Biosystems Engineering 243 (2024) 57–69

Contents lists available at ScienceDirect

Biosystems Engineering

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

# 3D pose estimation of tomato peduncle nodes using deep keypoint detection and point cloud

Jianchao Ci *, Xin Wang, David Rapado-Rincón, Akshay K. Burusa, Gert Kootstra

*Agricultural Biosystems Engineering Group, Department of Plant Sciences, Wageningen University and Research, P.O. Box 16, Wageningen, 6700AA, the Netherlands*

ARTICLE INFO

*Keywords:*
Deep learning
Peduncle
Keypoint detection
Point cloud
Detectron2
Pose estimation

ABSTRACT

Greenhouse production of fruits and vegetables in developed countries is challenged by labour scarcity and high labour costs. Robots offer a good solution for sustainable and cost-effective production. Acquiring accurate spatial information about relevant plant parts is vital for successful robot operation. Robot perception in greenhouses is challenging due to variations in plant appearance, viewpoints, and illumination. This paper proposes a keypoint-detection-based method using data from an RGB-D camera to estimate the 3D pose of peduncle nodes, which provides essential information to harvest the tomato bunches. Specifically, this paper proposes a method that detects four anatomical landmarks in the colour image and then integrates 3D point-cloud information to determine the 3D pose. A comprehensive evaluation was conducted in a commercial greenhouse to gain insight into the performance of different parts of the method. The results showed: (1) high accuracy in object detection, achieving an Average Precision (AP) of AP@0.5=0.96; (2) an average Percentage of Detected Joints (PDJ) of the keypoints of PhDJ@0.2 = 94.31%; and (3) 3D pose estimation accuracy with mean absolute errors (MAE) of 11° and 10° for the relative upper and lower angles between the peduncle and main stem, respectively. Furthermore, the capability to handle variations in viewpoint was investigated, demonstrating the method was robust to view changes. However, canonical and higher views resulted in slightly higher performance compared to other views. Although tomato was selected as a use case, the proposed method has the potential to be applied to other greenhouse crops, such as pepper, after fine-tuning.

## 1. Introduction

Growing vegetables in greenhouses can significantly extend the production period of plants, which increases yields and brings economic benefits to the owner of the greenhouse. Most greenhouse crops, such as tomato, cucumber, and bell pepper require selective maintenance and harvesting, making production activities labour-intensive. This poses a huge challenge for production because the labour cost is high and labour is scarce (Benavides et al., 2020). A higher level of automation and robotisation is seen as a good solution to replace human labour in greenhouse production, and many studies have been proposed to more effectively involve robotics in greenhouse operations, such as harvesting (Bac et al., 2014; Ji et al., 2012; Yoshida et al., 2018), monitoring (Halstead et al., 2018), and phenotyping (Boogaard et al., 2020; Virlet et al., 2016; Vit et al., 2020).

Most greenhouse operations require the robotic perception system to detect the relevant parts of the plant (e.g., fruits, peduncles, and main stem) and to collect sufficient information about them to perform manipulations, such as harvesting, deleafing, and pruning. The perception process is adversely affected by variation in the appearance of the plant and occlusions that are significantly present in complex greenhouse environments (Afonso et al., 2019; Kootstra et al., 2021). Variation results from natural variation in the growth of plants, creating differences in the morphology and appearance of the plant parts, as well as from environmental influences, such as changing illumination. Occlusions happen frequently in the cluttered greenhouse environments, where plant parts are frequently entirely or partially obstructed by other plant parts.

To overcome the challenge of variation and improve perception performance, in recent years, deep-learning-based vision systems have been extensively used in agricultural scenarios. Compared to traditional methods that use handcrafted features, deep-learning methods include the feature extraction as part of the end-to-end learning process, which has been shown to perform much better in terms of accuracy and robustness to variation. Kamilaris and Prenafeta-Boldú (2018) reviewed 40 studies using deep learning in agricultural applications and

* Corresponding author.
*E-mail address:* jianchao.ci@wur.nl (J. Ci).

https://doi.org/10.1016/j.biosystemseng.2024.04.017
Received 2 September 2023; Received in revised form 26 April 2024; Accepted 27 April 2024
Available online 13 May 2024

concluded that deep neural networks are superior to other methods. Bargoti and Underwood (2017), for instance, used a deep neural network for object detection of several different fruits, including mangoes, apples, and almonds in orchards, while Sa et al. (2016) used a similar method on a combination of colour and near infra-red images to successfully detect seven different fruit types in greenhouse environments. Boogaard et al. (2020) used deep object detection to detect the leaf and fruit nodes of cucumber plants. A deep neural network for instance segmentation of grape bunches in orchards was used in the work of Santos et al. (2020). The same method was used by Shi et al. (2019) to segment images of tomato seedlings in stem, node, and the individual leaf. Kang and Chen (2020) proposed a new network for instance segmentation of apples. These detection algorithms are able to locate different plant parts in 2D images. However, to perform a robotic operation, 3D information about the position and orientation of the objects is needed so that the robot can bring a tool to the desired location and orientation.

