Robust pollination for tomato farming using deep learning and visual servoing

Rajmeet Singh ⬤, Lakmal Seneviratne and Irfan Hussain

Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University, Abu Dhabi, UAE
Corresponding author: Irfan Hussain; Email: irfan.hussain@ku.ac.ae

Abstract

In this work, we propose a novel approach for tomato pollination that utilizes visual servo control. The objective is to meet the growing demand for automated robotic pollinators to overcome the decline in bee populations. Our approach focuses on addressing this challenge by leveraging visual servo control to guide the pollination process. The proposed method leverages deep learning to estimate the orientations and depth of detected flower, incorporating CAD-based synthetic images to ensure dataset diversity. By utilizing a 3D camera, the system accurately estimates flower depth information for visual servoing. The robustness of the approach is validated through experiments conducted in a laboratory environment with a 3D printed tomato flower plant. The results demonstrate a high detection rate, with a mean average precision of 91.2 %. Furthermore, the average depth error for accurately localizing the pollination target is impressively minimal, measuring only 1.1 cm. This research presents a promising solution for tomato pollination, showcasing the effectiveness of visual-guided servo control and its potential to address the challenges posed by diminishing bee populations in greenhouses.

1. Introduction

Greenhouse farming plays a crucial role in modern agriculture by enabling controlled and efficient crop cultivation. However, the scarcity of natural pollinators, such as bees, poses a significant hurdle to achieving successful pollination within greenhouse environments. The controlled environments of greenhouses make them reliant on effective crop pollination, which cannot solely be achieved through natural pollinators. Despite efforts to increase honey bee colonies, the growth rate has been insufficient to meet the growing demand, resulting in pollination deficits and escalating prices for pollination services [1, 2]. The tomato is a globally popular and extensively cultivated crop, ranking at the forefront of global crop production [3]. Pollination, the transfer of pollen from the stamens to the pistils, plays a vital role in the development of seeds and fruits. Unlike certain plants that rely on cross-pollination between different strains, tomatoes possess the ability to self-pollinate within a single flower once pollen is produced. In greenhouse tomato cultivation, three widely used methods for pollination are employed. These methods consist of insect pollination, artificial pollination through manual flower vibration, and hormonal pollination utilizing plant growth regulators. However, ensuring effective insect pollination can be challenging, particularly in high-temperature conditions during the summer when insect activity decreases, leading to reduced pollination efficiency. Furthermore, the use of commercial bumblebees, which are commonly employed for pollination, may be restricted in certain countries like Japan and Australia due to concerns regarding potential ecological risks [3, 4]. Figure 1 shows the tomato flower pollination overview process (a) bee pollination (b) manual pollination. Insect-mediated pollination follows natural processes, where honeybees and bumblebees shake tomato flowers to facilitate pollination through the transfer of pollen. However, managing and raising insects for pollination can be challenging,
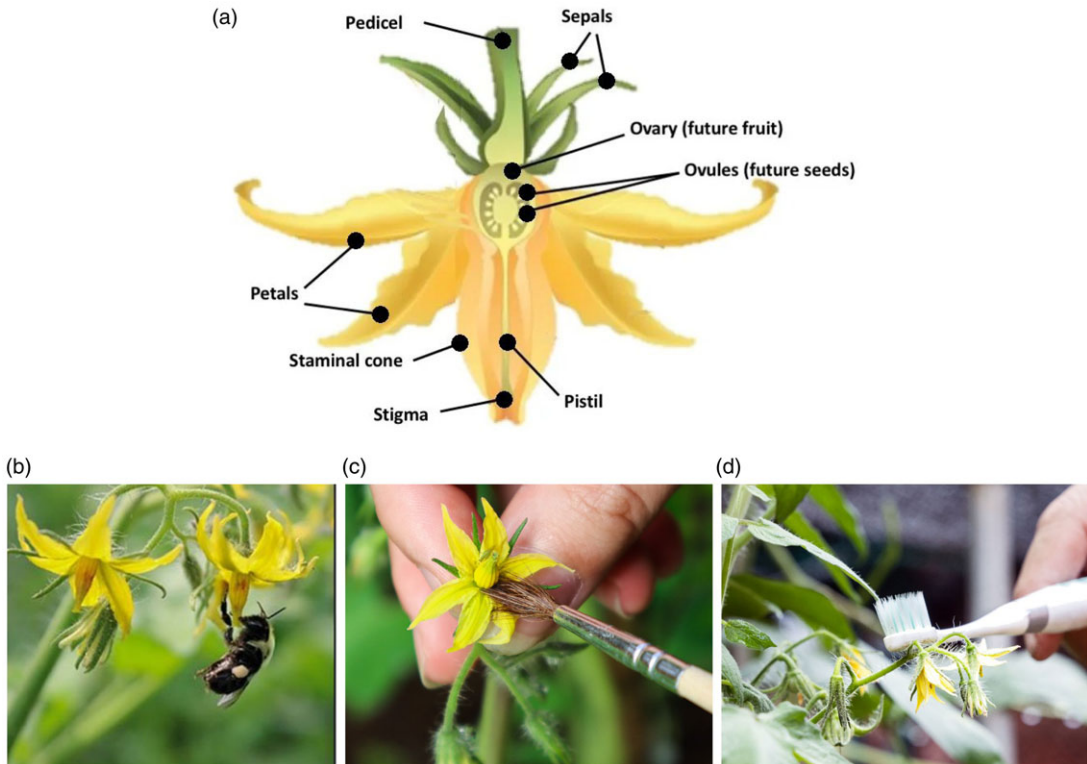
Figure 1. *(a) Tomato flower structure, (b) bee pollination, (c) hand brush pollination, and (d) vibrator toothbrush pollination.*

and the effectiveness of pollination declines when bees become inactive in high temperatures and moisture. Consequently, artificial pollination methods, such as manual flower vibration, are often employed. In this approach, farm workers visually identify mature flowers based on their shape and use vibrating tools to shake them for pollination. However, the accurate classification of flowers for artificial pollination requires experienced and skilled workers. Consequently, a significant number of skilled laborers are required, resulting in increased expenses for the cultivation process.

The decline in pollinators has led to an economic impact, resulting in a rising demand for pollination services within the agriculture sector [5, 6].This surge in demand necessitates the need for robust solutions. In the past ten years, the pressing matter of pollination has gained attention from various researchers, small companies, and start-ups, as evidenced by the increasing number of patents focused on artificial pollination devices. [7]. The devices include hand brush, vibrators, and robotic pollinators [8]. The development of advanced technologies, including robots, visual servoing, and artificial intelligence, is crucial for achieving automatic pollination. Industrial robots, known for their contribution to automation, have historically been limited to specific and fixed tasks while demonstrating high precision [9]. Visual servoing, a branch of robot control, has evolved from the manipulation control of arm-robot manipulators [10]. A pneumatic-based gripper for harvesting was proposed and designed [11]. Specific attention has been focused on tomato horticulture for a practical design and prototyping of a device for laboratory experiments. With advancements in computer vision, the utilization of cameras mounted on the robot end effector, known as eye-in-hand servoing, has become more prevalent across various fields, including agriculture. This technology aims to enhance productivity for commercial farmers [12]. Additionally, there have been instances where eye-to-hand servoing has been employed, which involves mounting the camera on the workspace to provide a global view of the robot's actions [13].

The prevailing method for categorizing classical visual servoing approaches is to divide them into two categories: image-based visual servoing (IBVS) and position-based visual servoing (PBVS) [14, 15]. In IBVS, visual feedback is obtained by extracting image visual features, allowing for the computation of the minimum error between the desired and observed image features captured by the camera. Conversely, PBVS utilizes the reference camera's pose coordinate system to estimate the target's pose using camera geometric model. Additionally, fused approaches, like 2.5D visual servoing, combine elements of both IBVS and PBVS [16]. A visual servoing using convolutional neural network (CNN) approach is proposed in ref. [17].This approach enables the estimation of the robot's relative pose with respect to a desired image. As agricultural robots, mobile platforms, and artificial intelligence progressed in tandem, robots became increasingly adaptive and proficient in resolving complex tasks with enhanced accuracy. While artificial intelligence was initially limited to simple rule-based systems [18] lacking versatility, [19] the Mask R-CNN based apple flower detection for precise pollination was proposed in ref. [20]. The computer vision-based pollination monitoring algorithm called Polytrack was proposed to tracks multiple insects simultaneously in complex agricultural environments [21]. The fusion of CNNs as feature extractors and classification with machine learning algorithms was proposed for classifications of the Almond in the farm in ref. [22]. This evolution paved the way for the successful implementation of Deep Learning across various robotic domains. Deep learning has demonstrated remarkable robustness and accuracy across diverse fields, including classification and object detection. This progress has been achieved through the introduction of novel model architectures such as You Only Look Once (YOLO), Inception, and Vision Transformer, which have significantly expanded the capabilities of deep learning.

The accurate estimation of depth plays a crucial role in the advancement of efficient robotic pollination systems. Estimating depth from a single image is a complex problem in computer vision, using triangulation [23], monocular cues [24], and deep learning models [25]. Nevertheless, the existing methods for depth estimation have limitations, including the need for multiple images, human annotations, or extensive training data. In light of these challenges, our study introduces depth estimation using RGB-D camera. The deep learning and intelligent visual-guided servoing-based pneumatic pollination system for watermelon is proposed in ref. [26].

Tomato is a widely cultivated crop, occupying the top position in global crop production [3]. Pollination is the process by which pollen from the stamens is transferred to the pistils, leading to the development of seeds and fruits. While some plants require cross-pollination from different strains, tomatoes can easily self-pollinate within a single flower once pollen is produced. The paper by Masuda et al. [27] presents the development of a multi-degrees-of-freedom robotic pollinator designed for precise flower pollination. The robot employs a unique approach of simulating the effect of wind blowing to facilitate pollination. To meet cultivation restrictions, the robot incorporates a collision-free motion design that ensures the end-effector reaches the desired blowing position without causing any damage to the crops or greenhouse structure. For tomato flowers, compressed air is utilized as the pollination method. The pollination arm consists of a linear motion mechanism, a flexible mechanism, and a pneumatic system. At the arm's tip, a soft tube is attached, which gently shakes the flowers using compressed air during the pollination process. A mobile robot-based robotic kiwi pollinator specifically designed for kiwi farm is tested in real field conditions [28]. The authors utilized YOLOv4 with transfer learning to detect kiwi flowers and buds. In subsequent works [29], YOLOv4 was employed to detect flowers, followed by identifying the operating position of the robotic pollinator's mechanical arm and the orientation of the kiwi flower. An ground vehicle (Bramblebee) is proposed for pollinating flowers on blueberry plants [8]. However, the work focused on a limited number of flower poses and did not account different flower orientations. This limitation is not suitable for the tomato case, especially considering the diverse orientations of flowers.

