

Article

Real-Time Constrained Visual Servoing for Agricultural Harvesting Robots via MPC-Guided Reinforcement Learning

Liangzheng Gao ^{1,2}, Qingchun Feng ^{2,*} , Shiqi Chen ², Zhijie Yang ³, Fengcui Fan ³, Lin Chen ⁴ and Chunjiang Zhao ^{1,2,*} 

¹ College of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing 100083, China; glzzz2024@163.com

² Intelligent Equipment Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China; chen2752933052@163.com

³ Hebei Academy of Agricultural and Forestry Sciences, Shijiazhuang 050031, China; yzjsz@163.com (Z.Y.); njsffc@163.com (F.F.)

⁴ Hebei Institute of Agricultural Mechanization Co., Ltd., Shijiazhuang 050031, China; nongji.0501@163.com

* Correspondence: fengqc@nercita.org.cn (Q.F.); zhaocj@nercita.org.cn (C.Z.)

Abstract

With the intensification of global agricultural labor shortage and scaled development of facility agriculture, autonomous precision harvesting robots for unstructured greenhouse environments have become an urgent need. For cluster-picking crops such as tomatoes, visual servoing enables real-time closed-loop control of the end-effector pose, addressing challenges of random fruit distribution and variable stem orientations. However, existing methods struggle to balance constraint handling with real-time efficiency. This paper proposes an MPC-Guided Reinforcement Learning visual servoing framework, innovatively combining the planning capability of optimal control with the adaptive learning ability and real-time inference advantages of reinforcement learning. The approach adopts a teacher–student paradigm: expert trajectories from the MPC controller warm-start the reinforcement learning policy through behavior cloning, followed by PPO-based fine-tuning with adaptive gain regulation and stagnation-enhanced exploration mechanisms. Simulation experiments demonstrate a 95% success rate with average positioning and orientation errors of 13.6 mm and 0.009 rad respectively. Compared to MPC baseline, task steps are reduced by 53.4%; compared to Standard PPO, success rate improves by 6%. Greenhouse field validation achieves 85.3% picking success rate and 5.63 s per fruit operation time, confirming the framework’s excellent balance among control precision, robustness, and efficiency for high-precision robotic harvesting in unstructured agricultural environments.

Keywords: agricultural harvesting robot; visual servoing; reinforcement learning; teacher–student learning; Model Predictive Control



Academic Editor: Juraj Gregaň

Received: 28 February 2026

Revised: 23 March 2026

Accepted: 27 March 2026

Published: 1 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Fresh fruit harvesting is one of the most labor-intensive and time-consuming processes in agricultural production. With the intensification of global population aging and increasing agricultural labor shortage, traditional manual harvesting methods can no longer meet the urgent demands for efficiency and cost control in modern large-scale agriculture [1,2]. Therefore, developing agricultural harvesting robots with autonomous operation capabilities to achieve the transformation from “labor-intensive” to “technology-

intensive” modes [3] has become both a research hotspot and an inevitable trend in the field of agricultural robotics [4].

Currently, harvesting robots primarily adopt two operational modes: single-fruit picking and cluster harvesting. Compared to single-fruit picking, cluster harvesting (taking tomatoes as an example) requires the end-effector to complete high-precision harvesting operations within the confined space formed by fruits, main stems, and peduncles. This operational mode not only demands high positioning accuracy but also imposes stringent requirements on approach path planning and end-effector operation poses to achieve effective obstacle avoidance and safe operation in complex agricultural environments [5,6]. Therefore, introducing visual servoing to enhance close-range positioning accuracy and operational stability is one of the critical steps for transitioning harvesting robots toward practical field applications [7].

Visual servoing tightly couples visual perception with motion control to construct a closed-loop feedback system, enabling real-time correction of perception errors and execution deviations, thereby significantly improving robot positioning accuracy and operational stability in unstructured environments [8,9]. Early studies mainly verified the feasibility of applying vision-guided closed-loop control to fruit harvesting robots. For example, Zhao De-An et al. [10] demonstrated the integration of visual perception, manipulator control, and harvesting execution in an autonomous apple-picking system. Mehta et al. [11] and Mehta et al. [12] further extended this line of research by introducing robust and adaptive visual servo control strategies for harvesting robots under uncertainty and fruit motion. More recently, [9] Lehnert et al. advanced harvesting-oriented visual servoing toward active perception by seeking better viewpoints in occluded scenes.

However, as discussed by Zhao Y. et al. [13] and Magalhães et al. [14], most existing harvesting-oriented visual servo studies mainly improve performance through target recognition, viewpoint adjustment, or local feedback regulation, while only limited attention has been paid to the explicit coordination of picking-point accuracy, end-effector orientation alignment, collision avoidance, and kinematic feasibility within a unified closed-loop control framework. This limitation becomes particularly critical in greenhouse tomato harvesting, where the robot operates in a confined workspace formed by fruits, stems, and peduncles, and must simultaneously satisfy stringent geometric constraints and high-precision motion requirements. Therefore, achieving robust and efficient visual servoing under multiple coupled constraints remains a key challenge in robotic harvesting.

Model Predictive Control (MPC) has become an important paradigm for robotic manipulation because it optimizes future control actions over a receding horizon while explicitly handling state, input, and safety constraints within a unified framework. In contrast to instantaneous feedback controllers, MPC is particularly suitable for manipulation tasks that involve strong nonlinear coupling, limited workspace, and strict motion feasibility requirements. In visual-servo applications, Qiu et al. [15] incorporated visibility and actuator-related constraints into an MPC-based IBVS scheme, showing the advantage of predictive optimization for constrained robot motion. For articulated manipulators, Carron et al. [16] combined data-driven model compensation with MPC to improve trajectory tracking accuracy, while Kleff et al. [17] demonstrated high-frequency nonlinear MPC on a real 7-DoF manipulator, highlighting the potential of MPC for reactive control in dynamic environments. Beyond pure motion tracking, Gold et al. [18] proposed model predictive interaction control for robotic manipulation tasks with force and motion coupling, extending MPC to contact-rich operations. However, MPC applications in visual servoing still face two limitations [19]: first, its performance heavily depends on the accuracy of system and prediction models, with control effectiveness potentially degrading significantly under perception errors, environmental variations, or model mismatch conditions [20]; second,

MPC requires solving optimization problems at each control cycle, incurring substantial computational overhead, making it difficult to maintain high-frequency closed-loop control in high-DoF manipulators and highly constrained scenarios over extended periods, which is particularly pronounced in close-range, high-precision visual servoing tasks [17,21].

In recent years, Deep Reinforcement Learning (DRL) has provided new solutions for motion planning and control problems of robotic manipulators due to its powerful nonlinear mapping capability and adaptability to high-dimensional state spaces [22,23]. RL can learn end-to-end mapping strategies from visual perception to joint actions through environmental interaction. Addressing different challenges in visual servoing, Zhong et al. [24] proposed a DQN-based uncalibrated IBVS method to handle FOV constraints; Zhang et al. [25] developed a TD3-based multi-perspective closed-loop visual servoing framework; Li et al. [26] designed an SAC-based end-to-end visual servoing controller achieving sub-pixel level positioning accuracy. Despite demonstrating RL's potential, these methods still face significant challenges in high-precision, highly constrained tasks such as tomato cluster harvesting: (1) low sample efficiency, often requiring extensive interactions to converge to stable policies with high training costs [27]; (2) early exploration phases in confined spaces prone to ineffective sampling or even collisions, posing high safety risks [28]; (3) complex constraints difficult to naturally internalize through reward functions, making reward shaping challenging and susceptible to local optima [29]; and (4) RL tends to focus on learning policies from empirical data while often neglecting available prior knowledge such as system dynamics models and kinematic constraints [30]. For the aforementioned challenges faced by reinforcement learning in complex manipulation tasks, providing heuristic and near-optimal prior knowledge or prior experience is of significant importance for improving learning efficiency, accelerating policy convergence, and enhancing decision quality. In this context, Model Predictive Control (MPC), as a mature optimal control method, demonstrates complementary advantages to reinforcement learning.

Based on the above context, this paper proposes an MPC-Guided Reinforcement Learning Framework. The method adopts a “teacher–student” learning paradigm, utilizing a nonlinear MPC controller with explicit geometric constraint modeling capability as the “teacher” to generate high-quality, collision-free expert demonstration trajectories. Subsequently, the reinforcement learning policy is warm-started through Behavior Cloning (BC), enabling it to inherit safe and feasible motion priors, thereby effectively avoiding the blind exploration and high-risk collision issues in the early stages of reinforcement learning. Building upon this foundation, the Proximal Policy Optimization (PPO) algorithm is introduced, combined with adaptive gain regulation and stagnation-enhanced exploration mechanisms, to interactively fine-tune the policy for correcting covariate shift and improving policy robustness in dynamic environments. The core innovation of this framework lies in two aspects: on one hand, complex geometric constraints and physical feasibility knowledge are implicitly injected into the reinforcement learning policy through MPC expert demonstrations, significantly improving sampling efficiency and reducing training time; on the other hand, the finally trained policy network requires only a single forward pass during inference to output control increments, thereby achieving visual servoing control that simultaneously satisfies strict geometric constraints and provides millisecond-level real-time response efficiency.

The main contributions of this paper are as follows:

- (1) A keypoint-based 3D pose representation method is proposed to accurately describe tomato peduncle and stem geometry, providing reliable spatial constraints for visual servoing within a high-fidelity simulation environment.

- (2) An MPC teacher controller is developed to generate collision-free expert trajectories while satisfying hard constraints including cutting angle, joint limits, and obstacle avoidance.
- (3) A behavior cloning-based policy learning mechanism with adaptive gain regulation and stagnation-triggered exploration is proposed to efficiently inherit motion priors from expert demonstrations while enhancing convergence and robustness.
- (4) A comprehensive reward function is designed to effectively balance task completion, orientation alignment, control smoothness, and safety, addressing the challenge of internalizing complex constraints in PPO learning.
- (5) The effectiveness and robustness of the proposed method are validated through high-fidelity simulation experiments and real greenhouse environment deployment.