Often, the 3D pose of an object is defined by the 3D position and the 3D orientation of the object. This was used, for instance by Eizentals and Oka (2016) and Lehnert et al. (2016) to represent the pose of bell peppers. In both studies, the pose of the fruit was estimated by fitting a 3D template to the acquired partial point cloud of the fruit. Li et al. (2018) estimated the pose of peppers by finding the symmetry axes. A deep neural network was proposed in Wagner et al. (2021) to estimate the 3D pose of strawberries directly from the colour-and-depth (RGB-D) image. Kang et al. (2020) proposed a point-based neural network to learn to estimate the grasp pose directly from the point cloud of the fruit.

In the animal domain, a different definition of object pose is used. There, computer-vision methods, inspired by human-pose detectors, estimate the pose of the animal by a set of keypoints, representing anatomical landmarks on the animal (X. Li et al., 2019; Mathis et al., 2020; Pereira et al., 2019; Russello et al., 2022). The pose provides information on the location of these keypoints, as well as the relations between the keypoints. The advantages of this representation are that it can be used for articulated objects and that they are more robust to occlusion, as the non-occluded keypoints can still be detected to represent the object pose. This makes it interesting for pose estimation of plant parts, but it has hardly been explored to date, with the exception of Zhang et al. (2022) who used keypoint detection to get the pose of tomato bunches using a combination of stacked hourglasses and an object-detection network. The method localised a set of 2D keypoints on the tomatoes, peduncle and stem and then converting them into 3D keypoints using spatial information provided by the depth camera. The accuracy of the 3D pose estimation, however, was not quantitatively evaluated, so it is unclear if the method can be used for robotic operation.

This study focusses on the 3D pose estimation of peduncle nodes on tomato plants to provide information for fruit harvesting. The peduncle connects the fruit with the main stem. To avoid fruit damage and extend shelf life, tomato trusses are normally detached from the plant by cutting the peduncle. Some other studies tried to detect and localise the peduncle relying on the detection of other plant parts (such as the fruits and branches), then determining the peduncle according to specific morphological relationships between them (Sa et al., 2017; Yoshida et al., 2020). The downside of these methods is that they pose assumptions on the pose of the peduncle. Some applications, for instance, assumed the peduncle to be on top of the fruit, requiring the fruit cluster to be downwards and vertical (Liang et al., 2020; Luo et al., 2016). This limits the application, as the tomato peduncles can be oriented in any direction. In the work of Boogaard et al. (2020), nodes on cucumber plants were directly detected in the images by a deep neural network (DNN) without assumptions on the relations to other plant parts. In the work of Rong et al. (2022), a DNN for instance-segmentation method was used to predict an image mask for tomato peduncles, after which a handcrafted algorithm was used to determine the 3D cutting position. However, both approaches focused solely on the peduncle without considering the relationship to the main stem, which can lead collisions between the robot and the plant during harvesting (G. Lin et al., 2019). Kim et al. (2023) used a more elaborate definition of the pose of a tomato truss, by detecting keypoints at the centre of all tomatoes on the truss, at each calyx, and at the pedicels. They developed a method to detect these points in cropped 2D images of a truss but did not evaluate the accuracy in 3D. Furthermore, their work is of interest for harvesting individual tomatoes, while the focus of this study is harvesting tomato trusses. Zhang et al. (2023) developed the Tomato Pose Method (TPMv2), utilising a deep neural network for simultaneous 2D and 3D keypoint detection on tomato trusses. Their method includes the pose of the peduncle and local direction of the stem in 3D, needed for robotic harvesting of the truss. However, the direct prediction of 3D positions requires additional effort in the annotation of the training data and the more complex neural network requires additional data to train the model. Because the results were not compared to ground-truth geometrical measurements in the real world, it is unclear what the achieved pose accuracy was. If sufficient training data is available and when the additional information about the location of the tomatoes on the truss is needed, the work of Zhang et al. (2023) offers a useful approach. However, in the light of variation in the appearance of plants of different cultivars, the detection system will likely need to be retrained for specific situations, requiring less data-hungry methods.

