Method for the fruit tree recognition and navigation in complex environment of an agricultural robot

Xiaolin Xie[1,2,3*], Yuchao Li[1], Lijun Zhao[4], Xin Jin[1,3], Shengsheng Wang[1], Xiaobing Han[1]

(1. *College of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang 471003, Henan, China*;
2. *Collaborative Innovation Center of Machinery Equipment Advanced Manufacturing of Henan Province, Luoyang 471003, Henan, China*;
3. *Longmen Laboratory, Luoyang 471003, Henan, China*;
4. *College of Intelligent and Manufacturing Engineering, Chongqing University of Arts and Sciences, Chongqing 402160, China*)

Abstract: To realize the visual navigation of agricultural robots in the complex environment of orchards, this study proposed a method for fruit tree recognition and navigation based on YOLOv5. The YOLOv5s model was selected and trained to identify the trunks of the left and right rows of fruit trees; the quadratic curve was fitted to the bottom center of the fruit tree recognition box, and the identified fruit trees were divided into left and right columns by using the extreme value point of the quadratic curve to obtain the left and right rows of fruit trees; the straight-line equation of the left and right fruit tree rows was further solved, the median line of the two straight lines was taken as the expected navigation path of the robot, and the path tracing navigation experiment was carried out by using the improved LQR control algorithm. The experimental results show that under the guidance of the machine vision system and guided by the improved LQR control algorithm, the lateral error and heading error can converge quickly to the desired navigation path in the four initial states of [0 m, −0.34 rad], [0.10 m, 0.34 rad], [0.15 m, 0 rad] and [0.20 m, −0.34 rad]. When the initial speed was 0.5 m/s, the average lateral error was 0.059 m and the average heading error was 0.2787 rad for the navigation trials in the four different initial states. Its average driving was 5.3 m into the steady state, the average value of steady state lateral error was 0.0102 m, the average value of steady state heading error was 0.0253 rad, and the average relative error of the robot driving along the desired navigation path was 4.6%. The results indicate that the navigation algorithm proposed in this study has good robustness, meets the operational requirements of robot autonomous navigation in orchard environment, and improves the reliability of robot driving in orchard.

Keywords: fruit tree recognition, visual navigation, YOLOv5, complex environments, orchards
DOI: 10.25165/j.ijabe.20241702.8031

Citation: Xie X L, Li Y C, Zhao L J, Jin X, Wang S S, Han X B. Method for the fruit tree recognition and navigation in complex environment of an agricultural robot. Int J Agric & Biol Eng, 2024; 17(2): 221–229.

1 Introduction

After entering the 21st century, with the continuous development of control theory, sensor technology, and processor performance improvement for China's industrial, agricultural, and service industries automation and intelligent development have laid the foundation[1]. With the increasing level of automation of agricultural machinery in China, the study of agricultural robots has also received increasing attention[2,3]. Given that all of China's fruit production is at the top of the world, fruits, and their by-products have become a major source of income for farmers and are gradually becoming an important pillar of the rural economy. However, the highly manual-dependent production mode and rapidly increasing labor costs have become the main factors limiting fruit production[4]. To further improve production efficiency and fruit quality, the intelligence of agricultural machinery and equipment is an inevitable trend for future development[5].

Agricultural robots are indispensable for ploughing[6], pruning[7], fertilizing and weeding[8], and picking[9,10] in the production and management of orchards. Agricultural robots want to operate in this unstructured environment without perception of the orchard environment[11], they must first know the exact location of the fruit trees[12,13]. Visual navigation has the advantages of simple equipment, wide detection range, low cost, and autonomy in real-time, and is gradually becoming a mainstream navigation method, widely used for local path planning problems in unmanned[14]. The key to this technology is the reliable and stable recognition of the navigation path by means of image processing, which in turn guides the travel of the machine[15]. Many commercial field robots use GNSS (Global Navigation Satellite System) data as a global positioning source and operate using field maps measured by aerial imaging or seed maps generated by seeders[16]. Radcliffe et al.[17] segment the sky from the canopy background and fit a navigation path based on the difference between the sky and the canopy. However, for most orchards, the GNSS signal and the sky background are blocked by the canopy. Sharifi et al.[18] segmented the clustered images based on graph segmentation theory and applied the Hough to extract path features. Yang et al.[19] proposed an algorithm to detect the center line of the maize rows under the canopy by extracting the bottom of the plants. Yi et al.[20] proposed a night hare monitoring method based on infrared thermography and improved YOLOV3, compared with

Received date: 2022-11-10 **Accepted date:** 2023-08-29

Biographies: Yuchao Li, MS candidate, research interest: control algorithm for intelligent agricultural equipment, Email: 200320041528@stu.haust.edu.cn; **Lijun Zhao**, PhD, Professor, research interest: intelligent manufacturing, Email: 20190005@cqwu.edu.cn; **Xin Jin**, PhD, Associate Professor, research interest: precision operation and detection system for intelligent agricultural equipment, Email: jx.771@163.com; **Shengsheng Wang**, Lecturer, research interest: intelligent agriculture equipment for fruit harvest, Email: ws@haust.edu.cn; **Xiaobing Han**, MS candidate, research interest: control algorithm for intelligent agricultural equipment, Email: 2903352328@qq.com.

***Corresponding author: Xiaolin Xie**, PhD, Lecturer, research interest: new pattern transmission system and intelligent control unit. College of Equipment Agricultural Engineering, Henan University of Science and Technology, Luoyang 471003, Henan, China. Email: xiexiaolin@haust.edu.cn.

Faster-RCNN and RFCN-RESNET101, which increased the detection rate by 45% and 20%, and the detection speed increased by 30fps and 45ps. Zhang et al.[21] proposed a weed detection method for vegetable seedlings with improved YOLOv5 algorithm and image processing, which improved the accuracy of 5.8%, 6.9%, 10.3%, 13.1%, 9.0% and 5.2% compared with Faster R-CNN, SSD, EfficientDet, RetinaNet, YOLOv3 and YOLOv4. However, the inconsistent nature of lighting and shadows in farmland and orchards poses a challenge to these color image-based algorithms.

At present, scholars at home and abroad have carried out more research related to path recognition, but in the orchard environment, the use of traditional image processing methods is susceptible to interference from factors such as the varying morphology of fruit trees, complex environmental backgrounds, and changes in lighting conditions[22]. In recent years, convolutional neural networks[23] have made great research progress in the direction of image processing. YOLOv5 (You Only Look Once) is a target detection network based on a fully convolutional neural network, uses the Pytorch framework, which can easily train its own dataset is very friendly to users, and has the advantages of high precision, fast speed, and easy deployment compared with other models such as RCNN and SSD. For the limitations of existing methods, this paper proposes a navigation method based on YOLOv5 for fruit tree recognition and navigation path fitting. The objective of this study is to devise a reliable visual navigation methodology tailored for autonomous navigation systems operating within intricate orchard settings.

