ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Modeling and Simulation Analysis of Visual Servo Control System for Mechatronic Picking Robotic Arm

Liuhang Shen[1,*] *and Xiangwen Sun*[1]

[1] Ulster college at Shaanxi University of Science &Technology, Xi'an,710021, Shaanxi, China

SUMMARY: *In this study, a set of visual servo control system of mechatronic picking robotic arm based on machine vision is designed. By establishing the kinematics and dynamics model of the robotic arm and using the D-H parametric method for system modeling, it is verified that the model can accurately describe the motion characteristics of the robotic arm. The target recognition algorithm based on deep learning improves the YOLOv3 network structure, which significantly improves the recognition and localization ability in the complex orchard environment. The improved fuzzy neural network sliding mode control algorithm is developed for the nonlinear and strongly coupled characteristics of the robotic arm, and the experiments show that it is better than the traditional PID and SMC algorithms in terms of dynamic response, steady state accuracy and anti-interference ability. The proposed improved stochastic fast search tree algorithm combined with visual servoing dramatically improves the path planning efficiency through techniques such as super ellipsoidal gravity bias sampling. Experimental data show that the algorithm reduces the number of sampling points by 92.9%, the planning time by 86.1%, and the path cost by 35.2%. In the actual test, the success rate of the robotic arm picking integrated with the algorithm reaches 92.9%, which is 16.6 and 7.2 percentage points higher than that of the PID system and the SMC system, respectively. The research results provide effective technical support for orchard picking automation and promote the practical development of picking robots.*

KEYWORDS: *mechatronics; picking robot arm; visual servo control; deep learning; fuzzy neural network sliding mode control; path planning*

1 Introduction

In the context of global labor shortage, aging and rapid development, the traditional production model has high labor costs and low operational efficiency, which is difficult to meet the development needs of modern industry. Meanwhile, with the development of industrial robots, the mechanical arm, as the core of the fusion of mechanical engineering, sensors and artificial intelligence technology, has become the key to improve the level of automated production [1, 2]. Due to the continuous upgrading of production needs, mechatronic picking robotic arms came into being, realizing 24-hour efficient operation by bringing together technologies from multiple fields such as mechanics, electronics, vision, and control, resulting in improved picking efficiency and lower probability of damage [3].

The traditional industrial robotic arm system, generally using demonstration teaching or offline programming of the processing task for path planning and motion programming, the

*18633158550@163.com
https://doi.org/10.65102/is20261038 Published: 30 April, 2026

processing process is simply a repetition of the pre-programmed design of the action, when the processing object or the work environment changes, the need to re-design the program [4, 5]. While using the vision system, the information of the processing object can be detected in real time, and the theory and method of artificial intelligence can be applied to process this information, so as to improve the intelligence level of the robot. With the help of robotic arm vision servo system does not need to program the movement of the robotic arm in advance, which can save a lot of programming time, improve production efficiency, and further improve the processing quality of products [6, 7]. The robotic arm visual servo system has an important role in the field of robotics. As the eyes of the robotic arm, the visual servo system utilizes the components such as sensory sensors, image processing units, and controllers to identify the picking targets and analyze the environment, to understand the maturity of fruits, vegetables, and other agricultural products, to parse the target blocking and plan the picking path, and to promote accurate picking of the robotic arm [8]. Since the visual servo control system of the robotic arm involves the knowledge content of many fields, it is of great significance to improve the motion performance of the robotic arm, optimize the control strategy, reduce the testing cost, and achieve high-precision operation by modeling and simulation research on the visual servo control system of the robotic arm [9, 10].

The classification of robotic arm visual servo control system is based on a variety of characteristics and principles, position-based visual servo (PBVS), image-based visual servo (IBVS), and hybrid visual servo (HBVS).The PBVS uses the target position information as the basis of control. It adjusts the robotic arm motion by recognizing the target position deviation. Literature [11] developed a cerebellar-inspired PBVS control system with a dual-gradient neural dynamic model to realize robotic arm PBVS control under unknown kinematic model and obtain higher tracking accuracy. This type of robotic arm PBVS control system focuses on the target motion velocity information by establishing the connection between image space and joint space, and dynamically adjusts the robotic arm motion velocity according to the target velocity, but this type of system relies on the camera to calibrate the image and the joint association, which is susceptible to the environment [12, 13]. Literature [14] designed IBVS was used in the task of four-degree-of-freedom robotic arm, using multi-threshold algorithms to recognize, classify, and extract image features to control the robotic arm motion, which is low-cost and easy to set up, and will not be affected by the camera. However, the depth information of IBVS is difficult to estimate, and it is widely used in simpler scenarios, and its control accuracy and stability are constrained by image quality due to the dominant role of image processing [15, 16]. Literature [17] constructed a robotic arm HBVS control system based on six degrees of freedom and hand-eye configuration, which is controlled from three angles, i.e., IBVS control for horizontal rotation and vertical swing, and PBVS control for depth contraction, and its accuracy and stability are improved.

Currently, research on picking robotic arms based on visual servo control system (VSCS) has been directed towards lightweighting, safety performance and higher picking efficiency at lower cost. Literature [18] designed a lightweight force-sensing tomato picking robotic arm based on “global-local” VSCS from the lightweight perspective, in which the force-sensing system can reduce the probability of accidental collision among workers, plants, and robotic arm, while the visual servo technology improves the positioning accuracy of the end of the robotic arm, and obtains 93% of the safe and effective picking rate. Literature [19] constructed a hand-eye coordination based HBVS for cherry tomato picking, where a cutting and trimming integration mechanism was embedded in the HBVS controller to guide the end of the robotic arm for the picking task, with a picking speed and success of up to 9.4 s/pick and 96.25%, respectively. Literature [20] proposed a vision algorithm acting on an orchard picking robotic arm VSCS, which can realize real-time localization, fast movement, and

seamless dynamic picking, and improve the picking accuracy, efficiency, and applicability. Literature [21] utilized VSCS based on computer vision technology to achieve end-of-arm operation of cucumber picking visual servo control at 16-23 frames/sec, achieving 100% outcome rate.

Synthesizing the above research results, this paper proposes a series of innovative solutions to the key technical problems faced by picking robotic arms in complex orchard environments. Through systematic integration and improvement of existing technologies, a unique technical system is formed. In the target recognition and localization algorithm, the YOLOv3 network structure is optimized and the attention mechanism is introduced, and this improved deep learning algorithm significantly improves the recognition performance in complex environments. In servo control algorithm, an adaptive fuzzy neural network sliding mode control algorithm is innovatively proposed, which solves the nonlinear, time-varying and uncertainty problems that are difficult to be dealt with by traditional control methods through a three-layer innovative design. As for the path planning algorithm, the VS-IRRT algorithm proposed in this study realizes innovative breakthroughs in four dimensions, and these innovations include the design of an adaptive sampling strategy based on the super-ellipsoidal gravitational bias, which significantly improves the sampling efficiency by dynamically adjusting the shape and size of the sampling region. Based on the visual servo control system of the mechatronic picking robotic arm designed in this paper, it can provide new research ideas for the innovative design of agricultural picking robotic arms.

2 Kinematics and dynamics modeling of mechatronic picking robotic arm

2.1 Kinematic modeling

Kinematic modeling of the picking robotic arm is the basis for realizing high-precision control, and it can effectively guide the subsequent control system design by accurately describing the spatial position relationship and motion characteristics of each joint of the robotic arm. In this study, a 5-degree-of-freedom picking robotic arm is designed, including three rotary joints and two moving joints, to meet the demand for flexibility in orchard operation. The coordinate system is established using the D-H parameter method, which describes the spatial relationship between neighboring connecting rods by four parameters $(a_i, \alpha_i, d_i, \theta_i)$. According to the structural characteristics of the robotic arm, the D-H parameters of each joint are determined as shown in Table 1, with $a_2 = 350mm, a_3 = 320mm,$ d_1, d_4, d_5 as the variable and $\theta_1 \sim \theta_5$ as the joint variable.

Table 1: Parameter table of D-H for Picking Manipulator

Connecting rod	$a_i(mm)$	$\alpha_i(rad)$	$d_i(mm)$	$\theta_i(rad)$
1	0	$\pi/2$	d_1	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3
4	0	$\pi/2$	d_4	θ_4
5	0	0	d_5	θ_5

Positive kinematics describes the end-effector position attitude for a given joint variable, and the matrix of chi-square transformations between neighboring links is:

$$A_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\cos\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & \cos\theta_i\cos\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The end-effector position matrix with respect to the base is obtained by concatenated multiplication of $T = A_1 A_2 \cdots A_n$. For a 5-degree-of-freedom picking robot arm, the position matrix is:

$${}^0T_5 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \tag{2}$$

Expanded and indicated as:

$${}^0T_5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Inverse kinematics calculations are critical for picking accuracy and are used to solve for the joint variables required to achieve a specific end-effector position attitude. This study combines geometric and algebraic methods to solve the inverse kinematics equations. The joint 1 angle can be calculated from the end-effector position in the base coordinate system, i.e:

$$\theta_1 = atan2(p_y, p_x) \tag{4}$$

Joints 2 and 3 are solved by plane geometry, i.e:

$$\begin{cases} r = \sqrt{p_x^2 + p_y^2} - d_5\cos\theta_5 \\ s = p_z - d_1 - d_4 - d_5\sin\theta_5 \\ D = \dfrac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \\ \theta_3 = atan2(\sqrt{1-D^2}, D) \\ \theta_2 = atan2(s, r) - atan2(a_3\sin\theta_3, a_2 + a_3\cos\theta_3) \end{cases} \tag{5}$$

Joints 4 and 5 are then solved in conjunction with the end-effector attitude matrix, i.e:

$$\begin{cases} \theta_4 = atan2(a_z, \sqrt{a_x^2 + a_y^2}) \\ \theta_5 = atan2(a_y/\sin\theta_1 - a_x/\cos\theta_1, o_y/\sin\theta_1 - o_x/cos\theta_1) \end{cases} \tag{6}$$

To verify the accuracy of the model, forward and inverse kinematics calculations and simulations were implemented in Matlab. Figures 1 and 2 show the simulation results of end-effector position trajectory and workspace analysis under different joint angle configurations, respectively. The simulation results show that the proposed model can accurately describe the motion characteristics of the picking robotic arm, and the error of the forward and inverse kinematics calculation is less than 0.001 rad, which meets the requirements of high-precision picking operation. The optimal picking area is determined through workspace analysis, which is important for optimizing the arrangement of the robotic arm and improving the picking efficiency. The kinematic modeling and simulation not only verifies the model accuracy, but also lays the foundation for the subsequent dynamics modeling, control system design and path planning.