In this paper, a keypoint-detection-based method was proposed, combining the benefits of a 2D keypoint detection network with point-cloud processing techniques to estimate the full 3D pose of peduncle nodes of tomato plants, enabling robotic operations. As shown in Fig. 1, the method was built upon three steps: (1) a deep keypoint detector was used for robust detection of the peduncle node and its pose in 2D images, (2) the detected keypoints were projected to the aligned 3D point cloud, and (3) the 3D pose was estimated. To provide relevant information about the pose and nearby stem segments needed for robotic harvesting, four keypoints was defined: one on the node, one on the peduncle, and two on the main stem. The method was thoroughly evaluated on three aspects: the node detection, the 2D pose estimation, and the 3D pose estimation, each aspect impacting the accuracy of the final estimated 3D pose. The first two analyses were done on the image level, while the last analysis evaluated the accuracy of estimated 3D pose in real-world coordinates compared to manual measurements. Furthermore, because the success of peduncle node detection might depend on the viewpoint, the result for different camera poses relative to the target object was analysed. This provides insight for optimal viewpoint planning, such as implemented in, e.g., Lehnert et al. (2019) and Burusa et al. (2023), who aimed to find the optimal viewpoint and utilised a robot arm with a sensor mounted on the end effector to reach this viewpoint for data collection in downstream tasks. All images used in this project (both training and validation) are raw images directly obtained from the camera, which provided a realistic evaluation of the model's performance in real-world applications.

## 2. Materials and methods

### 2.1. Materials

#### 2.1.1. Camera setup and dataset acquisition

The data were collected at a commercial tomato greenhouse located in Beek en Donk, Netherlands. This greenhouse is equipped with a fully automatic climate control system, including temperature, humidity, and lighting control. The tomato plants were grown in a high-wire pattern and the main stem was trained to grow vertically along the wire. The tomato variety used in this research was 'Tasty Tom', which has around 8 fruits per tomato truss, a peduncle diameter of around 7 mm at maturity. Standard operational management was applied, including the removal of lower leaves around the ripe tomato trusses (de-leafing) to reduce pests and diseases, and to remove leaves that are no longer contributing to photosynthesis. No modifications of the crop were made

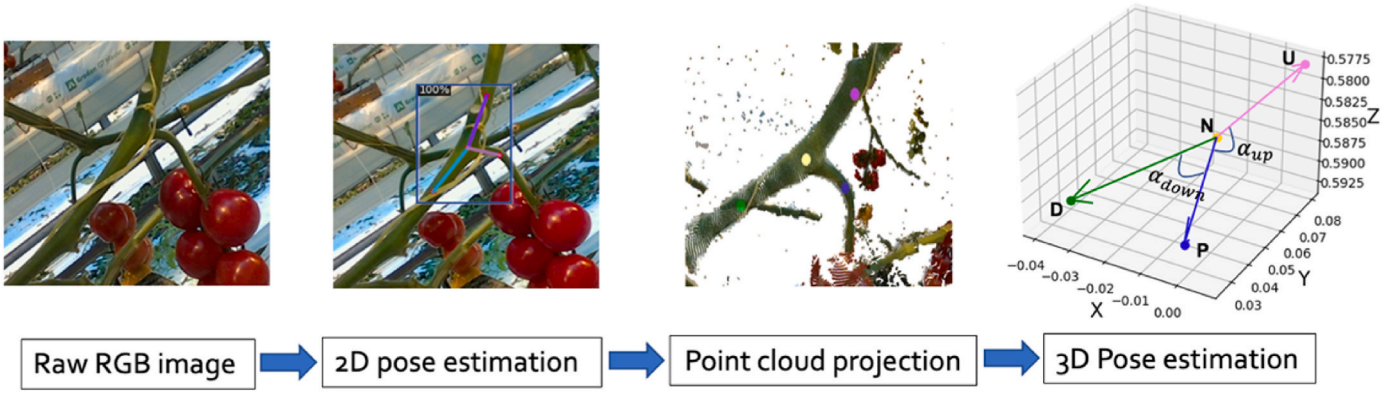


**Fig. 1.** Three stages, 2D pose estimation, 3D point cloud projection, and 3D pose estimation.

for the purpose of this project. Although de-leafing reduces the level of occlusion, the task to detect the pose of the peduncle nodes is still challenging due to different viewing poses, variation in the appearance of the nodes, self-occlusion, and clutter of other neighbouring plants.