2 Materials and methods

2.1 Robot

This section introduces our team's self-developed robot, which was used for the data acquisition and final navigation test of this experiment. As shown in Figure 1, the robot is equipped with various sensors for navigation tasks, including a TOF (Tine-of-Flight) camera, an RGB camera, an RTK (Real-time kinematic) module, and two inertial measurement units. The main sensor for the vision-based navigation function in this article is the RGB camera, which has a viewing angle of 90°×59° and a resolution of 1280×960.

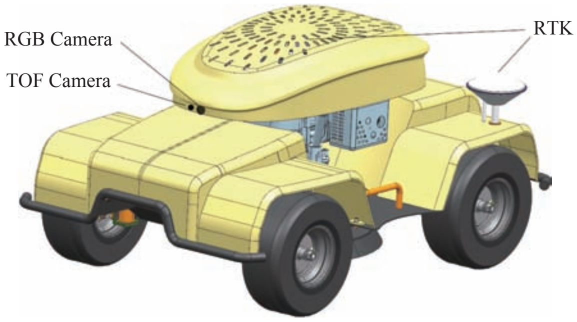


Figure 1 Arched waist agricultural robot

This article focuses on the vision module shown in Figure 2, and other modules in the navigation system will be described in other studies. In the visual navigation system developed by the author's team of this study, the main functions are divided into three parts, the first is to collect data, make datasets, and train network models. The second is to establish the kinematic model of the test platform and solve the relevant parameters. Finally, the model is used for object detection, and then the path planning is carried out, combined with the pose and kinematic model of the test platform to complete the navigation.

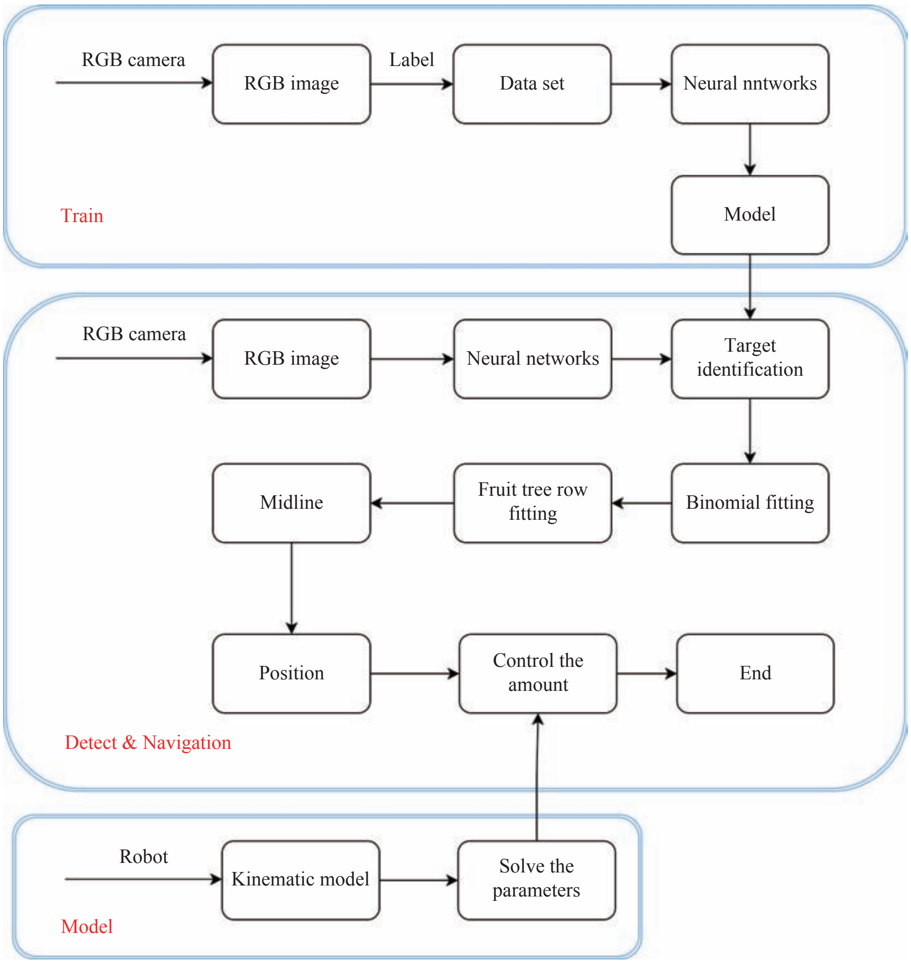


Figure 2 Flowchart of vision-based navigation system module

2.2 Introduction to the YOLOv5 network model

The YOLO model is a fast and compact object detection model with better performance for the same size than other networks and has been steadily improving. YOLOv5 can be easily and conveniently deployed to small mobile devices and its network structure is divided into four parts: Input, Backbone, Neck, and Prediction. As shown in Figure 3, the stage of Input includes the preprocessing of an image, such as scaling the image to the appropriate pixel size, normalization, and so on. The Backbone part is used to extract the representation of some common characteristics of fruit trees. Neck can further improve feature diversity and robustness. The Head part includes a classification branch and a regression branch, which is used to complete the output of fruit tree target detection results.