2. Materials and Methods

2.1. Robotic Prototype

The robotic system developed in this study is designed for greenhouse tomato harvesting operations, with the objective of achieving autonomous localization, approach, and picking of entire fruit clusters on the main stem. As illustrated in Figure 1, the system primarily comprises a rail-mounted mobile platform, a 6-degree-of-freedom (6-DOF) robotic manipulator (xMate SR5, Rokae, Beijing, China), a custom-designed integrated gripper-cutter end-effector, an Intel RealSense D405 RGB-D camera (Intel Corporation, Santa Clara, CA, USA), an industrial personal computer (IPC), and other auxiliary components.

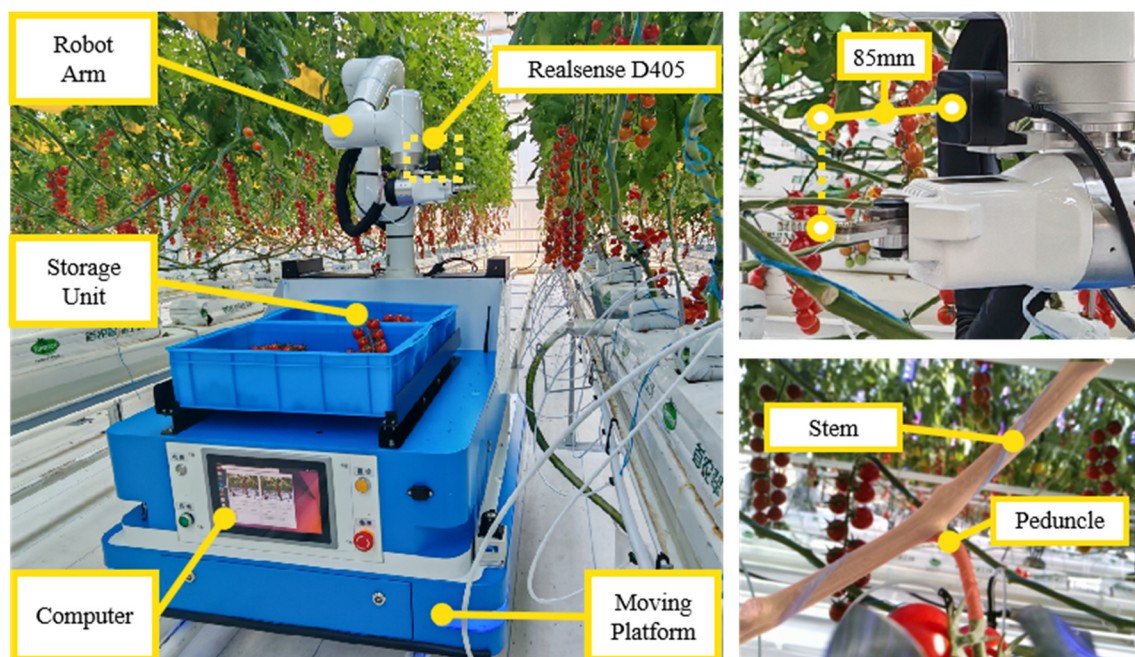


Figure 1. Robotic prototype.

The manipulator is mounted at the front end of the mobile chassis, with a body mass of 16.5 kg, a rated payload capacity of 5 kg, and a maximum working radius of 919 mm. The end-effector employs electric-driven gripping and cutting mechanisms, with a maximum dynamic mass of approximately 500 g and a length of 154 mm. The dynamic parameters of the manipulator, including geometric dimensions, mass/inertia parameters, and joint motion constraints, are extracted from the official URDF file. These parameters are utilized to construct the robot–environment interaction simulation platform and serve as state inputs for the Model Predictive Control (MPC) and reinforcement learning strategies.

The vision system adopts an “eye-in-hand” configuration. An RGB-D camera (RealSense D405, Intel Corporation, Santa Clara, CA, USA) is mounted above the end-effector to synchronously acquire color images and aligned depth images at a resolution of 640×480 pixels. As shown in Figure 1, the engineering design fully considers the imaging characteristics of the camera: although the D405 camera has a minimum sensing depth of 70 mm, through structural optimization, the gripping center point of the end-effector is positioned approximately 85 mm from the camera plane. This design ensures that even during the extremely close-range operational phase when the manipulator approaches and grasps the fruit peduncle, the camera remains within its effective working range, thereby stably acquiring depth information of the target and surrounding obstacles.

2.2. Sensing

Accurate acquisition of three-dimensional spatial information of tomato fruits and their surrounding obstacles (especially the main stem and fruit cluster) is a prerequisite for manipulator motion planning and visual servoing control. To accommodate both global task planning and close-range precision servoing, this study divides the perception pipeline into two stages: far-range and close-range operation.

In the far-range stage (Figure 2a), the system performs fruit cluster recognition, ripeness assessment, and coarse localization to guide the manipulator’s approach toward the target region. However, slender structures such as the main stem and fruit peduncles have limited resolution at this distance, making stable extraction of geometric details for picking point localization difficult. Therefore, this stage provides only coarse-grained guidance without directly participating in visual servoing.

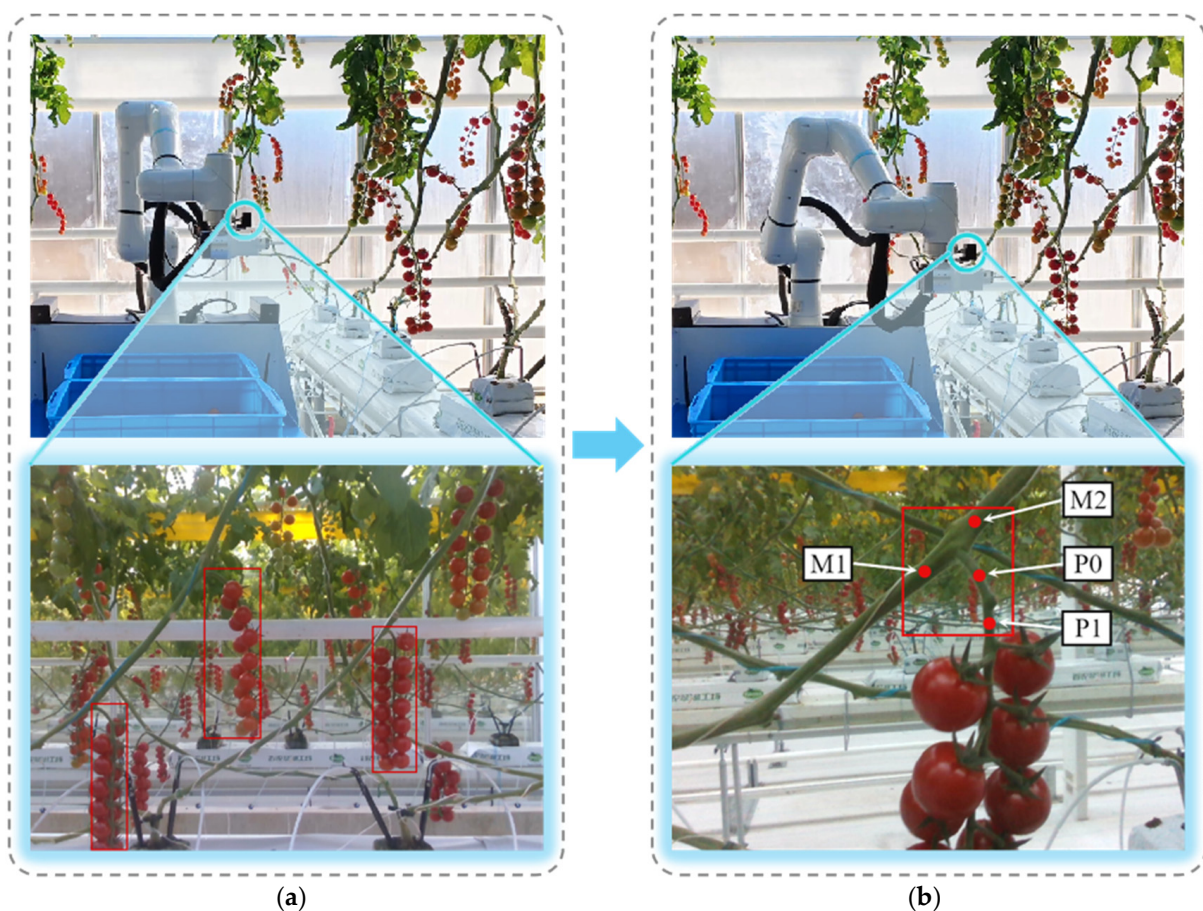


Figure 2. Identification process. (a) Far-range fruit cluster localization; (b) close-range keypoint detection.

When the manipulator enters the target fruit cluster neighborhood, the system switches to the close-range stage (Figure 2b), focusing on keypoint detection of the main stem and fruit peduncles. The core objective is to simultaneously obtain (1) the three-dimensional spatial position of the peduncle picking point, and (2) directional and planar constraints for constructing the desired end-effector pose.

This study employed YOLOv8-Pose for keypoint detection, reusing a close-range greenhouse tomato dataset from prior work [31] to ensure stability and real-time performance. The dataset comprises 1000 close-range greenhouse tomato images, partitioned into training, validation, and test sets at an 8:1:1 ratio. The model performance metrics are summarized in Table 1, confirming sufficient perceptual accuracy as input to the visual servoing framework.

Table 1. The model performance metrics.

