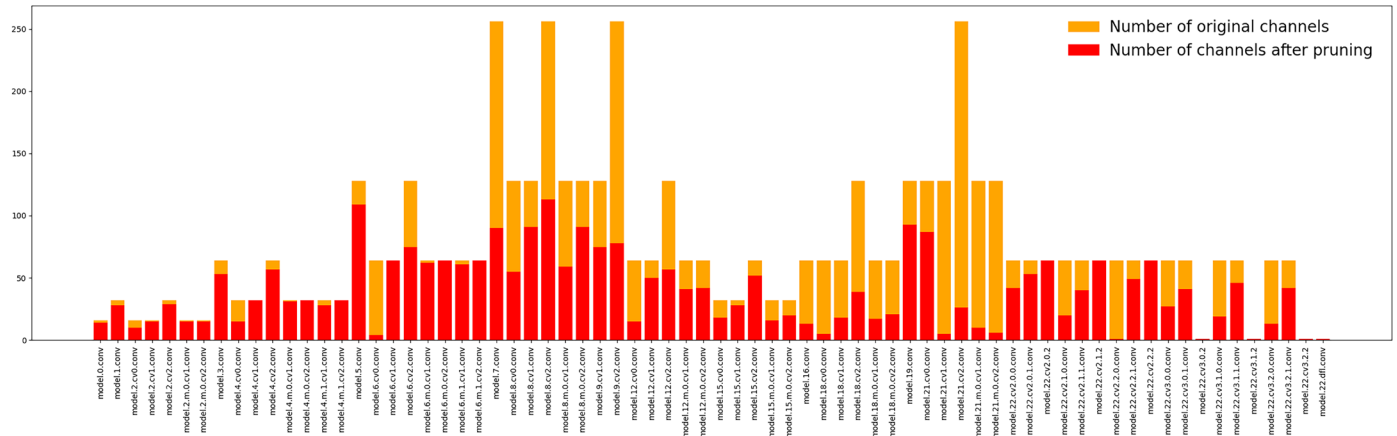


Figure 10. Convolutional channel number variation graph.

3.5. Ablation Study

To investigate the effect of TIoU loss and model slimming on model performance, ablation experiments were conducted, as presented in Table 3. Compared to the original YOLOv8n model, the mAP@0.5:0.95 with TIoU loss increased by 2.3%, suggesting that the TIoU-YOLOv8n model achieves better overlap between predicted boxes and ground truth boxes. By pruning the model, the GFLOPs were reduced by 56.1%, resulting in a slight decrease in detection performance. The model that uses both TIoU loss and pruning has a lower recall compared to the original YOLOv8n model, but it outperforms the original model in terms of the number of parameters, GFLOPs, precision, and mAP@0.5:0.95, demonstrating the feasibility of combining TIoU loss with pruning.

Table 3. Comparison of ablation experiments.

Model	Precision	Recall	mAP@0.5:95	Parameters	GFLOPs
YOLOv8n	93.8%	91.6%	57.0%	3.01×10^6	8.2
TIoU-YOLOv8n	93.8%	90.3%	59.3%	3.01×10^6	8.2
YOLOv8n-Pruned	91.1%	87.5%	53.4%	7.59×10^5	3.6
TIoU-YOLOv8n-Pruned	94.1%	89.2%	57.2%	9.50×10^5	3.6

3.6. Comparative Experimental Analysis of Different Lightweighting Models

To evaluate the superior detection performance of the TIoU-YOLO-Pruned model, it was compared with YOLOv8n models that incorporated lightweight backbones such as MobileNetV3, FasterNet, GhostNet, EfficientNet, ShuffleNetV2, GhostNetV3 and MobileNetV4 under identical testing conditions. The results are presented in Table 4. Recall for the TIoU-YOLOv8n-Pruned model is 0.5% lower than that of the Ghost-YOLOv8n model, but the TIoU-YOLOv8n-Pruned model outperforms all other models in precision, recall, mAP@0.5:0.95, parameters, and GFLOPs.

Table 4. Comparison of different lightweight models.