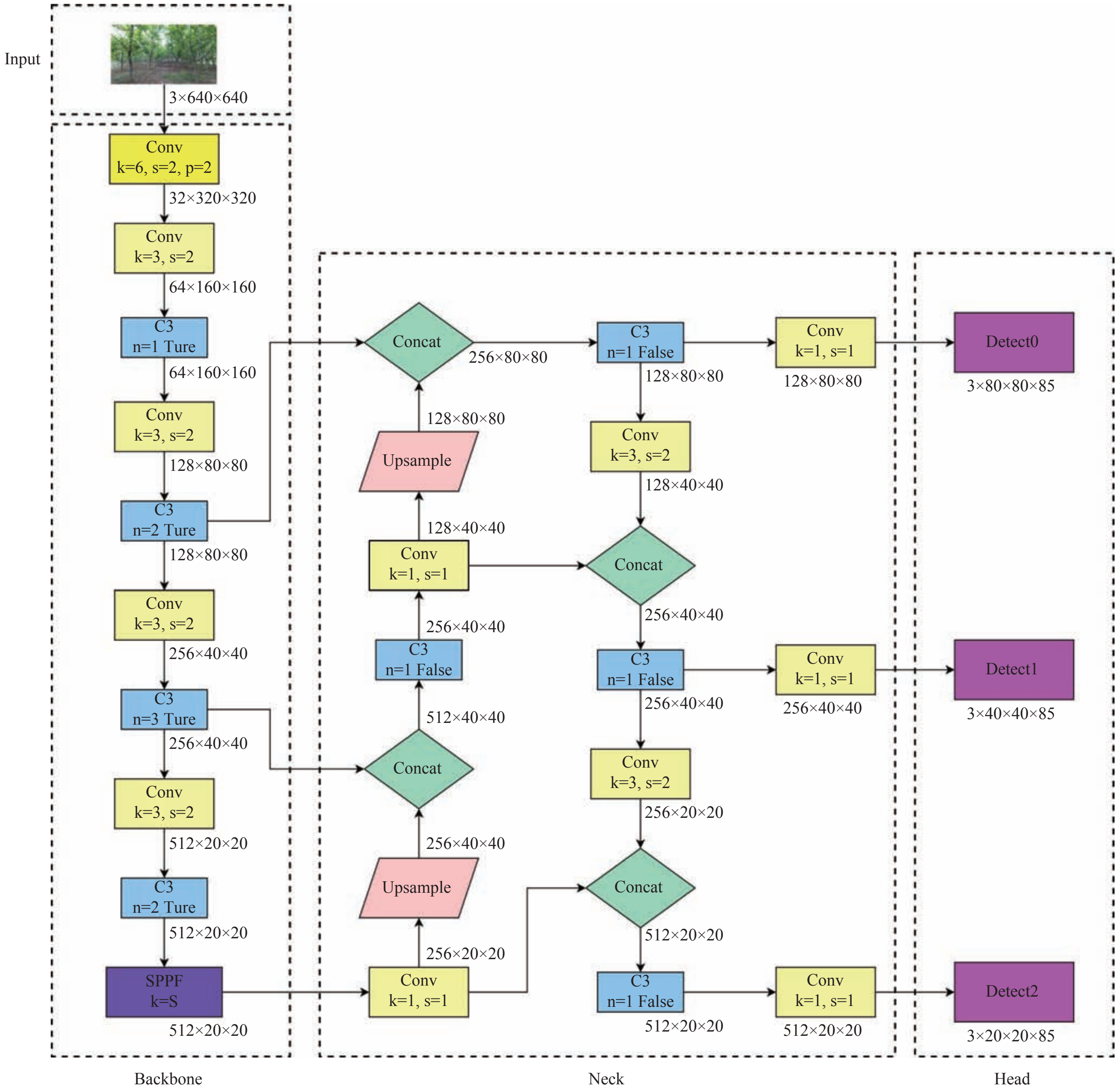


Figure 3 RPN structure

2.3 Image and data acquisition and production

The image acquisition site was on a walnut plantation at the Yuanjian Research Base in Luoyang City, Henan Province, and the image acquisition took place in May 2022. The data acquisition was carried out by remote control of the arched waist agricultural robot walking without mounted operational implements to simulate the normal working conditions of the robot. Video data acquisition was carried out via an Azure Kinect DK vision sensor mounted on the front of the robot. The vision sensor was mounted 0.6 m above the ground at the front of the robot and the camera was oriented in the direction of the robot's advance. The robot travels at a speed of approximately 0.5 m/s and the vision sensor captures images at a resolution of 1280×720 pixels at a frame rate of 30 fps. The captured video was divided into frames, and to avoid excessive image similarity in the time series, an interval extraction method was used to extract one frame every 30 frames to form the image sample data set, as shown in Figure 4a.

2990 images were selected from the collected data for the orchard dataset. The LabelImg tool was used to label the image sample dataset. The trunk of the walnut tree had many branches and no obvious boundaries, so it was not possible to label the smallest outer rectangle, therefore, this study selected the bottom of the walnut tree and the ground articulation to the emergence of the branch part of the labeled external rectangle, and the nearest three to five trees in the left and right columns were labeled as "tree". This is shown in Figure 4b.

a. Original image

b. Labeled images

Figure 4 Examples of the dataset production for the test

The data was automatically saved as a text file after annotation, the contents of which are listed in Table 1. 70% of these annotated images were used as the training set and 30% as the test set.

Table 1 Labeled data properties

Category	*x*/W	*y*/H	*w*/W	*h*/H
0	0.224 609 375	0.589 583 333	0.038 281 25	0.1625
0	0.310 937 5	0.545 833 333	0.021 875	0.086 111 111
0	0.340 625	0.520 833 333	0.0125	0.066 666 667
0	0.911 328 125	0.827 777 778	0.060 156 25	0.255 555 556
0	0.601 171 875	0.603 472 222	0.028 906 25	0.126 388 889
0	0.538 671 875	0.555 555 556	0.014 843 75	0.077 777 778

Each row in Table 1 represents the location of a tree, *x*, *y*, *w*, and *h* are normalized values relative to the scale of the image, and the value is between 0 and 1. *x* and *y* are the coordinates of the center point of the label box on the *x* and *y* axes respectively, and *w* and *h* are the width and height of the label box, respectively.

Each row in Table 1 represents the position information of a tree, which is the normalized value relative to the picture scale, ranging from 0 to 1. Where *x* and *y* represent the coordinates of the center points of the label box in the direction of the *x*-axis and *y*-axis respectively; *w* and *h* represent the width and height of the label box respectively. *W* and *H* mean that the size of the image is *W*×*H* pixels.

2.4 Model selection and training

YOLOv5 provides users with four models YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x according to the complexity of the architecture, and the performance of these four models is shown in Figure 5. YOLOv5s has the smallest network and the fastest detection speed, although the AP precision is lower, considering the large target of the detection object in this study, the small model for mobile deployment friendly and other characteristics, so the YOLOv5s model was chosen for training.

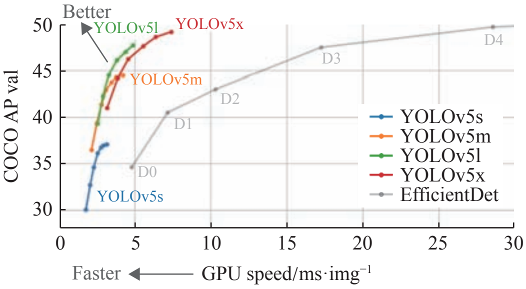


Figure 5 Performance diagram of different models of YOLOv5[21]

The Pytorch architecture was used for model training, and a GeForce RTX 2060 Super was used for GPU acceleration. The specific model training configuration is listed in Table 2.