A RealSense L515 LiDAR camera was used to capture both RGB images and aligned point clouds, both of which had a resolution of 1280x720. The camera was connected to an Alienware M15 R3 laptop with an Intel i7 10750H CPU and Nvidia GeForce RTX-2070s GPU, running on Windows 10 using the Pyrealsense2 package for image acquisition. Data was collected under ambient lighting, with the camera being hand-held between 50 cm and 80cm from a peduncle node, which was usually centred in the image. In addition to this node, a large number of peduncle nodes from other plants were present in the background of each image. Data collection occurred between October to December, in the 6th, 10th, and 11th weeks after planting.

The dataset included 648 RGB images, out of which 503 were used for model training and 145 for testing (Table 1). The training set included 614 annotated peduncle nodes, with 1–3 nodes per image, while the test set contained 145 annotated peduncle nodes, with a single peduncle node annotated per image. Although the test images regularly contain multiple nodes and the proposed method is also able to detect multiple peduncle nodes, the test focused on the detection of the peduncle closest to the robot, hence only one peduncle was annotated. Apart from the peduncle nodes, also leaf nodes (petiole node) were visible in the images. The detection algorithm had to ignore those and detect only the peduncle nodes.

The test set contained a subset of 95 images of 19 nodes from five different viewpoints to be able to analyse the detection performance for different poses of the peduncle in the image. These viewpoints were named V1 (canonical view), V2 (higher view), V3 (lower view), V4 (leftwards view), and V5 (rightwards view), all directed toward the peduncle node (see Fig. 2). V1 is horizontal and perpendicular to the direction of the plant rows, while the positions of V2–V5 were approximately determined, each having an angle of around $60^o \pm 15^o$ with respect to V1. Although the positions of the five viewpoints were determined approximately in this study, they still provide insights into the impact of camera pose relative to the peduncle node on the model's performance.

#### 2.1.2. Ground-truth annotation in 2D

The RGB images were annotated using COCO annotator, an open-source web-based image annotation tool for object detection, segmentation, and keypoint detection (Brooks, 2019). As shown in Fig. 3, four keypoints were labelled to express the pose of the peduncle node, where three points 'U', 'N', and 'D' were set at the upper, node, and lower positions of the main stem, respectively, and one point 'P' was set on the peduncle where it begins to bend. These four keypoints provide comprehensive pose information about the peduncle and the stem. The position and orientation of the peduncle are derived from the vector $\overrightarrow{NP}$, while those of the stem are determined by the vectors $\overrightarrow{DN}$ and $\overrightarrow{NU}$. Only the visible keypoints were labelled. It is important to note that due to the lack of visible features, the positions of points 'U' and 'D' were determined based on Euclidean distance to 'N' at 80 pixels. The distance was determined arbitrarily and did not consider the distance from the camera lens to the object. The images shown in Fig. 3 are not raw images with a resolution of 1280 x 720, but are cropped images of single peduncle nodes to give a clear presentation of the annotation results.

**Table 1**
The composition of the data set.

| | Total RGB | Total Nodes | Align point cloud |
|---|---|---|---|
| **Training set** | 503 | 614 (1–3 per image) | No |
| **Test set** | 145 | 145 | Yes |

#### 2.1.3. Ground-truth measurement in 3D

To evaluate the pose-estimation method, the ground-truth angles between peduncle and main stem were manually measured. As shown in Fig. 1, the upper angle between the peduncle and the main stem was defined as $\alpha_{up}$, which was calculated as $\alpha_{up} = \angle\left(\overrightarrow{NP}, \overrightarrow{NU}\right)$, while the lower angle was defined as $\alpha_{down}$, calculated as $\alpha_{down} = \angle\left(\overrightarrow{NP}, \overrightarrow{ND}\right)$. Three people measured the angles with protractors, and the average value was used. The standard deviations between the observers for all node measurements ranged between 1.73°-7.64° for $\alpha_{up}$, and between 0°-8.66° for $\alpha_{down}$.