Model	Precision	Recall	mAP@0.5:95	Parameters	GFLOPs
MobileNetV3-YOLOv8n	89.2%	84.1%	50.5%	2.35×10^6	5.7
FasterNet-YOLOv8n	87.6%	82.1%	47.7%	1.75×10^6	5
Ghost-YOLOv8n	89.9%	89.7%	54.3%	1.71×10^6	5
EfficientNet-YOLOv8n	89.7%	85.9%	50.6%	1.91×10^6	5.6
ShuffleNetv2-YOLOv8n	85.6%	79.4%	42.5%	1.71×10^6	5
GhostNetV3-YOLOv8n	88.9%	84.5%	54.5%	1.72×10^6	5
MobileNetV4-YOLOv8n	88.4%	88.5%	53.5%	4.30×10^6	8.0
TIoU-YOLOv8n-Pruned	94.1%	89.2%	57.2%	9.50×10^5	3.6

3.7. Edge Device Deployment

To validate the deployment of the TIoU-YOLOv8n-Pruned model on edge devices, the model was deployed on the NVIDIA Jetson Xavier NX for object detection on the test set. The model's average preprocessing time $T_{pre-process}$ per image is 8.6 ms, the inference time $T_{inference}$ is 36.7 ms, and the non-maximum suppression time T_{NMS} is 3.9 ms. The model's average frame rate is 20.3 FPS, exceeding the required threshold of 18 FPS. The average frame rates of the evaluated models were as follows: MobileNetV3-YOLOv8n (16.4 FPS), FasterNet-YOLOv8n (17.8 FPS), Ghost-YOLOv8n (17.2 FPS), EfficientNet-YOLOv8n (16.7 FPS), ShuffleNetv2-YOLOv8n (17.8 FPS), MobileNetV4-YOLOv8n (17.5 FPS) and GhostNetV3-YOLOv8n (13.8 FPS). None of these models achieved the frame rate required for this specific task.

3.8. Comparison of Recognition Error

Twenty walnut trees were measured using both a binocular camera and manual methods, and the 3D coordinate results are presented in Table 5. Among these, 1–10 represent direct recognition and positioning; 11–20 represent recognition and positioning after coordinate correction, where the camera captured 20 photos within one second. Without coordinate correction, the average recognition error ratio in the x-direction was 27.53%. After capturing 20 additional images within one second and applying coordinate correction, the average x-axis recognition error ratio decreased to 5.15%. Experimental results demonstrate that trajectory fitting and coordinate correction using multiple frames captured within one second significantly enhance positioning accuracy, validating the effectiveness of this method.

Table 5. Comparison of Recognition Error Table.

Number	Actual Detection Value (mm) (x, y, z)	Visual Measurement Value (mm) (x, y, z)	Error Ratio % (x, y, z)
1	(569, 338, 261)	(501, 302, 249)	(11.95, 10.65, 4.60)
2	(−37, 629, 2157)	(−49, 673, 2173)	(32.43, 7.00, 0.74)
3	(−118, 507, 2115)	(−139, 478, 2107)	(17.80, 5.72, 0.38)
4	(−142, 663, 2337)	(−216, 693, 2358)	(52.11, 4.52, 0.90)
5	(−237, 347, 2206)	(−299, 368, 2243)	(26.16, 6.05, 1.68)
6	(136, 453, 1899)	(177, 419, 1954)	(30.15, 7.51, 2.90)
7	(251, 573, 2495)	(205, 543, 2398)	(18.33, 5.24, 3.89)
8	(−351, 356, 2389)	(−289, 388, 2234)	(17.66, 8.99, 6.49)
9	(135, 475, 2486)	(195, 449, 2401)	(44.44, 5.47, 3.42)
10	(−276, 379, 2884)	(−209, 401, 2783)	(24.28, 5.80, 3.50)

Table 5. Cont.

Number	Actual Detection Value (mm) (x, y, z)	Visual Measurement Value (mm) (x, y, z)	Error Ratio % (x, y, z)
11	(−168, 335, 2000)	(−175, 304, 1924)	(4.17, 9.25, 3.80)
12	(47, 299, 2759)	(38, 318, 2863)	(19.15, 6.35, 3.77)
13	(−899, 357, 2346)	(−917, 379, 243)6	(2.00, 6.16, 3.84)
14	(−739, 275, 1941)	(−768, 298, 1872)	(3.92, 8.36, 3.55)
15	(−344, 792, 2923)	(−328, 771, 2989)	(4.65, 2.65, 2.26)
16	(396, 345, 2657)	(412, 320, 2677)	(4.04, 7.25, 0.75)
17	(−247, 465, 2787)	(−254, 443, 2821)	(2.83, 4.73, 1.22)
18	(−463, 683, 2357)	(−488, 726, 2421)	(5.40, 6.30, 2.72)
19	(386, 567, 2854)	(375, 599, 2941)	(2.85, 5.64, 3.05)
20	(573, 756, 2597)	(559, 724, 2532)	(2.44, 4.23, 2.50)