Table 2 System environment configuration for model training

Name	Configuration information
Operating systems	Windows 11
CPU	AMD Ryzen 5 3600X 6-Core Processor 3.80 GHz
GPU	GeForce RTX 2060 Super
Video memory	8 G
Memory	32 G
Development environment	Pycharm2021,Python 3.6
Accelerated environment	CUDAv10.2,CUDNNv7.6.5

The epochs were 300, the learning rate was 0.01, the batch size was 8, the momentum was 0.937, and YOLOv5 was adaptive to image scaling, with a default image size of 1280×720.

To verify the validity of the model, both qualitative and quantitative evaluations were carried out. Qualitative evaluation means comparing whether there are any missed or false detections in the target box selection. The main indicators selected for the quantitative evaluation are Precision (*P*), Recall (*R*), and mean Average Precision (mAP). The equations are as follows:

$$P=\frac{T_p}{T_p+F_p}\times 100\% \tag{1}$$

$$R=\frac{T_p}{T_p+F_N}\times 100\% \tag{2}$$

$$\text{mAP}=\frac{\sum AP}{N} \tag{3}$$

where, *P* is the proportion of all positive predictions that are correct, %; *R* is the proportion of all real positive observations that are correct, %; AP is the average precision, %; mAP is the mean average precision, %; T_p is the positive category judged to be positive; F_p is the negative category judged to be positive; F_N is the positive category judged to be negative; *N* is the sum of different categories of targets for this image.

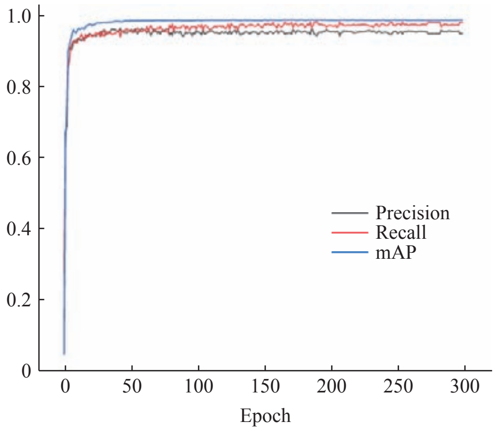
As shown in Figure 6, after 65 Epoch iterations of the model, P reaches 0.912 60, R reaches 0.939 09, and mAP reaches 0.958 89, indicating that the model has high prediction accuracy.

2.5 Fruit tree row detection and fitting of navigation paths

Use the trained network model to detect 1000 images of the collected fruit trees, the detection effect images are shown in Figure 7, in which 963 images were correctly detected and 37 images had false detections. The values at the top right of the target box in the figure indicate the confidence level of the different class labels, and it can be seen from the figure that the model achieves a confidence level of about 90% for the detection of the left and right rows of trees.

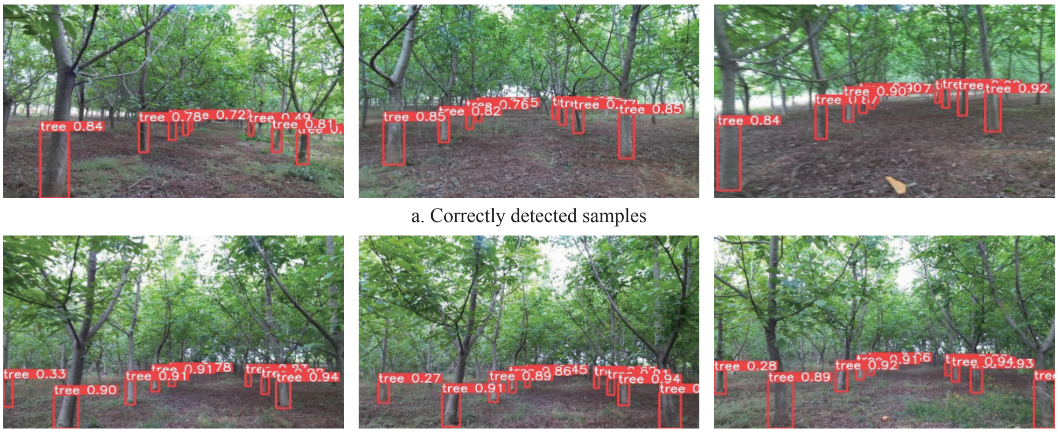
To address the impact of the occurrence of false detections on the fruit tree row fit, the confidence levels of the data from the 1000 images mentioned above were counted and the results are shown in Figure 8.

Based on the statistical results, the confidence level of correctly identified targets ranged from 0.72-0.98, while the confidence level of misidentified targets ranged from 0.20-0.60, so a threshold of 0.70 was chosen. Targets with confidence levels below the threshold were considered as misidentified targets and the data were not used for quadratic polynomial fitting and fruit tree row line fitting.



Note: mAP is the mean average precision

Figure 6 The relationship between Epoch and Model precision



a. Correctly detected samples

b. Samples with false detections

Figure 7 Examples of the detection results of the test

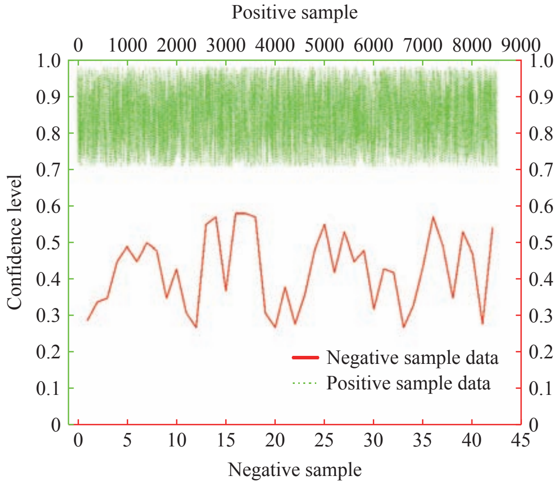


Figure 8 Confidence level distribution of detection results

As shown in Figure 9, the midpoint of the bottom of the bounding box is close to the bottom of the trunk of the fruit tree, so the final navigation path is obtained by fitting a straight line to the bottom of the fruit tree on the left and right sides respectively, and then finding the midline. The detected targets are divided into two parts based on their spatial location using quadratic polynomial extreme points. First, a quadratic polynomial is fitted to the midpoint of the base of all identified targets. The equation of the fitted curve is set to:

$$y = ax^2 + bx + c \tag{4}$$

Use Python’s least-squares functional leastsq to find the three parameters a_1, b_1, and c_1, respectively.

