DEVELOPMENT OF AUTOMATIC SUB-CROWN DATA COLLECTION EQUIPMENT FOR CORN BREEDING

Yongliu WANG [1, *]

With the growth of smart agriculture, deep perception technologies for crop monitoring and pest/disease prediction are becoming essential for precise crop growth control. This paper presents the development of an automatic data collection system for crop canopy sensing and a vision-based navigation control algorithm to enhance smart agriculture. The YOLOv8 object detection model, optimized with TensorRT, ensures real-time image processing on the platform's main controller. A fuzzy control algorithm utilizes crop root target information and IMU heading deviations to adjust motor speed for stable navigation and furrow switching. To mitigate motion blur, a method based on optical flow method is employed to improve image clarity and detection accuracy. Experimental results show that the model achieves an inference time of 25ms per image with 88% accuracy. The fuzzy control algorithm effectively reduces heading deviations and lateral errors. This system improves real-time performance, supports decision-making, reduces human intervention, and promotes precision agriculture.

Keywords: smart agriculture, machine vision, navigation control, YOLOv8, optical flow method.

1. Introduction

Maize, as a key food crop, has a growing sown area and the greatest potential for yield increase among cereal crops, driving grain output. With the rapid advancement of technology, smart agriculture enhances real-time monitoring of crops and farm environments, aiming to boost grain yield, reduce costs, and minimize resource wastage.

Researchers have developed various crop data collection platforms. Velásquez et al. [1] created a cyber-physical system for collecting data from coffee crop test plots. The University of Illinois [2] designed a tracked robotic platform with an adjustable mast to capture canopy data at different crop growth stages. Zhang et al. [3] combined drone imagery with the YOLOv3 algorithm for rapid field crop data collection, enabling automatic identification of crop types from drone images.

* Corresponding author

[1] School of Instrument Science and Engineering, Harbin Institute of Technology, Harbin 150000, China, e-mail: 18263159379@163.com

Image processing in agricultural robot navigation involves analyzing captured images to identify crops and then applying techniques like Hough transforms, linear regression, or template matching to fit crop rows. Yang [4] applied the least squares method to collect robot image data and used the linear judgment dimensionality reduction algorithm to cluster and analyze the trajectory data of robot visual navigation. Higuti et al. [5] designed an autonomous navigation system using 2D LiDAR to process raw data, extract crop row information, and use a PID algorithm to control the robot's movement along the crop rows.

Recent years have seen deep learning-driven target recognition algorithms, such as YOLO, widely used in image processing. Dos Reis et al. [6] developed a vision system based on YOLO to recognize static obstacles using a Microsoft Kinect sensor, with an Nvidia Jetson TX2 GPU for improved image processing. Yu et al. [7] explored five deep learning-based methods for field navigation line extraction, deploying them on an embedded system for autonomous robot navigation. They used semantic segmentation to segment field roads and polygon fitting to extract navigation lines.

This study proposes a solution to address the high cost of manual data collection, incomplete data, and navigation instability using deep learning and computer vision. By combining YOLOv8 target detection, IMU heading angle data, and a fuzzy control algorithm, a vision-based navigation control system is implemented. Additionally, an optical flow method reduces image motion blur caused by camera jolts. A platform vehicle is designed to autonomously navigate a maize planting environment while performing crop data collection.

2. Overall modular design of the platform vehicle

This study adopts a modular design approach for the overall system, which consists of three parts: a drive-by-wire chassis system, a navigation system, and a data collection system. The overall structural schematic is given in Fig. 1.

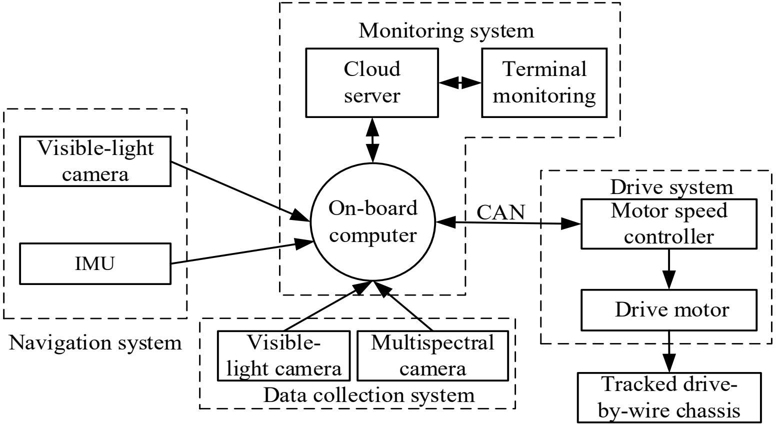

Fig. 1. Schematic diagram of the overall structure

2.1 Drive-by-wire chassis system design

In this study, a tracked drive-by-wire chassis is used with design indicators including a minimum travel speed of 0.3m/s, a climbing angle of 10°, obstacle-crossing width of 100 mm, height of 50 mm, and a battery life over 1 hour. The design process includes selecting the drive motor based on speed and track radius, designing a transmission system for climbing and obstacle-crossing, calculating and selecting the battery, and designing the structure to meet ground contact and weight distribution requirements.