The tomato is a highly cultivated and economically significant crop worldwide [30]. However, tomato cultivation faces several challenges, including issues with pollination [31]. To address these challenges, incorporating robots for pollination in smart greenhouses has become crucial. Robotic pollination offers benefits such as labor cost reduction and improved pollination efficiency. Therefore,
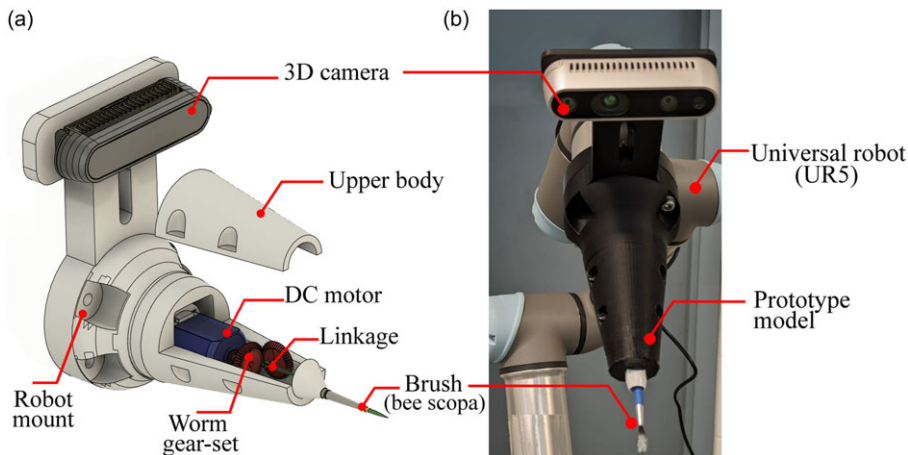
Figure 2. *(a) CAD model of pollinator and (b) prototype model of pollinator mounted on UR5 robot.*

there is a need for an accurate detection model that identifies the flower and bud and the location of flowers to enhance tomato production and quality in greenhouses. When it comes to tomato pollination, several primary challenges need to be addressed. These include the presence of diverse orientations of flowers caused by their location on the plant, the detection of relatively small-sized flowers, and the complications arising from occlusions caused by the abundance of leaves and other plant components.

In this study, we present the design of our Intelligent Tomato Flower Pollination (ITFP) system, which incorporates a deep-learning image feedback guided visual servoing software. This software enables the system to achieve high precision in the pollination process. It is important to note that image-guided visual servoing represents a novel approach to robotic pollination control, distinct from classical methods. The ITFP system proposed in this study utilizes deep learning (DL) techniques to accurately detect flowers in various pose and estimate the depth using 3D camera information.In addition to flower localization, our ITFP system excels at accurately determining the precise position and size of flowers and buds. It also achieves high-speed and accurate differentiation between flowers and buds during the pollination process. Furthermore, the system incorporates robust algorithms to handle challenges related to illumination variations, domain shifts, and geometric transforms. By integrating all these capabilities into a single module, our system effectively resolves the orientation, size, and location of the flowers. To accomplish these achievements, we utilized the Ultralytics version of YOLOv8 [32]. The model was trained to extract and learn orientation features specifically for tomato flowers.

2. Materials and methods

2.1. System overview

The robust working of the ITFP depends upon the coordination between mobile platform (Clearpath Husky) and Universal robot (UR5) 6-DoF robot arm. The proposed pollinator design with Intel realsense D435 camera is mounted on the robotic arm. Figure 2 provides a visual representation of the CAD model of the pollinator and prototype is attached to the end effector of the UR5 robotic arm. The novel pollinator design includes a DC motor coupled with gear mechanism to make end link vibratory. The brush is attached to the end of the link of the pollinator act as bee scopa (bundles of fine hairs on their abdomens) [33] to hold collected pollen.The mobile platform helps in accessing the plants, while the 6 DoF arm offers flexibility with movement in 3D space. This adaptability allows the robotic arm to navigate through the greenhouse. Figure 3 presents the UR5 robot various parameters such as size and joint coordinates. It is important to note that the current mobile platform (Husky) is autonomous,
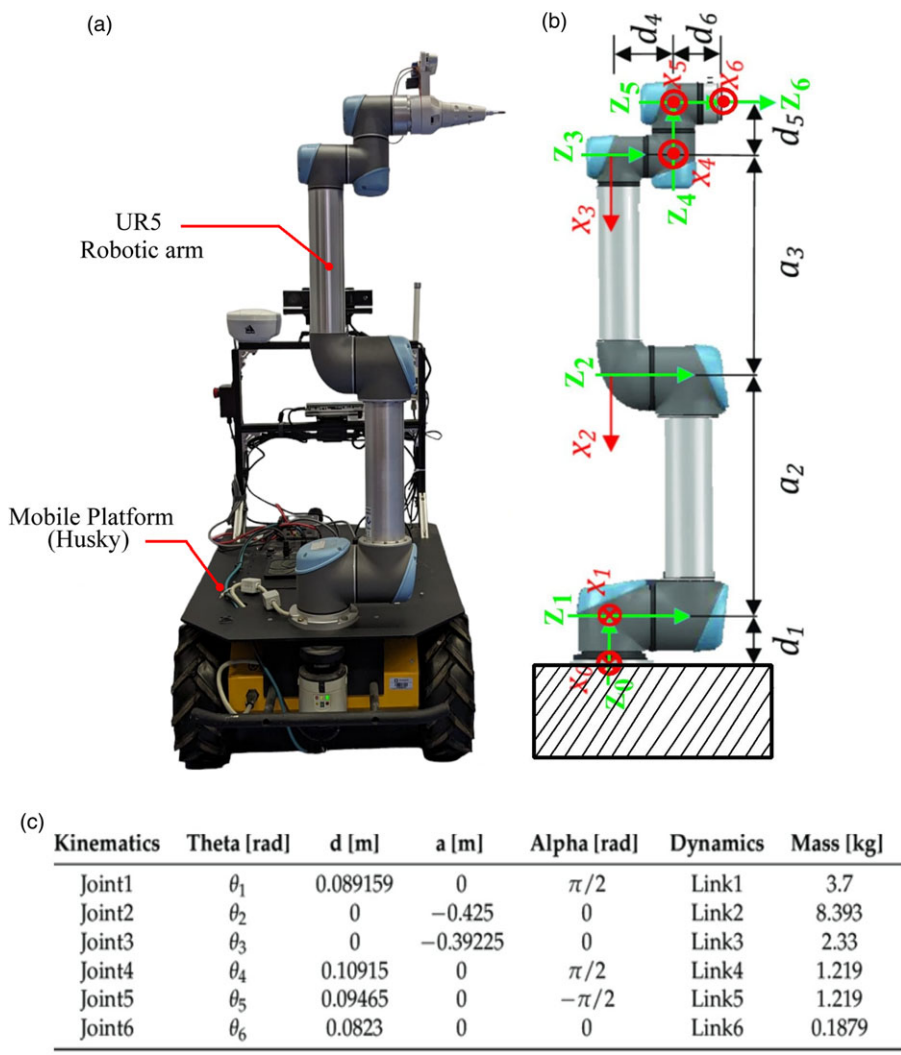
Figure 3. *Universal robot (UR5) joint configuration and Denavit–Hartenberg (DH) parameters.*

but in this work, our primary focus lies on the pollination process itself. The intel realsense D435 camera captures RGB and depth information that is essential for the visual servoing system. It is important to highlight that the realsense camera provides various resolution settings, and for our study, we utilized the 720p setting to maintain a balance between image quality and processing speed.

2.2. Pollination using deep learning-guided UR5 robotic arm control

2.2.1. Pollination controller

The precise control of the robot wrist's position and orientation is important because the pollinator is mounted on the wrist of the robot. The control visual servoing loop scheme is shown in Figure 4. The wrist position is represented by a vector (x, y, z), and orientation by vector $(\theta_x, \theta_y, \theta_z)$, indicating angles relative to the x, y, and z axes. The position and orientation information are combined and referred to as the "pose vector" $(x, y, z, \theta_x, \theta_y, \theta_z)$. Additionally, the current pose vector are denoted as $(x', y', z', \theta'_x, \theta'_y, \theta'_z)$, representing the "current pose vector" of the robot. The difference between the two
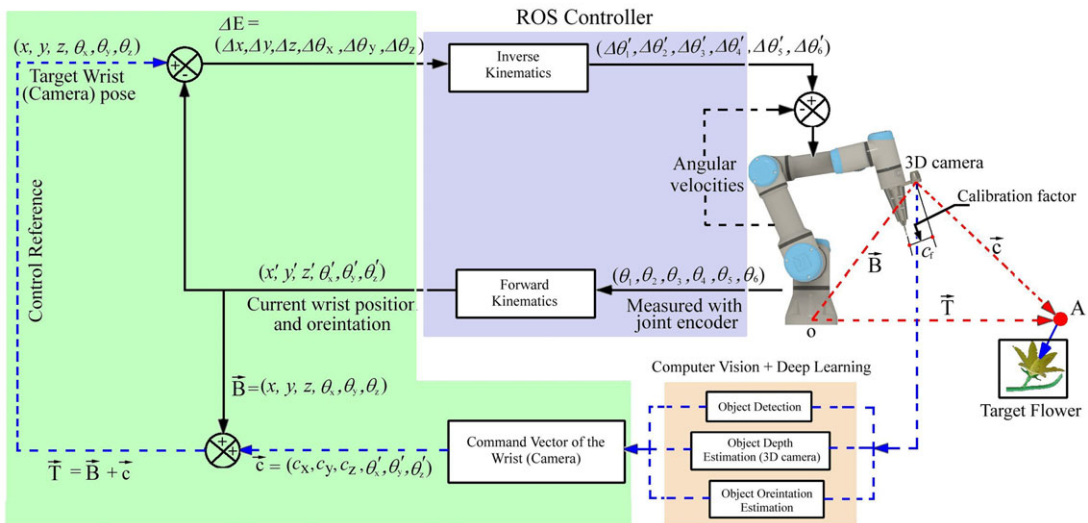
Figure 4. *Control visual servoing loops.*

vectors (ΔE) is calculated.

$$(\Delta E) = \left(x, y, z, \theta_x, \theta_y, \theta_z\right) - \left(x', y', z', \theta'_x, \theta'_y, \theta'_z\right) \tag{1}$$

By employing control loops, the disparity among reference and current pose vector is progressively diminished until it converges to zero. This convergence signifies that the desired control direction has been successfully achieved. To provide a more detailed explanation, the pose error vector, which represents the difference between two input pose vectors, undergoes a transformation using inverse kinematics. This transformation converts the pose error vector into an angular error vector, denoted $\Delta\zeta$.

$$\Delta\zeta = \left(\Delta x, \Delta y, \Delta z, \Delta\theta_x, \Delta\theta_y, \Delta\theta_z\right) I - K \tag{2}$$

$$= \left(\Delta\theta'_1, \Delta\theta'_2, \Delta\theta'_3, \Delta\theta'_4, \Delta\theta'_5, \Delta\theta'_6\right) \tag{3}$$

In the Eq. (2), *I-K* denotes the inverse kinematics transform, while $\Delta\zeta$ represents the angular representation of the pose difference. The angular difference, $\Delta\zeta$, is utilized in the robot inner loop. It is compared with the angular velocity of the robot joints encoders. This enables the robot to track the reference velocity values, treating the angle vector $\Delta\zeta$ as a velocity vector for control purposes.