Then the coordinates of the extreme value point x_1 can be expressed as Equation (5):

$$x_1 = -\frac{b_1}{2a_1} \tag{5}$$

Use Python’s least-squares functional leastsq, and bring in the bottom midpoints of the targets on each side to obtain a line at the bottom of the left side:

$$y = a_2 x + b_2 \tag{6}$$

Right bottom line:

$$y = a_3 x + b_3 \tag{7}$$

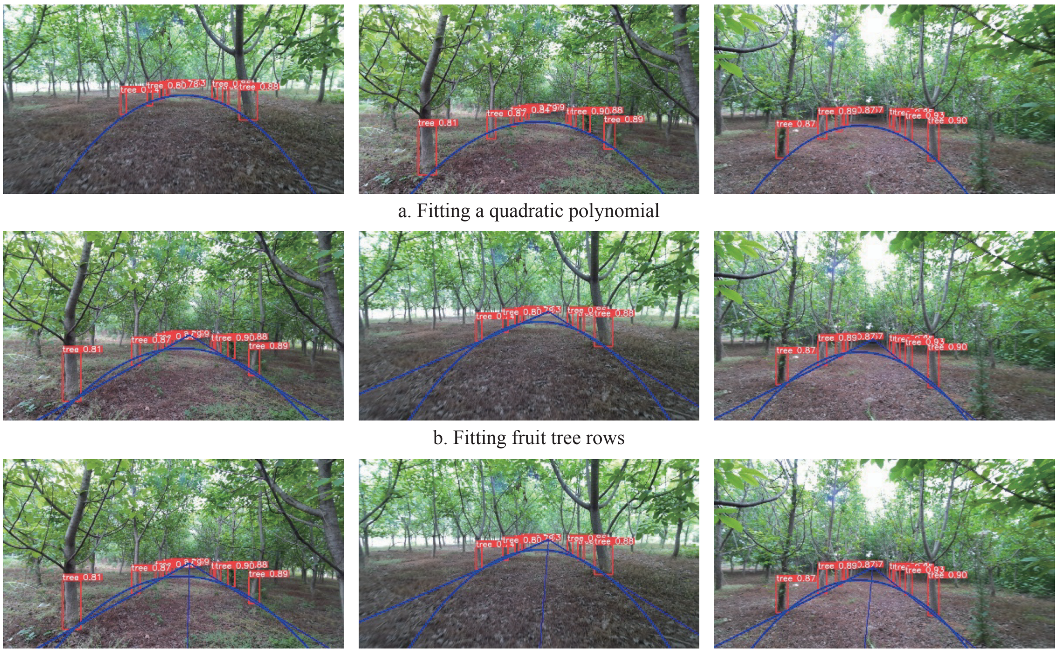
Slope k is obtained by the slope of the median Equation (8):

$$k = \tan\left(\frac{\arctan(a_2) + \arctan(a_3)}{2}\right) \tag{8}$$

The slope k is found by Equation (8), and the intersection of the two lines is found by combining Equation (6) with Equation (7) to finally find the midline Equation (9):

$$y = kx + b \tag{9}$$

The resulting midline $y = kx + b$ is used as the final navigation path.



a. Fitting a quadratic polynomial

b. Fitting fruit tree rows

c. Find the center line

Figure 9 Navigation path fitting results

2.6 Analysis of navigation path fitting results

To verify the precision of the navigation path fitting method, the data from 3000 images were randomly divided into 3 groups and tested separately, and the detection results are listed in Table 3. The average precision remained above 96% and the detection frame rate was above 38 fps. The precision and performance of the navigation path fitting method fully met the requirements for use in navigation.

Table 3 Statistical table of test results

Group	Number of samples	Detected correctly	Detected time	Precision	Average precision
A	1000	988	26.31	98.8%	97.73%
B	1000	965	28.57	96.5%	
C	1000	979	29.41	97.9%	

2.7 Control algorithms

In practice, the chassis needs to be able to eliminate pose errors (heading error and lateral error) in the shortest possible time and the output angle power should also be as small as possible, hence the LQR control algorithm introduced in this study. The mobile platform model is simplified, as shown in Figure 10. Based on the geodetic coordinate system OXY, the position of the mobile platform at this moment is defined as (x, y), the speed is v (m/s), the wheelbase is l (m), the heading angle is φ (rad), and the output angle is δ (rad).

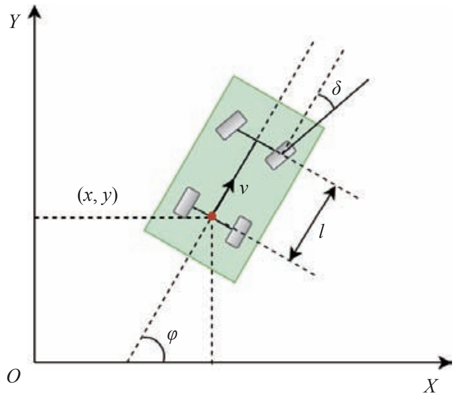


Figure 10 Kinematic model of the experiment platform

Establish the kinematic equations for the chassis:

$$\dot{x} = v_x = v\cos\varphi \tag{10}$$

$$\dot{y} = v_y = v\sin\varphi \tag{11}$$

$$\dot{\varphi} = \frac{v\tan\delta}{l} \tag{12}$$

where, $\dot{x}$ is the amount of change in the position of the mobile platform in the X-axis direction, $\dot{y}$ is the amount of change in the position of the mobile platform in the Y-axis direction and φ is the change in the heading angle of the mobile platform.