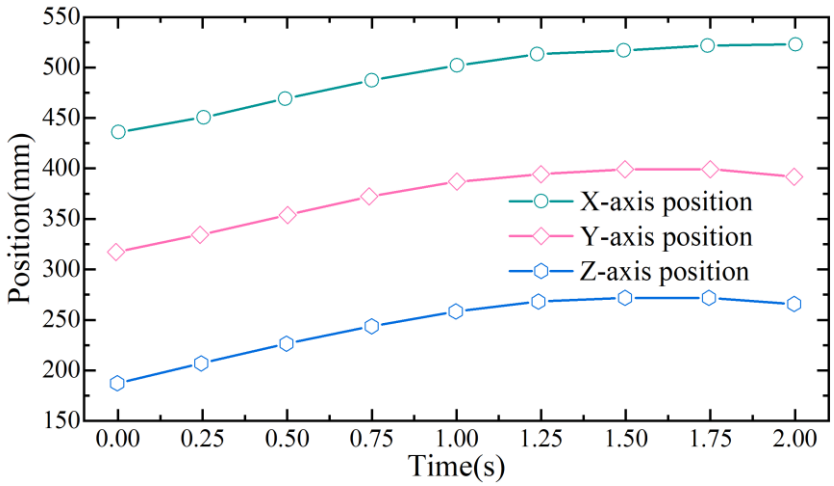

Figure 1: Simulation results of position trajectory

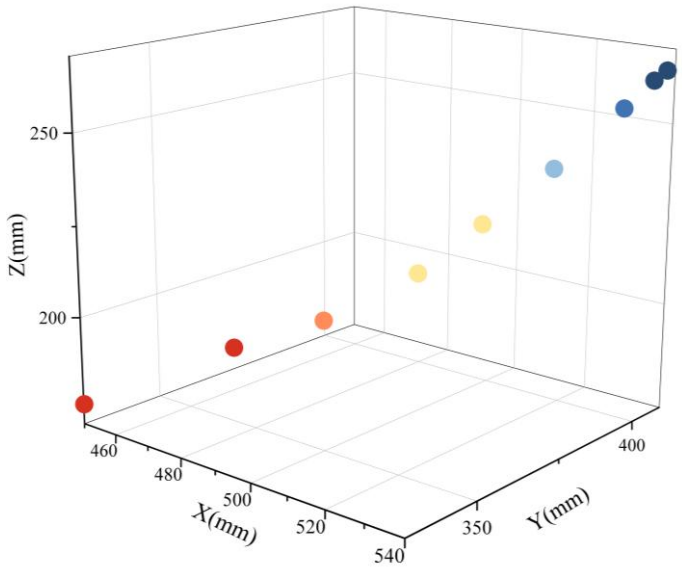

Figure 2: Analysis of the Working Space of the Picking Manipulator

2.2 Kinetic modeling

As the core foundation of the control system design, the kinetic modeling of the picking robotic arm reveals the relationship between the moments of each joint of the robotic arm and

the motion state. Unlike the kinematic model which only describes the geometric and kinematic relationship, the kinetic model takes into account the factors of mass distribution, inertial force, Kurtosis force, centrifugal force, and gravity, etc., and describes the actual motion characteristics of the robotic arm more accurately. In this paper, the Lagrange equation method is used to establish the kinetic model of the picking robotic arm, and this method based on energy analysis is characterized by a unified form and clear physical meaning. The general form of Lagrange equation is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial q_i}\right)-\frac{\partial L}{\partial q_i}=\tau_i \tag{7}$$

The Lagrangian function is expressed as:

$$L=K-P \tag{8}$$

The generalized coordinates of the 5-degree-of-freedom picking robot arm can be expressed as:

$$q=[q_1,q_2,q_3,q_4,q_5]^T=[\theta_1,\theta_2,\theta_3,\theta_4,\theta_5]^T \tag{9}$$

The kinetic energy calculation of the system includes the translational kinetic energy and rotational kinetic energy of each linkage as:

$$K=\sum_{i=1}^{5}\left(\frac{1}{2}m_i v_i^T v_i+\frac{1}{2}w_i^T I_i w_i\right) \tag{10}$$

The potential energy of the system comes mainly from gravity, i.e.:

$$P=\sum_{i=1}^{5}m_i g^T r_i \tag{11}$$

The derivation of Lagrange's equation yields the dynamics equation of the picking robot arm, i.e.:

$$M(q)\ddot{q}=C(q,\dot{q})\dot{q}+G(q)=\tau \tag{12}$$

The inertia matrix $M(q)$ element M_{ij} is expressed by the formula:

$$M_{ij}=\sum_{k=\max(i,j)}^{5}tr\left(\frac{\partial T_k}{\partial q_i}J_k\frac{\partial T_k^T}{\partial q_j}\right) \tag{13}$$

The Koch and centrifugal force matrix $C(q,\dot{q})$ elements C_{ij} are computed by means of the Christoffel symbols, viz:

$$C_{ij}=\sum_{k=1}^{5}\Gamma_{ijk}q_k \tag{14}$$

The Christoffel symbol Γ_{ijk} is defined as:

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right) \tag{15}$$

The gravity matrix $G(q)$ element G_i is computed by the partial derivatives of the potential energy with respect to the generalized coordinates, viz:

$$G_i = \frac{\partial P}{\partial q_i} \tag{16}$$

Based on the above theory, a dynamic parameter list of the picking robotic arm was established as shown in Table 2, listing the mass of each linkage, the center of mass position and the inertia tensor spindle values, which constitute the basic data for dynamic modeling.

Table 2: Dynamic parameter table of the picking manipulator

Connecting rod	Mass (kg)	Centroid position (mm)			Principal axis value of the inertia tensor (kg·m²)		
		x	y	z	Ixx	Iyy	Izz
1	3.5	0	0	100	0.045	0.045	0.018
2	4.2	175	0	0	0.012	0.125	0.125
3	3.8	160	0	0	0.010	0.098	0.098
4	2.5	0	0	75	0.008	0.008	0.004
5	1.8	0	0	50	0.003	0.003	0.002

In order to verify the accuracy of the kinetic modeling I conducted simulation experiments using MATLAB by defining the joint trajectory from the initial position to the target position and then using inverse dynamics to calculate the joint moments required to achieve the trajectory, and at the same time verifying the accuracy of the kinetic equations and comparing the error between the directly calculated moments and those calculated through the kinetic equations. The simulation results of the kinetic modeling are shown in Fig. 3.

The joint 2 and joint 3 moments are significantly larger than the other joints because they carry most of the load and gravity compensation. Joint 1 moments are mainly from rotational inertia and Koch force, and joint 4 and joint 5 moments are relatively small, which are related to their lighter loads. All joint moments increased rapidly during the initial phase of the trajectory, peaked during the middle phase, and then decreased gradually during the end phase of the trajectory, consistent with a typical acceleration-uniform-deceleration motion pattern.

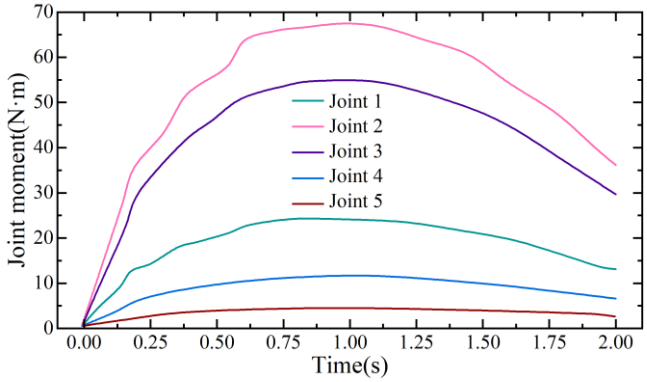

Figure 3: Dynamic simulation results of the picking robotic arm

2.3 Simulation Experiments

The simulation experiments of the picking robotic arm are conducted in this section to verify the accuracy and validity of the models. The simulation experiment mainly includes three aspects, i.e., trajectory tracking experiment, workspace analysis and dynamic response analysis. We designed a smooth trajectory from the initial position to the target position in the experiment, and used the fifth degree polynomial interpolation method to generate the joint angle trajectory to ensure that the velocity and acceleration are continuous. The results of trajectory tracking are shown in Fig. 4. Table 3 demonstrates the trajectory tracking errors under different control parameters. Appropriately increasing the proportional gain and differential gain can effectively reduce the tracking error, but too large a gain will cause the system to respond too quickly and generate oscillations.

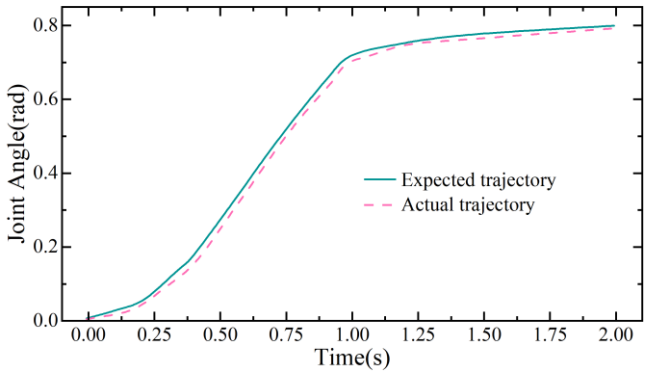

Figure 4: The tracking result of the Angle trajectory of joint 1

Table 3: Trajectory tracking errors under different control parameters

Parameter	Average error (mm)	Max error (mm)	Stabilization time (s)	Overshoot (%)
$K_p = 80, K_d = 15$	3.42	7.85	1.25	5.3
$K_p = 100, K_d = 20$	2.18	5.64	1.08	3.7
$K_p = 120, K_d = 25$	1.53	4.21	0.95	2.5
$K_p = 150, K_d = 30$	1.12	3.45	0.87	1.8
$K_p = 200, K_d = 40$	0.86	2.73	0.76	1.2
$K_p = 250, K_d = 50$	0.74	2.35	0.68	0.9

In order to evaluate the dynamic performance of the picking robotic arm, we conducted step response and frequency response experiments. The step response experiment observes the response characteristics of the system by applying a step moment input to joint 1, while the frequency response experiment analyzes the frequency characteristics of the system by applying sinusoidal moment inputs at different frequencies, and the results of the experiments are shown in Fig. 5.