The drive motor's power and the transmission system design must ensure that the maximum speed $v_{\max}$ satisfies Equation (1), where ω is the motor's angular velocity in rad/s, and r is the radius of the drive wheel in meters.

$$v_{\max} = \omega \cdot r \tag{1}$$

The track's driving force must be sufficient to overcome the additional resistance caused by the incline. The condition for this is expressed in Equation (2), where F_{drive} represents the driving force of the track, G is the component of gravitational force acting along the incline(both in N), and θ is the slope angle. Equation (3) defines the driving force F_{drive} in terms of the motor torque τ (in $N \cdot m$) and the radius of the drive wheel r (in meters). Combining these, the required motor torque can be calculated using Equation (4), where m is the mass of the vehicle (in kg), and g is the acceleration due to gravity (in m/s^2).

$$F_{drive} \geq G \cdot \sin(\theta) \tag{2}$$

$$F_{drive} = \tau \cdot r \tag{3}$$

$$\tau \geq \frac{m \cdot g \cdot \sin(\theta)}{r} \tag{4}$$

The motion of the tracked chassis is controlled by its linear and angular velocities, with track speeds determining straight or circular movement. Fig. 2 shows the kinematic model of the tracked chassis, with COM representing the centroid and ICR the instantaneous center of rotation.

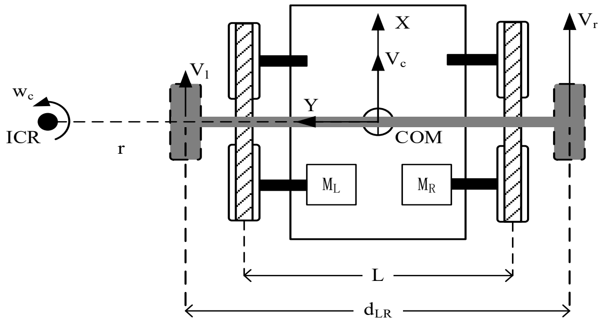

Fig. 2. Kinematic model of the tracked drive-by-wire chassis

Equation (5) describes the relationship between track speed and motor speed, where i is the transmission ratio, n is the motor speed (r/min), and d is the wheel diameter (m).

$$v = \frac{n}{i} \times \pi \times D \tag{5}$$

Assuming no track slip and the COM along the symmetry line, the system simplifies to a two-wheel differential model. The linear velocity, angular velocity, and turning radius can be obtained using Equations (6), (7), and (8), respectively.

$$v_c = \frac{v_r + v_l}{2} \tag{6}$$

$$\omega_c = \frac{v_r - v_l}{d_{LR}} \tag{7}$$

$$r = \frac{v_c}{\omega_c} = \frac{d_{LR}}{2} \times \frac{v_r + v_l}{v_r - v_l} \tag{8}$$

where, d_{LR} is the virtual track width (m), v_r is the speed of the right track (m/s), and v_l is the speed of the left track (m/s). In Equation (8), there is a parameter that changes with operating conditions, which can be calculated using Equation (9):

$$d_{LR} = \gamma L \tag{9}$$

In this equation, L is the width (m), and γ is a dimensionless parameter related to the load, the ground, and other factors.

Two M3508 DC brushless geared motors are used as the power source. Each motor in the platform chassis is connected to a C620 brushless motor speed controller. The onboard computer communicates with the speed controllers via the CAN bus. The M3508's rated voltage is 24 V, motor weight is 365 g, and the reduction ratio is 3591/187. Under rated voltage conditions, paired with the C620, the motor's no-load/nominal speed is 482/469 rpm, no-load/nominal current is 0.78/10 A, rated torque is 3 N·m, and maximum efficiency can reach 70%.

2.2 Navigation system design

In this study, the YOLOv8 target detection algorithm is used for maize crop recognition and detection. The crops appear green and soil appears brown, creating a clear contrast. The green plant and soil regions have strip-like features, and crops are planted in rows along straight or nearly straight lines, facilitating crop row recognition for unmanned agricultural operations. YOLOv8 is a variant of the YOLO family of target detection algorithms [8].

This study proposes a method to address image motion blur caused by rapid camera movement. The method, based on optical flow and information from the previous frame, improves blurred images. The processed images are used as input to the fuzzy control algorithm, which adjusts the platform vehicle's rotational speed to ensure it stays centered in the furrow. If no root targets are detected, the platform performs a turning and row-switching operation, signaling the end of the crop row. The chosen camera for this study is the SY011HD, featuring a resolution of 1920× 1080, a frame rate of 30 Hz, a focal length of 3.5 mm, and a horizontal field of view of 85°and vertical field of view of 60°.