From Equations (10)-(12), it follows that:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} v\cos\varphi \\ v\sin\varphi \\ \dfrac{v\tan\varphi}{l} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{13}$$

The state quantity $\dot{x} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix}$ is chosen and the control quantity is $u = \begin{bmatrix} v \\ \delta \end{bmatrix}$. Then for any reference point of the reference trajectory, denoted by r, the above equation can be rewritten as:

$$\dot{x}_r = f(x_r, u_r) \tag{14}$$

where, $\dot{x}_r = \begin{bmatrix} x_r \\ v_r \\ \varphi_r \end{bmatrix}$, $u = \begin{bmatrix} v \\ \delta \end{bmatrix}$. For the above equation using Taylor series expansion at the reference point and ignoring higher order terms:

$$\dot{x} = f(x_r, u_r) + \frac{\partial f(x_r, u_r)}{\partial x}(x - x_r) + \frac{\partial f(x, u)}{\partial u}(u - u_r) \tag{15}$$

Finding Jacobi matrices for $\dfrac{\partial f(x_r, u_r)}{\partial x}$ and $\dfrac{\partial f(x, u)}{\partial u}$, we have:

$$\frac{\partial f(x_r,u_r)}{\partial x}=\begin{bmatrix}\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \varphi}\\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \varphi}\\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \varphi}\end{bmatrix}=\begin{bmatrix}0 & 0 & -v_r\sin\varphi_r\\ 0 & 0 & v_r\cos\varphi_r\\ 0 & 0 & 0\end{bmatrix} \tag{16}$$

$$\frac{\partial f(x,u)}{\partial u}=\begin{bmatrix}\frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \delta}\\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \delta}\\ \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \delta}\end{bmatrix}=\begin{bmatrix}\cos\varphi_r & 0\\ \sin\varphi_r & 0\\ \frac{\tan\varphi_r}{l} & \frac{v_r}{l\cos^2\delta_r}\end{bmatrix} \tag{17}$$

Amount of change in state volume error:

$$\dot{\tilde{x}}=\begin{bmatrix}\dot{x}-\dot{x}_r\\ \dot{y}-\dot{y}_r\\ \dot{\varphi}-\dot{\varphi}_r\end{bmatrix}=\begin{bmatrix}0 & 0 & -v_r\sin\varphi_r\\ 0 & 0 & v_r\cos\varphi_r\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x-x_r\\ y-y_r\\ \varphi-\varphi_r\end{bmatrix}+$$
$$\begin{bmatrix}\cos\varphi_r & 0\\ \sin\varphi_r & 0\\ \frac{\tan\varphi_r}{l} & \frac{v_r}{l\cos^2\delta_r}\end{bmatrix}\begin{bmatrix}v-v_r\\ \delta-\delta_r\end{bmatrix}\Rightarrow\dot{\tilde{x}}=A\tilde{x}+B\tilde{u} \tag{18}$$

The above equation shows that state error quantities can form a linear state space. A forward Eulerian discretization of the above equation yields:

$$\dot{\tilde{x}}=\frac{\tilde{x}(k+1)-\tilde{x}(k)}{T}=A\tilde{x}+B\tilde{u} \tag{19}$$

Collation gives:

$$\tilde{x}(k+1)=(TA+E)\tilde{x}(k)+TB\tilde{u}(K)=\begin{bmatrix}1 & 0 & -Tv_r\sin\varphi_r\\ 0 & 1 & Tv_r\cos\varphi_r\\ 0 & 0 & 1\end{bmatrix}\tilde{x}(k)+$$
$$\begin{bmatrix}T\cos\varphi_r & 0\\ T\sin\varphi_r & 0\\ T\frac{\tan\varphi_r}{l} & T\frac{v_r}{l\cos^2\delta_r}\end{bmatrix}\tilde{u}(K)=\tilde{A}\tilde{x}(k)+\tilde{B}\tilde{u}(K) \tag{20}$$

Consider the following two characteristics of the desired system response:

1) The ability to quickly and steadily approach 0 and maintain balance in terms of pose errors;

2) The front wheel angle control output is as small as possible.

Therefore, this study introduces a loss function, expressed as the weighting of the accumulated tracking error of the tracking process and the accumulated control output, as in the following equation:

$$J=\sum_{k+1}^{N}(X^TQX+u^TRu) \tag{21}$$

where, ***Q*** is the state weighting matrix of the semi-positive definite matrix and ***R*** is the control weighting matrix of the positive definite matrix. ***Q*** larger matrix elements imply that the tracking error is expected to converge to zero quickly; ***R*** larger matrix elements imply that the control output is expected to be as small as possible.

Therefore, the former optimization objective represents the cumulative size of the tracking error, and the second optimization objective represents the loss of energy in the tracking process, which transforms the trajectory tracking control problem into an optimization control problem. For the optimization solution of the objective Equation (21), the optimal control law ***u*** is a linear function with respect to the state variable ***X***:

$$u=-[(R+B^TPB)^{-1}B^TPA]X=-KX \tag{22}$$

where, ***P*** is the solution of the following Riccati equation of Equation (23):

$$P=A^TPA-A^TPB(R+B^TPB)^{-1}B^TPA+Q \tag{23}$$

It is obtained by MATLAB R2020a simulation that the positional error can quickly and steadily converge to 0 when $Q=\begin{bmatrix}0.5 & 0 & 0\\ 0 & 0.5 & 0\\ 0 & 0 & 1\end{bmatrix}$ and $R=\begin{bmatrix}0.5 & 0\\ 0 & 1\end{bmatrix}$. Currently, the body system meets our design requirements and works best.