	F1-Score (%)	mAP-50 (%)	Precision (%)	Recall (%)
YOLOv8-Pose	89.1	93.1	92.4	93.6

To achieve structured and interpretable three-dimensional localization, this study further constructs a tomato geometric prior model based on keypoints. Considering that the geometric dimensions of mature tomatoes are relatively stable [5], measurements were conducted on 100 plants, yielding maximum cross-sectional diameters of approximately 14 mm, 6 mm, and 50 mm for the main stem, fruit peduncle, and fruit, respectively. As shown in Figure 3a, the three-dimensional reconstruction process is as follows: First, the keypoint detection network obtains the positions of each keypoint in the pixel coordinate system; subsequently, the corresponding depth is read from the depth map, and the three-dimensional coordinates of keypoints in the camera coordinate system are calculated through back-projection based on camera intrinsic parameters, as shown in Equation (1).

$$\begin{cases} X = (u - c_x) \frac{Z}{f_x} \\ Y = (v - c_y) \frac{Z}{f_y} \\ Z = Z \end{cases} \quad (1)$$

where (X, Y, Z) denotes the three-dimensional coordinates of the keypoint in the camera frame; (u, v) denotes the corresponding pixel coordinates of the detected keypoint in the image plane; c_x and c_y denote the principal point coordinates (i.e., the pixel coordinates of the optical center); f_x and f_y denote the focal lengths along the horizontal and vertical axes, respectively, expressed in pixel units; and Z denotes the depth value at the corresponding pixel, directly obtained from the aligned depth image of the RealSense D405 RGB-D camera.

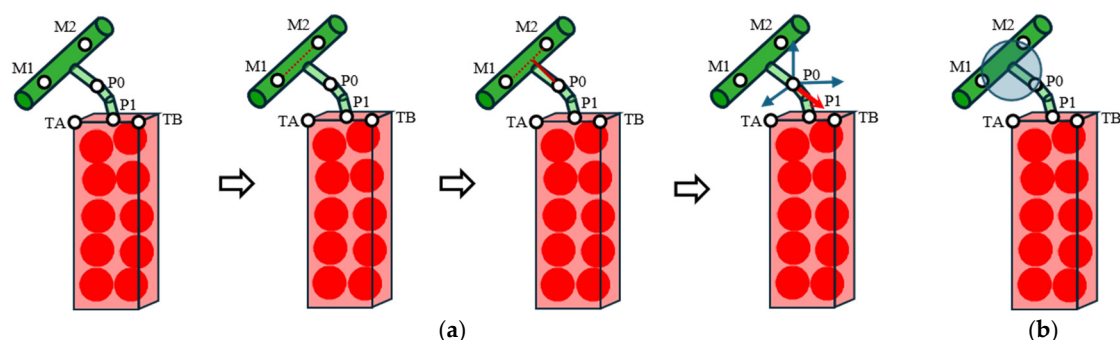


Figure 3. Three-dimensional reconstruction of fruits. (a) Sequential steps of the 3D geometric reconstruction process; (b) optimal picking plane.

After obtaining the three-dimensional coordinates, the cutting point keypoint P_0 is directly used as the position target input for visual servoing. Based on the measurement data, the following geometric approximations are adopted: the main stem is approximated as a cylinder with its axis determined by the three-dimensional line connecting two stem keypoints M_1 and M_2 , combined with the stem diameter prior to form a cylindrical constraint; the fruit peduncle is approximated as two connected cylindrical segments with local pose expressed by keypoint constraints. To complete the geometric information of the desired end-effector pose, this study utilizes three keypoints $\{M_1, M_2, P_0\}$ to define an “optimal picking plane” (as shown in Figure 3b), and constructs a pose vector reference v_{pick} within this plane: the in-plane direction indicates the desired cutting direction of the end-effector, while the plane normal constrains the remaining degrees of freedom of the end-effector about the cutting direction, thereby extending the two-dimensional pose information provided by the direction vector into a reference frame capable of generating a complete end-effector pose. Finally, combined with the P_1 and fruit diameter priors, critical points T_A and T_B for fruit occupancy boundaries are inferred through bounding volume approximations, which are used to provide obstacle avoidance constraints for neighboring fruits during motion planning and servoing processes.

To consistently represent the spatial relationships between tomato fruits and the main stem/fruit peduncles in the simulation environment, this study employs hand-eye calibration to transform the three-dimensional positions of keypoints from the camera coordinate system to the robot base coordinate system [32,33]. Through the extrinsic transformation matrix obtained from calibration, the keypoints in the above geometric model can be uniformly expressed in the base coordinate system, based on which the three-dimensional morphology of the tomato main stem–fruit peduncle–fruit is reconstructed (corresponding to Figure 3). This three-dimensional reconstruction result is used both to construct the robot–environment interaction simulation environment and to provide unified and interpretable geometric inputs for subsequent motion planning, MPC constraint modeling, and reinforcement learning strategy training.

2.3. Visual Servoing Framework

The core task of visual servoing (VS) is to construct closed-loop control through visual feedback, minimizing the error between the current visual features $s(m(t))$ and the desired reference features s^* , as shown in Equation (2). In the field of agricultural robotics, existing visual servoing methods are primarily divided into Image-Based Visual Servoing (IBVS) and Position-Based Visual Servoing (PBVS).

$$e(t) = s(m(t), a) - s^* \quad (2)$$

where $e(t)$ denotes the visual feature error, $s(m(t), a)$ denotes the current visual feature vector extracted from the live image, and s^* denotes the desired reference feature vector corresponding to the target configuration.

Although IBVS exhibits robustness to camera calibration errors, it is unsuitable for greenhouse tomato harvesting due to: (1) unstable feature extraction caused by morphological variation and occlusions, making image-plane feedback signals prone to jitter or loss; and (2) inability to explicitly model complex three-dimensional approach vectors and pose alignment requirements within two-dimensional image space, leading to control divergence under feature instability.

This study adopts the Position-Based Visual Servoing (PBVS) framework, which utilizes three-dimensional geometric information from the perception module to construct control laws directly in $SE(3)$ Cartesian space. This paradigm transforms the task into a 6-DoF pose regulation problem, enabling decoupled optimization of position and orienta-

tion errors while providing a rigorous mathematical foundation for integrating obstacle avoidance and dynamic constraints.

To describe the spatial relationships among the robot, vision system, and operational objects, the following coordinate frames are defined in this study:

Robot Base Frame $\{B\}$: Fixed at the manipulator base, serving as the inertial reference frame for robot kinematics and global control.

Tool Frame $\{E\}$: Fixed at the Tool Center Point (TCP) of the end-effector. The z_E axis is along the end-effector approach direction, the x_E axis is along the gripper cutting direction, and the y_E axis is determined by the right-hand rule.

Camera Frame $\{C\}$: Fixed at the optical center of the eye-in-hand camera, following the standard camera coordinate system definition.

Target Frame $\{G\}$: Bound to the peduncle picking point, determined by position and approach direction.

The rigid body transformation from any coordinate frame $\{A\}$ to $\{B\}$ in space is represented by the homogeneous transformation matrix ${}^B T_A$:

$${}^B T_A = \begin{bmatrix} {}^B R_A & {}^B t_A \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3)$$

where ${}^B R_A \in SO(3)$ is the rotation matrix and ${}^B t_A \in \mathbb{R}^3$ is the translation vector.

To achieve precise cutting operations in three-dimensional space, the geometric primitives output by the perception module must be transformed into a complete six-degree-of-freedom (6-DoF) target pose, and corresponding control variables and errors must be defined.

The construction of task targets in this study is obtained through the outputs of the perception module and geometric reconstruction. The geometric construction of task targets primarily includes the following two key quantities:

- (1) Picking Point Position: Three-dimensional spatial coordinates (x, y, z) of the target.
- (2) Picking Pose Vector: Characterizes the ideal orientation of the end-effector.

Since a single 3D vector constrains only pitch and yaw angles, leaving roll undetermined, this study combines the picking plane normal to lock the roll angle, constructing a uniquely determined 6-DoF target pose.

The core task of PBVS is to generate control commands based on the deviation between the current end-effector pose ${}^B T_E$ and target pose ${}^B T_G$. We define this error $\mathbf{e}$ in the Lie algebra $se(3)$ space:

$$\mathbf{e} = \text{Log}\left({}^B T_G^{-1} {}^B T_E\right) = \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_r \end{bmatrix} \in \mathbb{R}^6 \quad (4)$$

where $\mathbf{e}_p$ is the translational error and $\mathbf{e}_r$ is the orientation error represented as a rotation vector.

The control objective is defined as the end-effector pose increment rather than direct joint angle regulation. At each control cycle t , the control strategy outputs a pose correction $\mathbf{a}_t$ based on the current error state:

$$\mathbf{a}_t = \begin{bmatrix} \Delta \mathbf{p}_t \\ \Delta \boldsymbol{\omega}_t \end{bmatrix} \in \mathbb{R}^6 \quad (5)$$

where $\Delta \mathbf{p}_t$ represents the Cartesian-space translational increment and $\Delta \boldsymbol{\omega}_t$ represents the rotation vector increment. This increment is applied to the end-effector reference target, driving the manipulator through inverse kinematics or dynamic tracking mechanisms to

progressively eliminate the error e , ultimately converging to the target pose under the premise of satisfying geometric constraints.

2.4. MPC Controller

In greenhouse tomato harvesting, the end-effector must reach the target precisely while completing cutting with specific spatial pose to avoid collisions in confined spaces. Traditional PBVS methods lack trajectory foresight, making them prone to infeasible solutions under conflicting constraints.

This work employs Model Predictive Control (MPC) to construct the visual servoing controller. MPC optimizes future control sequences over a receding horizon, explicitly incorporating pose constraints, approach restrictions, collision avoidance, and joint limits into a unified optimization problem, generating safe and feasible trajectories under complex constraints.

In the proposed MPC-guided RL framework, the MPC controller serves as a “teacher”: its collision-free expert trajectories provide feasible region priors for subsequent RL policies, enabling the agent to inherit geometric constraint handling while improving generalization through interactive learning.

2.4.1. Discrete Dynamics and Decision Variables

The system state is defined as the joint angle vector $\mathbf{q}_k \in \mathbb{R}^n (n = 6)$, and the control input as the joint angular velocity $\mathbf{u}_k = \dot{\mathbf{q}}_k \in \mathbb{R}^n$. With a fixed control period $\Delta t = 0.05$ s, the discrete dynamics over prediction horizon N is:

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{u}_k \Delta t, k = 0, \dots, N - 1 \quad (6)$$

The end-effector pose $T_E(\mathbf{q}_k) \in \text{SE}(3)$ is derived through forward kinematics:

$$T_E(q_k) = \begin{bmatrix} \mathbf{R}_E & \mathbf{p}_E \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (7)$$

where $\mathbf{R}_E \in \text{SO}(3)$ is the end-effector orientation rotation matrix, and $\mathbf{p}_E \in \mathbb{R}^3$ is the position vector.