2.3 Data collection system design

The data collection system consists of a multispectral camera and a visible-light camera. The multispectral camera is the RedEdge-MX five-band multispectral camera developed by Micasense, USA. It is capable of capturing five distinct spectral bands at once: red, green, blue, red-edge, and near-infrared. By analyzing the collected multispectral image data, indices such as the vegetation coefficient of the crop can be obtained. The characteristic parameters of the RedEdge-MX are shown in Table 1.

Table 1

RedEdge-MX characteristic parameters

Name	Parameter
WEIGHT	232g
Dimensions	8.7cm×5.9cm×4.54cm
Power Supply	4.2V-15.8V
Power Consumption	4/8W(Normal/Peak)
Sensor Resolution	1280×960
resolution	47.2°
Field of View	HFOV
Capture Rate	1 capture/second

The system controls the multispectral camera via its HTTP interface by sending a POST request to capture and store images. The images are downloaded through their URLs and saved locally. Afterward, a DELETE request removes the files from the camera. Meanwhile, the USB visible-light camera captures images continuously using OpenCV and stores them locally.

3. Design and implementation of the visual navigation algorithm

The proposed visual navigation algorithm uses the YOLOv8 target detection model to identify crop roots. After deploying the engine model to the development board, GPU-accelerated inference with TensorRT boosts detection speed. To address motion blur from rapid camera movement, a method based on optical flow and adjacent frame information is used to improve image quality.

3.1 YOLOv8 algorithm

The YOLOv8 algorithm is a lightweight improvement based on YOLOv5. Compared to YOLOv5, it improves the model's ability to adapt to targets of various sizes and shapes, striking a balance between accuracy and speed, and is suitable for real-time-critical scenarios [9-11]. The network structure of YOLOv8 is primarily composed of three components: Backbone, Neck, and Head.

The Backbone utilizes a combination of convolution and deconvolution layers for feature extraction, incorporating residual connections and bottleneck structures to reduce the network size and enhance performance.

The Neck part consists of an SPPF module, a PAA module, and two PAN modules, which fuse feature maps from various levels of the Backbone through multi-scale feature fusion.

The Head part extracts information features at three different scales through three detection layers. It comprises a detection head and a classification head, responsible for the final target detection and classification tasks [12-14].

3.2 YOLOv8 model conversion and deployment

When training the YOLOv8 model with a crop root dataset using multiple GPUs, distributed training speeds up the process. However, deploying the model on a single GPU or embedded platform introduces challenges due to the performance difference. The large model size can lead to slow inference and high latency, which may not meet the real-time requirements for navigation tasks.

To improve deployment inference speed, the trained YOLOv8 model is optimized using TensorRT. First, the model is converted to ONNX format for deployment, ensuring framework interoperability. Then, the ONNX model is converted into a TensorRT engine format, optimizing inference speed on edge platforms by generating an efficient inference engine for GPU and network configurations.

During inference, the saved engine file is loaded and deserialized to create a TensorRT inference engine. On the Nvidia Jetson Xavier NX platform, this engine performs real-time inference on images from the camera, providing target detection results for the visual navigation control module [15].

3.3 Navigation control algorithm combining vision and IMU

This study's navigation control algorithm consists of two parts: processing YOLOv8 target detection results and designing the fuzzy control algorithm. In the first part, the nearest left and right root targets are identified, and the lateral deviation (center offset) is calculated based on the detection results. The flow chart for computing lateral deviation is shown in Fig. 3.

For the second part, a fuzzy controller must be designed first. The structure of the fuzzy controller is shown in Fig. 4. The fuzzy control system performs fuzzification on input variables, converting them into fuzzy variables. These are used by the fuzzy inference module, which relies on a knowledge base (database and rule base) to make decisions. The rule base defines the fuzzy relationships between input and output variables. After fuzzy inference, a defuzzification step is applied to convert fuzzy control quantities into precise control outputs for the system [16-18].