### 2.2. Peduncle node pose estimation

#### 2.2.1. Overview

As shown in Fig. 1, the pose estimation system was designed with three stages: (1) 2D pose estimation, (2) point cloud projection, and (3) 3D pose estimation. The system takes the 2D colour images and the aligned point cloud as input and estimates the 3D pose of the peduncle node as output. The first stage uses a deep keypoint detector on colour images to localise the 4 pre-defined keypoints ('U', 'N', 'D', 'P'). In the second stage, the 2D keypoints are projected onto the point cloud to convert them into 3D Cartesian space. Finally, the 3D poses are estimated in the third stage. More details about the three stages are provided in the next subsections.

#### 2.2.2. Pose estimation in 2D

*2.2.2.1. Keypoints R–CNN architecture.* This study employed the Keypoints R–CNN network (He et al., 2017), based on the Detectron2 implementation (Wu et al., 2019), for pose estimation on colour images. Keypoint R–CNN takes a camera image as input and predicts the

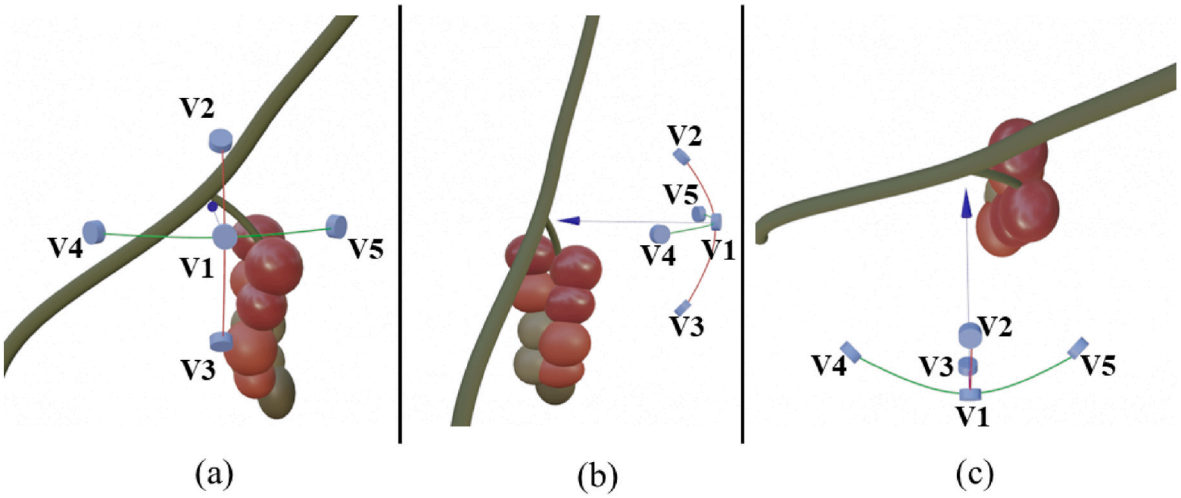


**Fig. 2.** The distribution of the 5 viewpoints from (a) the front view, (b) the side view, (c) the top view. V1 represents the canonical view, V2 represents the higher view, V3 represents the lower view, V4 represents the leftwards view, and V5 represents the rightwards view. All viewpoints are directed towards the peduncle node.