Meanwhile, the angles of each joint are denoted as ζ''.

$$\zeta'' = \left(\theta''_1, \theta''_2, \theta''_3, \theta''_4, \theta''_5, \theta''_6\right) \tag{4}$$

The angular vector ζ'' is transformed to base coordinate of the robot as

$$\overline{B} = \left(\theta''_1, \theta''_2, \theta''_3, \theta''_4, \theta''_5, \theta''_6\right)^K = \left(x', y', z', \theta x', \theta y', \theta z'\right) \tag{5}$$

where the *K* denotes the forward kinematics transformation. Therefore, the robot pose vector $= (x', y', z', \theta x', \theta y', \theta z')$ approaches the reference pose $(x, y, z, \theta x, \theta y, \theta z)$ provided from external sources.

To facilitate robotic pollination, two movements are required: the approach (command) stage and the pollination stage. During the approach (command) stage, the robot's wrist swiftly moves from its initial position to the designated pollination point, labeled as A. This stage involves both approaching the flower and making necessary pose adjustments. While the robot arm is positioned around the flower, it actively seeks the optimal pollination spot through positional adjustments. To support this
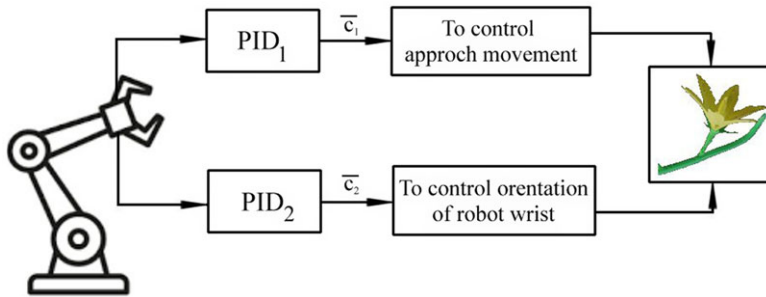
Figure 5. *Two PID controllers scheme to effectively coordinate the robot's approach, alignment, and orientation for successful pollination.*

action, technologies such as flower detection and tracking are employed. The pollination point A is positioned in close proximity to the target flower. The pollinator is used for pollination action (as describe in Section 2.1).

The command vector is defined as $(c_x, c_y, c_z, \theta'_x, \theta'_y, \theta'_z$. The new pose vector of the robot arm is summation of base vector $\overline{B}$ and an command pose vector $\overline{c}$.

$$\overline{G} = \overline{B} + \overline{c} \qquad (6)$$

Calculating the pose vector B, which represents the transformation from the robot's base to the wrist, is a straightforward task using robot kinematics. However, the challenge lies in calculating the command vector $\overline{c}$ for position and orientation. To address this issue, we introduced a deep learning-based approach in our work, which is discussed in detail in the respective section. This method aims to determine the components of the command vector $\overline{c}$, enabling more accurate and efficient control of the robot's position and orientation.

2.2.2. Pollinator pose control

To successfully pollinate a tomato flower using a robot, it is crucial to position the wrist in front of the detected flower, referred to as D, with the wrist facing the pistil. Additionally, the wrist should be apart from the flower equal to the length of the pollination tool (our case 5 cm). The control process begins with the initiation of the approaching control mechanism, which focuses on bringing the robot closer to the target flower. Simultaneously, precise orientation adjustment is implemented to ensure alignment with the flower pistil. The combined approach and orientation control strategies enable the robot to approach the flower closely while maintaining the correct alignment for effective pollination.

The PID controllers utilize two control signals, namely $\overline{c_1}$ and $\overline{c_2}$, to perform their respective tasks. The first PID controller, with control signal $\overline{c_1}$ to regulate the approach control of the robot, enabling efficient and swift movement towards the target flower. Simultaneously, the second PID controller utilizes the control signal $\overline{c_2}$ to manage the wrist orientation control, ensuring precise alignment with the flower pistil. By utilizing these control signals, the PID controllers effectively coordinate the robot's approach and alignment for successful pollination. Figure 5 shows the detail control approach for controlling of the robot wrist using PID controllers. The estimation of the flower's position is accomplished using deep learning techniques, as described in next section.

2.2.3. Deep learning-based robot pose estimation

As depicted in Figure 6, the camera utilized a deep learning-based detection model to recognize tomato flowers. This recognition enables the identification of the flowers, providing visual information such as size and orientation. These elements play a vital role in the control loops of the proposed robot control
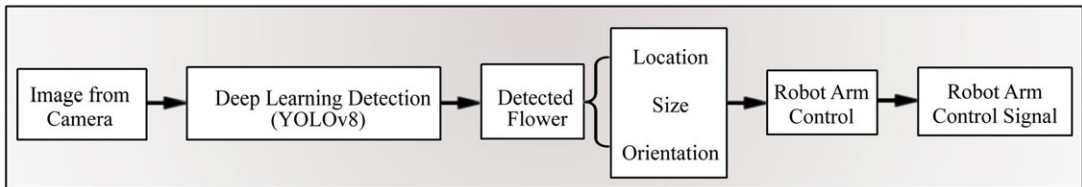
Figure 6. *Components of the image feedback visual-servoing system.*
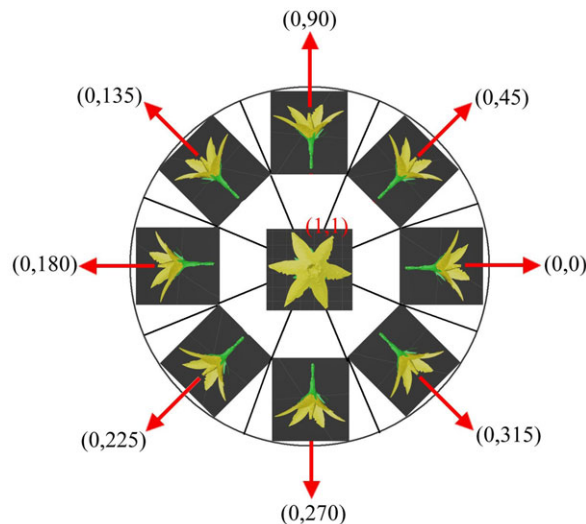

Figure 7. *Flower orientation labelling based on ITFP annotation wheel system.*

system. In this study, this methodology is referred to as image feedback guided visual servoing, where visual information guides the robot's actions and decision-making processes. The RGB-D camera is used to detect the depth of detected flower from camera. A selection of flower is determined on the basis of confidence value. The priority is given to the highest detection confidence value. Ensuring the correct orientation of the flower is crucial for accurately aligning the pollinator to the flower center. This alignment is essential to capture a clear and focused image of the flower during the robotic pollination process. The robot operating system (ROS) controller generates the control signal for robot arm's to reach the current target end effector position. The primary purpose of the detector is to accurately predict the flower orientation. Flower orientation, in this context, refers to the direction in which the flower stigma faces. The assigned classes, representing different directions of the flower, hold significant importance for achieving the objectives of the study, particularly in the context of efficient flower pollination using the ITFP system. To precisely characterize the orientations of the flowers, a classification approach based on ITFP annotation wheel system (orientation angle based) was proposed and adopted, as illustrated in Figure 7.

To enable the robot's wrist to accurately reach the center of each identified flower, the pose estimation is performed for each flower individually. The flower position is extracted from the pixel coordinates obtained from image processing, while the corresponding depth information is estimated using an RGB-D camera. This combined approach of extracting position and depth data allows for precise localization of each flower, enabling the robot to effectively navigate and position its wrist at the center of each flower for the pollination process. Since the exact 3D flower orientation is unknown, a simplification is made by dividing the flower orientations into nine distinct classes. These classes are determined based on the direction in which the flowers are positioned on a polar coordinate system, as depicted in Figure 7.
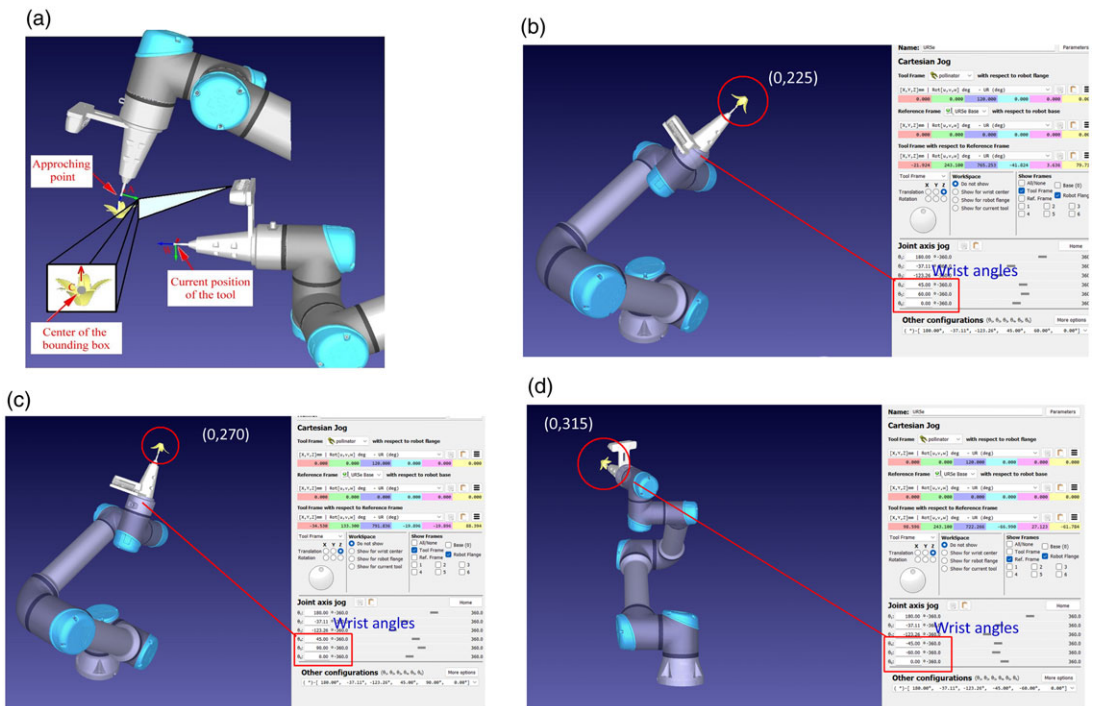
Figure 8. *(a) Robot wrist (position and orientation) for pollination control, (b) Class (0,255), (c) Class (0,270), and (d) Class (0,315).*

The defined classes are as follows: (1,1), (0, 0), (0, 45), (0, 90), (0, 135), (0, 180), (0, 225), (0, 270), and (0, 315). This classification scheme allows for a practical representation of flower orientation.