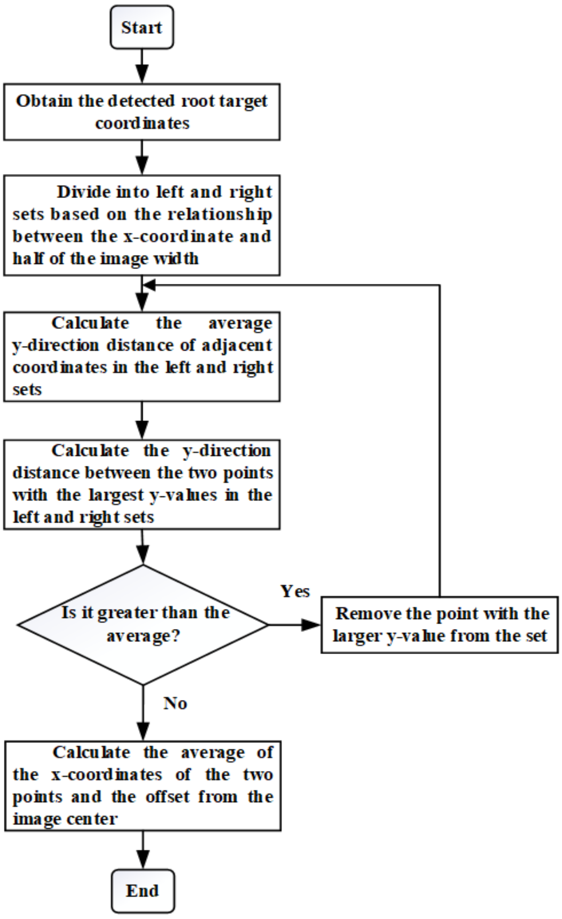
Start
Obtain the detected root target coordinates
Divide into left and right sets based on the relationship between the x-coordinate and half of the image width
Calculate the average y-direction distance of adjacent coordinates in the left and right sets
Calculate the y-direction distance between the two points with the largest y-values in the left and right sets
Is it greater than the average?
Yes
Remove the point with the larger y-value from the set
No
Calculate the average of the x-coordinates of the two points and the offset from the image center
End

Fig. 3. Lateral deviation calculation flow chart

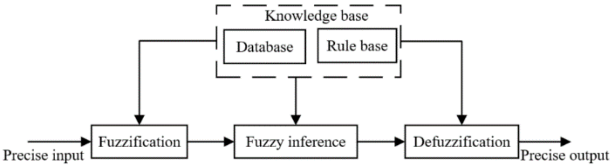

Fig. 4. Structure of the fuzzy controller

The fuzzy controller in this study uses lateral deviation (processed from YOLOv8 target detection) and heading angle (measured by the IMU) as input variables. The heading angle ranges from [-20°, 20°], and lateral deviation ranges from [-250, 250] pixels. Both variables are divided into 7 levels, with fuzzy set elements {LB, LM, LS, ZE, RS, RM, RB}. The membership functions for the heading angle and lateral deviation are shown in Fig. 5 and Fig. 6.

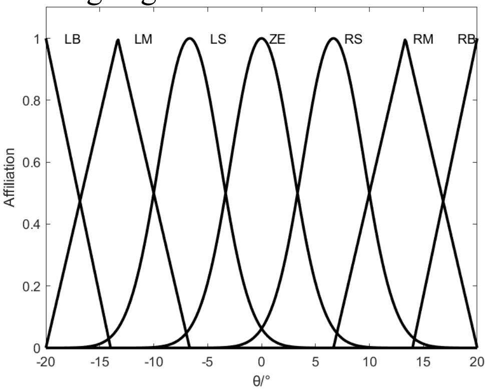

Fig. 5. Heading angle membership function

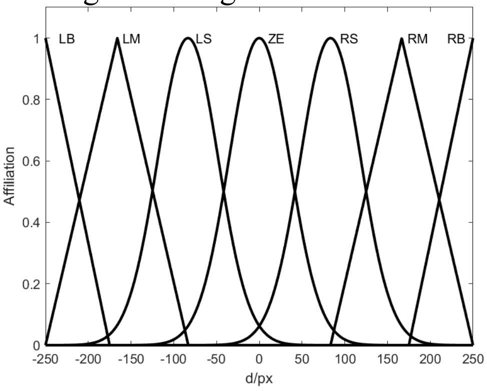

Fig. 6. Lateral deviation membership function

The fuzzy control system's output is the rotational speed difference between the vehicle's right and left motors, ranging from [-2000 rpm, 2000 rpm]. The fuzzy set has 7 elements: {LB, LM, LS, ZE, RS, RM, RB}, representing left turns, no adjustment, and right turns with decreasing degrees of turn. The membership function of the speed difference is shown in Fig. 7.

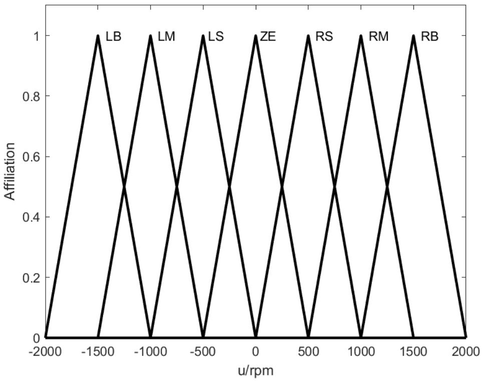

Fig. 7. Output speed difference membership function

The fuzzy control algorithm uses fuzzy control rules to compute the output fuzzy sets. Since both the heading angle and lateral deviation fuzzy sets contain seven fuzzy quantities, there are 49 fuzzy control rules in total. The fuzzy control rules designed in this study are shown in Table 2.