2.4.2. Objective Function

The MPC controller minimizes cumulative pose error while regularizing control input for smoothness and efficiency. The pose error $\mathbf{e}_k \in \mathbb{R}^6$ is defined in Lie algebra $\text{se}(3)$ space:

$$\mathbf{e}_k = \begin{bmatrix} \mathbf{e}_k^p \\ \mathbf{e}_k^r \end{bmatrix} = \begin{bmatrix} \mathbf{p}_G - \mathbf{p}_E(\mathbf{q}_k) \\ \text{Log}(\mathbf{R}_G^T \mathbf{R}_E(\mathbf{q}_k)) \end{bmatrix} \quad (8)$$

where $\mathbf{e}_k^p \in \mathbb{R}^n$ is position vector, $\mathbf{e}_k^r \in \mathbb{R}^n$ is orientation error, $\text{Log}(\cdot) : \text{SO}(3) \rightarrow \text{so}(3)$ is the logarithmic map, and $T_G = [\mathbf{R}_G, \mathbf{p}_G]$ is the target pose.

The optimization problem is:

$$\min_{\mathbf{u}_{0:N-1}} J = \sum_{k=0}^{N-1} \left(\mathbf{e}_k^T \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + \Delta \mathbf{u}_k^T \mathbf{S} \Delta \mathbf{u}_k \right) + \mathbf{e}_N^T \mathbf{Q}_f \mathbf{e}_N \quad (9)$$

where $\mathbf{Q}$ weights pose tracking, $\mathbf{R}$ penalizes control energy, $\mathbf{S}$ enforces smoothness ($\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$), and $\mathbf{Q}_f \gg \mathbf{Q}$ accelerates terminal convergence.

2.4.3. Constraint Set

The MPC framework explicitly handles multi-level constraints for safe operation in narrow, cluttered environments (Figure 4).



Figure 4. Coordinate system and constraints. (a) Tool frame and orientation constraint; (b) cone constraint for approach path.

(1) Approach Cone Constraint

The end-effector's approach direction must lie within a conical region centered on the optimal picking plane normal $\mathbf{n}_{\text{plane}}$. Let the current end-effector position be $\mathbf{p}_E^k$ and the target picking point position be $\mathbf{p}_G$. The approach vector is defined as:

$$\mathbf{d}^k = \mathbf{p}_G - \mathbf{p}_E^k \quad (10)$$

The mathematical expression for the cone constraint is:

$$\frac{\mathbf{d}^k \cdot \mathbf{n}_{\text{plane}}}{\|\mathbf{d}^k\| \|\mathbf{n}_{\text{plane}}\|} \geq \cos(\theta_{\text{cone}}) \quad (11)$$

where θ_{cone} is the cone half-angle. This ensures approach from the front of fruit growth direction, minimizing collision risk with main stem and neighboring fruits.

(2) Orientation Alignment Constraint

The end-effector y_E axis (cutting plane normal) must align with the peduncle axis $\mathbf{v}_{\text{pick}}$. Let $\mathbf{y}_E^N = \mathbf{R}_{E_y}^k \mathbf{e}_y$ be the y_E axis in base frame. Terminal constraint:

$$|\mathbf{y}_E^N \cdot \mathbf{v}_{\text{pick}}| \geq \cos(\alpha_{\text{tol}}) \quad (12)$$

where α_{tol} is the allowable orientation deviation, ensuring the cutting plane remains perpendicular to the peduncle.

(3) Collision Avoidance Constraints

Based on the tomato geometric prior model constructed in Section 2.2, the main stem is modeled as a cylinder, fruits as capsules. Let the robot capsule set be $\mathcal{C}_{\text{robot}} = \{C_1^{\text{robot}}, C_2^{\text{robot}}, \dots\}$ and the environmental obstacle capsule set be $\mathcal{C}_{\text{obs}} = \{C_1^{\text{stem}}, C_1^{\text{fruit}}, C_2^{\text{fruit}}, \dots\}$. The collision avoidance constraint is expressed as:

$$\text{dist}(C_i^{\text{robot}}, C_j^{\text{obs}}) \geq d_{\text{safe}}, \forall C_i^{\text{robot}} \in \mathcal{C}_{\text{robot}}, \forall C_j^{\text{obs}} \in \mathcal{C}_{\text{obs}} \quad (13)$$

where $d_{\text{safe}} = 5 \text{ mm}$ (accounting for depth sensing noise and model uncertainty).

(4) Joint Feasibility Constraints

To ensure control command executability and avoid joint saturation, joint angle and joint velocity constraints are imposed:

$$\mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}, \dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}} \leq \dot{\mathbf{q}}_{\max} \quad (14)$$

$$\dot{\mathbf{q}}_{\min} \leq \frac{\mathbf{q}^{k+1} - \mathbf{q}^k}{\Delta t} \leq \dot{\mathbf{q}}_{\max} \quad (15)$$

where limits are extracted from the xMate SR5 manipulator's URDF file.

(5) Field-of-View (FOV) Constraint

To ensure the target remains within the eye-in-hand camera's effective range, let $p_{G,z}^C$ be target coordinates in camera frame:

$$p_{G,z}^C \geq z_{\min} \quad (16)$$

$$\left| \arctan\left(\frac{p_{G,x}^C}{p_{G,z}^C}\right) \right| \leq \frac{\text{FOV}_x}{2}, \left| \arctan\left(\frac{p_{G,y}^C}{p_{G,z}^C}\right) \right| \leq \frac{\text{FOV}_y}{2} \quad (17)$$

where $z_{\min} = 70$ mm (minimum working depth of RealSense D405), $\text{FOV}_x = 87^\circ$, $\text{FOV}_y = 58^\circ$ (based on camera specifications).

2.4.4. CasADi Implementation and Constraint Handling Strategy

The MPC problem is formulated using CasADi and solved with IPOPT interior-point solver. A warm-start strategy (shifting previous optimal trajectory as initial guess) reduces solver iterations and improves convergence. Constraints are divided into two priority levels: 1. Hard Constraints (Hard Constraints, Non-Violable): Collision avoidance, Joint limits. 2. Soft Constraints (Soft Constraints): Approach cone, Orientation alignment, FOV.

For soft constraints, when infeasibility occurs, slack variables $\epsilon = [\epsilon_{\text{cone}}, \epsilon_{\text{orient}}, \epsilon_{\text{fov}}]^T$ introduced with penalty terms:

$$J_{\text{slack}} = \sum_i w_i e_i^2 \quad (18)$$

where w_i is the penalty weight coefficient (typically set on the order of 10^3 – 10^4). This strategy enables the solver to dynamically balance constraint satisfaction and control performance while ensuring safety, thereby seeking optimal feasible solutions in complex constrained environments.

Table 2 summarizes the key implementation parameters for the MPC controller. Through this integrated constraint hierarchy, the MPC controller generates expert trajectories satisfying geometric safety, kinematic feasibility, and task accuracy requirements, serving as the foundation for behavior cloning in the RL framework (Section 2.5).

Table 2. The key implementation parameters for the MPC controller.

Parameters	Values
Δt	0.05
w_p	100
w_r	10
w_u	0.01
w_a	0.1
a	0.524 rad
θ_{tol}	0.0875 rad
$d_{\min}$	0.03 m

2.5. RL Design

2.5.1. MDP Formulation

We formulate the tomato harvesting visual servoing task as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

1. State Space ($\mathcal{S}$)

The state vector s_t is designed to provide a compact description of task deviation and system status for the policy network. To avoid singularities associated with Euler angles and maintain input continuity, we parameterize pose errors using Lie algebra representations:

$$\mathbf{s}_t = [\mathbf{e}_p^T \mathbf{e}_r^T \mathbf{q}_t^T \dot{\mathbf{q}}_t^T d_t c_t]^T \in \mathbb{R}^{20} \quad (19)$$

Position error ($\mathbf{e}_p \in \mathbb{R}^3$): The translational deviation of the end-effector relative to the target point.

Orientation error ($\mathbf{e}_r \in \mathfrak{so}(3)$): Let $\mathbf{R}_{\text{current}} \in \text{SO}(3)$ denote the current end-effector rotation and $\mathbf{R}_{\text{target}} \in \text{SO}(3)$ the target rotation. The error is defined as:

$$\mathbf{e}_r = \text{Log}(\mathbf{R}_g^T \mathbf{R}_e) \quad (20)$$

where $\text{Log}(\cdot) : \text{SO}(3) \rightarrow \mathfrak{so}(3)$ is the logarithmic map. The direction of this vector represents the error rotation axis, while its magnitude represents the error angle. Compared to quaternion-based error representations, this formulation exhibits superior linearity in the vicinity of zero error, facilitating neural network learning.

Proprioceptive state ($\mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^6$): Joint angles and angular velocities, respectively.

Distance scalar ($d \in \mathbb{R}^+$, $c \in \{0, 1\}$): d is the distance scalar and c is the collision indicator.

2. Action Space ($\mathcal{A}$)

To accommodate the unbounded real-valued outputs characteristic of neural networks, we define the action as a spatial velocity increment in the tool coordinate frame of the end-effector, i.e., an element of the Lie algebra $\mathfrak{se}(3)$:

$$\mathbf{a}_t = [\mathbf{v}_t^T \ \boldsymbol{\omega}_t^T]^T \in \mathbb{R}^6 \quad (21)$$

where $\mathbf{v}_t \in \mathbb{R}^3$ is the translational velocity, and $\boldsymbol{\omega}_t \in \mathbb{R}^3$ is the rotational velocity.