4. Discussion

This study addresses the challenges of low recognition accuracy caused by significant variations in trunk aspect ratios and the impact of vibration from tractor equipment during intelligent walnut harvesting. By comparing the effects of different loss functions on model performance, the proposed TIoU loss function—which dynamically adjusts the quality classification criteria for anchor boxes—optimizes the CIoU aspect ratio penalty term to improve the lateral position accuracy of anchor boxes. This makes the model more effective for detecting trunks with substantial aspect ratio differences and oblique growth, resulting in higher lateral positioning accuracy for walnut trunks and improving its suitability for vibration-prone harvesting operations. The results demonstrate that the TIoU loss function achieves a 59.3% mAP, outperforming WIoU and other loss functions by enhancing the alignment accuracy between predicted bounding boxes and actual tree trunk boundaries. Given the computational constraints of visual development boards, model pruning was employed to improve detection frame rates. The study found that pruning within the 0–55% range had minimal impact on model accuracy, with performance beginning to decline slightly at a pruning rate of 55%. When pruning rates exceeded 55%, significant performance degradation was observed, leading to the selection of 55% as the optimal pruning rate. Despite the reduced performance of the pruned model, its detection capability still outperforms other lightweight models, demonstrating its significant practical potential for walnut trunk detection in orchards. Furthermore, coordinate calibration using multi-frame image acquisition per second reduced lateral positioning errors caused by tractor PTO shaft vibrations to 5.15%. It should be noted that the trunk data used in this study were collected exclusively during the pre-harvest stage in Yecheng County, Kashgar, Xinjiang. Regional factors, such as variations in lighting and background conditions, may introduce a risk of overfitting, limiting the model's direct applicability to walnut trunk recognition tasks in other areas.

5. Conclusions

This study proposes a TIoU loss function to address large variations in the aspect ratio of the bounding box around walnut tree trunks under real field conditions. Additionally, model pruning reduces the number of parameters and computational complexity, thereby improving inference speed. Localization accuracy is further enhanced through trajectory fitting and coordinate correction. Based on these improvements, the following conclusions are drawn:

- (1) Using the TIoU loss, the mAP@0.5:0.95 is 59.3%, which is higher than that of other loss functions. This indicates that for objects such as tree trunks with large variations in

aspect ratios, the predicted bounding boxes of YOLOv8n using the TIoU loss have a higher overlap with the ground truth bounding boxes. However, the recall is slightly lower compared to the YOLOv8n model trained with the WIoU loss.

- (2) When pruning the model, the performance remains almost unchanged at low pruning rates, but it decreases rapidly once the pruning rate exceeds a certain threshold. In this experiment, this rapid decrease occurs when the pruning rate reaches 55%. With a 55% pruning rate, after fine-tuning the model, the computation is reduced by 56.1%, and the frame rate increases by 1.45 times compared to the original. Meanwhile, the model's precision, recall, and mAP@0.5:0.95 are only reduced by 2.7%, 4.1%, and 3.6%, respectively, which demonstrates the effectiveness of the pruning method.
- (3) The improved model proposed in this paper has lower computational cost and higher precision and mAP@0.5:0.95 compared to other lightweight networks. The model achieves 94.1% precision, 57.2% mAP@0.5:0.95, and 3.6 GFLOPs. Although the recall rate of 89.2% is slightly lower than that of Ghost-YOLOv8n by 0.5%, the TIoU-YOLOv8n-Pruned model is significantly smaller in size compared to Ghost-YOLOv8n. Additionally, the recognition frame rate is 20.3 frames per second on the NVIDIA Jetson Xavier NX, which is sufficient to meet the operational requirements of walnut harvesting equipment, including shaking and vibrating.
- (4) By applying least squares fitting for coordinate calibration based on tractor PTO shaft vibrations, the lateral positioning error in the x-direction was reduced to 5.15%.

In summary, the TIoU-YOLOv8-Pruned model developed in this study meets the operational requirements for walnut vibration-based harvesting. Additionally, it provides reliable technical support for other non-fresh produce crops that employ similar trunk vibration harvesting methods.