During the process of approaching the flower, it is common for the wrist (camera) of the robot to not be aligned with the center of the flower's pistil. Figure 8(a) provides an example where the wrist (camera) is not aligned with the flower center. To facilitate effective pollination, it becomes necessary to adjust the position of the wrist (camera) in the direction of the frontal face of the flower's, as indicated by the arrow labeled as A. This adjustment ensures that the camera is properly oriented toward the desired target area of the flower for accurate pollination. The few flower orientation classes are shown in Figure 8(b)–(d) with respective desired wrist angles.

As per Section 2.2.2, the combined control approach emphasizes the significance of orientation adjustment control in aligning the wrist with the flower center. In this context, if the wrist current position is denoted as W, the point where the wrist and the flower center bounding box (c) are aligned is represented by A. The pollinating pose of the robot arm at A is calculated by integrating the position information of the flower extracted from pixel coordinates, depth information, and the flower orientation. The flower orientation is determined through deep learning techniques and subsequently converted into the corresponding wrist orientation vector using Table I. This integration of positional and orientation data enables the robot to achieve the precise pose required for effective pollination. Table I illustrates the transformation of the detected polar coordinate information into 3D coordinate data from Figure 8, which aids in facilitating the rotation of the end effector of the Intelligent Tomato Flower Pollination (ITFP) system. Additionally, the depth of the flower from the wrist is calculated using the 3D camera. These transformations and depth estimations provide essential data for accurately positioning and manipulating the robot's end effector (pollinator) during the pollination process. After determining the

Table I. *Flower orientation classes with desired robot wrist angle values.*

Classes	x-axis(Roll angle) in degree	y-axis(Pitch angle) in degree	z-axis(Yaw angle) in degree
(1,1)	0	0	0
(0,0)	90	0	0
(0,45)	−90	0	0
(0,90)	−90	45	0
(0,135)	0	−90	0
(0,180)	0	90	0
(0,225)	45	60	0
(0,270)	45	90	0
(0,315)	−45	−60	0

pollination point A and its corresponding orientation, the robot arm can achieve the desired pose, as demonstrated in Figure 7.

2.2.4. Depth estimation using 3D camera

In order to execute the pollination action, it is crucial to have information about the orientation and depth of the flower. Determining the direction of the flower can be achieved easily by utilizing the object's coordinates within the image. However, obtaining accurate depth information for objects at short distances, such as in the case of pollination tasks, poses a challenge. In this work, we are using intel realsense 3D camera for measuring the depth distance of the detected flower. The intel realsense 3D camera has the capability of auto adjustment of high dynamic range (HDR) property. HDR imaging refers to the capability of an imaging system to capture scenes with both very dark and very bright areas. To overcome the limited dynamic range, many designs employ a technique of capturing multiple images with different exposures and combining them to create a single HDR image. The Intel RealSense D435 camera has a maximum HDR range of 10 m. To utilize this feature in the Intel RealSense Viewer, users must enable the depth and infrared streams. To ensure the generation of an HDR output, the device's frame rate is configured to twice the normal speed. The HDR function merges consecutive frames, such as high exposure + low exposure, and low exposure + high exposure, to generate an HDR depth. Although the resultant frame rate remains 60 fps, there will be an effective latency of 2 frames [34].

Camera model.   The projection of the camera onto the image plane is depicted in Figure 9, where the pixel point p(u,v) is represented in the 2D coordinate system. This projection illustrates the mapping of the three-dimensional (3D) world onto the two-dimensional (2D) image plane captured by the camera. The 3D scene point and camera coordinate system are represented as P_c and (x_c, y_c, z_c). The c is the center point of the detected flower on image plane O_i, and f is focal length of the camera. The relationship between the 3D coordinates and the corresponding image coordinates is established using the pinhole model. This model describes the geometric principles governing the projection of three-dimensional points onto a two-dimensional image [35].

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f\frac{x_c}{z_c} \\ f\frac{x_c}{z_c} \end{bmatrix} \tag{7}$$

However, it should be noted that in practical scenarios, the principal point of the camera may not align perfectly with the center of the image. Therefore, it becomes necessary to consider the offset (u_o, v_o) in the image plane. Consequently, Eq. (7) is modified to incorporate this offset and account for any deviation from the ideal alignment.
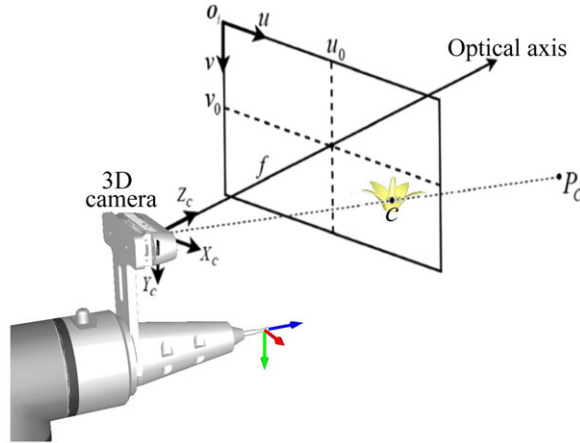
Figure 9. *Camera model for flower depth calculation.*

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \qquad (8)$$

Hand-to-eye camera robot calibration.   In this study, a ROS-based hand-to-eye camera robot calibration approach is employed. The calibration process involves using a printed Aruco marker board (shown in Figure 10(a)) to calibrate both the camera and the robot. To illustrate the setup, consider a camera that is mounted on the robot's end effector, as depicted in Figure 10(b). The *visphand2eyecalibration* ROS package is used for calibration process. The package contains the two transformation (i) camera-to-object and (ii)world to hand transformation. The *calibrator node* is used to calibrate the hand-to-eye calibration. For hand-to-eye calibration process, place the Aruco plane inside the camera view and run the *arucorealsense.launch* file. Two transformation frames, namely /cameralink and /armarker, are published in the system. These frames represent the spatial relationships between the camera and the marker board attached to the robot.

The visualization of these frames can be observed in *RVIZ*, as depicted in Figure 10(c) and (d). This calibration ensures accurate perception and coordination between the robot's hand and the camera during pollination tasks.

2.3. ITFP operational procedure

The robot-based pollination task is initiated by capturing an image of the flower using the camera mounted to the end effector of the 6-DoF robot arm. As shown in Figure 11, the image is processed by adeep learning detector, which provides outputs containing information about the flower's size, and orientation. When the identified flower is determined to be frontal-facing ((1,1) class), the robot engages approach control and proceeds to estimate the depth of the target in Z direction. After pollination, the robot arm returns to its home position and proceeds to locate the next flower. If estimation of pollinator depth is unsuccessful, the process is redirected to the deep learning detector for further analysis in the loop. If the flower is not initially frontal facing, the software initiates an orientation control. This control mechanism directs the robot arm to reposition itself and continues detecting the flower's orientation until the frontal orientation is detected, ensuring accurate pollination.
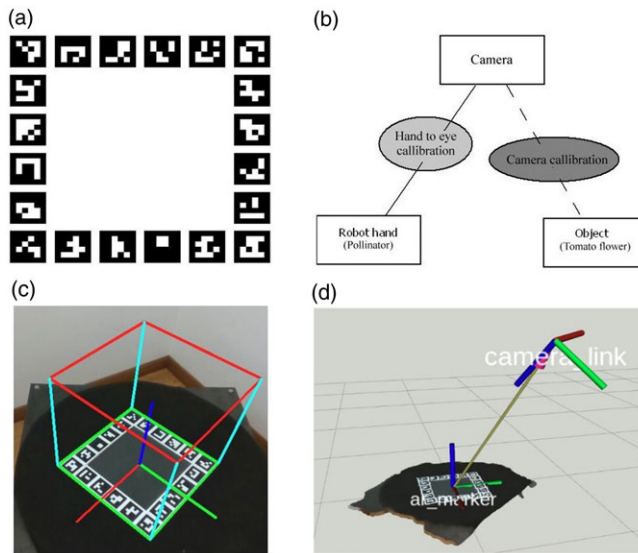
Figure 10. *Camera-robot calibration (a) Aruco marker, (b) hand-to-eye camera robot calibration model, (c) Aruco marker visualization in RVIZ, and (d) Camera-aruco marker hand to eye calibration.*

3. Results

3.1. Data collection

A dataset of tomato flowers was obtained from two sources: roboflow [36] and CAD synthetic data generated using Solidworks. The dataset comprised a total of 1,150 images, incorporating images from both roboflow (950) and the synthetic dataset (200). This approach ensured a diverse collection of images with varied characteristics and perspectives, capturing different lighting conditions. In Figure 12, we present the class distribution of our dataset, illustrating the number of instances for each class. The data reveal an imbalanced distribution among the Orientation Classes, with Class (0, 315) being the most prevalent, while Class (0, 90) is the least represented. To overcome this imbalance, we have applied data augmentation techniques to improve the overall generalization of the model. Table II mentioned the detailed information regarding the number of buds, orientation, and flower data in each dataset.

3.2. Data labeling

The main focus of the labeling strategy is on the orientation and classification of tomato flowers and buds. To ensure precise deep learning model training, a controlled labeling strategy called Refinement Filter Bank was employed. This strategy was previously utilized for pest recognition and plant disease [37]. By utilizing the Refinement Filter Bank approach, the labeling process aimed to enhance the accuracy and reliability of the deep learning model in accurately identifying and classifying tomato flowers and buds. This comprehensive labeling strategy aimed to train the model to accurately differentiate between various aspects of the tomato flower (orientation, flower, and buds).