3. State Transition Mechanism

Although the action space is defined in the tangent space, pose updates in the simulation environment must be performed on the manifold to preserve geometric constraints (such as rotation orthogonality). Upon receiving action $\mathbf{a}_t$, the environment converts it into a rigid body transformation increment via the exponential map and updates the Mocap target pose in MuJoCo:

(1) Position update:

$$\mathbf{a}_t = [\mathbf{v}_t^T \ \boldsymbol{\omega}_t^T]^T \in \mathbb{R}^6 \quad (22)$$

(2) Orientation update:

Using quaternion exponential mapping, the rotational increment $\boldsymbol{\omega}_t$ is converted to a unit quaternion:

$$\Delta \mathbf{Q}_t = \exp(\boldsymbol{\omega}_t \Delta t) \quad (23)$$

(3) Subsequently, the orientation is updated via quaternion multiplication:

$$\mathbf{Q}_{t+1} = \Delta \mathbf{Q}_t \otimes \mathbf{Q}_t \quad (24)$$

This process ensures that the updated orientation $\mathbf{Q}_{t+1}$ always resides on the unit quaternion sphere, effectively preventing numerical drift that would arise from direct integration of orientation parameters.

2.5.2. Network Architecture

To achieve high-frequency stable control in continuous action spaces, we adopt an Actor–Critic architecture for joint parameterization of the policy and value functions. This asymmetric structure allows the value network (Critic) to assist optimization during training, while only the policy network (Actor) is retained for lightweight inference during deployment. The PPO network architecture is shown in Figure 5.

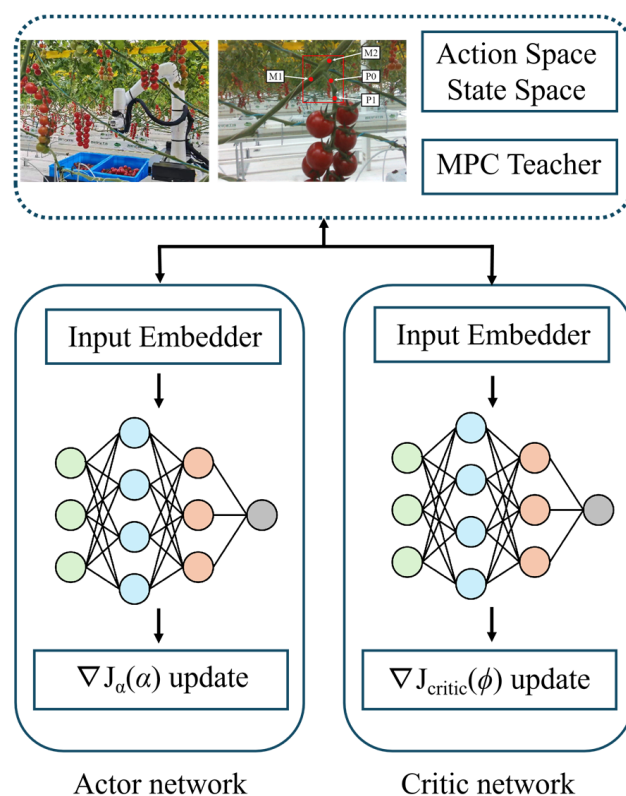


Figure 5. The PPO network architecture.

Actor (Policy Network): The Actor represents the stochastic policy $\pi_{\phi}(a_t | s_t)$. For continuous action spaces, we model the policy as a diagonal Gaussian distribution:

$$\pi_{\phi}(\mathbf{a}_t | \mathbf{s}_t) = \mathcal{N}\left(\mu_{\phi}(\mathbf{s}_t), \text{diag}\left(\sigma_{\phi}^2\right)\right) \quad (25)$$

where $\mu_{\theta}(s_t)$ is the policy mean predicted by the neural network based on state input, and σ_{θ} is the standard deviation (which can be either state-independent learnable parameters or state-dependent outputs from the network). During training, actions a_t are sampled from this distribution to maintain necessary stochasticity and promote exploration. During deployment, to achieve higher execution consistency and control smoothness, a deterministic execution mode is typically adopted, directly using the mean action $\mu_{\theta}(s_t)$ (or its boundary-mapped form) as the control output.

Critic (Value Network): The Critic estimates the state value function $V_\psi(\mathbf{s}_t)$, representing the expected discounted return starting from the current state:

$$V_\psi(\mathbf{s}_t) = \mathbb{E} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid \mathbf{s}_t \right] \quad (26)$$

In PPO, the Critic's value estimates are primarily used to construct the advantage function, which is computed as the immediate reward plus the discounted value of the next state minus the estimated value of the current state, thereby providing low-variance update directions for policy gradients and significantly improving training stability and sample efficiency.

2.5.3. MPC-Driven Behavior Cloning

Given the characteristics of narrow environments and severe occlusions in close-range tomato picking visual servoing tasks, randomly initialized policies in continuous action spaces are prone to ineffective exploration and may even cause collisions between the robotic arm and branches or fruits, resulting in low training sample efficiency and unstable convergence. To address this cold start problem, we introduce a supervised learning phase that leverages expert trajectories generated by the NMPC teacher to “warm-start” the policy network.

1. Expert Dataset Construction

We execute the MPC expert controller in the simulation environment across multiple episodes with randomly generated picking scenarios, recording trajectory data from successfully completed tasks. At each time step t , we collect the environment observation state $\mathbf{s}_t$, map the MPC-generated optimal joint increment to the end-effector increment $\mathbf{a}_t^{\text{mpc}}$ via the manipulator's geometric Jacobian, and simultaneously record the changes in the six joints for updating the reward function, to construct the expert dataset:

$$\mathcal{D}_{\text{expert}} = \left\{ \left(\mathbf{s}_t, \mathbf{a}_t^{\text{mpc}} \right) \right\}_{t=1}^N \quad (27)$$

A total of 200 expert trajectories were collected, comprising approximately 40,000 state-action pairs, which capture the teacher model's behavioral patterns when handling cone approach and collision avoidance constraints.

2. Pretraining (Critic-First Strategy)

To mitigate policy update oscillations in the early training phase of the Actor–Critic architecture caused by value estimation bias, we adopt a “Critic-first” pretraining strategy:

- (1) Critic Pretraining: Calculate discounted returns on expert trajectories according to the task reward function to construct supervision signals, where T denotes the episode horizon:

$$G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}, \gamma = 0.99 \quad (28)$$

The value function is then regressed using mean squared error:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(\mathbf{s}_t, \cdot) \sim \mathcal{D}_{\text{expert}}} \left[(V_\psi(\mathbf{s}_t) - G_t)^2 \right] \quad (29)$$

During this phase, Actor parameters are frozen while only the Critic is updated, allowing the value function to first converge on the “expert distribution,” thereby providing low-variance advantage estimates for subsequent PPO.

(2) Actor Behavior Cloning (Supervised Warm-Start)

After completing Critic pretraining, we employ behavior cloning to perform supervised regression on the Actor's mean output. Let the Actor output mean action be $\mu_\theta(s_t)$. The BC loss is defined as:

$$\mathcal{L}_{BC}(\phi) = \mathbb{E}_{(s_t, \mathbf{a}_t^{\text{mpc}}) \sim \mathcal{D}_{\text{expert}}} \left[\left\| \mu_\phi(s_t) - \mathbf{a}_t^{\text{mpc}} \right\|_2^2 \right] \quad (30)$$

This training enables the policy to generate end-effector increment commands close to those of MPC in the initial stage, significantly reducing collision probability and invalid sampling proportion caused by random exploration. This process equips the agent with prior policies for “safe approach” and “basic collision avoidance,” denoted as π_{θ_0} , before entering the reinforcement learning phase.

3. PPO Fine-tuning with Adaptive Mechanisms

Although BC can imitate expert trajectories, it suffers from the covariate shift problem. The second phase introduces the PPO algorithm, allowing the agent to interact with the environment and focusing on addressing “pose refinement” and “dynamic environment robustness.” We design two key mechanisms to enhance the algorithm's adaptability in tomato picking scenarios.

(1) Dynamic Gain Module: Staged Translation Step Size and Stagnation-Triggered Exploration Enhancement

The visual servoing process for tomato peduncles cutting exhibits distinct phases: in the long-range phase, the primary objective is to rapidly reduce the spatial distance between the end-effector and the peduncle; upon entering the peduncle's neighborhood, the system should prioritize completing cutting angle/pose alignment, then approach smoothly with small step sizes to avoid branch-leaf collisions and close-range oscillations. If the policy employs a fixed step size throughout, two failure modes readily emerge:

- a. Insufficient long-range advancement, leading to low approach efficiency;
- b. Continued large translational step sizes in the close-range phase, causing overshoot, oscillation, or collision, preventing stable pose alignment.

We propose an adaptive gain scheduling mechanism that acts solely on translational degrees of freedom. Let the policy's raw action be $\mathbf{u}_t^p$. The actual executed translational increment $\Delta \mathbf{p}_t$ is defined as:

$$\Delta \mathbf{p}_t = g_p(d_t) \cdot \lambda_{\text{pos}} \cdot \mathbf{u}_t^p \quad (31)$$

where the gain coefficient $g_p(d_t)$ follows a sigmoid decay law as a function of distance d_t :

$$g_p(d_t) = g_{\min} + (g_{\max} - g_{\min}) \cdot \sigma\left(\frac{d_t - \phi}{\kappa}\right) \quad (32)$$

where the neighborhood threshold is set to $\phi = 150$ mm. When $d_t \gg \phi$, the gain approaches $g_{\max} = 1.0$ for full-speed advancement; when entering the close-range operation zone $d_t < \phi$, the gain automatically shrinks to $g_{\min} = 0.2$. This nonlinear regulation strategy endows the system with dexterity similar to human operation, significantly reducing steady-state error and dynamic overshoot upon reaching the target position while ensuring real-time performance.

(2) Stagnation Detection and Noise Enhancement (Stagnation-Induced Exploration)

In close-range confined spaces, the policy may become trapped in local minima (i.e., oscillating slightly near the target without error reduction). We introduce a stagnation detection mechanism based on a sliding window ($N = 10$ steps): if improvements in

both distance and orientation errors simultaneously fall below thresholds $(\epsilon_d, \epsilon_\theta)$, stagnation is declared. Once stagnation is triggered, the algorithm immediately amplifies the standard deviation σ of the policy distribution to enhance random exploration and break the deadlock:

$$\sigma \leftarrow \eta_\sigma \cdot \sigma, \eta_\sigma > 1 \quad (33)$$

where η_σ is the noise enhancement coefficient. Subsequently, σ gradually returns to normal levels following an exponential decay law. This mechanism grants the agent the ability to “fine-tune and escape” in narrow spaces, avoiding instability caused by prolonged high noise while effectively resolving local convergence issues.