Author Contributions: Conceptualization, Y.X.; methodology, C.Y.; validation, C.Y., C.L., F.F., and Z.J.; resources, Y.X.; writing—original draft preparation, C.Y.; writing—review and editing, C.Y.; visualization, C.Y.; project administration, Y.X. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Xinjiang Uygur Autonomous Region, grant number Key R&D Program Projects of Xinjiang Uygur Autonomous Region (2024B02017) and 2025 Xinjiang Autonomous Region Graduate Innovation Project (XJ2025G090).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Wang, R.; Zhong, D.; Wu, S.; Han, Y.; Zheng, Y.; Tang, F.; NI, Z.; Liu, Y. The phytochemical profiles for walnuts (*J. regia* and *J. sigillata*) from China with protected geographical indications. *Food Sci. Technol.* **2021**, *41* (Suppl. 2), 695–701. [\[CrossRef\]](#)
- Meng, J.; Fang, X.P.; Shi, X.M.; Zhang, Y.; Liu, J. Situation, problems and suggestions on the development of walnut industry in China. *Mag. Press China Oils Fats* **2023**, *48*, 84–86.
- Fukasawa, R.; Miyazawa, T.; Abe, C.; Bhaswant, M.; Toda, M. Quantification and Comparison of Nutritional Components in Oni Walnut (*Juglans ailanthifolia* Carr.), Hime Walnut (*Juglans subcordiformis* Dode.), and Cultivars. *Horticulturae* **2023**, *9*, 1221. [\[CrossRef\]](#)
- Wang, Y.; Xu, L.; Zhang, Y.; Zhu, Y.; Zhou, H.; Cui, W.; Zhang, A. Study of vibration patterns and response transfer relationships in walnut tree trunks. *Sci. Hortic.* **2024**, *337*, 113567. [\[CrossRef\]](#)
- Yu, R.; Fan, G.; Xu, G.; Xu, L.; Zhou, H.; Chen, J. Research Status and Development Trend of Walnut Vibration Harvesting. *J. For. Eng.* **2024**, *9*, 21–31.
- Zhuo, P.; Li, Y.; Wang, B.; Jiao, H.; Wang, P.; Li, C.; Niu, Q.; Wang, L. Analysis and experimental study on vibration response characteristics of mechanical harvesting of jujube. *Comput. Electron. Agric.* **2022**, *203*, 107446. [\[CrossRef\]](#)