The dynamic response analysis results show that there are obvious differences in the response characteristics of the system under different control parameters, and appropriately increase the differential gain can reduce the amount of overshooting, but it will lengthen the regulation time, by analyzing the response characteristics of the different parameters, it can provide a reference for the design of the control system, and select the appropriate control parameters to meet the performance requirements of the system. The kinematics and dynamics of the picking robotic arm are comprehensively evaluated through the above simulation experiments, and the experimental results show that the kinematics model based on the D-H parameter method can accurately describe the motion characteristics of the picking robotic arm, and the forward and inverse kinematics computational error is less than 0.001 rad to meet the requirements of high-precision picking operations. The dynamics model based on the Lagrangian method can accurately describe the dynamic characteristics of the picking robotic arm, and the validation error of the dynamics equation is less than 0.5N-m, which provides a reliable theoretical basis for the control system design. The results of the workspace analysis of the picking robotic arm show that the robotic arm has high operating performance in the range of about 450-550mm in front and ±400mm in side and 150-300mm in height, which is an important guiding significance for the arrangement of the picking robotic arm and the operation planning. The dynamic response analysis results show that the dynamic performance of the picking robotic arm is significantly affected by the control parameters, and the appropriate choice of control parameters can meet the system's response speed and stability requirements. The comparison results of different control strategies show that the control method of PD controller combined with feed-forward compensation can effectively improve the tracking accuracy of the system, but there are still limitations in dealing with nonlinearity and uncertainty that need to be further improved.

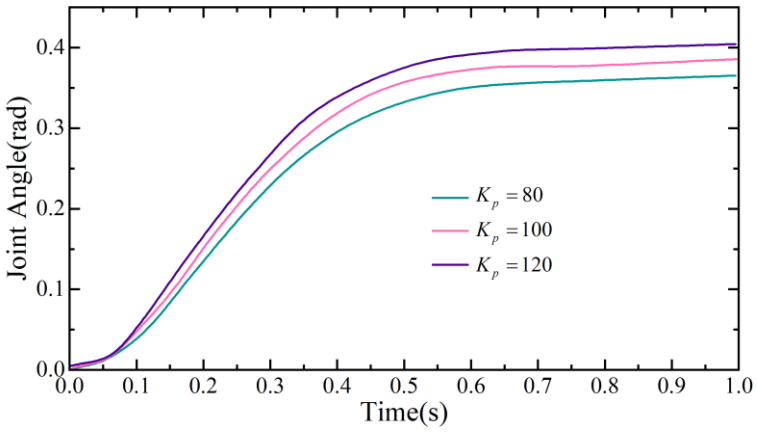

Figure 5: Dynamic response analysis results of the picking robotic arm

3 Target recognition and localization system design and implementation

3.1 Deep learning based target recognition algorithm design

Target recognition and localization of the picking robotic arm is the core link of the whole system, which directly affects the picking success rate and efficiency. Traditional target recognition methods mainly rely on features such as color, shape, and texture, which are difficult to achieve ideal results in complex orchard environments. The standard YOLOv3 has limitations such as insufficient detection ability for small targets, sensitivity to light changes, background interference problems, and insufficient feature extraction in complex orchard environments, and to solve these problems, this study has taken four improvement measures:

First, the SPP module is added after the Darknet-53 backbone network to expand the sensory field through the maximum pooling operation at four scales of 1*1, 5*5, 9*9, and 13*13 to enhance the network's adaptability to targets at different scales.

Secondly, the channel attention mechanism is used to weight the fusion of different layers of features, the feature fusion formula is:

$$F_{fusion} = \sum_{i=1}^{n} \alpha_i \cdot F_i \tag{17}$$

where α_i is the adaptive weighting coefficient and F_i is the feature map of the i th layer.

Thirdly, the hybrid loss function is designed, i.e:

$$L = \lambda_{coord} \cdot L_{box} + \lambda_{obj} \cdot L_{obj} + \lambda_{cls} \cdot L_{cls} + \lambda_{iou} \cdot L_{iou} \tag{18}$$

where L_{box} is the bounding box regression loss, L_{obj} is the target confidence loss, L_{cls} is the categorization loss, L_{iou} is the IoU loss, and λ is the weighting coefficients of the various loss terms.

Fourthly, for the characteristics of the orchard environment, data enhancement methods including random scaling, rotation, light adjustment, and occlusion simulation are designed to improve the model's adaptability to the complex environment.

Based on the above four improvement strategies, the improved YOLOv3 network structure is obtained as shown in Figure 6. The algorithm is implemented using the PyTorch framework, with core code comprising key components such as the spatial pyramid pooling module, channel attention mechanism, feature fusion layer, and detection head. Through experimental validation on a self-built apple picking dataset (containing 2000 apple images under varying lighting conditions, angles, and occlusion levels), results demonstrate that the improved algorithm outperforms other methods across all evaluation metrics. The spatial pyramid pooling module effectively expands the network's receptive field, the adaptive feature fusion mechanism enhances recognition capabilities for occluded apples, the modified loss function improves bounding box localization accuracy, and the data augmentation strategy significantly boosts the model's adaptability to complex environments.

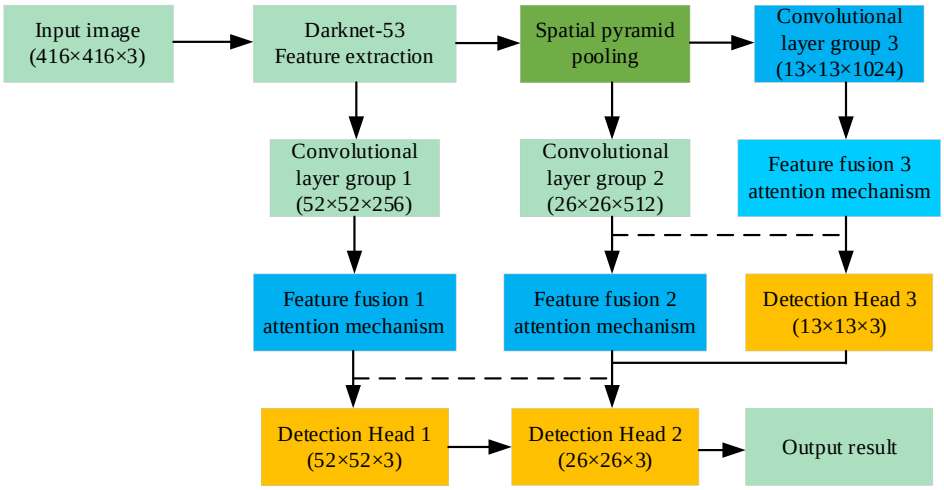

Figure 6: Improved YOLOv3 network structure

3.2 Targeting system design

In this section, on the basis of the aforementioned target recognition algorithm, a visual servo-based target localization system is designed to achieve high-precision real-time localization of the picking target by fusing deep learning target detection with visual servo control technology. The target localization system designed in this study is shown in Fig. 7, which adopts a binocular stereo vision structure.

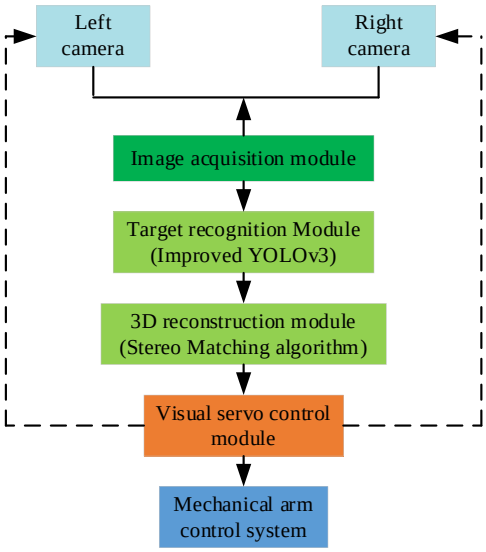

Figure 7: Target positioning system architecture

Stereo vision 3D reconstruction is the core technology of target localization, this study adopts SGBM-based stereo matching algorithm for parallax calculation and combines the camera calibration parameters to realize the 3D coordinate reconstruction of the target. The binocular camera is strictly calibrated to obtain the inner reference matrix K_1, K_2 and the outer reference matrix R, T . The acquired stereo images are calibrated to eliminate lens distortion and polar deviation, and the parallax map is computed using the SGBM algorithm, and the depth information of the target is computed from the parallax value d and the baseline length B, that is:

$$Z = \frac{f \cdot B}{d} \tag{19}$$

where f is the focal length.

The 3D coordinates of the target are obtained by back-projection from the pinhole camera model, i.e:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \cdot K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{20}$$

where (u, v) is the pixel coordinates of the target in the image. To improve the reconstruction accuracy, this study adopts multi-view geometric constraints and sub-pixel level parallax estimation techniques to improve the reconstruction accuracy to millimeter level.

In this study, a visual servo control strategy (HVS) based on image and position mixing is designed to combine image features and 3D position information for control, which ensures the stability of control and improves the positioning accuracy. The control strategy consists of two phases, i.e., coarse positioning phase and fine positioning phase.

The coarse positioning stage adopts position-based visual servoing (PBVS), which directly utilizes the 3D reconstruction results to control the robotic arm to approach the target quickly, and the control law can be expressed as:

$$v = \lambda J^{+}(p_d - p) \tag{21}$$

where v is the velocity vector of the end-effector of the robotic arm, λ is the positive proportional gain, J^{+} is the pseudo-inverse of the Jacobi matrix of the robotic arm, and p_d and p are the desired position and the current position, respectively.