2.5.4. Reward Function

Given the specific requirements of the tomato picking task—which must simultaneously balance collision avoidance, precise pose alignment, and smooth control—we design a hybrid reward function comprising four core components.

In the visual servoing process for tomato picking, the timing of end-effector pose regulation significantly impacts overall motion efficiency and operational safety. If strict pose constraints are imposed too early during the long-range approach phase, the feasible solution space for inverse kinematics will be severely compressed, limiting end-effector movement speed and potentially causing path planning deadlock or singularity issues, which seriously compromises operational efficiency. Conversely, if pose adjustment is excessively delayed to the close-range phase, large-scale end-effector reorientation within dense foliage will generate substantial swept volumes, readily causing collisions between the end-effector and the peduncle, stem, or neighboring fruits, thereby reducing system safety.

The overall reward function is defined as:

$$R_t = r_{dist} + r_{pose} + r_{smooth} + r_{collision} \quad (34)$$

1. Target Approach Reward (r_{dist})

To balance training stability in the far field with positioning accuracy in the near field, we adopt a Huber loss-like piecewise reward function. This function applies a linear penalty at long distances to prevent gradient explosion and transitions to a quadratic penalty within a threshold φ to ensure smooth convergence and avoid chattering. Setting the distance threshold $\varphi = 150$ mm:

$$r_{goal} = \begin{cases} -\frac{1}{2}d_g^2 & d_g \leq \varphi \\ -\varphi\left(|d_g| - \frac{1}{2}\varphi\right) & d_g > \varphi \end{cases} \quad (35)$$

When distance $d_g > \varphi$ is large: a linear penalty is applied. At this stage, the reward function gradient remains constant, providing continuous and stable driving signals that avoid gradient explosion at long distances while efficiently driving the robotic arm toward the target neighborhood.

When distance $d_g \leq \varphi$ is small: the reward function smoothly transitions to a quadratic penalty. Leveraging the characteristic that its gradient decays linearly with error reduction, this implements a “soft constraint” on control inputs, focusing on suppressing overshoot and high-frequency chattering near the zero point to ensure the robotic arm converges smoothly to the operational position.

2. Orientation Alignment Reward (r_{pose})

This reward r_{pose} is triggered only when the end-effector enters the target neighborhood ($d_g \leq \varphi$), aiming to guide the robotic arm to approach the peduncle with the correct

pose. Let v_{pick} denote the picking vector of the optimal cutting plane obtained by the visual perception module, representing the best end-effector approach direction that meets agronomic requirements. When the robotic arm end-effector is within the target picking zone, we calculate the cosine similarity between the unit vector v_y^e of the tool coordinate frame Y-axis (the gripper plane normal direction) and the target vector. The smaller the angular deviation between them, the lower the penalty, thereby guiding the end-effector to autonomously adjust its pose during approach until precise alignment with the target peduncle is achieved.

To ensure safety and kinematic feasibility in close-range operations, we introduce a discrete region constraint on top of the continuous orientation reward. If the end-effector meets the distance requirement but has not yet entered the predefined valid picking region T (positioned correctly but with severe pose deviation or collision risk), the system imposes a fixed penalty to force the policy network to correct the path and enter the feasible region. The mathematical expression is:

$$r_{pose} = -\mathbb{I}[d_g \leq \varphi \wedge \mathbf{p}_e \in \mathcal{T}] \left(\arccos \left(\frac{(\mathbf{v}_y^e)^\top \mathbf{v}_{pick}}{\|\mathbf{v}_y^e\| \|\mathbf{v}_{pick}\|} \right) \right)^2 - 10\mathbb{I}[d_g \leq \varphi \wedge \mathbf{p}_e \notin \mathcal{T}] \quad (36)$$

where $\mathbb{I}[\cdot]$ is an indicator function that equals 1 when the condition in brackets is satisfied and 0 otherwise; d_s denotes the Euclidean distance between the end-effector and the target picking point; φ denotes the neighborhood threshold defining the close-range operation zone (consistent with the threshold defined in Section 2.5.3); $\mathbf{p}_e$ denotes the current position of the end-effector; T denotes the predefined valid picking region, within which the end-effector pose satisfies both distance and orientation feasibility requirements; $\mathbf{v}_y^e$ denotes the unit vector along the y -axis of the end-effector tool frame (i.e., the cutting plane normal); $\mathbf{v}_{pick}$ denotes the optimal picking direction vector of the peduncle obtained from the visual perception module; and $\|\cdot\|$ denotes the Euclidean norm.

The first term applies a continuous quadratic penalty proportional to the squared angular deviation between the end-effector approach axis and the target picking direction, activated only when the end-effector has entered the neighborhood ($d_s \leq \varphi$) and is within the valid picking region ($\mathbf{p}_e \in T$). The second term imposes a fixed penalty of -10 when the end-effector is within the neighborhood but outside the valid region, encouraging the policy to re-enter the feasible approach corridor.

3. Smoothness Reward (r_{smooth})

To avoid high-frequency control signals that cause robotic arm jitter and wear, we penalize joint angle variations:

$$r_{smooth} = -k_s \sum_{j=1}^n \|q_t^{(j)} - q_{t-1}^{(j)}\|^2 \quad (37)$$

where $q_t^{(j)}$ represents the angle of the j -th joint at time t . Larger variations result in larger negative rewards, thereby encouraging the policy network to output smooth action sequences.

4. Collision Avoidance Reward ($r_{collision}$)

Given that the BC phase has already endowed the agent with basic collision avoidance capability, the RL phase employs sparse penalties as a safety baseline:

$$r_{collision} = \begin{cases} -100 & \text{if Collision} = \text{True} \\ 0 & \text{otherwise} \end{cases} \quad (38)$$

If a collision is detected between the end-effector (capsule geometry) and obstacles, an extremely large negative reward is immediately applied, and the current episode is terminated.

2.6. Experimental Setup

We evaluate the MPC-PPO framework through high-fidelity simulations and real-world greenhouse experiments. All implementations run on Ubuntu 22.04 with Python 3.10 and PyTorch 2.7.0,

using an Intel Xeon Platinum 8352 V CPU (Intel Corporation, Santa Clara, CA, USA), 64 GB RAM, and NVIDIA RTX 4090 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

Simulation Environment: A MuJoCo-based platform comprises three components: a base platform, the ROKAE xMate SR5 manipulator (Section 2.1) with a custom gripper-cutter, and tomato plants reconstructed from geometric priors and keypoint positions (Section 2.2). Tomato targets are randomly placed in a $400 \times 400 \times 200$ mm³ workspace centered at (460, 0, 720) mm in the base frame, with Cartesian step size ± 50 mm, as shown in Figure 6.

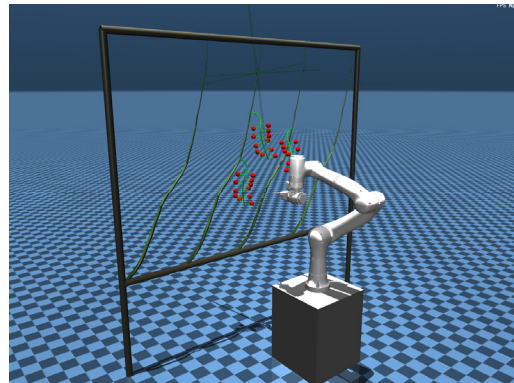


Figure 6. Simulation Environment.

Real-World Validation: Field experiments use the cherry tomato harvesting robot (Figure 1) to assess Sim-to-Real transfer. The mobile platform integrates recognition, picking, collection, and transport capabilities for autonomous greenhouse operation.

3. Results

3.1. Experimental Setup and Evaluation Metrics

We evaluated the proposed MPC-Guided RL framework through extensive experiments in high-fidelity simulation and on real robot platforms, focusing on control accuracy, safety, convergence, and real-time performance in densely constrained harvesting scenarios.

All methods were tested under identical perception inputs and task settings with randomized episode initialization. For each episode, tomato position, peduncle pose, and stem occlusion patterns were randomized, with the end-effector initialized at random poses within the camera field of view. Episodes were terminated upon (1) successful reach of harvestable configuration, or (2) collision with stem or obstacles.

- (1) **Success Rate:** Proportion of collision-free trials reaching the target with valid approach pose. Success requires:
 - (i) The Euclidean distance between the origin of the end-effector reference coordinate system and the picking point is less than 0.03 m.
 - (ii) The angle between the Y-axis of the end-effector and the optimal posture plane is less than 0.05 rad.
 - (iii) collision-free execution.
- (2) **Final Position Error (mm):** End-effector-to-target Euclidean distance at termination.
- (3) **Final Orientation Error (rad):** Angular deviation between approach axis and desired peduncle direction.
- (4) **Average Steps:** The average number of control steps required to complete a successful harvesting attempt, reflecting temporal efficiency.

3.2. Comparative Experiment

We compared the proposed framework against three representative baselines: (1) Traditional PBVS (proportional control-based classical PBVS), (2) NMPC-PBVS (nonlinear MPC with explicit geometric and joint constraints requiring online optimization), (3) Standard PPO and (4) Standard PPO with DGM (dynamic gain mechanisms).

Results from 200 randomized episodes (Table 3) show MPC-Guided PPO achieves optimal overall performance. Despite a slightly higher positioning error than NMPC (13.6 mm vs. 12.1 mm), it attains 95% success rate, significantly exceeding Traditional PBVS (74%) and Standard PPO (89%), demonstrating successful transfer of constraint-handling capability via behavior cloning.

Table 3. The results from 200 randomized episodes.