Table 2

Fuzzy control rule table

u		θ						
		LB	LM	LS	ZE	RS	RM	RB
d	LB	LB	LB	LB	LB	LB	LB	LB
	LM	LB	LM	LM	LM	LM	LM	LB
	LS	LB	LM	LS	LS	LS	LM	LB
	ZE	RB	RM	RS	ZE	LS	LM	LB
	RS	RB	RM	RS	RS	RS	RM	RB
	RM	RB	RM	RM	RM	RM	RM	RB
	RB	RB	RB	RB	RB	RB	RB	RB

The centroid defuzzification method is used to convert fuzzy inference results into control values. The desired rotational speeds for the left and right motors are determined, and these values are input to the motor speed controllers to achieve timely adjustments to the platform vehicle's motion state.

3.4 Motion blur improvement algorithm based on optical flow

To improve motion blur and enhance the resilience of the navigation control algorithm, this study proposes an algorithm based on the pyramid LK optical flow method. This approach helps mitigate the impact of fast movement or bumps on uneven road surfaces, which can affect target detection.

The LK optical flow method tracks a small number of feature points to represent the overall motion of the camera. These feature points are computed using the Shi-Tomasi corner detection algorithm. The basic idea of the LK optical flow method is based on the following three assumptions:

1. **Brightness constancy**: The pixel appearance of a target in the scene does not change during frame-to-frame motion. For grayscale images, this means the pixel intensity remains constant during tracking.

2. **Temporal continuity**: The camera's motion on the image plane changes gradually over time, meaning pixel positions do not change drastically, allowing the pixel intensity to correspond to the partial derivatives of position.

3. **Spatial consistency**: Neighboring points on the same surface in the scene exhibit similar motion, and their projections on the image plane are close to each other. Based on the first two assumptions, the image constraint equation is obtained as shown in Equation (10), where I(x, y, t) represents the brightness of the image at position (x, y) at time t.

$$I(x,y,t)=I(x+\delta x,y+\delta y,t+\delta t) \tag{10}$$

Using the Taylor series expansion of the function $I(x+\delta x,y+\delta y,t+\delta t)$ at (x, y) and combining it with Equation (10), we derive Equation (11). Here, u and v are the velocity components in the x and y directions, respectively.

$$I_x u+I_y v+I_t=0 \tag{11}$$

Based on the third assumption, the optical flow within a window of size $n = m^2$ is assumed to be constant. This leads to the matrix form of the equation, as shown in Equation (12):

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tn} \end{bmatrix} \tag{12}$$

Using the least squares method, the final optical flow matrix can be solved, as shown in Equation (13). Here, u and v are the velocities of matching feature points in the x and y directions, respectively [19-21].

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} I_{xi}^2 & \sum_{i=1}^{n} I_{xi} I_{yi} \\ \sum_{i=1}^{n} I_{xi} I_{yi} & \sum_{i=1}^{n} I_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{i=1}^{n} I_{xi} I_{ti} \\ -\sum_{i=1}^{n} I_{yi} I_{ti} \end{bmatrix} \tag{13}$$

To address errors from large object motion, the pyramid LK optical flow algorithm constructs a three-level pyramid starting with low resolution. It refines optical flow vectors at each level, iteratively propagating the results down to the original resolution to accurately estimate large displacements [22,23].

Using the optical flow vectors, pairs of matched feature points between consecutive frames can be obtained. The average displacements in the x and y directions (avg_dx and avg_dy) are calculated to estimate the overall motion direction and magnitude. The standard form of the affine transformation matrix is a 2x3 matrix, as shown in Equation (14). Where, a,b,c,d control rotation and scaling, and t_x , t_y control translation. Based on the offsets in the x and y directions, the affine transformation matrix is created as shown in Equation (15). a and d are set to 1 to indicate no scaling, b and c are set to 0 to indicate no rotation, and $t_x = avg\_dx$ and $t_y = avg\_dy$ represent the offsets in the x and y directions, respectively.

$$\begin{bmatrix} a & b & t_x \\ c & d & t_y \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} 1 & 0 & avg\_dx \\ 0 & 1 & avg\_dy \end{bmatrix} \tag{15}$$

After applying affine transformation, pixel values at the image boundaries are copied to uncovered areas. If a pixel exceeds the original image boundaries, the excess area is filled with the nearest valid pixel value, ensuring no blank areas or abrupt edges in the transformed image.

4. Experimental analysis and discussion

4.1 Training and deployment of YOLOv8 target detection model

The dataset used in the study includes 556 images with 4080 annotated root labels, collected under various lighting conditions in real crop environments. It was split into a training set (445 images, 3277 labels) and a validation set (111 images, 803 labels). Training was conducted on a system with Ubuntu 18.04, Intel Xeon Platinum 8362 CPU, RTX 3090 GPU, Python 3.10, PyTorch 2.10, and CUDA 12.1. A pre-trained YOLOv8 nano model was used for 200 iterations, with an input size of 640×640, batch size of 8, and learning rates of 0.01. The training set optimized parameters, and the validation set assessed accuracy.