7. Zhou, J.; Xu, L.; Zhang, A.; Hang, X. Finite element explicit dynamics simulation of motion and shedding of jujube fruits under forced vibration. *Comput. Electron. Agric.* **2022**, *198*, 107009. [\[CrossRef\]](#)
8. Hoshyarmansh, H.; Dastgerdi, H.R.; Ghodsi, M.; Khandan, R.; Zareinia, K. Numerical and experimental vibration analysis of olive tree for optimal mechanized harvesting efficiency and productivity. *Comput. Electron. Agric.* **2017**, *132*, 34–48. [\[CrossRef\]](#)
9. He, L.; Liu, X.; Du, X.; Wu, C. In-situ identification of shaking frequency for adaptive vibratory fruit harvesting. *Comput. Electron. Agric.* **2020**, *170*, 105245. [\[CrossRef\]](#)
10. de Gonzaga Ferreira Júnior, L.; da Silva, F.M.; Ferreira, D.D.; de Souza, C.E.P.; Pinto, A.W.M.; de Melo Borges, F.E. Dynamic behavior of coffee tree branches during mechanical harvest. *Comput. Electron. Agric.* **2020**, *173*, 105415. [\[CrossRef\]](#)
11. Du, X.; Jiang, F.; Li, S.; Xu, N.; Li, D.; Wu, C. Design and experiment of vibratory harvesting mechanism for Chinese hickory nuts based on orthogonal eccentric masses. *Comput. Electron. Agric.* **2019**, *156*, 178–186. [\[CrossRef\]](#)
12. Fang, J.; Shi, Y.; Cao, J.; Sun, Y.; Zhang, W. Active Navigation System for a Rubber-Tapping Robot Based on Trunk Detection. *Remote. Sens.* **2023**, *15*, 3717. [\[CrossRef\]](#)
13. Juman, M.A.; Wong, Y.W.; Rajkumar, R.K.; Goh, L.J. A novel tree trunk detection method for oil-palm plantation navigation. *Comput. Electron. Agric.* **2016**, *128*, 172–180. [\[CrossRef\]](#)
14. Shen, Y.; Zhuang, Z.; Liu, H.; Jiang, J.; Ou, M. Fast recognition method of multi-feature trunk based on realsense depth camera. *Trans. Chin. Soc. Agric. Mach.* **2021**, *53*, 304–312.
15. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
16. de Aguiar, A.S.P.; dos Santos, F.B.N.; dos Santos, L.C.F.; de Jesus Filipe, V.M.; de Sousa, A.J.M. Vineyard trunk detection using deep learning—An experimental device benchmark. *Comput. Electron. Agric.* **2020**, *175*, 105535. [\[CrossRef\]](#)
17. Da Silva, D.Q.; Dos Santos, F.N.; Sousa, A.J.; Filipe, V. Visible and Thermal Image-Based Trunk Detection with Deep Learning for Forestry Mobile Robotics. *J. Imaging* **2021**, *7*, 176. [\[CrossRef\]](#) [\[PubMed\]](#)
18. da Silva, D.Q.; dos Santos, F.N.; Filipe, V.; Sousa, A.J.; Oliveira, P.M. Edge AI-Based Tree Trunk Detection for Forestry Monitoring Robotics. *Robotics* **2022**, *11*, 136. [\[CrossRef\]](#)
19. Xu, Z.; Li, X. Research on tree trunk detection and navigation line fitting algorithm in orchard. *J. Chin. Agric. Mech.* **2024**, *45*, 217–222.
20. Zhang, J.; Tian, M.; Yang, Z.; Li, J.; Zhao, L. An improved target detection method based on YOLOv5 in natural orchard environments. *Comput. Electron. Agric.* **2024**, *219*, 108780. [\[CrossRef\]](#)
21. Wang, H.; Liu, Y.; Luo, H.; Luo, Y.; Zhang, Y.; Long, F.; Li, L. *Camellia oleifera* trunks detection and identification based on improved YOLOv7. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8265. [\[CrossRef\]](#)
22. Brown, J.; Paudel, A.; Biehler, D.; Thompson, A.; Karkee, M.; Grimm, C.; Davidson, J.R. Tree detection and in-row localization for autonomous precision orchard management. *Comput. Electron. Agric.* **2024**, *227*, 109454. [\[CrossRef\]](#)
23. Zhu, X.; Chen, F.; Zheng, Y.; Chen, C.; Peng, X. Detection of *Camellia oleifera* fruit maturity in orchards based on modified lightweight YOLO. *Comput. Electron. Agric.* **2024**, *226*, 109471. [\[CrossRef\]](#)
24. Fan, X.; Sun, T.; Chai, X.; Zhou, J. YOLO-WDNet: A lightweight and accurate model for weeds detection in cotton field. *Comput. Electron. Agric.* **2024**, *225*, 109317. [\[CrossRef\]](#)
25. Liu, Z.; Abeyrathna, R.R.D.; Sampurno, R.M.; Nakaguchi, V.M.; Ahamed, T. Faster-YOLO-AP: A lightweight apple detection algorithm based on improved YOLOv8 with a new efficient PDWConv in orchard. *Comput. Electron. Agric.* **2024**, *223*, 109118. [\[CrossRef\]](#)
26. Zhao, S.; Zhang, S.; Lu, J.; Wang, H.; Feng, Y.; Shi, C.; Li, D.; Zhao, R. A lightweight dead fish detection method based on deformable convolution and YOLOV4. *Comput. Electron. Agric.* **2022**, *198*, 107098. [\[CrossRef\]](#)
27. Shannon, C.E. Communication in the presence of noise. *Proc. IRE* **1984**, *72*, 1192–1201. [\[CrossRef\]](#)
28. Jin, W.; Zhao, J.; Zhuang, T.; Liu, L.; Zhao, E.; Yang, X. Energy Transfer Characteristics of Walnut Trunk and Branches in Mechanical Vibration Picking. *Trans. Chin. Soc. Agric. Mach.* **2024**, *55*, 2031.
29. Yu, R.; Fan, G.; Xu, L.; Zhang, H.; Zhou, H.; Shi, M.; Wang, Y.; Cui, W.; Xu, G. Design and Vibration Performance of a Flexible Rocker-Arm Walnut Vibrating Harvester. *Sci. Silvae Sin.* **2025**, *61*, 180–195.
30. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [\[CrossRef\]](#)
31. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
32. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [\[CrossRef\]](#)
33. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.

34. Wang, D.; He, D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. *Biosyst. Eng.* **2021**, *210*, 271–281. [[CrossRef](#)]
35. Zhang, R.; Yan, K.; Ye, J. Lightweight YOLO-v7 for Digital Instrumentation Detection and Reading. *Comput. Eng. Appl.* **2024**, *60*, 192–201.
36. Wang, F.; He, Z.; Zhang, Z.; Xie, K.; Zeng, Y. Lightweight Object Detection Method for *Panax notoginseng* Based on Edge. *Trans. Chin. Soc. Agric. Mach.* **2024**, *55*, 171–183.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.