The implementation of the nested label strategy is visualized in Figure 13. By utilizing this nested label strategy, the model training process encompasses the ability to accurately classify and differentiate not only the orientation of the flowers but also distinguish between flowers and buds. The label strategy was applied to both the roboflow dataset and the CAD synthetic dataset. This means that for both sets of data, the labeling process was carried out to classify and annotate the relevant features, such as flower orientation, flower, and buds. By applying the label strategy to both datasets, the resulting labeled
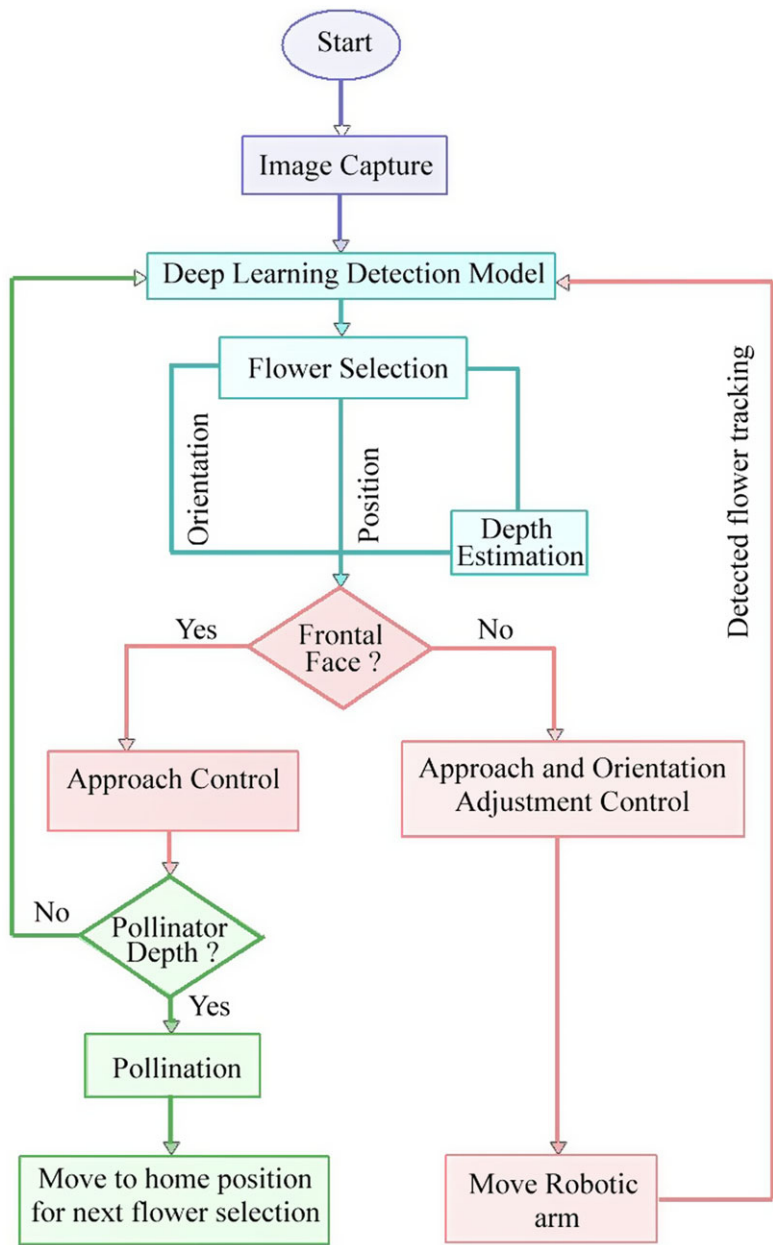
Figure 11. *Flow diagram of ITFP workflow system.*

datasets were enriched with accurate and comprehensive annotations, ensuring consistency in the training and evaluation of the deep Learning model across both sources of data. Through the incorporation of diverse classification types, the deep learning model is trained to effectively detect and classify between various aspects of tomato flowers and buds. This inclusive labeling approach enhances the model's comprehension of the tomato flower's visual characteristics and orientation. As a result, the model's performance is significantly improved, enabling it to accurately identify and classify tomato flowers in real-world scenarios.

Table II. *Amount of data for respective classes values.*

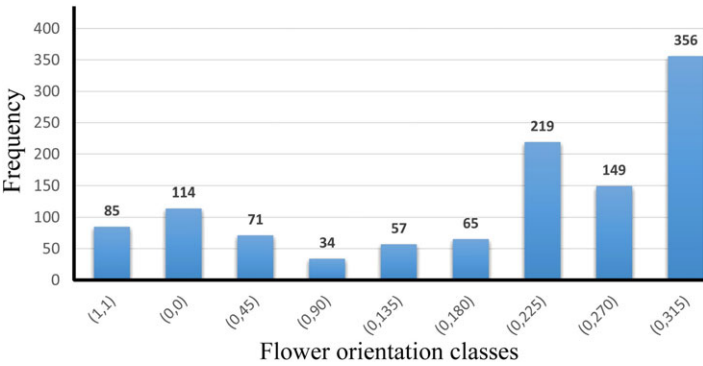| Classes	Dataset (No's)	
	Roboflow	**Synthetic CAD**
Buds	250	50
Flowers	700	150
Orientation	950	200

Figure 12. *Distribution graph for all the classes in the flower dataset (roboflow and synthetic data).*
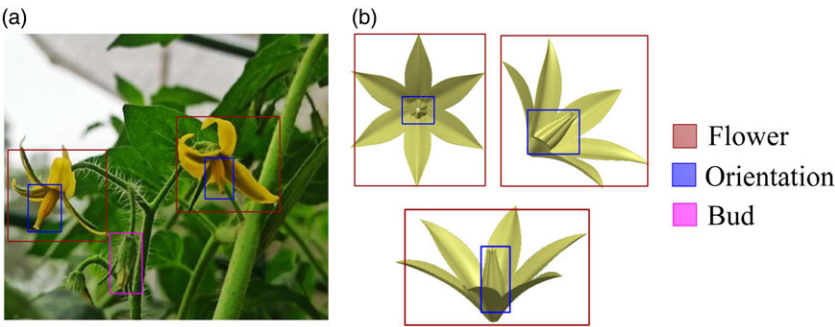

Figure 13. *Example of labeling the dataset (a) roboflow dataset and (b) CAD synthetic dataset).*

3.3. Results and validation

3.3.1. Deep learning model

YOLOv8, a deep-learning model, embodies the cutting-edge progress in object detection. It offers a remarkable combination of high accuracy, real-time inference speed, and a compact size, making it an appealing option for object detection applications. Illustrated in Figure 14 [38], the architecture of this detector is based on the CSPDarknet53 feature extractor, enhanced with a novel C2f module.

In our work, we made optimizations to the detection model to enable parallel processing, enabling concurrent execution of tasks such as tomato flower orientation and classification. By integrating the detector into our ITFP system, we equip it with detection capabilities, facilitating precise and efficient flower pollination. This integration enhances the system's performance and enables it to carry out its tasks effectively and efficiently.

The YOLOv8 model was set to undergo a maximum of 1000 training epochs. However, during the training process, the model reached its criteria for early stopping after 151 epochs. This early stopping mechanism allowed the training to conclude prematurely, indicating that the model had achieved satisfactory performance. To optimize the training process and facilitate effective parallelization, a batch size

Table III. *Performance evaluation comparison with different backbone architectures.*

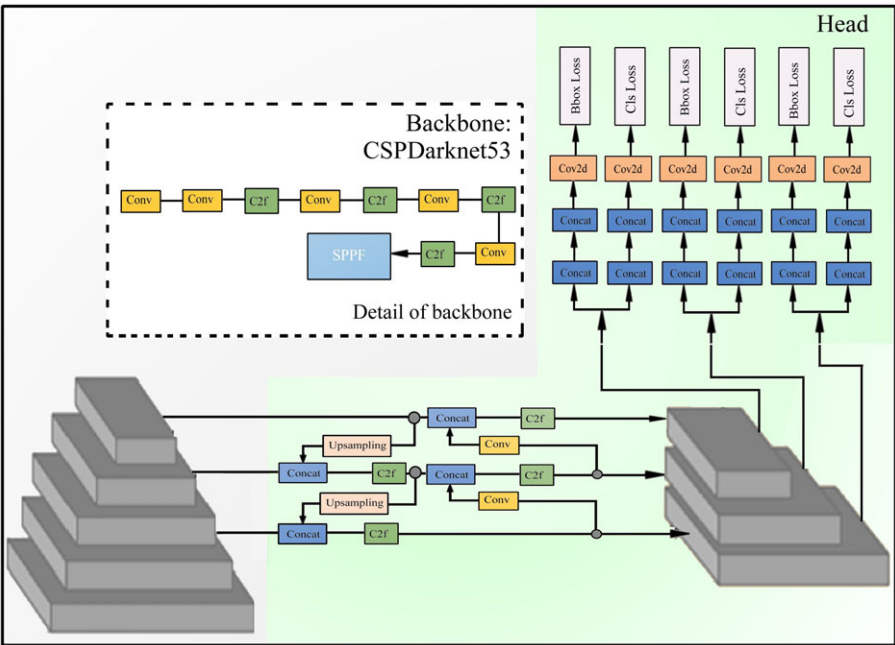| Architecture (Backbone)	mAP50
Yolo8n (default backbone)	0.912
Yolo8n (ResNet-50)	0.835
Yolo8n (ResNet-101)	0.849

Figure 14. *YOLOv8 diagram of model architecture.*

of 16 was utilized. This batch size enabled efficient parallel processing, enhancing the training speed and overall effectiveness of the training procedure.

In order to mitigate the risk of overfitting, a combination of online and offline data augmentation techniques was employed. Offline augmentation methods involve rotating the images within a specified range to simulate different flower orientations. The corresponding bounding box labels were updated accordingly to ensure accurate annotations. Additionally, other online augmentation techniques such as color augmentation, translation, and scaling were applied to introduce further variations into the dataset. These techniques increased the dataset's diversity and aided in training a more versatile model. The dataset was divided into training, validation, and testing sets in an 8:1:1 ratio. The 8:1:1 ratio allowed to allocate a significant portion of the data for training the model, while reserving separate subsets for rigorous testing and validating the model's generalization capabilities. This approach ensured a comprehensive evaluation of the model's performance and its readiness for deployment in real-world scenarios for pollination purposes. In order to determine the most suitable backbone architecture, a comparative experiment was conducted to evaluate the performance of ResNet-50 and ResNet-101 with the default backbone of the YOLOv8-nano, as indicated in Table III. The experimental design maintained consistency across all hyperparameters to ensure a fair comparison. Each backbone architecture was assessed using the same evaluation criteria to measure its performance.