The accuracy of the trained model is typically evaluated using **precision**, which is calculated using Equation (16). In this formula, the numerator represents the number of correctly predicted positive samples, and the denominator is calculated by adding true positives and false positives.

$$\text{precison} = \frac{\text{NP}}{\text{TP+FP}} \tag{16}$$

Recall, shown in Equation (17), measures the model's ability to identify positive samples. A higher recall implies fewer false negatives (FN) and a lower miss rate, indicating the model's capability to more accurately recognize actual positive samples.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

The **mAP** is an aggregate metric that summarizes the AP across all categories. It evaluates the overall predictive performance of the model. The mAP is calculated using Equation (18).

$$\text{mAP=}\frac{1}{N}\sum_{\text{i=1}}^{N} AP_i \tag{18}$$

The performance metrics of the trained model are shown in Table 3. The experimental results demonstrate that the model achieves 83.9% precision, 84.1% recall, and 88.1% mAP, indicating strong detection performance. The model is lightweight, with 3,005,843 parameters, a computational complexity of 8.1 GFLOPS, and a compact weight size of 6.1 MB, making it efficient and suitable for deployment in resource-constrained environments.

Table 3

Experimental results

Metric Name	Result
Precision/%	83.9
Recall/%	84.1
mAP/%	88.1
Parameters	3005843
GFLOPS	8.1
Weights	6.1MB

The .pt model was converted to ONNX and deployed on an edge platform. It was further converted into TensorRT engine format and loaded to create an inference engine on an Nvidia Jetson Xavier NX platform. The average inference speed for a single image was 25ms, enabling real-time processing suitable for applications like autonomous navigation and crop monitoring. This performance ensures timely decision-making and efficient operation in smart agriculture. The inference results are given in Fig. 8.

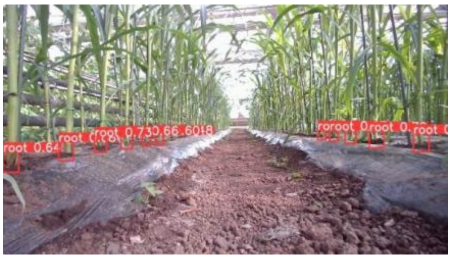

Fig. 8. Inference results

4.2 Field visual navigation experiment

The proposed visual navigation algorithm was tested in a maize planting environment at the Harbin Agricultural Science Academy. The average ridge width was 80 cm, furrow width 65 cm, and flat width within the furrows 50 cm. The autonomous driving of the platform vehicle within the furrows is given in Fig. 9.

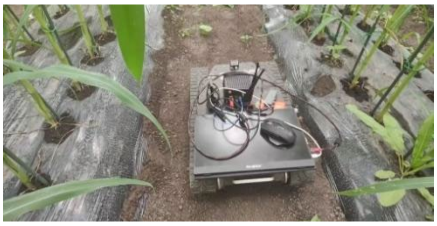

Fig. 9. Platform vehicle operation diagram

During testing, the platform vehicle's navigation path was defined as the straight line at the center of the furrow's flat ground. Heading angle and lateral deviation were used to measure accuracy, with true values obtained manually. The lateral deviation is measured in pixels (px) and represents the lateral displacement of the vehicle in the navigation path relative to the target centerline. A pixel is the basic unit in an image that describes the offset of a vehicle in a visual navigation image.The vehicle continuously adjusted its position and posture, maintaining stable travel in the crop row's center using the visual navigation algorithm. As shown in Table 4, during visual navigation, the relative errors for both the heading angle and lateral deviation were less than 10%.

Table 4

Navigation path recognition results in field environment

Parameter	Index	True Value	Measured Value	Absolute Error	Relative Error
Heading Angle	1	15.0°	16.5°	1.5°	10.0%
	2	-13.0°	-11.9°	1.1°	8.5%
	3	-15.0°	-13.8°	1.2°	8.0%
	4	13.0°	12.1°	-0.9°	6.9%
Lateral Deviation	1	199.6px	218.5px	18.9px	9.5%
	2	-51.1px	-55px	-3.9px	7.6%
	3	46.9px	50px	3.1px	6.6%
	4	180.7px	196.2px	15.5px	8.6%

The real-time target detection results from the front-facing camera during autonomous navigation are illustrated in Fig. 10. The calculated lateral deviation and heading angle are displayed in the upper left corner of the image.