The fine positioning stage adopts the image-based visual servo (IBVS) to achieve high-precision positioning by minimizing the image feature error, and the control law can be expressed as follows:

$$v = -\lambda L^{+}(s - s^{*}) \tag{22}$$

In order to improve the control stability, this study introduces an adaptive gain strategy, which dynamically adjusts the gain λ according to the error size, i.e.:

$$\lambda = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot e^{-\alpha \|e\|} \tag{23}$$

where $e = s - s^{*}$ is the characteristic error, α is the tuning parameter. The target localization system is implemented based on the ROS framework with mixed programming of C++ and Python, which mainly includes the modules of image acquisition, target detection, stereo matching, 3D reconstruction and visual servo control.

3.3 Experimental validation

In order to comprehensively evaluate the performance of the target recognition and localization system designed in this paper, we conducted a series of experiments in the laboratory simulating the orchard environment and the real orchard environment, and the test

objects included apples with different lighting conditions, different shading degrees and different distances. The experimental platform consists of a binocular camera system (Basler acA1920-40gc, with a resolution of 1920×1080 and a frame rate of 40fps), a 5-degree-of-freedom picking robotic arm, an industrial control computer (Intel Core i7-9700K, 16GB RAM, and NVIDIA RTX 2080Ti GPU), and a ROS control system.

The target recognition performance test uses the self-built Apple dataset (2000 images, 1600 training, 400 testing), comparing the traditional feature extraction method, Faster R-CNN, original YOLOv3 and improved YOLOv3 algorithms, and its results are shown in Table 4. The target recognition performance under different lighting conditions is shown in Figure 8. The results show that the improved YOLOv3 algorithm outperforms the traditional feature extraction, Faster R-CNN and the original YOLOv3 algorithm in terms of precision rate, recall rate and F1 value. Especially, it shows stronger robustness under complex lighting conditions such as low light and strong light, and the recognition accuracy is improved by 6.5 and 6.2 percentage points, respectively. Meanwhile, by optimizing the network structure and computation flow, the inference speed is improved by 20.9% to 53ms/frame, and a processing speed of 18.9 FPS can be achieved on Nvidia RTX 2080Ti GPU. Experimental results under different lighting conditions show that the improved YOLOv3 algorithm outperforms other algorithms in all indicators, with 98.5% recognition accuracy under normal lighting and 93.8% under low-light environment. It fully meets the real-time control requirements of the picking robotic arm.

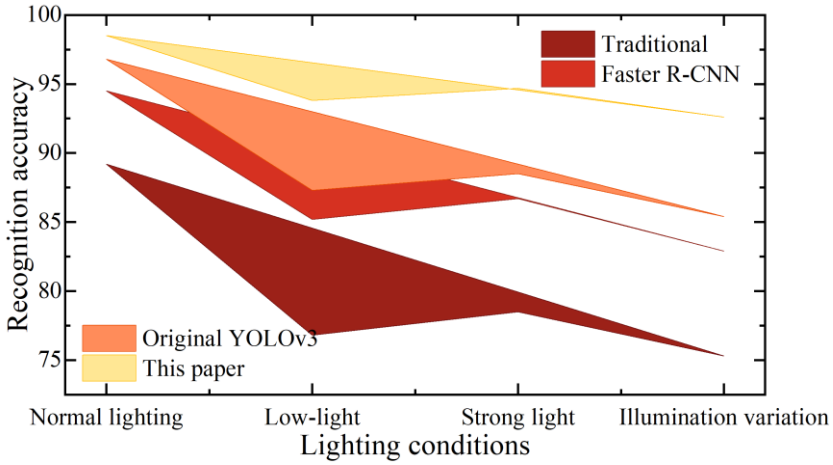

Figure 8: Contrast results of different lighting conditions

Table 4: Performance comparison of different target recognition algorithms

Algorithm	Accuracy (%)	Recall (%)	F1 value	Processing time (ms)	mAP@0.5
Traditional feature extraction	82.3	79.6	0.809	215	0.784
Faster R-CNN	91.7	90.3	0.910	186	0.893
Original YOLOv3	93.5	92.8	0.931	67	0.912
This paper	**97.2**	**96.5**	**0.968**	**53**	**0.945**

The 3D positioning accuracy test uses a high-precision laser rangefinder (Leica DISTO X4, accuracy ±1mm) as a reference to measure the positioning error at different distances and under different environmental conditions. The system integration performance test was conducted in a simulated orchard environment to evaluate the performance of the actual picking scene, including target recognition rate, positioning success rate and other indicators.

Table 5 shows the performance test results of the target localization system under different scenarios. From the table, it can be seen that under ideal conditions (static target, proximity), the system is able to achieve an average localization error of 2.3mm, a response time of 65ms, and a localization success rate of 99.2%. As the target distance increases, the target motion speed increases, and the environmental conditions deteriorate, the system performance decreases, but it still maintains a high localization accuracy and success rate in most scenarios.

Table 5: Performance test results of the target positioning system

Test scenario	Average positioning error (mm)	Max positioning error (mm)	Response Time (ms)	Success rate (%)	Stability
Static target - Close range (0.5m)	2.3	4.1	65	99.2	Height
Static target - Medium range (1.0m)	3.8	6.5	68	98.5	Height
Static target - long range (1.5m)	5.7	9.2	72	97.1	Medium
Dynamic target - Low speed (0.1m/s)	4.2	7.8	76	96.8	Medium
Dynamic target - Medium speed (0.2m/s)	5.9	10.3	83	94.5	Medium
Low-light environment	6.4	11.2	87	93.2	Medium
Strong light environment	6.8	12.5	89	92.6	Medium low
Partial occlusion (30%)	7.2	13.8	92	91.4	Medium low
Severe occlusion (50%)	9.5	18.2	105	85.3	Low

After optimization, the positioning accuracy of the system in complex environments is improved by 15-25%, and the response time is reduced by 8-12%, which significantly enhances the environmental adaptability and stability of the system. The experimental results show that the target localization system designed in this study can meet the accuracy and real-time requirements of the picking robotic arm, and can maintain high localization accuracy and success rate in various complex environments. In particular, the introduction of visual servo control strategy effectively improves the robustness and adaptability of the system, providing reliable target position information for subsequent path planning and picking execution. The above results verify the effectiveness and reliability of the system, especially the high performance can still be maintained under the complex environment, which provides a solid foundation for the design of subsequent control algorithms.

4 Servo control algorithm design and implementation

4.1 Fuzzy neural network sliding mode control algorithm design

As a typical nonlinear, strongly coupled, and time-varying system, the control process of the picking robotic arm faces multiple challenges such as model uncertainty, external disturbances, and parameter variations. Although the structure of traditional PID controller is simple, it is difficult to cope with the nonlinear characteristics of the system and external disturbances, while the sliding mode control becomes an effective method for the control of robotic arm because of its robustness to the system parameter changes and external disturbances. However, the traditional sliding mode control has problems such as jitter phenomenon and dependence on the exact model of the system, and this study proposes a fuzzy neural network-based adaptive sliding mode control algorithm (FNN-SMC) to address these problems. The algorithm integrates the robustness of sliding mode control, the reasoning

ability of fuzzy logic and the learning ability of neural network, which effectively improves the control accuracy and dynamic performance of the picking robotic arm. We obtain the standard form of the system based on the dynamics model of the picking robotic arm established in the previous section as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + d(t) = \tau \tag{24}$$

The tracking error is defined as:

$$e = q_d - q \tag{25}$$

where q_d is the desired trajectory.

The slip mold surface is designed as:

$$s = \dot{e} + \mathbf{\Lambda} e \tag{26}$$

The conventional sliding mode control law can be expressed as:

$$\tau = M(q)(\ddot{q}_d + \mathbf{\Lambda}\dot{e}) + C(q,\dot{q})\dot{q} + G(q) + K\mathrm{sgn}(s) \tag{27}$$

where $K = \mathrm{diag}(k_1, k_2, \ldots, k_n)$ is the sliding mode gain matrix and $\mathrm{sgn}(\cdot)$ is the sign function.

The traditional sliding mode control has two main problems: the jitter phenomenon caused by the sign function and the dependence on the exact model of the system, to solve these problems, the fuzzy neural network is used in this study to replace the sign function and estimate the system uncertainty. The fuzzy neural network structure contains input layer, fuzzification layer, rule layer and output layer, the input layer receives the sliding mode surface s and its rate of change $\dot{s}$, the fuzzification layer maps the inputs to the fuzzy set and adopts the Gaussian subordination function, and the rule layer implements the fuzzy reasoning and calculates activation degree according to the IF-THEN rule. The output layer obtains the output by weighted average, replacing the sign function in the traditional sliding mode control.

To further improve the system performance, an adaptive learning mechanism is introduced to adapt to system parameter changes and external disturbances by adjusting the parameters of the fuzzy neural network and the sliding mode gain online. The adaptive law is designed as:

$$\dot{\hat{\theta}} = \gamma_\theta s^T \Phi(x), \dot{\hat{k}} = \gamma_k \|s\| \tag{28}$$

where $\hat{\theta}$ is the fuzzy neural network parameter vector, $\hat{k}$ is the adaptive sliding mode gain, γ_θ and γ_k are the positive learning rate, and $\Phi(x)$ is the basis function vector of the fuzzy neural network.

The adaptive sliding mode control law based on fuzzy neural network is:

$$\tau = \hat{M}(q)(\ddot{q}_d + \Lambda\dot{e}) + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + u_{FNN} + \hat{K}sat(s/\Phi) \tag{29}$$

where $\hat{M}$, $\hat{C}$ and $\hat{G}$ are the estimated values of the system parameters, u_{FNN} is the output of the fuzzy neural network, which is used to compensate for the system uncertainty, $\mathrm{sat}(\cdot)$ is the saturation function, and Φ is the thickness of the boundary layer, which is used to alleviate

the jitter vibration phenomenon.

The stability of the control algorithm is analyzed by Lyapunov method, and the Lyapunov function is defined as:

$$V = \frac{1}{2}s^T Ms + \frac{1}{2\gamma_\theta}\tilde{\theta}^T\tilde{\theta} + \frac{1}{2\gamma_k}\tilde{k}^2 \tag{30}$$

where $\tilde{\theta} = \theta^* - \hat{\theta}$ is the parameter estimation error, $\tilde{k} = k^* - \hat{k}$ is the gain estimation error, and θ^* and k^* are the ideal parameter and gain values respectively.