Method	Avg. $e + \sigma_e$	Avg. $\theta_e + \sigma_{\theta}$	Success Rate	Average Steps
Traditional PBVS	28.3 ± 13.4	0.074 ± 0.031	0.74	39.1
NMPC-PBVS	12.1 ± 6.1	0.013 ± 0.05	0.96	91.7
PPO	17.6 ± 7.8	0.024 ± 0.11	0.89	39.4
PPO+DGM	16.6 ± 7.4	0.021 ± 0.09	0.92	36.6
MPC-Guided PPO	13.6 ± 6.2	0.009 ± 0.04	0.95	42.7

The Dynamic Gain Module (DGM) substantially improves PPO performance during fine manipulation. DGM reduces positioning error from 17.6 mm to 16.6 mm, increases success rate from 89% to 92%, and counter-intuitively decreases average steps from 39.4 to 36.6. This confirms DGM resolves steady-state oscillation and overshoot issues in near-field regions through adaptive gain decay, enhancing convergence precision without sacrificing exploration efficiency.

Both MPC-Guided PPO and NMPC maintain low orientation errors (0.009 rad and 0.013 rad), ensuring strict end-effector alignment in unstructured environments. Traditional PBVS, lacking explicit constraints, achieves only 74% success rate.

While NMPC achieves highest precision (12.1 mm, 96% success), it requires 91.7 steps versus MPC-Guided PPO’s 42.7 steps—a 53.4% efficiency gain. This demonstrates effective “distillation” of online optimization into lightweight neural policies, enhancing real-time performance while preserving safety.

3.3. Training Efficiency and Convergence Characteristics

Figure 7 illustrates the evolution of average episode return for the proposed method (MPC-Guided PPO, blue curve) and the baseline approach (PPO Baseline, red curve) over 2500 training episodes. By examining the trajectory characteristics of both curves, the significant reshaping effect of expert prior knowledge on the reinforcement learning process can be clearly observed.



Figure 7. Training convergence comparison over 2500 episodes. MPC-Guided PPO (blue) benefits from behavior cloning warm-start, achieving stable high performance around Episodes 500–600. PPO Baseline (red) experiences prolonged oscillations, converging around Episode 2000. Shaded regions indicate standard deviation.

Benefiting from the behavior cloning-based warm-start, MPC-Guided PPO exhibits an initial return stably around -10 , completely circumventing the extremely low returns (approximately -50) and high-variance blind exploration dilemma faced by PPO Baseline during the early stage (Episodes 0–200). From the perspective of curve gradient, MPC-Guided PPO demonstrates a smooth and rapid monotonic ascent, entering a high-reward plateau region around Episodes 500–600, indicating that the policy has rapidly taken shape. In contrast, although PPO Baseline experiences a brief period of rapid improvement during the Episode 200–300 interval, it subsequently falls into prolonged severe oscillations and adjustments, converging to near-optimal performance levels only around Episode 2000.

Integrating training time statistics, the proposed method not only reduces the number of episodes required to reach convergence but also shortens the actual training time by 52.4% (596 s vs. 1253 s). This confirms that introducing behavior cloning constraints the action space, significantly enhancing sample utilization efficiency.

3.4. Field Experiments

To validate the engineering feasibility of the proposed algorithm, we deployed the trained policy network on a six-DOF tomato harvesting robot equipped with an eye-in-hand RGB-D camera, as shown in Figure 8. The lightweight policy network requires only a single forward pass per control cycle, achieving a stable 20 Hz control frequency on the industrial computing platform.



Figure 8. Field experiments.

Field results demonstrate excellent sim-to-real transferability. Despite lighting fluctuations and foliage occlusion, the controller reliably guides the end-effector to avoid the main stem and neighboring fruits while reaching the target peduncle with correct cutting pose. The system achieves 85.3% harvesting success rate with an average operation time of 5.63 s per fruit. Compared to traditional open-loop methods, the closed-loop visual feedback improves success rate by 35%, confirming robustness in complex unstructured agricultural scenarios.

Figure 9a compares end-effector Cartesian trajectories generated by the MPC teacher (blue) and MPC-Guided PPO (orange) in a challenging scenario. Unlike standard RL policies that produce jittery paths, MPC-Guided PPO generates highly smooth trajectories with continuous curvature changes, benefiting from the jerk penalty term and successfully internalizing the MPC teacher's smoothness prior—crucial for reducing wear and preventing fruit damage. Convergence curves show faster error reduction and significantly lower cross-episode variance compared to baseline PPO, indicating superior consistency and reliability.

Figure 9b,c show end-effector pose evolution (coordinate frames along trajectory) for MPC teacher and learned policy, respectively. The learned policy achieves synchronized position-orientation coupling rather than sequential adjustment. As the end-effector approaches the target, the frame rotates smoothly, ensuring the cutting plane (Y-axis) becomes precisely perpendicular to the peduncle upon arrival. This complex six-DOF coordination demonstrates capability to handle strong geometric constraints beyond simple position servoing.

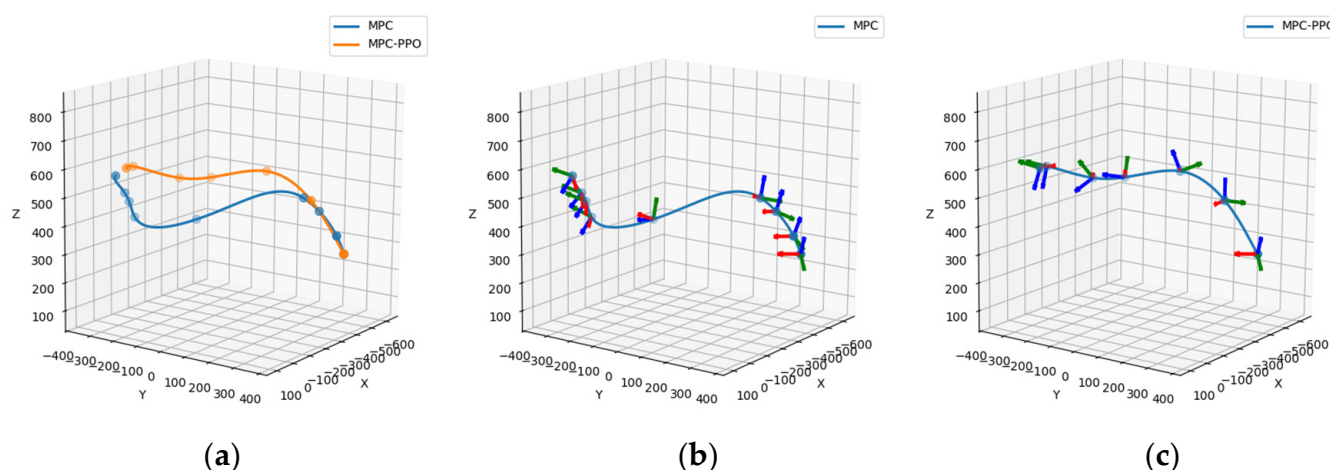


Figure 9. Field trajectory analysis. (a) Cartesian trajectories: MPC teacher (blue) vs. MPC-Guided PPO (orange) showing smooth obstacle avoidance. (b,c) End-effector pose evolution demonstrating synchronized position-orientation coupling.

The arc-shaped trajectory (rather than straight-line) indicates the agent learned active three-dimensional obstacle avoidance (main stem and neighboring fruits), highly consistent with the NMPC solver's "safe corridor" under hard constraints. Although MPC-PPO avoids online optimization, it replicates equivalent safe decision-making through neural network nonlinear mapping.

3.5. Failure Analysis

Figure 10 illustrates several representative failure cases. The primary causes of failure can be categorized as follows:



Figure 10. Several representative failure cases.

1. Depth Perception Noise and Keypoint Jitter:

The stability of visual model outputs directly determines the upper performance bound of servo control. In certain failure cases, although YOLOv8-Pose successfully detected the peduncle keypoints, the slender geometry of the peduncle, combined with challenging illumination conditions or limited viewing angles (as shown in Figure 10), induced fluctuations in depth measurements. These depth inconsistencies resulted in abrupt jumps in the estimated target pose. Such high-frequency perception noise propagates into the state space s_t , causing the policy network to misinterpret the current error magnitude e_{pos} . Although we incorporated smoothness reward terms r_{smooth} during training to mitigate this issue, under extreme noise conditions, the end-effector may still exhibit micro-scale oscillations. Consequently, at the instant of gripper closure, the end-effector deviates from the optimal cutting point, resulting in harvest failure.

2. Severe Occlusion in Dynamic Environments:

Occlusion represents the dominant external factor leading to task termination, particularly in the eye-in-hand configuration. As the camera approaches the target, the field of view (FOV) progressively narrows, drastically reducing contextual environmental information. In densely clustered fruit scenarios, when the manipulator executes the cutting motion, surrounding leaves or non-target fruits may inadvertently enter the camera frame and occlude the target peduncle. Once critical visual features are lost or erroneously associated with nearby obstacles, the PBVS system loses its feedback signal, triggering either safety-stop mechanisms or incorrect obstacle avoidance maneuvers.

3. Kinematic Singularities and Unreachable Configurations:

Beyond visual factors, a subset of failures stems from target spatial distributions that exceed the manipulator's dexterous workspace. When target fruits are located behind the main stem or deep within foliage, satisfying the strict "perpendicular approach" pose constraint C_{pose} may force the manipulator toward singular configurations. Under such circumstances, even when the policy network generates reasonable Cartesian-space increments, the underlying inverse kinematics (IK) solver struggles to execute the corresponding joint motions, resulting in task timeout or deadlock.

4. Conclusions

This paper presents a visual servoing framework combining Nonlinear Model Predictive Control (NMPC) with Proximal Policy Optimization (PPO) for precise tomato harvesting in unstructured greenhouse environments. Using a teacher–student paradigm, we warm-start PPO with NMPC expert demonstrations via behavioral cloning, avoiding the blind exploration and convergence instabilities inherent to model-free RL in complex 6-DOF tasks. Expert priors improve sample efficiency, reducing training time by 52.4% versus Standard PPO while maintaining safe exploration. The framework achieves 13.6 mm steady-state positioning accuracy and 95% simulation success rate—28% better than conventional PBVS—effectively handling obstacle avoidance and pose alignment in cluttered scenes. A composite reward penalizing joint velocities and end-effector increments yields smooth, kinematically feasible trajectories, enhancing stability and hardware durability. Field trials validate sim-to-real transfer and robustness under illumination changes and variable target poses, achieving 85.3% success with 5.63 s average operation time. Despite the aforementioned progress, the performance ceiling of the current system remains constrained by depth perception noise from the RGB-D camera and severe occlusion issues. Future work will focus on integrating active viewpoint selection mechanisms to address these occlusion challenges.