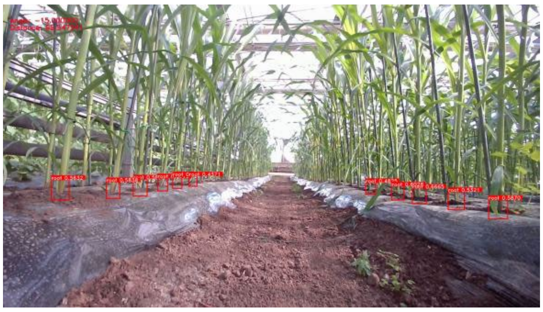

Fig. 10. Real-time visual navigation results

Fig. 11 and Fig. 12 show the tracking performance of the heading angle and lateral deviation during testing. The navigation control system reduced both values, with the maximum heading angle at 14.3°and the maximum lateral deviation at 80 pixels.

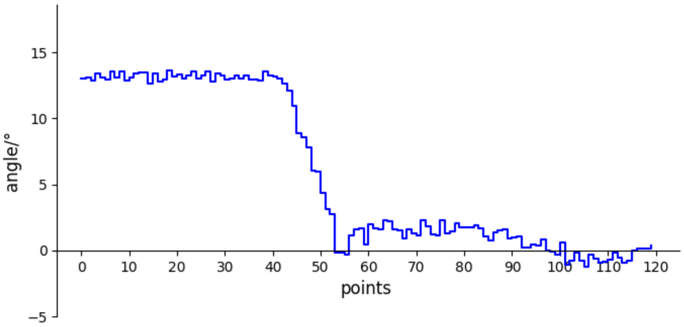

Fig. 11. Heading angle adjustment results

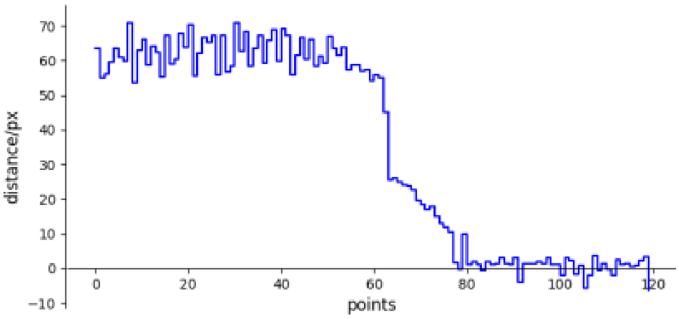

Fig. 12. Lateral deviation adjustment results

4.3 Motion blur improvement experiment using optical flow

Fig. 13 illustrates the process of improving motion blur caused by camera jolts using information from the previous clear image. The pyramid optical flow method was used to estimate that the blurred image moved 0.56 pixels in the positive x-direction and 0.055 pixels in the negative y-direction relative to the previous clear image. In Fig. 13(c), red points depict the detected feature points in the previous frame using the corner detection method, while green points connected to them represent the corresponding matched points in the blurred image.

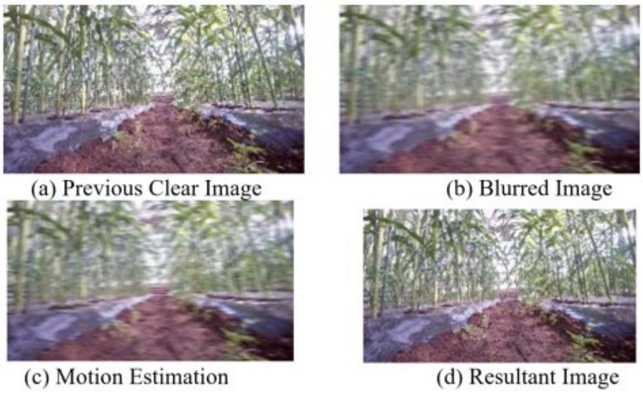
(a) Previous Clear Image
(b) Blurred Image
(c) Motion Estimation
(d) Resultant Image

Fig. 13. Process of improving motion blur

5. Conclusion

This study developed an automatic under-canopy data collection device and a vision-based navigation control algorithm to address the underutilization of data in open agricultural environments. The YOLOv8 model, optimized with TensorRT, was deployed on the platform's main computer, achieving a 25 ms inference speed.

Crop root target information and IMU-based heading angle were used by the fuzzy control algorithm to adjust motor speeds, ensuring stable navigation and row-switching. A motion blur improvement method using optical flow was proposed to enhance detection accuracy. Future work will focus on enhancing the algorithm's robustness and integrating high-precision sensors for better environmental perception.

REFERENCES

[1] *Velásquez D, Sánchez A, Sarmiento S, et al.* A cyber-physical data collection system integrating remote sensing and wireless sensor networks for coffee leaf rust diagnosis. Sensors, Vol. **21**, Iss. 16, pp. 5474, 2021.

[2] *Young S N, Kayacan E, Peschel J M.* Design and field evaluation of a ground robot for high-throughput phenotyping of energy sorghum. Precision Agriculture, Vol. **20**, Iss. 4, pp. 697-722, 2019.