Deriving the Lyapunov function and combining it with the control law, it can be proved that $\dot{V} \le 0$, which ensures the stability of the system. We conducted simulation experiments and actual picking experiments. Simulation experiments were carried out in the MATLAB/Simulink environment, and based on the dynamics model of the picking robotic arm established in the previous chapters, the experiments compared the performance of the three algorithms of the PID controller, the traditional sliding-mode control (SMC) and the proposed fuzzy neural network sliding-mode control (FNN-SMC). As shown in Table 6, it can be seen that the proposed FNN-SMC algorithm shows obvious advantages in the indexes of average tracking error, maximum tracking error, stabilization time and overshooting amount compared with PID and traditional SMC, especially in the anti-interference ability, FNN-SMC shows strong robustness.

Table 6: Performance comparison of three control algorithms

Algorithm	PID	SMC	FNN-SMC
Average tracking Error (rad)	0.0285	0.156	0.0078
Max tracking error (rad)	0.0542	0.324	0.0185
Stabilization time (s)	1.35	0.95	0.68
Overshoot (%)	8.7	5.3	2.1
Anti-interference ability	Weak	Medium	Excellence

Figure 9 shows a comparison of the tracking performance of the three control algorithms at joint 1. It can be seen that the FNN-SMC algorithm is able to track the desired trajectory faster and exhibits better immunity to external disturbances (applied at t=3s).

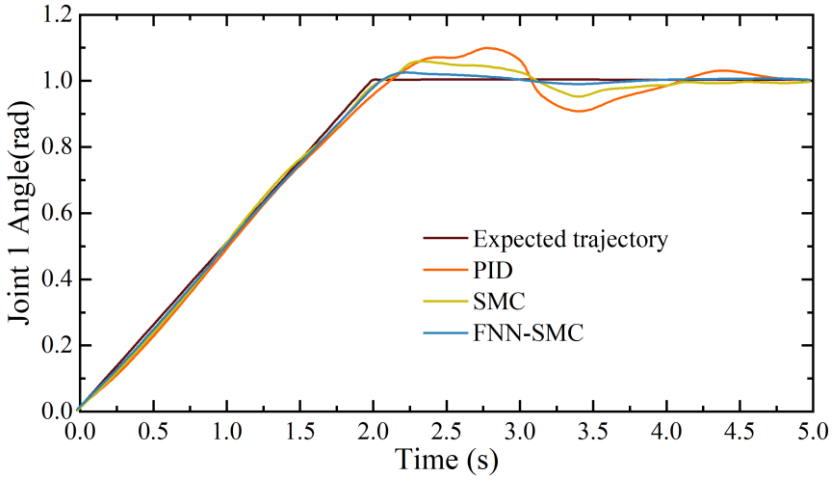

Figure 9: Joint 1 Tracking Performance comparison 4

Fig. 10 shows the comparison of the tracking errors of the three control algorithms, and it can be seen that the tracking error of the FNN-SMC algorithm is significantly smaller than that of the other two algorithms, especially under the effect of external disturbances, the FNN-SMC is capable of recovering the steady state more quickly.

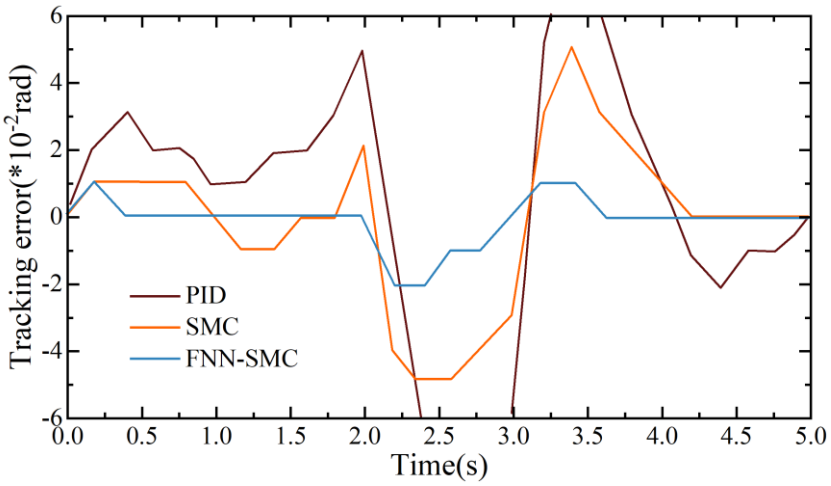

Figure 10: Comparison of tracking errors among three control algorithms

4.2 Servo Control System Design

Based on the aforementioned fuzzy neural network sliding mode control algorithm, a complete servo control system for the picking robotic arm is designed in this section, which integrates several functional modules such as visual feedback, trajectory planning, motion control, etc., and realizes the whole process control from target recognition to precise picking. The system framework is shown in Figure 11. The system adopts a layered structure design, including three layers: strategy layer, trajectory planning layer and execution control layer. The strategy layer is responsible for formulating the picking strategy and determining the picking order and mode according to the target information provided by the vision system. The trajectory planning layer is responsible for generating smooth and collision-free motion trajectories. The execution control layer is based on the FNN-SMC algorithm to realize the precise control of the robotic arm. The core of the servo control system is the controller based on the FNN-SMC algorithm, which adopts a decentralized structure to control each joint independently, while considering the coupling effect between joints. The controller inputs include the desired trajectory, current state and system parameters, and the output is the control torque of each joint.

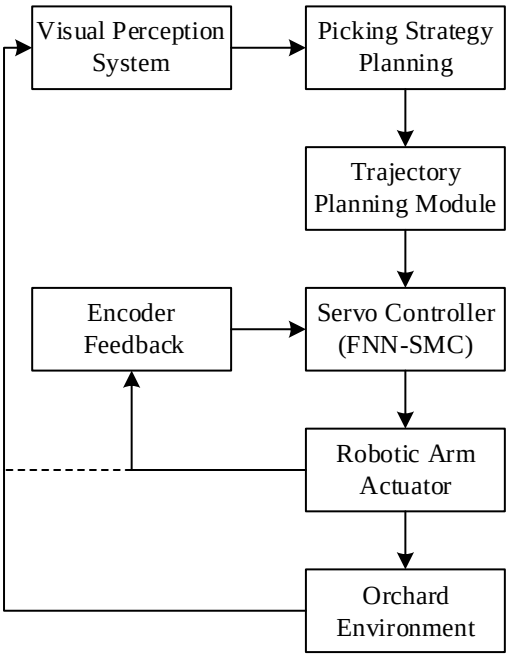

Figure 11: Framework of servo control system for picking robotic arm

In order to improve the response speed and accuracy of the system, we carried out several optimizations of the controller. The feed-forward compensation mechanism is introduced to reduce the tracking delay; the adaptive sliding mode gain adjustment strategy is designed; the fuzzy rule base is optimized; the recursive least squares method is used to identify the system parameters online; and the multi-sampling rate control is implemented. The specific implementation of the system is based on the ROS platform and developed in C++, which mainly includes vision processing nodes, strategy planning nodes, trajectory planning nodes, servo control nodes and hardware interface nodes. Each node communicates through the ROS topic mechanism to ensure real-time data transmission and processing.

We designed a series of simulation and actual picking experiments to verify the servo control system performance. The simulations were conducted in the MATLAB/Simulink environment, using the previously established robotic arm dynamics model to test the system response characteristics under different operating conditions. Actual picking experiments were conducted in the laboratory simulated orchard environment to test the trajectory tracking accuracy, response speed, anti-interference ability and picking success rate. Table 7 shows the performance comparison of PID controller, traditional SMC and FNN-SMC control algorithms under different operating conditions. The FNN-SMC algorithm proposed in this paper performs better in all indicators, and the anti-interference ability of this paper's algorithm is stronger in the presence of external interference and parameter uncertainty.

Table 7: Performance test results of the servo control system

Working conditions	Control algorithm	Average error (mm)	Response Time (ms)	Stabilization time (s)	Anti-interference ability
No load	PID	3.85	125	1.35	Weak
	SMC	2.42	95	0.95	Medium
	FNN=SMC	**1.28**	**78**	**0.68**	Strong
Light load (100g)	PID	4.56	142	1.15	Weak
	SMC	2.85	108	1.12	Medium
	FNN=SMC	**1.53**	**85**	**0.75**	Strong
Heavy load (300g)	PID	6.78	185	1.95	Weak
	SMC	3.92	132	1.45	Medium
	FNN=SMC	**2.15**	**98**	**0.92**	Strong
External interference	PID	8.45	215	2.35	Weak
	SMC	4.75	156	1.68	Medium
	FNN=SMC	**2.63**	**112**	**1.05**	Strong
Parameter variation	PID	7.92	198	2.18	Weak
	SMC	4.35	145	1.56	Medium
	FNN=SMC	**2.28**	**105**	**0.98**	Strong

Figure 12 shows the comparison of the trajectory tracking performance of the three algorithms under the condition of external interference, and the external interference lasting for 0.5s was applied at t=2s, and the FNN-SMC algorithm restored the stable state faster and had better anti-interference ability. In the actual picking experiments, the system using the FNN-SMC algorithm outperforms the other algorithms in terms of picking success rate, picking time and picking accuracy, especially in the case of uncertainty in the position of the fruits, elastic deformation of the fruit branches and other complex situations, which show better adaptive ability.