The study primarily concentrated on utilizing the YOLOv8-n architecture (default backbone), which demonstrated superior performance compared to the other tested architectures. Specifically, the default

Table IV. *Mean average precision (mAP) values.*

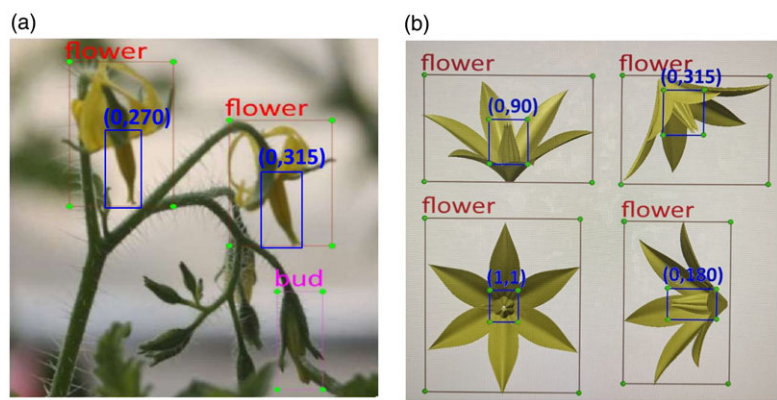| Orientation		Classification	
Class	**AP(%)**	**Class**	**AP(%)**
(1,1)	91.2	Flower	94.8
(0,0)	93.6		
(0,45)	85.4	Bud	86.1
(0,90)	77.8		
(0,135)	83.7		
(0,180)	97.1		
(0,225)	96.3		
(0,270)	84.6		
(0,315)	98.8		
mAP(%)	**91.2**	**mAP(%)**	**89.4**

Figure 15. *Some examples of correctly classified results (a) real flower images dataset, and (b) CAD generated synthetic dataset.*

backbone achieved a mean average precision (mAP50) of 91.2 %. In contrast, the ResNet-50 and ResNet-101 backbones yielded mAP50 scores of 83.5 % and 84.9 % respectively. These results highlight the effectiveness of the YOLOv8n architecture's default backbone in achieving higher accuracy in object detection tasks compared to the alternative backbone architectures.

The evaluation results, presented in Table IV, provide additional insights into the overall mAP of 91.2 %. Specifically, the flower orientation classification achieved a mAP of 94.8 %, while flower and bud classification achieved a mAP of 86.1 %, as indicated in Table IV. This outcome can be attributed to the higher prevalence of instances related to flower orientation within the dataset.

The evaluation of the model's performance, as depicted in Figure 15, showcases the accurate classification results. In the first instance, Figure 15(a) demonstrates the correct identification of right-facing flowers with orientations of (0,270) and (0,315). Additionally, both the flowers and buds were accurately recognized from real plant image data, indicating the model's effectiveness in detecting and distinguishing these elements. In the second case, as shown in Figure 15(b), the model accurately classified the flower orientations according to the labels provided for the CAD synthetic dataset. This successful classification further reinforces the model's capability to accurately identify and classify flower orientations, even when working with synthetic data.

Previous studies on robotic pollination did not report specific findings related to the detection of tomato flower orientations similar to our research. However, a related study conducted by Strader et al. [8] involved the classification of bramble flowers into three classes based on their position relative

Table V. *Comparison of YOLOv8n model weight size with different backbone architectures models.*

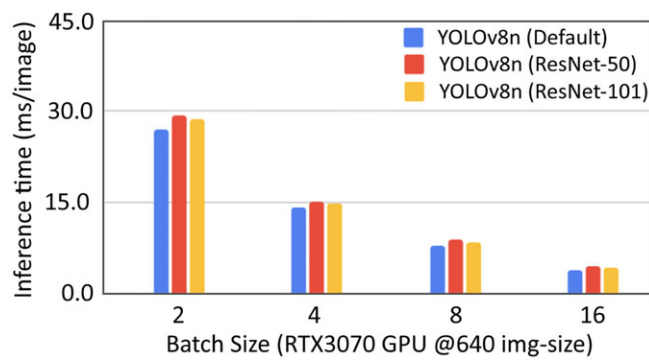| Algorithm	Custom trained model weight size (MB)
YOLOv8n	10.5
YOLOv8n-ResNet50	97.8
YOLOv8n-ResNet101	170.2

Figure 16. *Inference time vs batch size.*

to the camera (front, left, and right side). Although this work also dealt with flower classification, the classification categories and plant species differ from our research on tomato flowers. In Table IV, we present the accuracy results for our model's classification of the nine different flower orientation classes. These findings demonstrate the performance and effectiveness of our approach in accurately classifying tomato flower orientations, filling a gap in the existing literature regarding this specific aspect of robotic pollination. Figure 16 illustrates the relationship between inference time and batch size for the proposed YOLOv8n model, considering different backbone architectures such as ResNet-50 and ResNet-101. It provides a visual representation of how the model's inference time varies as the batch size changes, offering insights into the performance characteristics of the YOLOv8n model with different backbone architectures. Figure 16 clearly demonstrates that the YOLOv8n (Default) model exhibits shorter inference times compared to other models when considering batch sizes of 2, 4, 8, and 16. The graph provides a visual representation of the superior performance of the YOLOv8n (Default) model in terms of faster inference times across these batch sizes. Table V provides a comparison of the weight sizes between the YOLOv8n model and models with different backbone architectures. It clearly demonstrates that the weight size of the YOLOv8n model is relatively smaller compared to the weight sizes of the ResNet-50 and ResNet-101 models. This comparison highlights the advantage of the YOLOv8n model in terms of a more compact weight size, which can have implications for storage requirements, computational efficiency, and easy to deploy in hardware.

Table VI presents a comparison of the overall classification performances between our proposed method and the approach by [8]. In their study, Strader et al. achieved a precision of 79.3 % for the frontal view of bramble flower detection, while the precision for the other classes was notably lower at 74.3 % and 59.5 %, respectively. On contrary, our proposed work achieved significantly higher precision rates of 90.8 %, 90.6 %, and 93.5 % for the classes. Furthermore, the recall rates in Strader et al.'s study were also comparatively lower than those in our research. Our approach demonstrated higher recall rates, indicating a better ability to detect and capture the desired flower orientations accurately. These findings highlight the enhanced performance and suitability of our method for accurately detecting and classifying flower orientations. Figure 17 presents the average performance metrics of the proposed

Table VI. *Comparison of orientation detection accuracies and recall: Our approach versus Strader et. al (2019) [8].*

| Work done	Detection model	Class		Precision (%)	Average Precision(%)	Recall (%)	Average Recall (%)
[8]	Inception-v3	Center		79.3		83.3	
		Left direction		74.3		59.1	
		Right direction		59.5		61.1	
Proposed model	Yolov8-n	Center	(1,1)	90.8	90.8	89.6	89.6
		Left direction	(0,135)	79.2	90.6	77.8	87.2
			(0,180)	98.9		89.3	
			(0,225)	93.7		94.7	
		Right direction	(0,45)	86.2	93.5	80.2	92.1
			(0,0)	97.5		98.8	
			(0,315)	96.8		97.3	
		Up direction	(0,90)	69.3	69.3	71.5	71.5
		Down Direction	(0,270)	83.2	83.2	79.9	79.9
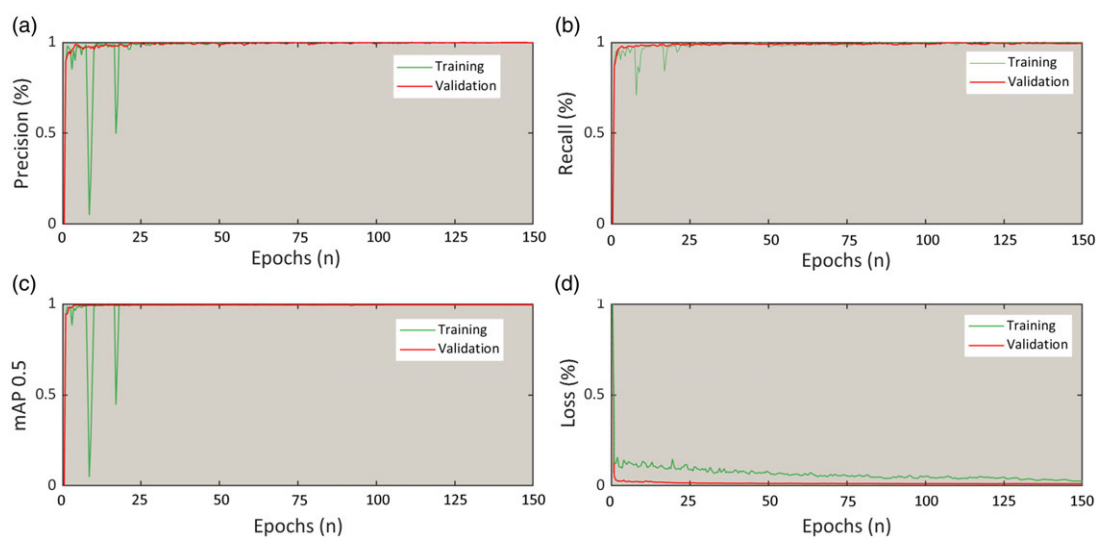
Figure 17. *Average performance parameters of proposed model (a) precision, (b) recall, (c) mAP 0.5, and (d) loss.*

model for the training and validation dataset. As depicted from Figure 17, the model performs good during training stage as compared to the validation dataset.