3 Results and analysis

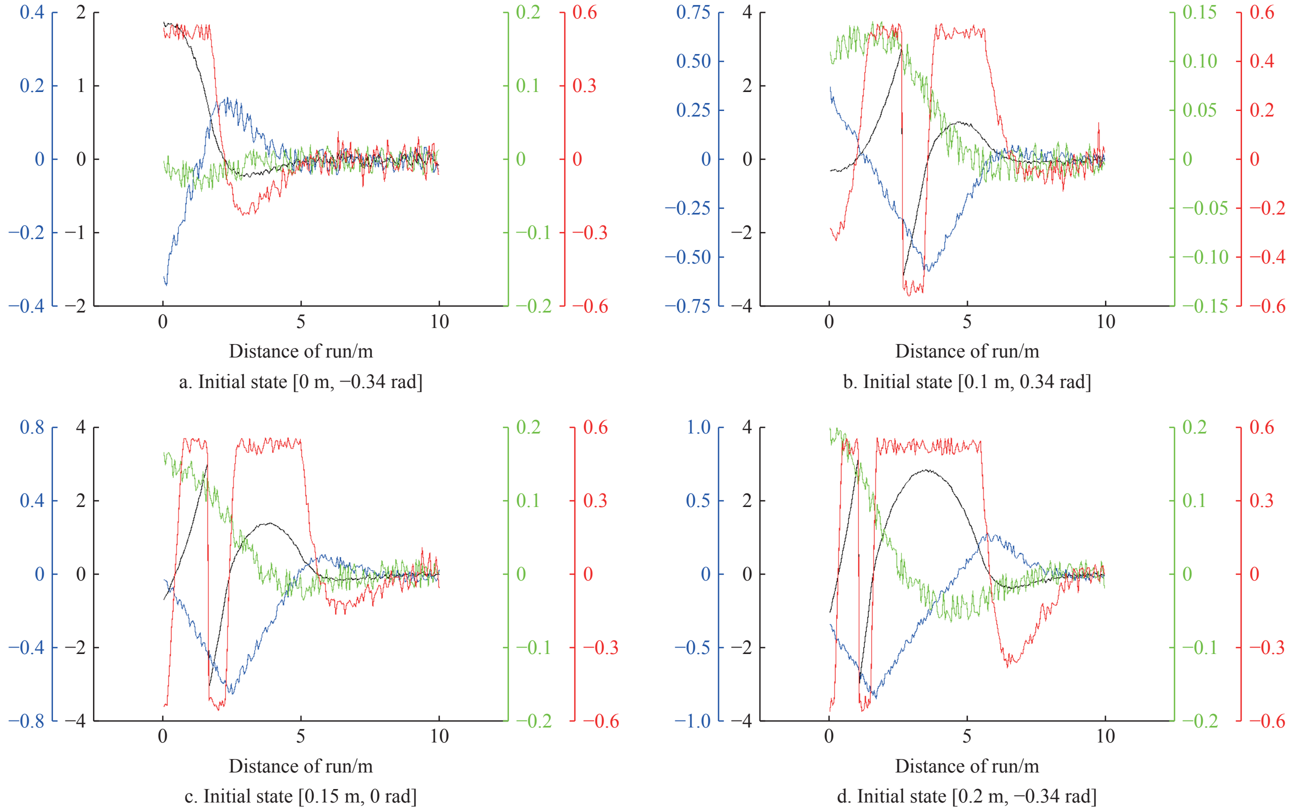
3.1 Experimental process

To verify the precision of the algorithms in this study, visual navigation experiments were conducted on the experiment platform in a walnut orchard at the Yuanjian Research Base in Luoyang City. Walnut forest stands are spaced 4 m apart in rows and 2 m apart in plants, with walnut trees about 5 m high. The camera was positioned at the front of the center line of the longitudinal axis of the experiment platform, at a height of 0.6 m above the ground, with a pitch angle of 0 rad, a sampling frame rate of 30 fps, and a forward speed of 0.5 m/s. The experiment platform was operated in four initial states with lateral and heading errors of [0 m, −0.34 rad], [0.1 m, 0.34 rad], [0.15 m, 0 rad], and [0.2 m, −0.34 rad], and was fully autonomous under the control of the machine vision system. The average errors, standard deviation, steady-state error, adjustment time, and adjustment efficiency were used to quantify the navigation experimental effects. The experimental effects and experimental data are presented in Figure 11, Table 4, and Table 5. Travel distance is the distance of the current position of the experiment platform from the starting point; errors are the pose errors of the experiment platform, the sum of lateral error and heading error; steady state is the state when the experiment platform converges to a lateral error of ±0.03 m and heading error of ±0.05 rad; adjustment time is the time required to adjust the experiment platform from the starting point to the steady state; adjustment efficiency is the average attitude error corrected per second.

3.2 Analysis and discussion

1) The average lateral errors of the experiment platform in the 4 different initial states were 0.0163, 0.0807, 0.0789, and 0.0607 m, and the average heading error was 0.1312, 0.2747, 0.3473, and 0.3615 rad, respectively. The average lateral error for the different initial states was 0.059 m and the average heading error was 0.2787 rad. When the experiment platform travels to a steady state, the steady state lateral errors are 0.0101, 0.0102, 0.0118, and 0.0088 m, with a mean value of 0.0102 m; the steady state heading error is 0.0178, 0.0258, 0.0379, and 0.0195 rad, with a mean value of 0.0253 rad. Based on the above analysis, it can be concluded that the experiment platform can quickly correct errors and quickly reach a stable state by driving autonomously along the desired navigation path of the orchard, and when it reaches a stable state, its steady-state error can be maintained at around 0.01 m and 0.026 rad, meeting the requirements of orchard navigation precision.

2) In data statistics, the standard deviation reflects the degree of dispersion of the data. In this experiment, the driving stability of the experiment platform can be reflected. The standard deviation of the lateral error in the four initial states were 0.0115, 0.0441, 0.0539,



a. Initial state [0 m, −0.34 rad]

b. Initial state [0.1 m, 0.34 rad]

c. Initial state [0.15 m, 0 rad]

d. Initial state [0.2 m, −0.34 rad]

Note: —— Heading error (rad); —— Lateral error (m); —— Actual angles (rad); —— Expect angles (rad).

Figure 11 Navigation effect of the experiment platform in different initial states

Table 4 Error information in different initial states

Initial state	Positioning	Average errors	Standard deviation	Steady-state error	Steady-state standard deviation
[0 m, −0.34 rad]	Heading error/rad	0.1312	0.0798	0.0178	0.0119
	Lateral error/m	0.0163	0.0115	0.0101	0.0059
[0.1 m, 0.34 rad]	Heading error/rad	0.2747	0.1587	0.0258	0.0183
	Lateral error/m	0.0807	0.0441	0.0102	0.0067
[0.15 m, 0 rad]	Heading error/rad	0.3473	0.1804	0.0379	0.0290
	Lateral error/m	0.0789	0.0539	0.0118	0.0075
[0.2 m, −0.34 rad]	Heading error/rad	0.3615	0.2454	0.0195	0.0169
	Lateral error/m	0.0607	0.0580	0.0088	0.0054

Table 5 Control efficiency under different initial states

Initial state	Adjustment time/s	Adjustment efficiency	Steady distance/m
[0 m, −0.34 rad]	6.8	0.050	3.678
[0.1 m, 0.34 rad]	10.5	0.042	5.490
[0.15 m, 0 rad]	8.9	0.017	4.526
[0.2 m, −0.34 rad]	15.3	0.035	7.658

and 0.0580 m, respectively, and the standard deviation of the heading error was 0.0798, 0.1587, 0.1804, and 0.2454 rad, respectively; while the steady-state standard deviation of the lateral error was 0.0059, 0.0067, 0.0075, and 0.0054 m, respectively, and the steady-state standard deviation of heading error was 0.0119, 0.0183, 0.0290, and 0.0169 rad, respectively. From the above data, the standard deviation of the experiment platform is 0.0418 m and 0.1662 rad, respectively, while the steady-state standard deviation is 0.0064 m and 0.0190 rad, both of which have a low degree of dispersion, indicating that the experiment platform can travel stably under various initial states and has a high degree of stability.