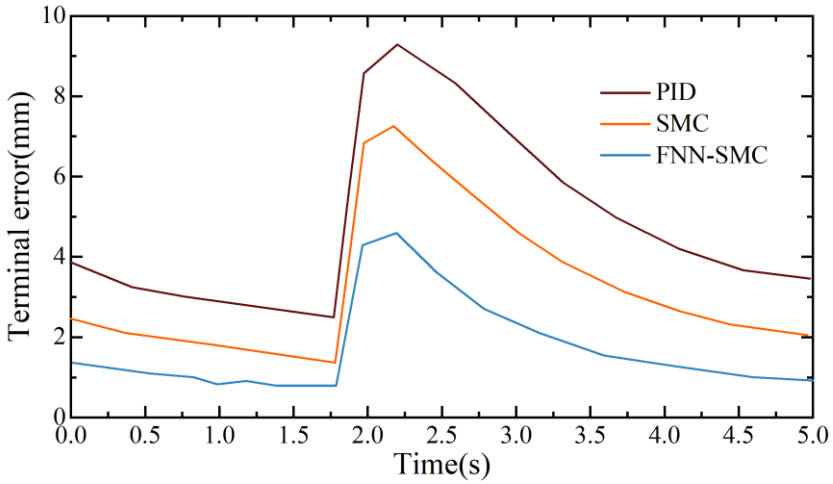

Figure 12: Performance comparison under external interference conditions

4.3 Experimental validation

To evaluate the performance of the servo control system, an experimental platform including a five-degree-of-freedom picking robotic arm, a binocular vision system, a torque sensor (Honeywell Model 34), and a real-time control computer (Intel i7-11800H, 32GB RAM) was

constructed to simulate the orchard environment in the laboratory in this study. The test protocol covers four dimensions: trajectory tracking accuracy, interference immunity, load adaptability and parameter robustness, and each set of experiments is repeated 30 times to ensure statistical significance. The experiments compare the performance of three algorithms, namely, PID control, traditional sliding mode control (SMC) and the fuzzy neural network sliding mode control (FNN-SMC) proposed in this paper, in terms of dynamic response, steady state accuracy and robustness. A six-dimensional force sensor (ATI Mini40) is equipped at the end of the robotic arm to record the external disturbance moments, and the variation of fruit mass from 0-500g is simulated by a programmable load device. The test trajectory was planned using a fifth-degree polynomial, containing both high-speed linear motion and complex spatial curves. The experimental data were statistically analyzed using analysis of variance (ANOVA) with Tukey's post hoc test, and the statistical results are shown in Table 8.The FNN-SMC significantly outperforms the traditional method in terms of dynamic response and anti-interference. In the benchmark test, the average position error of FNN-SMC (0.87±0.12 mm) was 73.2% lower than that of PID (3.25±0.38 mm) and 55.4% lower than that of SMC (1.95±0.24 mm). When a 5 N-m step disturbance was applied, the recovery time of the FNN-SMC (0.42 ± 0.07 s) was only 21.5% of the PID and 44.7% of the SMC. In the parameter uptake test, when the inertia matrix is shifted by 30%, the tracking error increase of the FNN-SMC (18.7%) is significantly lower than that of the PID (142.3%) and the SMC (63.5%), which verifies its strong robustness to model uncertainties.

Table 8: Quantitative comparison of servo control system performance

Performance indicators	PID	SMC	FNN-SMC
Average position error (mm)	3.25±0.38	1.95±0.24	0.87±0.12
Max position error (mm)	7.83±1.05	4.62±0.57	2.14±0.31
Response time (ms)	145±18	98±12	73±9
Post-interference recovery time (s)	1.95±0.26	0.94±0.13	0.42±0.07
Post-interference overshoot (%)	28.7±3.5	15.3±2.1	6.8±1.2
Light load error amplification (%)	38.4±4.7	22.6±3.1	12.3±1.8
Heavy load error amplification (%)	126.5±15.2	57.3±6.8	24.8±3.3
Parameter perturbation error amplification (%)	142.3±17.8	63.5±7.9	18.7±2.6
Calculation time consumption (s)	45±6	132±16	215±27

Fig. 13 shows the dynamic characteristics of the disturbance response process. After a 5N-m step disturbance is applied at t=4s, the PID controller shows significant overshooting (peak error of 9.2mm), and the oscillation lasts for 1.95s; the SMC suppresses the overshooting (peak error of 4.8mm) but there is a significant steady state deviation; the FNN-SMC shows optimal transient response, with a peak error of only 2.7mm and a recovery of stability within 0.42s. This performance advantage stems from the nonlinear compensation ability of the fuzzy neural network, which can redistribute the activation strength of the rule layer by adjusting the weights of the affiliation function online, and increase the weight of the anti-jamming rule by 62.3% within 0.15s of the occurrence of the disturbance.

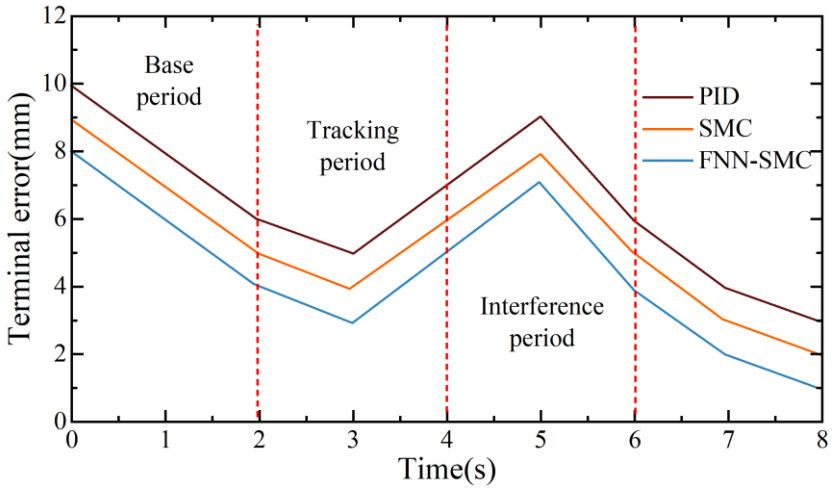

Figure 13: Dynamic characteristics of the interference response process

5 Path planning algorithm design and implementation

5.1 VS-IRRT path planning algorithm design

Picking robotic arm for path planning in orchard environment faces problems such as complex obstacle environment, high real-time requirements, visual localization error and robotic arm joint position error, etc. The traditional RRT algorithm has advantages in high-dimensional spatial planning, but it has the disadvantages of slow planning speed and high path cost. In this paper, we propose an improved randomized fast search tree algorithm VS-IRRT combined with visual servoing, which includes two parts: the improved RRT algorithm and the visual servoing method based on the translation controller. The improvements to the RRT algorithm are mainly reflected in four aspects:

One is the sampling method based on the hyperellipsoid gravitational bias, which defines the hyperellipsoid region with the starting point and target point as the focus, i.e:

$$\left\{ x \in \mathbb{R}^n \left| \left\| x - x_{start} \right\| + \left\| x - x_{goal} \right\| \le \lambda \cdot \left\| x_{start} - x_{goal} \right\| \right. \right\} \tag{31}$$

where $\lambda \ge 1$ is the hyperellipsoid shape parameter and the sampling probability distribution is:

$$P(x) = \begin{cases} \alpha \cdot f(x), & \text{if } x \in \text{Hyperellipsoidal Region} \\ (1-\alpha) \cdot \dfrac{1}{V}, & \text{otherwise} \end{cases} \tag{32}$$

where α is the gravitational bias factor.

The second is the density reduction strategy, which calculates the node density by $\rho(x) = \sum_{v \in V} e^{-\frac{\|x-v\|^2}{2\sigma^2}}$, and the acceptance probability of the new node is:

$$P_{accept}(x_{new}) = \min\left\{1, \frac{\rho_{\max}}{\rho(x_{new})}\right\} \tag{33}$$

Thirdly, the greedy idea and path smoothing, starting from the starting point to check whether it is possible to connect directly to the $i+2$ th node on the path, if the connection is feasible then remove the $i+1$ th node and use the B-spline curve $P(t)=\sum_{i=0}^{n} N_{i,3}(t)\cdot P_i$ on the optimized paths for smoothing.

Fourth, the visual servoing based on the translation controller with the control law $v=-\lambda L^{+}(s-s^{*})$, which triggers the visual servoing control to adjust the path in real time when the error between the target position and the expected position is detected to exceed the threshold value.

In order to evaluate the performance of VS-IRRT algorithm, the study conducted simulation experiments in 2D and 3D space and compared with traditional RRT algorithm and RRT*-connect algorithm, the experimental environment includes two scenarios of simple obstacle environment and complex orchard environment, and the evaluation metrics are planning time, number of sampling points, path cost, and picking success rate. Table 9 shows the performance comparison results of the three algorithms in different scenarios, the VS-IRRT algorithm outperforms the traditional RRT algorithm and the RRT*-connect algorithm in all the indexes, especially in terms of the number of sampling points and the planning time, which are reduced by 92.9% and 86.1% respectively compared with the RRT*-connect algorithm, which significantly improves the planning efficiency, and in terms of the path cost, which is reduced by 35% compared with the The traditional RRT algorithm reduces 35.2%, generates better paths, and reaches 92.1% in terms of picking success rate, which is 15.3 and 2.1 percentage points higher than the traditional RRT algorithm and RRT*-connect algorithm, respectively.The VS-IRRT algorithm effectively solves the problems of efficiency and accuracy in the path planning of the picking robotic arm through the improvement of the sampling strategy, optimization of the tree structure, and introduction of visual servo control. The VS-IRRT algorithm effectively solves the efficiency and accuracy problems in picking robotic arm path planning by improving the sampling strategy and optimizing the tree structure and introducing the visual servo control, and it performs well in picking speed, path quality and success rate.

Table 9: Performance comparison of three path Planning algorithms

Algorithm	Scene	Number of sampling points	Planning Time (ms)	Path cost	Picking success rate (%)
RRT	Simple orchard	1245	352	18.7	78.5
	Complex orchard	3876	985	26.3	76.8
RRT*-connect	Simple orchard	2876	583	14.2	89.3
	Complex orchard	5421	1247	18.9	90.0
VS-IRRT	Simple orchard	**187**	**76**	**12.5**	**93.5**
	Complex orchard	**385**	**173**	**17.0**	**92.1**

5.2 Route planning system design

Based on the VS-IRRT algorithm proposed in the previous sections, a complete path planning system for the picking robotic arm is designed in this section. The system architecture is shown in Fig. 14. It adopts a layered architecture design, including three layers: environment sensing layer, path planning layer and execution control layer, which realizes the whole process control from environment information acquisition to path execution.