Author Contributions: Conceptualization, L.G., Q.F. and C.Z.; methodology, L.G., S.C. and F.F.; software, L.G.; validation, L.G., Q.F., L.C. and C.Z.; formal analysis, L.G.; investigation, L.G., Q.F., S.C., Z.Y. and F.F.; resources, L.G., S.C. and L.C.; data curation, L.G. and Q.F.; writing—original draft preparation, L.G.; writing—review and editing, Q.F. and S.C.; visualization, L.G.; supervision, Q.F. and C.Z.; project administration, Q.F. and C.Z.; funding acquisition, Q.F. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Beijing Academy of Agricultural Artificial Intelligence and Robotics-R&D of Key Technologies and Products for Agricultural Robot; Major Science and Technology Support Plan Project Tasks of Hebei Province [242N7204Z], China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgments: The Agricultural Robotics Laboratory located at the Beijing Academy of Agriculture and Forestry Sciences is the institution that the authors would like to express their gratitude to, as it provided the necessary experimental site for conducting this experimental work.

Conflicts of Interest: Author Lin Chen was employed by the company Hebei Institute of Agricultural Mechanization Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, Z.; Xun, Y.; Wang, Y.; Yang, Q. Review of smart robots for fruit and vegetable picking in agriculture. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 33–54. [\[CrossRef\]](#)
2. Bac, C.W.; Van Henten, E.J.; Hemming, J.; Edan, Y. Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *J. Field Rob.* **2014**, *31*, 888–911. [\[CrossRef\]](#)
3. Shamshiri, R.R.; Weltzien, C.; Hameed, I.A.; Yule, I.J.; Grift, T.E.; Balasundram, S.K.; Pitonakova, L.; Ahmad, D.; Chowdhary, G. Research and development in agricultural robotics: A perspective of digital farming. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 1–11. [\[CrossRef\]](#)
4. Kootstra, G.; Wang, X.; Blok, P.M.; Hemming, J.; Van Henten, E. Selective harvesting robotics: Current research, trends, and future directions. *Curr. Rob. Rep.* **2021**, *2*, 95–104. [\[CrossRef\]](#)
5. Zhang, F.; Gao, J.; Zhou, H.; Zhang, J.; Zou, K.; Yuan, T. Three-dimensional pose detection method based on keypoints detection network for tomato bunch. *Comput. Electron. Agric.* **2022**, *195*, 106824. [\[CrossRef\]](#)
6. Li, Y.; Feng, Q.; Zhang, Y.; Peng, C.; Ma, Y.; Liu, C.; Ru, M.; Sun, J.; Zhao, C. Peduncle collision-free grasping based on deep reinforcement learning for tomato harvesting robot. *Comput. Electron. Agric.* **2024**, *216*, 108488. [\[CrossRef\]](#)
7. Shi, Y.; Jin, S.; Zhao, Y.; Huo, Y.; Liu, L.; Cui, Y. Lightweight force-sensing tomato picking robotic arm with a “global-local” visual servo. *Comput. Electron. Agric.* **2023**, *204*, 107549. [\[CrossRef\]](#)
8. Shamshiri, R.R.; Dworak, V.; ShokrianZeini, M.; Navas, E.; Käthner, J.; Höfner, N.; Weltzien, C. An overview of visual servoing for robotic manipulators in digital agriculture. In Proceedings of the 43. GIL-Jahrestagung, Resiliente Agri-Food-Systeme, Osnabrück, Germany, 13–14 February 2023; pp. 231–241. Available online: <https://dl.gi.de/handle/20.500.12116/40254> (accessed on 20 February 2026).
9. Lehnert, C.; Tsai, D.; Eriksson, A.; McCool, C. 3d move to see: Multi-perspective visual servoing towards the next best view within unstructured and occluded environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 3890–3897. [\[CrossRef\]](#)
10. Zhao, D.-A.; Lv, J.; Ji, W.; Zhang, Y.; Chen, Y. Design and control of an apple harvesting robot. *Biosyst. Eng.* **2011**, *110*, 112–122. [\[CrossRef\]](#)
11. Mehta, S.; MacKunis, W.; Burks, T. Nonlinear robust visual servo control for robotic citrus harvesting. *IFAC Proc. Vol.* **2014**, *47*, 8110–8115. [\[CrossRef\]](#)
12. Mehta, S.; Burks, T. Adaptive visual servo control of robotic harvesting systems. *IFAC-PapersOnLine* **2016**, *49*, 287–292. [\[CrossRef\]](#)
13. Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. A review of key techniques of vision-based control for harvesting robot. *Comput. Electron. Agric.* **2016**, *127*, 311–323. [\[CrossRef\]](#)
14. Magalhães, S.A.; Moreira, A.P.; Santos, F.N.d.; Dias, J. Active perception fruit harvesting robots—A systematic review. *J. Intell. Robot. Syst.* **2022**, *105*, 14. [\[CrossRef\]](#)
15. Qiu, Z.; Hu, S.; Liang, X. Model predictive control for uncalibrated and constrained image-based visual servoing without joint velocity measurements. *IEEE Access* **2019**, *7*, 73540–73554. [\[CrossRef\]](#)
16. Carron, A.; Arcari, E.; Wermelinger, M.; Hewing, L.; Hutter, M.; Zeilinger, M.N. Data-driven model predictive control for trajectory tracking with a robotic arm. *IEEE Rob. Autom. Lett.* **2019**, *4*, 3758–3765. [\[CrossRef\]](#)

17. Kleff, S.; Meduri, A.; Budhiraja, R.; Mansard, N.; Righetti, L. High-frequency nonlinear model predictive control of a manipulator. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 7330–7336. [\[CrossRef\]](#)
18. Gold, T.; Völz, A.; Graichen, K. Model predictive interaction control for robotic manipulation tasks. *IEEE Trans. Rob.* **2022**, *39*, 76–89. [\[CrossRef\]](#)
19. Xi, Y.-G.; Li, D.-W.; Lin, S. Model predictive control—Status and challenges. *Acta Autom. Sin.* **2013**, *39*, 222–236. [\[CrossRef\]](#)
20. Nauman, M.; Shireen, W.; Hussain, A. Model-free predictive control and its applications. *Energies* **2022**, *15*, 5131. [\[CrossRef\]](#)
21. Yaren, T.; Kizir, S. Real-time Nonlinear Model Predictive Control of a Robotic Arm Using Spatial Operator Algebra Theory. *J. Field Rob.* **2025**, *42*, 2337–2354. [\[CrossRef\]](#)
22. Zhang, T.; Mo, H. Reinforcement learning for robot research: A comprehensive review and open issues. *Int. J. Adv. Rob. Syst.* **2021**, *18*, 17298814211007305. [\[CrossRef\]](#)
23. Zhang, Y.; Li, Y.; Feng, Q.; Sun, J.; Peng, C.; Gao, L.; Chen, L. Compliant motion planning integrating human skill for robotic arm collecting tomato bunch based on improved DDPG. *Plants* **2025**, *14*, 634. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Zhong, X.; Zhou, Q.; Sun, Y.; Kang, S.; Hu, H. Deep reinforcement learning-based uncalibrated visual servoing control of manipulators with FOV constraints. *Appl. Sci.* **2025**, *15*, 4447. [\[CrossRef\]](#)
25. Zhang, L.; Pei, J.; Bai, K.; Chen, Z.; Zhang, J. A Closed-Loop Multi-perspective Visual Servoing Approach with Reinforcement Learning. In Proceedings of the 2023 IEEE International Conference on Robotics and Biomimetics (ROBIO), Koh Samui, Thailand, 4–9 December 2023; pp. 1–7. [\[CrossRef\]](#)
26. Li, Z.; Zhou, Y.; Wang, L.; Zhang, X.; Li, A.; Zhu, M.; Wu, Q. An end-to-end controller with image-based visual servoing of industrial manipulators with soft-actor-critic algorithm. *Knowl.-Based Syst.* **2025**, *311*, 112980. [\[CrossRef\]](#)
27. Liu, R.; Nageotte, F.; Zanne, P.; de Mathelin, M.; Dresch-Langley, B. Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review. *Robotics* **2021**, *10*, 22. [\[CrossRef\]](#)
28. Al Ali, A.; Zhu, Z.H. Reinforcement learning for path planning of free-floating space robotic manipulator with collision avoidance and observation noise. *Front. Control Eng.* **2024**, *5*, 1394668. [\[CrossRef\]](#)
29. Memarian, F.; Goo, W.; Lioutikov, R.; Niekum, S.; Topcu, U. Self-supervised online reward shaping in sparse-reward environments. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 2369–2375. [\[CrossRef\]](#)
30. Liang, J.; Lou, G.; Zhou, F.; Cai, Y.; Wang, C.; Zhou, Y. Residual Reinforcement Learning Based on Inverse Kinematic Modeling for Soft Robotic Arm Control. *J. Dyn. Syst. Meas. Control* **2025**, *147*, 061007. [\[CrossRef\]](#)
31. Gao, L.; Li, Y.; Wu, J.; Feng, Q.; Chen, L.; Shen, C.; Ma, Y.; Zhao, C. YOLO-GPP: End-to-end prediction of the grasp position and pose on tomato peduncle for robotic harvesting. *Artif. Intell. Agric.* **2026**, *16*, 713–724. [\[CrossRef\]](#)
32. Xie, B.; Jiao, W.; Wen, C.; Hou, S.; Zhang, F.; Liu, K.; Li, J. Feature detection method for hind leg segmentation of sheep carcass based on multi-scale dual attention U-Net. *Comput. Electron. Agric.* **2021**, *191*, 106482. [\[CrossRef\]](#)
33. Rong, J.; Wang, P.; Wang, T.; Hu, L.; Yuan, T. Fruit pose recognition and directional orderly grasping strategies for tomato harvesting robots. *Comput. Electron. Agric.* **2022**, *202*, 107430. [\[CrossRef\]](#)

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