[3] *Zhang H, Wang B, Tang Z, et al.* A rapid field crop data collection method for complexity cropping patterns using UAV and YOLOv3. Frontiers of Earth Science, pp. 1-14, 2024.

[4] *Yang H.* Visual navigation method for agricultural mobile robots based on spatial continuity clustering algorithm. International Journal on Interactive Design and Manufacturing (IJIDeM), pp. 1-13, 2024.

[5] *Higuti V A H, Velasquez A E B, Magalhaes D V, et al.* Under canopy light detection and ranging-based autonomous navigation. Journal of Field Robotics, Vol. **36**, Iss. 3, pp. 547–567, 2019.

[6] *Dos Reis D H, Welfer D, De Souza Leite Cuadros M A, et al.* Mobile robot navigation using an object recognition software with RGBD images and the YOLO algorithm. Applied Artificial Intelligence, Vol. **33**, Iss. 14, pp. 1290-1305, 2019.

[7] *Yu J, Zhang J, Shu A, et al.* Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction. Computers and Electronics in Agriculture, Vol. **209**, pp. 107811, 2023.

[8] *Shaikh I M, Akhtar M N, Aabid A, et al.* Enhancing sustainability in the production of palm oil: Creative monitoring methods using YOLOv7 and YOLOv8 for effective plantation management. Biotechnology Reports, Vol. **44**, pp. e00853, 2024.

[9] *Kumar S, Singh S K, Varshney S, et al.* Fusion of deep sort and Yolov5 for effective vehicle detection and tracking scheme in real-time traffic management sustainable system. Sustainability, Vol. **15**, Iss. 24, pp. 16869, 2023.

[10] *Nnadozie E C, Casaseca-de-la-Higuera P, Iloanusi O, et al.* Simplifying YOLOv5 for deployment in a real crop monitoring setting. Multimedia Tools and Applications, Vol. **83**, Iss. 17, pp. 50197-50223, 2024.

[11] *Jiang K, Pan S, Yang L, et al.* Surgical instrument recognition based on improved YOLOv5. Applied Sciences, Vol. **13**, Iss. 21, pp. 11709, 2023.

[12] *Al-Hasan T M, Bonnefille V, Bensaali F.* Enhanced YOLOv8-based system for automatic number plate recognition. Technologies, Vol. **12**, Iss. 9, pp. 164, 2024.

[13] *Song P, Zhao L, Li H, et al.* RSE-YOLOv8: An algorithm for underwater biological target detection. Sensors, Vol. **24**, Iss. 18, pp. 6030, 2024.

[14] *Duan P, Liang X.* An improved YOLOv8-based foreign detection algorithm for transmission lines. Sensors, Vol. **24**, Iss. 19, pp. 6468, 2024.

[15] *Li Z, Wang D, Yan Q, et al.* Winter wheat weed detection based on deep learning models. Computers and Electronics in Agriculture, Vol. **227**, pp. 109448, 2024.

[16] *Liu J, Wang C, Qiu S, et al*. Variable universe fuzzy control of once-through steam generator feedwater. Annals of Nuclear Energy, Vol. **207**, pp. 110699, 2024.

[17] *Dineshkumar C, Jeong J H, Joo Y H*. Observer-based fuzzy control for fractional order PMSG wind turbine systems with adaptive quantized-mechanism. Communications in Nonlinear Science and Numerical Simulation, Vol. **136**, pp. 108087, 2024.

[18] *Miron A, Cziker A C, Beleiu H G*. Fuzzy control systems for power quality improvement—A systematic review exploring their efficacy and efficiency. Applied Sciences, Vol. **14**, Iss. 11, pp. 4468, 2024.

[19] *Wang Y, Wang W, Li Y, et al*. Research on traditional and deep learning strategies based on optical flow estimation-A review. Journal of King Saud University-Computer and Information Sciences, pp. 102029, 2024.

[20] Wang J, Huang Z, Xu Y, et al. Gas–liquid two-phase flow measurement based on optical flow method with machine learning optimization model. Applied Sciences, Vol. **14**, Iss. 9, pp. 3717, 2024.

[21] *Guo Z, Guillen D P, Grimm J R, et al*. High throughput automated characterization of enamel microstructure using synchrotron tomography and optical flow imaging. Acta Biomaterialia, 2024.

[22] *Suni S S, Gopakumar K*. Fusing pyramid histogram of gradients and optical flow for hand gesture recognition. International Journal of Computational Vision and Robotics, Vol. **10**, Iss. 5, pp. 449-464, 2020.

[23] *Tehrani A K Z, Rivaz H*. MPWC-Net++: Evolution of optical flow pyramidal convolutional neural network for ultrasound elastography. Medical Imaging 2021: Ultrasonic Imaging and Tomography. SPIE, Vol. **11602**, pp. 14-23, 2021.