3.3.2. Depth estimation

For robotic pollination system, it is important to emphasize the significance of precision in depth measurement, particularly during the pollination operation. At this stage, the pollinator aligns itself to the flower center, denoted by the (1, 1) orientation, for the precise moment of pollination. In our study,
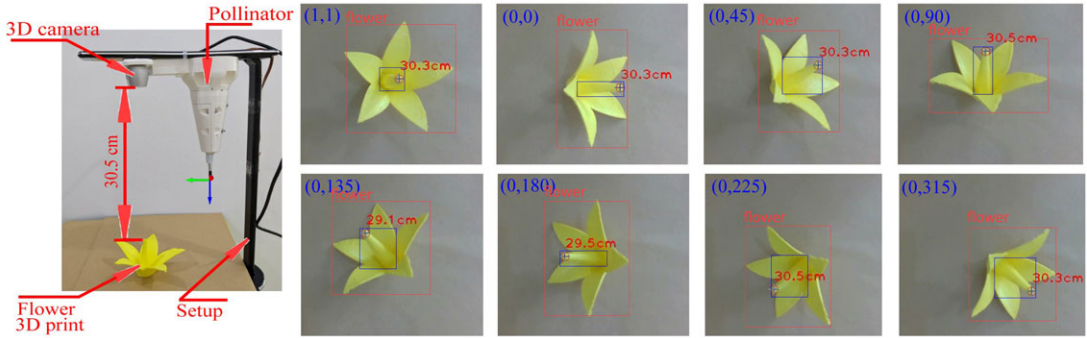
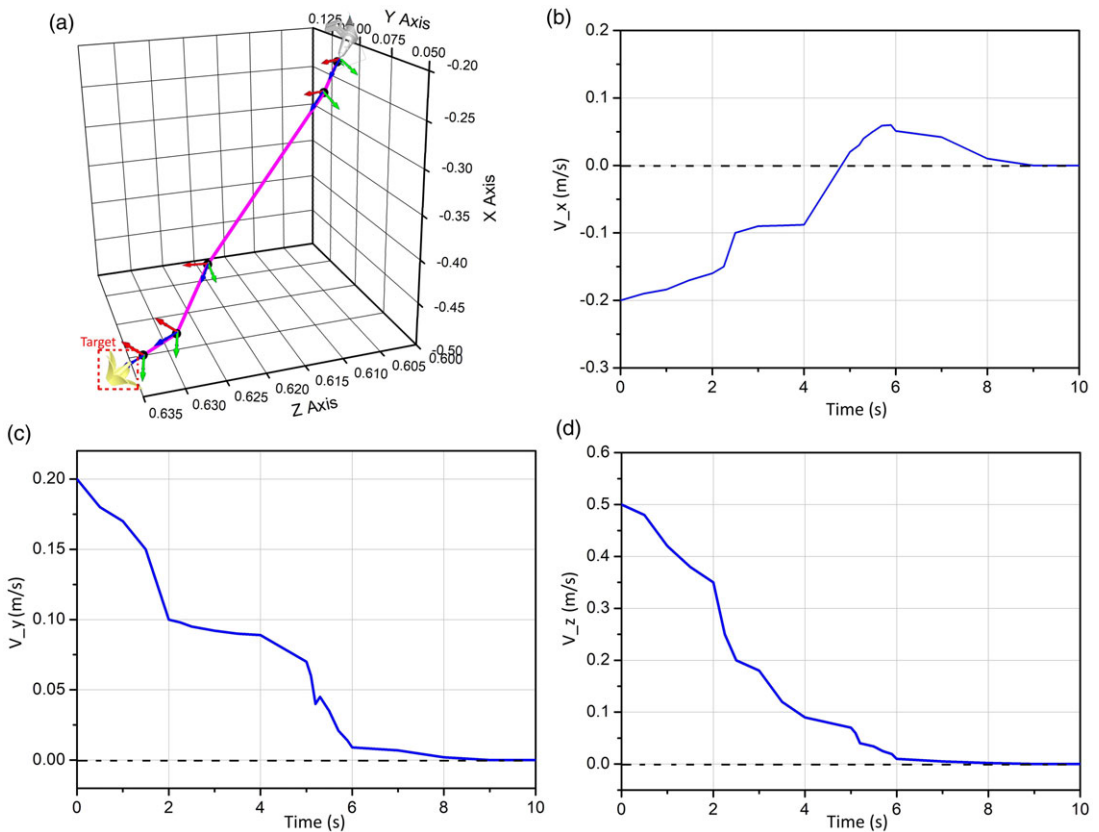**Figure 18.** *Flower orientation with depth estimation.*

Figure 19. *3D trajectory and translational velocity: starting and target orientations for the pollination process.*

we employed a laboratory setup to estimate depth while accounting for the variations in flower orientations, as depicted in Figure 18. However, these variations fall within an acceptable range for the initial stages of our pollination process. The focus during these preliminary stages was on approaching the flower rather than achieving absolute depth precision.
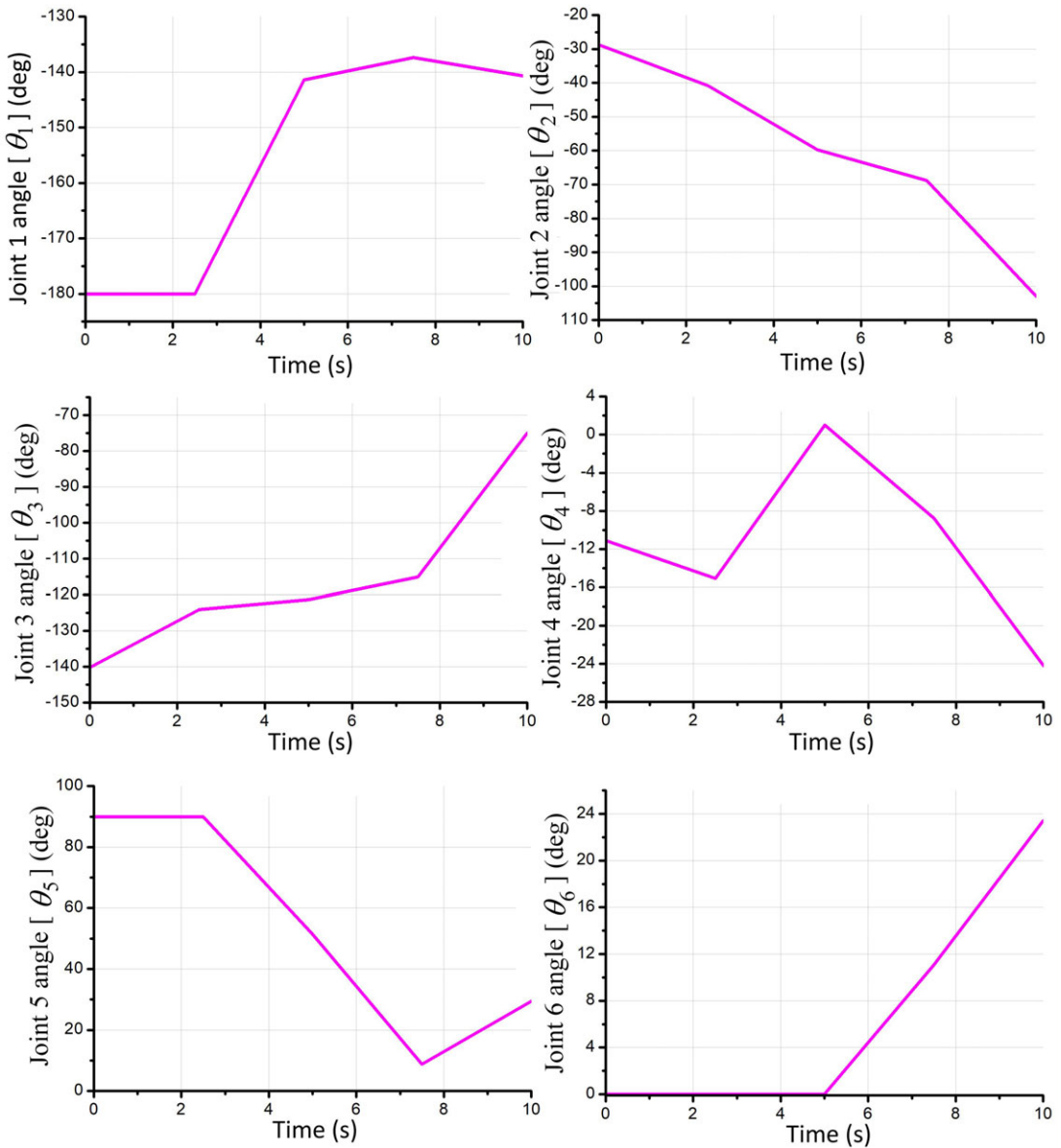
Figure 20. *3D trajectory and translational velocity: starting and target orientations for the pollination process.*

3.3.3. Results of proposed method

To assess the effectiveness of visual servo control, a controlled laboratory environment was created using 3D printed tomato flower plants and simulated tomato flowers in the robot simulation software (RoboDK and ROS). This setup allowed for the evaluation of visual servo control in the context of robotic pollination. The effectiveness of the visual servo method was evaluated by achieving a successful pose, defined as reaching the pollination point and aligning the axis of the pollinator precisely with the flower center. In each experiment, the flower was positioned arbitrarily with a distinct orientation, as illustrated in Figure 7.
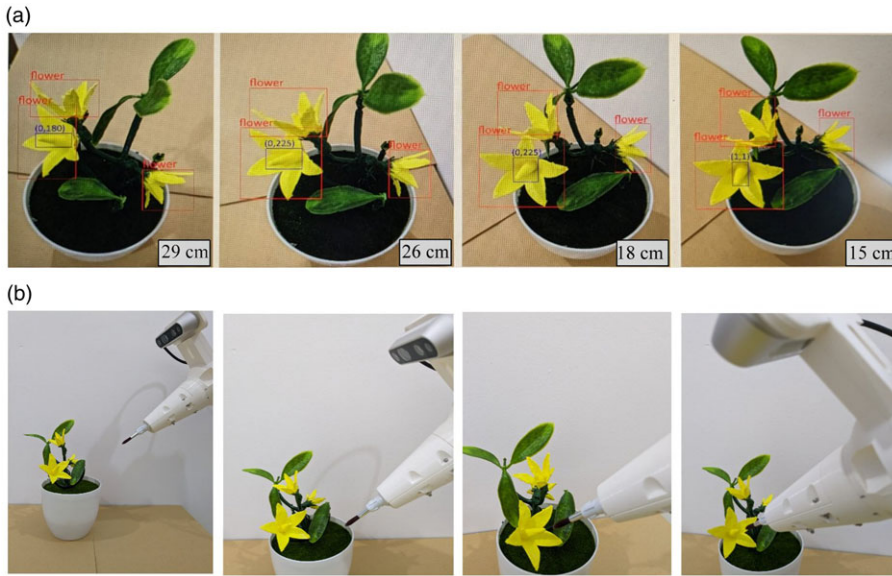
Figure 21. *(a) Sequential camera view images with corresponding depth information, and (b) Pollination done on 3D printed tomato flower plant.*

Figure 17 presents a comprehensive visualization of the pollination process for a tomato flower, showcasing the sequence from the initial stage to completion. In Figure 17(a), the path of the end effector's position in base coordinates is depicted, while Figure 17(b), (c), and (d) illustrate the translational velocity of the end effector. Analyzing the trajectory displayed in Figure 17(a), it is evident that the end effector approaches the target point in a smooth and rapid manner, predominantly in the downward direction. The translational velocity of the end effector, as depicted in Figure 17(b), (c), and (d), exhibits consistent and continuous movement without any notable drastic changes along the x, y, or z axes. The velocity profile demonstrates a smooth and steady approach towards the pollination point. Figure 17(b-d) illustrates that the translational velocity of the end effector (pollinator) gradually decreases after 6 s and approaches 0 m/s. The initial velocity from 0 s to 6 s represents the approach towards the target, after that the end effector detects the frontal face of the flower and aligns itself with the flower center.

Figure 19 provides a visual representation of the robot joint angles during the tracking of the 3D trajectory. As shown in the Figure 20, the robot joints 1, 2, and 3 exhibit smooth trajectories and play a significant role in approaching the target position. On the other hand, joints 4, 5, and 6 are primarily responsible for orienting the end effector with respect to the detected frontal face of the flower. These joints contribute to achieving the desired orientation necessary for the pollination process.