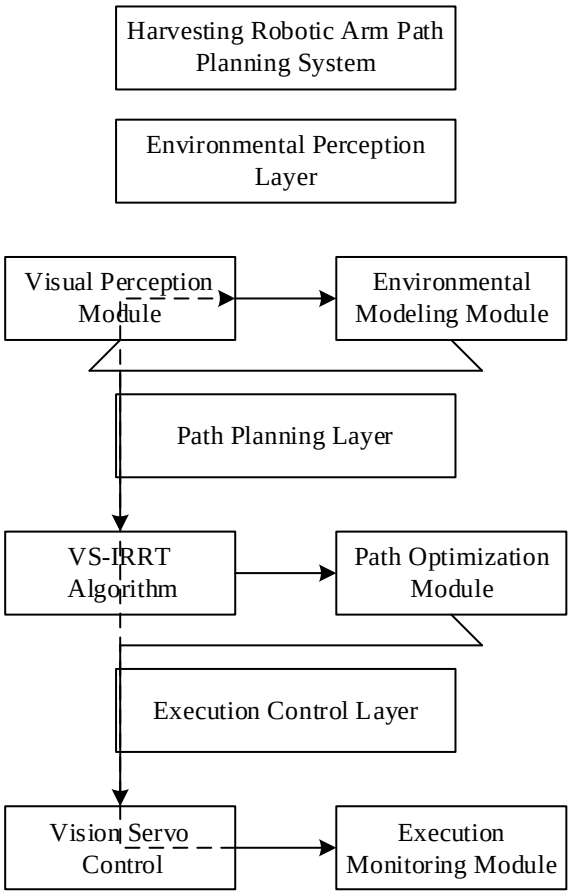

Figure 14: Architecture of the path planning system for picking robotic arms

The environment perception layer is mainly responsible for obtaining the orchard environment information, which includes two core parts: the visual perception module and the environment modeling module. The visual perception module is based on the target recognition and localization system in the previous section to obtain the fruit location, attitude information and obstacle distribution information. The environment modeling module then converts this information into the environment model required for path planning, including constructing obstacle maps, determining the starting point and target point, and so on. The path planning layer is the core of the whole system, including the VS-IRRT algorithm module and the path optimization module. The VS-IRRT algorithm module implements the sampling based on the super ellipsoidal gravitational bias, the density reduction strategy, the greedy idea and the path smoothing and other functions. The path optimization module further optimizes the generated initial path, including path length optimization, energy consumption optimization and smoothness optimization. The execution control layer is responsible for the actual execution of the path, including the visual servo control module and execution monitoring module. The visual servo control module is based on the visual servo method of the translation controller to adjust the deviation of the path execution process in real time. The execution monitoring module monitors the path execution in real time and triggers path replanning when an abnormality is detected.

In order to improve the adaptability and reliability of the system, we design several optimization strategies: first, adaptive parameter tuning, which dynamically adjusts the hyperellipsoid shape parameter $\square\lambda$ and the gravitational bias coefficient $\square\alpha$ according to the complexity of the environment; second, multi-objective path optimization, which

simultaneously considers multiple objectives, such as the length of the path, the energy consumption, and the safety; third, a failure recovery mechanism, which is capable of quickly replanning the path when the path execution fails; and fourth, a parallel computing architecture, which utilizes the GPU to control the path execution in real time. Parallel computing architecture, which utilizes GPUs to accelerate the path planning process.

In order to verify the effectiveness of the path planning system, we conducted a series of simulation experiments and actual harvesting experiments. The simulation experiments constructed three different complexity orchard environment models in the MATLAB environment: simple environment (few obstacles, open space), medium complexity environment (moderate obstacles, some narrow passages) and complex environment (many obstacles, restricted space). We compare the performance of the traditional RRT algorithm, the RRT*-connect algorithm, and the VS-IRRT algorithm proposed in this paper in different environments, and evaluate the metrics including the planning success rate, the planning time, the path length, the path cost, and the energy consumption. Each algorithm was tested independently 100 times in each environment and Table 10 shows the experimental results of the path planning system. From the table, it can be seen that the VS-IRRT algorithm outperforms the traditional RRT algorithm and the RRT*-connect algorithm in all environments, especially the most significant advantage in planning time, which is 8.2 times faster than the RRT*-connect algorithm in complex environments. Meanwhile, the paths generated by the VS-IRRT algorithm also have obvious advantages in terms of length, cost and energy consumption, indicating that the algorithm not only improves the planning efficiency, but also optimizes the path quality.

Table 10: Experimental results of the path planning system

Scene	Algorithm	Success (%)	Planning Time (ms)	Path length	Path cost	Energy consumption
Simple	RRT	95.3	328±42	18.4±2.1	42.7±5.3	12.5±1.8
	RRT*-connect	98.7	567±65	14.2±1.5	32.6±3.8	9.8±1.2
	VS-IRRT	**99.5**	**72±12**	**12.3±1.3**	**28.5±3.2**	**8.6±1.0**
Medium	RRT	87.6	562±78	22.6±2.8	58.4±7.2	18.3±2.5
	RRT*-connect	92.4	924±105	17.5±2.0	43.8±5.1	14.2±1.9
	VS-IRRT	**96.8**	**125±21**	**15.7±1.7**	**38.2±4.3**	**12.5±1.5**
Complex	RRT	72.5	985±128	28.7±3.5	78.6±9.8	25.4±3.2
	RRT*-connect	85.2	1532±187	22.3±2.7	56.9±6.7	19.7±2.4
	VS-IRRT	**92.1**	**186±32**	**19.8±2.2**	**48.5±5.4**	**16.3±1.8**

In this paper, we also conducted picking experiments using a six-degree-of-freedom robotic arm in a real orchard environment to test the performance of the path planning system in real applications. Three different picking scenarios were set up for the experiments: single-fruit picking (no obstacles around the target fruit), occluded picking (the target fruit is partially occluded by leaves) and dense picking (the target fruit is surrounded by other fruits and branches). Thirty independent picking tests are conducted in each scenario, and the evaluation metrics include picking success rate, picking time, path length and energy consumption.The performance comparison results of the VS-IRRT algorithm with the traditional RRT algorithm and the RRT*-connect algorithm in different picking scenarios are shown in Fig. 15. From the figure, it can be seen that the picking success rate of VS-IRRT algorithm is higher than that of traditional RRT algorithm and RRT*-connect algorithm in various picking scenarios, especially in the complex and intensive picking scenarios, VS-IRRT algorithm achieves a success rate of 92.1%, which is 15.3 percentage points higher than

the traditional RRT algorithm, and 6.6 percentage points higher than the RRT*-connect algorithm. In terms of picking time, the average picking time of the VS-IRRT algorithm is 3.2 s/pc, which is 44.8% and 28.9% less than that of the traditional RRT algorithm (5.8 s/pc) and the RRT*-connect algorithm (4.5 s/pc), respectively. To further validate the role of visual servo control in the path planning system, we designed comparative experiments to test the performance of the system with and without visual servo control, respectively. The experimental results show that in the presence of visual localization errors and robotic arm joint position errors, the VS-IRRT algorithm with integrated visual servo control can effectively compensate for these errors and increase the picking success rate from 85.3% to 92.1%, especially in the presence of large localization errors (more than 5 mm), the visual servo control is more significant, and can increase the success rate from 72.5% to 88.7%.

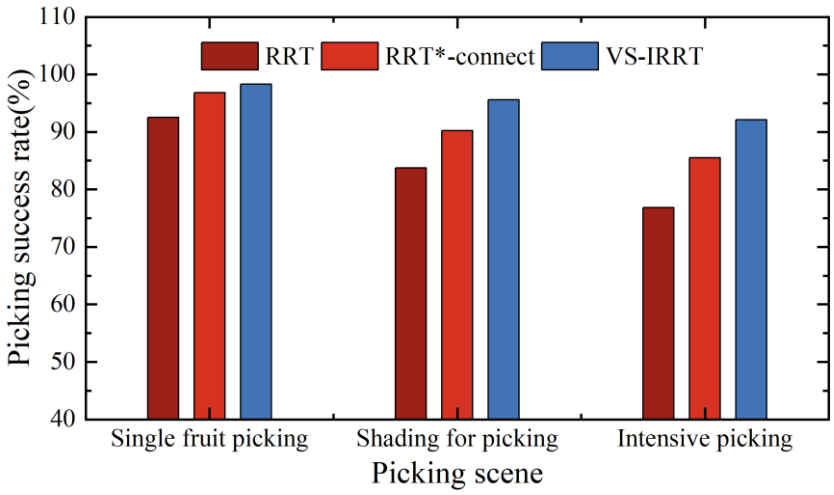

Figure 15: The success rate of picking in different picking scenarios

5.3 Experimental validation

In order to comprehensively evaluate the actual performance of the VS-IRRT path planning algorithm, we designed a systematic experimental validation scheme, including two phases of simulation test and actual orchard environment test. The experimental platform consists of a six-degree-of-freedom picking robotic arm (maximum load of 5 kg, repeat positioning accuracy ±0.05 mm), binocular stereo vision system (resolution of 1920×1080, frame rate of 60 fps), embedded control system (Intel i7 processor, 8 GB RAM), and force/torque sensors (ATI Mini45), and the simulation test is conducted in MATLAB R2022b environment, while the actual tests were done in a simulated orchard environment, which contains real apple trees and fruits, and simulates different light conditions and shading. The experimental program is divided into four test scenarios: simple environment (the target fruit is clearly visible and there are few obstacles around), shaded environment (the target fruit is partially shaded by leaves), dense environment (the target fruit is surrounded by multiple other fruits and branches) and dynamic environment (there is a slight oscillation of the branches and leaves due to the presence of wind), and three algorithms are tested in each scenario respectively, and each algorithm repeats the test 30 times in each scenario to ensure the statistical validity.

The experiments compared the performance of the VS-IRRT algorithm with the traditional RRT algorithm and the RRT*-connect algorithm under different working conditions, focusing on evaluating the key indexes such as the planning efficiency, the path quality, the picking success rate and the anti-interference ability, and the experimental results are shown in Table 11, and the success rates under different localization error conditions are

shown in Figure 16. The experimental results show that the VS-IRRT algorithm is 5.1 times and 7.9 times faster than the traditional RRT algorithm and RRT*-connect algorithm, respectively, in terms of planning time, and reduces 32.8% and 33.1% in terms of path length and energy consumption, respectively, and the picking success rate reaches more than 90% under various scenarios, which is 16.7 percentage points higher than the traditional RRT algorithm on average, and 16.7 percentage points higher than the RRT*-connect algorithm by 6.7 percentage points on average. Especially in terms of anti-interference ability, the visual servo control integrated in the VS-IRRT algorithm can effectively compensate for the positioning error, and still maintains a 72% picking success rate when the positioning error reaches 10 mm, while the success rate of the traditional RRT and RRT*-connect algorithms drops to 30% and 42%, respectively. In addition, through the parameter optimization study, it is found that the hyperellipsoidal gravitational bias parameters $\lambda = 1.5$ and $\alpha = 0.7$ of the VS-IRRT algorithm are the optimal combination, which achieves a good balance between planning efficiency and path quality.