3) High-efficiency performance is one of the important parameters to detect the goodness of the system, which reflects the size of the system’s ability to handle errors. The time consumed by the system from the appearance of the error to the next steady state and the adjustment efficiency, the attitude error corrected by the system per second can reflect the efficient performance of the system. In the 4 different initial states, the adjustment times were 6.8, 10.5, 8.9, and 15.3 s, with a mean value of 10.4 s, and their corresponding steady-state distances were 3.678, 5.49, 4.526, and 7.658 m, respectively; the adjustment efficiencies were 0.050, 0.042, 0.017 and 0.035, with a mean value of 0.036. In summary, the time required to reach a steady state for different initial states of the experiment platform is 10.4 s on average, with 0.036 attitude errors corrected per second, which has a highly efficient performance.

4 Conclusions

Aiming at the low accuracy of the current navigation method due to diverse tree shapes, complex orchard environments, and changing lighting conditions, a fruit tree detection and navigation control algorithm based on YOLOv5 was proposed to solve these problems. The test results in four different initial states show that the steady-state standard deviation of the test platform remains at 0.01 m and 0.026 rad. The steady-state standard deviation was 0.0064 m and 0.0190 rad, respectively, the regulation efficiency was 0.036, and the adjustment time was 10.4 s, which could meet the requirements of independent navigation operation in orchards. In this study, autonomous navigation between orchard rows of fruit trees was realized, but due to the limitations of image sensors, the boundary of fruit tree rows could not be judged, so the experimental platform could not independently complete the U-turn at the end of the fruit tree rows. In the follow-up study, it is proposed to use lidar to realize the autonomous navigation of orchards to make up for the shortcomings of this study.

Acknowledgements

This work was funded by the National Key Research and Development Program of China Project (Grant No. 2021YFD2000700), the National Natural Science Funds for Young Scholars of China (Grant No. 51905154), and the Luoyang Public Welfare Special Project (Grant No. 2302031A).

[References]

[1] Wang S S, Geng L X. Development situation and countermeasures of agricultural mechanization in hilly and mountain areas. Agricultural Engineering, 2016; 6(5): 1–4. (in Chinese)

[2] Zhang P, Zhang L N, Liu D, Wu H X, Jiao B. Research status of agricultural robot technology. Agricultural Engineering, 2019; 9(10): 1–12. (in Chinese)

[3] Yan Q T, Li L X, Qiu Q, Zhang Z G, Meng Z J. Research status and development trend of small-mobile agricultural robots. Journal of Chinese Agricultural Mechanization, 2019; 40(5): 178–186. (in Chinese)

[4] Nagle M, Intani K, Romano G, Mahayothee B, Sardsud V, Müller J. Determination of surface color of 'all yellow'mango cultivars using computer vision. Int J Agric & Biol Eng, 2016; 9(1): 42–50.

[5] Bi S, Wang Y H. Inter-line pose estimation and fruit tree location method for orchard robot. Transactions of the CSAM, 2021; 52(8): 16–26, 39. (in Chinese)

[6] Pedersen S M, Fountas S, Sørensen C G, Van Evert F K, Blackmore B S. Chapter Robotic seeding: Economic perspectives. In: Precision Agriculture: Technology and Economic Perspectives, Springer, 2017; pp.167–179.

[7] Zhang L, Zhang H D, Chen Y D, Dai S H, Li X M, Imou K J, Liu Z H, Li M. Real-time monitoring of optimum timing for harvesting fresh tea leaves based on machine vision. Int J Agric & Biol Eng, 2019; 12(1): 6–9.

[8] Steward B, Gai J Y, Tang L. The use of agricultural robots in weed management and control. Burleigh Dodds Science Publishing, 2019; pp.161–186.

[9] Zhang M J. Design of obstacle avoidance system of orchard lawn mower based on binocular vision technology. Master's dissertation. Baoding: Hebei Agricultural University, 2019; 63p. (in Chinese)

[10] Liu B, Yang C H, Xiong L Y, Wang K, Wang Y. Path recognition method of picking robot based on orchard natural environment. Jiangsu Journal of Agricultural Sciences, 2019; 35(5): 1222–1231. (in Chinese)

[11] Mao F D. Strcture design and vision navigation research on mobile platform of picking robot in the hills and mountains. Master's dissertation. Chongqing: Chongqing University of Technology, 2017; 103p. (in Chinese)

[12] Liu H, Zhu S H, Shen Y, Tang J H. Trunk fast segmentation algorithm based on multi-feature fusion. Transactions of the CSAM, 2020; 51(1): 221–229. (im Chinese)

[13] Shalal N, Low T, McCarthy C, Hancock N. Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion-Part A: Tree detection. Computers and Electronics in Agriculture, 2015; 119: 267–278.

[14] Zeng F, Wang C, Ge S S. A survey on visual navigation for artificial agents with deep reinforcement learning. IEEE Access, 2020; 8: 135426–135442.

[15] Han Z H, Li J, Yuan Y W, Fang X F, Zhao B, Zhu L C. Path recognition of orchard visual navigation based on U-Net. Transactions of the CSAM, 2021; 52(1): 30–39. (in Chinese)

[16] Bonadies S, Gadsden S A. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. Engineering in Agriculture, Environment and Food, 2019; 12(1): 24–31.

[17] Radcliffe J, Cox J, Bulanon D M. Machine vision for orchard navigation. Computers in Industry, 2018; 98: 165–171.

[18] Sharifi M, Chen X Q. A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards. 6th International Conference on Automation, Robotics and Applications (ICARA), IEEE, 2015; pp.251–255.

[19] Yang S J, Mei S L, Zhang Y E. Detection of maize navigation centerline based on machine vision. IFAC-PapersOnLine, 2018; 51(17): 570–575.

[20] Yi S, Li X R, Wu Z J, Zhu J M, Yuan X S. Night hare detection method based on infrared thermal imaging and improved YOLOv3. Transactions of the CSAE, 2019; 35(19): 223–229. (in Chinese)

[21] Zhang W K, Sun H, Chen X K, Li X B, Yao L G, Dong H. Research on weed detection in vegetable seedling field based on improved YOLOv5. Journal of Graphics, 2023; 44(2): 346–356. (in Chinese)

[22] Sivakumar A N, Modi S, Gasparino M V, Ellis C, Velasquez A E B, Chowdhary G, et al. Learned visual navigation for under-canopy agricultural robots. Agricultural Robots, 2021; arXiv: 2107. 02792, 2021.

[23] Kurtulmuş F, Alibaş İ, Kavdır I. Classification of pepper seeds using machine vision based on neural network. Int J Agric & Biol Eng, 2016; 9(1): 51–62.