3.3.4. Evaluation of ITFP

To assess the performance validation of the proposed system, a series of experiments were conducted within a laboratory environment. The evaluation utilized a 3D printed tomato flower plant to test the capabilities of the Intelligent Tomato Flower Pollination (ITFP) system. Figure 21 provides an overview of the experiments conducted in the lab environment. Figure 21(a) showcases the deep learning-based flower orientation detection, along with the measurement of depth distance from 3D camera to detected flower. The image-based visual servoing approach is depicted in Figure 21(b). In our experiments, the ITFP model achieved a success rate of 83 %, demonstrating its ability to accurately and successfully perform pollination. A pollination cycle was considered successful when the robot reached a significantly closer position to the flower, aligning the pollinator with the flower's pistil (desired pose), and establishing physical contact between the tip of the pollinator and the flower center, confirming precise

placement.The accuracy of the pollination process was quantified by calculating the mean error, which indicated a minimal discrepancy of 1.19 cm between the desired depth and the pollination tool depth achieved.

As a future scope, the proposed Intelligent Tomato Flower Pollination (ITFP) system will undergo testing in a greenhouse setting to validate the performance of the deep learning-based detection model and image-based visual servoing approach for the pollination process. This step aims to further assess and refine the system's performance in a more realistic and dynamic agricultural environment.

4. Conclusions

This research introduces a robust system for Intelligent Tomato Flower Pollination (ITFP) system. The Deep learning model detect a precise flower orientation and classification, utilizing RGB and CAD-generated synthetic images of tomato flowers, resulting in a high mAP alongside an accurate depth estimation technique. The accuracy of depth estimation was verified through laboratory experiments involving various flower orientations.The integration of precise depth information from the 3D camera is a crucial element within our visual servoing loop, demonstrating its effectiveness in autonomously guiding the Universal robot (UR5) mounted pollinator setup to attain precise alignment with the flower's front view class (1,1). Notably, our proposed ITFP system has exhibited exceptional practical usability by achieving a high rate of success in pollination tasks conducted within the laboratory setting. This accomplishment not only highlights the efficacy of our approach but also holds promise for driving advancements in the realm of robotic pollination, potentially contributing to sustainable agricultural practices. In terms of future scope, the authors plan to address challenges related to flower overlapping and occlusions in order to develop more robust solutions. These cases will be taken into consideration to enhance the effectiveness of the proposed method and ensure its applicability in scenarios where flowers may overlap or be occluded.

References

[1] M. A. Aizen, S. Aguiar, J. C. Biesmeijer, L. A. Garibaldi, D. W. Inouye, C. Jung, D. J. Martins, R. Medel, C. L. Morales, H. Ngo, et al., "Global agricultural productivity is threatened by increasing pollinator dependence without a parallel increase in crop diversification," *Glob. Change Biol.* **25**(10), 3516–3527 (2019).

[2] M. A. Aizen, L. A. Garibaldi, S. A. Cunningham and A. M. Klein, "Long-term global trends in crop yield and production reveal no current pollination shortage but increasing pollinator dependency," *Curr. Biol.* **18**(20), 1572–1575 (2008).

[3] A. Dingley, S. Anwar, P. Kristiansen, N. W. Warwick, C.-H. Wang, B. M. Sindel and C. I. Cazzonelli, "Precision pollination strategies for advancing horticultural tomato crop production," *Agronomy* **12**(2), 518 (2022).

[4] T. Nishimura, "The effect of greenhouse pollination methods on consumers' willingness to pay for tomatoes in Japan," *J. Agric. Appl. Econ.* **53**(2), 186–208 (2021).

[5] G. Colucci, L. Tagliavini, A. Botta, L. Baglieri and G. Quaglia, "Decoupled motion planning of a mobile manipulator for precision agriculture," *Robotica* **41**(6), 1872–1887 (2023).

[6] J. T. Murphy, T. D. Breeze, B. Willcox, S. Kavanagh and J. C. Stout, "Globalisation and pollinators: Pollinator declines are an economic threat to global food systems," *People Nat.* **4**(3), 773–785 (2022).

[7] M. A. Broussard, M. Coates and P. Martinsen, "Artificial pollination technologies: A review," *Agronomy* **13**(5), 1351 (2023).

[8] J. Strader, J. Nguyen, C. Tatsch, Y. Du, K. Lassak, B. Buzzo, R. Watson, H. Cerbone, N. Ohi, C. Yang, et al., "Flower Interaction Subsystem for a Precision Pollination Robot," **In:** *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2019) pp. 5534–5541.

[9] R. Ayres and S. Miller, "Industrial robots on the line," *J. Epsilon Pi Tau* **8**(2), 2–10 (1982).

[10] S. Hutchinson, G. D. Hager and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.* **12**(5), 651–670 (1996).

[11] M. Ceccarelli, G. Figliolini, E. Ottaviano, A. S. Mata and E. J. Criado, "Designing a robotic gripper for harvesting horticulture products," *Robotica* **18**(1), 105–111 (2000).

[12] T. Dewi, P. Risma, Y. Oktarina and S. Muslimin. "Visual Servoing Design and Control for Agriculture Robot: A Review," **In:** *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE (2018) pp. 57–62.

[13] G. Flandin, F. Chaumette and E. Marchand, "Eye-in-Hand/Eye-to-Hand Cooperation for Visual Servoing," **In:** *Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3 (IEEE, 2000) pp. 2741–2746.

[14] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robot. Autom. Mag.* **13**(4), 82–90 (2006).

[15] F. Janabi-Sharifi, L. Deng and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Trans. Mechatron.* **16**(5), 967–983 (2010).

[16] X. Sun, X. Zhu, P. Wang and H. Chen, "A Review of Robot Control with Visual Servoing," **In:** *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER) 2018* (IEEE) pp.116–121.

[17] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette and P. Corke. "Training Deep Neural Networks for Visual Servoing," **In:** *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2018) pp. 3307–3314.

[18] J. Arents and M. Greitans, "Smart industrial robot control trends, challenges and opportunities within manufacturing," *Appl. Sci.* **12**(2), 937 (2022).

[19] A. Farizawani, M. Puteh, Y. Marina and A. Rivaie, "A Review of Artificial Neural Network Learning Rule Based On Multiple Variant of Conjugate Gradient Approaches," **In:** *Journal of Physics: Conference Series*, vol. 1529 (IOP Publishing, 2020) p. 022040.

[20] X. Mu, L. He, P. Heinemann, J. Schupp and M. Karkee, "Mask R-CNN based apple flower detection and king flower identification for precision pollination," *Smart Agric. Technol.* **4**, 100151 (2023).

[21] M. N. Ratnayake, A. G. Dyer and A. Dorin, "Towards Computer Vision and Deep Learning Facilitated Pollination Monitoring for Agriculture," **In:** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021) pp. 2921–2930

[22] M. Yurdakul, İ. Ataباş and Ş. Taşdemir, "Flower Pollination Algorithm-Optimized Deep CNN Features for Almond (Prunus Dulcis) Classification," **In:** *2024 International Conference on Emerging Systems and Intelligent Computing (ESIC)*. IEEE (2024) pp. 433–438.

[23] J. Davis, R. Ramamoorthi and S. Rusinkiewicz. "Spacetime Stereo: A Unifying Framework for Depth from Triangulation," **In:** *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*. IEEE, vol. 2, (2003) pp. II–359.

[24] Y. Salih, A. S. Malik and Z. May, "Depth Estimation Using Monocular Cues from Single Image," **In:** *2011 National Postgraduate Conference*. IEEE (2011) pp. 1–4.

[25] B. A. Griffin and J. J. Corso, "Depth from Camera Motion and Object Detection," **In:** *In, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021) pp. 1397–1406.

[26] K. Ahmad, J.-E. Park, T. Ilyas, J.-H. Lee, J.-H. Lee, S. Kim and H. Kim, "Accurate and robust pollinations for watermelons using intelligence guided visual servoing," *Comput. Electron. Agr.* **219**, 108753 (2024).

[27] N. Masuda, M. M. Khalil, S. Toda, K. Takayama, A. Kanada and T. Mashimo, "Development of a Flexible Multi-degrees-of-Freedom Robot for Plant Pollination," **In:** *2024 IEEE/SICE International Symposium on System Integration (SII)*. IEEE (2024) pp. 416–421.

[28] G. Li, R. Suo, G. Zhao, C. Gao, L. Fu, F. Shi, J. Dhupia, R. Li and Y. Cui, "Real-time detection of kiwifruit flower and bud simultaneously in orchard using yolov4 for robotic pollination," *Comput. Electron. Agr.* **193a**, 106641 (2022a).

[29] K. Li, Y. Huo, Y. Liu, Y. Shi, Z. He and Y. Cui, "Design of a lightweight robotic arm for kiwifruit pollination," *Comput. Electron. Agr.* **198b**, 107114 (2022b).

[30] X. Cui, Z. Guan, K. L. Morgan, K.-M. Huang and A. M. Hammami, "Multitiered fresh produce supply chain: The case of tomatoes," *Horticulturae* **8**(12), 1204 (2022).

[31] T. Xu, X. Qi, S. Lin, Y. Zhang, Y. Ge, Z. Li, J. Dong and X. Yang, "A neural network structure with attention mechanism and additional feature fusion layer for tomato flowering phase detection in pollination robots," *Machines* **10**(11), 1076 (2022).

[32] G. Jocher, A. Chaurasia and J. Qiu, Yolo by Ultralytics (2023). https://github. com/ultralytics/ultralytics.

[33] N. Ohi, K. Lassak, R. Watson, J. Strader, Y. Du, C. Yang, G. Hedrick, J. Nguyen, S. Harper, D. Reynolds, et al., "Design of an Autonomous Precision Pollination Robot," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2018) pp. 7711–7718.

[34] M. H. Conde, K. Hartmann and O. Loffeld, "Adaptive high dynamic range for time-of-flight cameras," *IEEE Trans. Instrum. Meas.* **64**(7), 1885–1906 (2014).

[35] S. M. Ayaz, M. Y. Kim and J. Park, "Survey on zoom-lens calibration methods and techniques," *Mach. Vision Appl.* **28**(8), 803–818 (2017).

[36] H. Dwyer and Nelson, Roboflow (version 1.0) (2024).

[37] A. F. Fuentes, S. Yoon, J. Lee and D. S. Park, "High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank," *Front. Plant Sci.* **9**, 386083 (2018).

[38] RangeKing, Yolov8 (2023). https://github.com/ultralytics/ultralytics/issues/189.