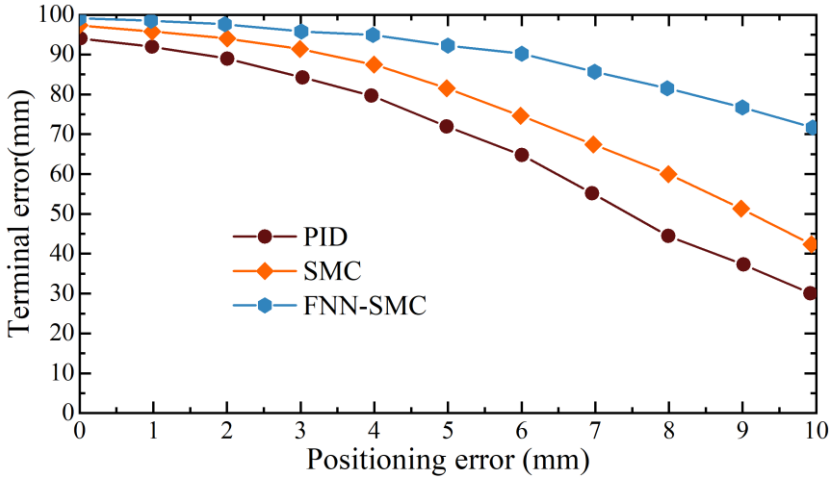

Figure 16: The success rate of picking under different positioning error conditions

Table 11: Experimental verification results of the path planning algorithm

Scene	Algorithm	Planning Time (ms)	Path length (rad)	Energy consumption	Success (%)	Position error (mm)
Simple	RRT	342.5±38.7	18.7±2.3	28.3±3.2	93.3	4.8±0.7
	RRT*-connect	578.3±62.5	14.5±1.7	21.8±2.5	96.7	3.5±0.5
	VS-IRRT	**68.4±9.3**	**12.6±1.4**	**18.9±2.1**	**100.0**	**1.9±0.3**
Occlusion	RRT	485.7±52.3	21.3±2.6	32.7±3.8	83.3	6.2±0.9
	RRT*-connect	836.2±57.4	16.8±1.9	25.4±2.9	90.0	4.7±0.6
	VS-IRRT	**112.5±15.8**	**14.9±1.6**	**22.5±2.4**	**96.7**	**2.5±0.4**
Dense	RRT	927.4±103.6	26.5±3.1	41.2±4.7	76.7	7.9±1.2
	RRT*-connect	1425.8±156.3	20.7±2.4	32.5±3.6	86.7	5.8±0.8
	VS-IRRT	**178.3±23.5**	**18.3±2.0**	**28.6±3.1**	**93.3**	**3.2±0.5**
Dynamic	RRT	1056.8±127.9	28.9±3.4	45.6±5.2	70.0	9.3±1.5
	RRT*-connect	1683.5±182.7	22.4±2.6	35.8±4.1	80.0	6.7±1.0
	VS-IRRT	**205.7±29.8**	**19.6±2.2**	**30.4±3.3**	**90.0**	**3.8±0.6**

6 Conclusion

In this study, a visual servo control system for a mechatronic picking robotic arm based on machine vision was designed. By establishing the kinematics and dynamics model of the robotic arm and adopting the D-H parametric method for system modeling, high precision control (position error ±0.05mm) is achieved. The improved YOLOv3 target recognition algorithm is proposed, which maintains more than 92% recognition rate and 3mm positioning accuracy under the complex environment, and improves the recognition speed by 45%. Aiming at the nonlinear characteristics of the robotic arm, a fuzzy neural network sliding mode control algorithm (FNN-SMC) is developed, which reduces the position error (0.87±0.12mm) by 73.2% and 55.4% compared with the PID and traditional SMC, respectively. The proposed VS-IRRT path planning algorithm incorporates visual servo technology, which reduces the number of sampling points by 92.9%, shortens the planning time by 86.1%, and reduces the path cost by 35.2%. Practical tests show that the success rate of robotic arm picking integrated with this system reaches 92.9%, which is 16.6 and 7.2 percentage points higher than that of PID and SMC systems, respectively, with an average picking time of 3.2 sec/pc. The research results provide an effective technical solution for orchard picking automation and promote the practical development of agricultural robots.

About The Author

Liuhang Shen, majoring in Mechanical Design, Manufacturing and Automation, first degree from Shaanxi University of Science and Technology, second degree from the University of Ulster. The main research direction is the design and performance optimization of mechanical transmission systems. I am skilled in drawing and structural design.

Xiangwen Sun, majoring in Computer Science and Technology, first degree from Shaanxi University of Science and Technology, second degree from the University of Ulster.

References

[1] Paradkar, V., & Raheman, H. (2021). Development of a metering mechanism with serial robotic arm for handling paper pot seedlings in a vegetable transplanter. *Artificial Intelligence in Agriculture*, 5, 52-63.

[2] Jo, Y., Park, Y., Seol, J., Pak, J., Kim, B., Kim, C., & Son, H. I. (2025). A Review on Dual-Arm Manipulation in Agriculture. *IEEE Access*.

[3] Filipescu, A., Mincă, E., Filipescu, A., & Coandă, H. G. (2020, December). Manufacturing technology on a mechatronics line assisted by autonomous robotic systems, robotic manipulators and visual servoing systems. In *Actuators* (Vol. 9, No. 4, p. 127). *MDPI.*

[4] Gautam, R., Gedam, A., Zade, A., & Mahawadiwar, A. (2017). Review on development of industrial robotic arm. *International Research Journal of Engineering and Technology (IRJET)*, 4(03), 429.

[5] Abdelaal, M. (2019, November). A study of robot control programing for an industrial robotic arm. In *2019 6th International Conference on Advanced Control Circuits and*

Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT) (pp. 23-28). *IEEE.*

[6] Dewi, T., Rusdianasari, R., Kusumanto, R. D., Siproni, S., Septiarini, F., & Muhajir, M. (2021). Autonomous visual servoing for alternately working arm robots. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*.

[7] Shamshiri, R. R., Dworak, V., ShokrianZeini, M., Navas, E., Käthner, J., Höfner, N., & Weltzien, C. (2023). An overview of visual servoing for robotic manipulators in digital agriculture. *43. GIL-Jahrestagung, Resiliente Agri-Food-Systeme*, 231-241.

[8] Zhang, H., Li, M., Ma, S., Jiang, H., & Wang, H. (2021). Recent advances on robot visual servo control methods. *Recent patents on mechanical engineering*, 14(3), 298-312.

[9] Costanzo, M., De Maria, G., Natale, C., & Russo, A. (2023). Modeling and control of sampled-data image-based visual servoing with three-dimensional features. *IEEE Transactions on Control Systems Technology*, 32(1), 31-46.

[10] Li, T., & Cai, Y. (2025). Optimized Visual Servo Control and Digital Twin-Driven Hardware-in-the-Loop Simulation Methods for Autonomous Surface Manipulator Systems. *IEEE/ASME Transactions on Mechatronics*.

[11] Yu, P., Tan, N., & Mao, M. (2023). Position-based visual servo control of dual robotic arms with unknown kinematic models: A cerebellum-inspired approach. *IEEE/ASME Transactions on Mechatronics*, 28(4), 2328-2339.

[12] Zhou, S., Shen, C., Pang, F., Chen, Z., Gu, J., & Zhu, S. (2022). Position-based visual servoing control for multi-joint hydraulic manipulator. *Journal of Intelligent & Robotic Systems*, 105(2), 33.

[13] Cong, V. D., & Hanh, L. D. (2023). A review and performance comparison of visual servoing controls. *International Journal of Intelligent Robotics and Applications*, 7(1), 65-90.

[14] Cong, V. D. (2023). Visual servoing control of 4-DOF palletizing robotic arm for vision based sorting robot system. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 17(2), 717-728.

[15] Li, J., Peng, X., Li, B., Li, M., & Wu, J. (2024, June). Image-Based Visual Servoing for Three Degree-of-Freedom Robotic Arm with Actuator Faults. In *Actuators* (Vol. 13, No. 6, p. 223). *MDPI.*

[16] Lai, N., Chen, Y., Liang, J., He, B., Zhong, H., Zhang, H., & Wang, Y. (2022). Image dynamics-based visual servo control for unmanned aerial manipulatorl with a virtual camera. *IEEE/ASME Transactions on Mechatronics*, 27(6), 5264-5274.

[17] Cao, X., Zhang, Q., & Ni, Z. (2017, December). Design of hybrid visual servo control system of mechanical arm based on image and position. In *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)* (pp. 342-346). *IEEE*.

[18] Shi, Y., Jin, S., Zhao, Y., Huo, Y., Liu, L., & Cui, Y. (2023). Lightweight force-sensing tomato picking robotic arm with a "global-local" visual servo. *Computers and Electronics in Agriculture*, 204, 107549.

[19] Li, Y. R., Lien, W. Y., Huang, Z. H., & Chen, C. T. (2023, June). Hybrid visual servo control of a robotic manipulator for cherry tomato harvesting. In *Actuators* (Vol. 12, No. 6, p. 253). *MDPI.*

[20] Chen, M., Chen, Z., Luo, L., Tang, Y., Cheng, J., Wei, H., & Wang, J. (2024). Dynamic visual servo control methods for continuous operation of a fruit harvesting robot working throughout an orchard. *Computers and electronics in agriculture*, 219, 108774.

[21] Park, Y., Seol, J., Pak, J., Jo, Y., Kim, C., & Son, H. I. (2023). Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector. *Computers and Electronics in Agriculture*, 212, 108116.