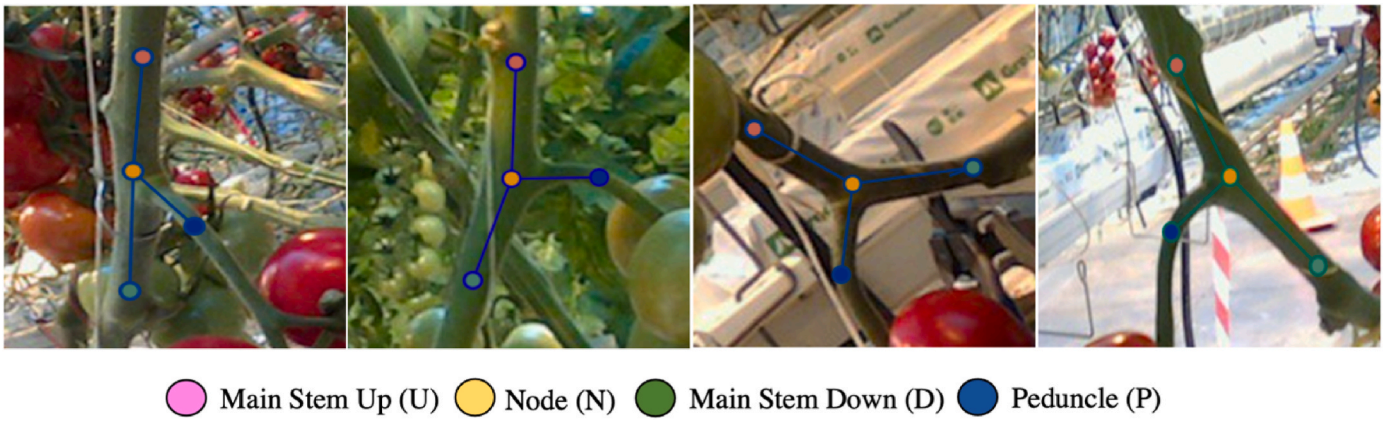


**Fig. 3.** Samples of annotated images. Four keypoints were annotated to express the pose of the peduncle node. The 3 points 'U', 'N', and 'D' were set at the upper, node, and lower positions of the main stem, while 1 point 'P' was set on the peduncle.

bounding box of each peduncle node, including the location of the four keypoints in the image. As shown in Fig. 4, the network is comprised of two stages. In the first stage, a ResNet50-FPN (50-layer Residual Network) (He et al., 2016) combined with a Feature Pyramid Network (T.-Y. Lin et al., 2017) was used as backbone to extract and fuse features from the input image. These feature maps were then processed by a region proposal network (RPN) (Ren et al., 2015) to generate a large set of regions of interest (ROI) that possibly contain target objects. In the second stage, the ROI proposals were used to crop the feature maps, which were subsequently processed by two independent networks: (1) an object-detection network featuring multiple Fully Connected (FC) layers to predict the object's bounding box, along with a class label and

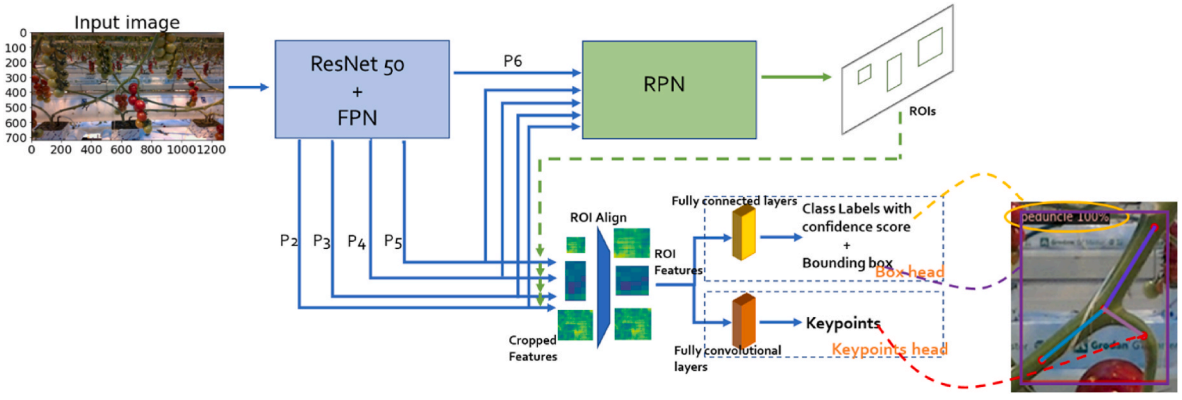


**Fig. 4.** The structure of the Keypoints-RCNN algorithm. ResNet50-FPN is used to extract features, and RPN is utilised to generate ROI proposals by taking feature maps as input. The cropped ROI features are fed into two branches to predict class labels, bounding boxes, and keypoints.

confidence score; and (2) a Fully Convolutional Network (FCN) with multiple convolutional layers to predict the location of the keypoints associated with the detected object.

*2.2.2.2. Training of the network.* The model was trained on a PC with an Nvidia-GPU RTX-3090 and Intel-CPU i9-10920X on Linux Ubuntu 18.04, utilising a training set of 503 images, as described in section 2.1.1. To speed up the process, transfer learning was applied by using a pre-trained network on the COCO dataset (T.-Y. Lin et al., 2014) and fine tuning on collected training set. The training process consisted of 30,000 iterations with a batch size of 8, resulting in approximately 477 epochs. The Stochastic Gradient Descent (SGD) optimiser was used with a learning rate of 0.005 and a decay rate of 0.1 to optimise the network weights and biases to reduce the loss. During training, an anchor was considered positive if it had the highest intersection-over-union (IoU) or an IoU higher than 0.7, while anchors with IoU less than 0.3 were considered negative. To reduce the duplication of detections corresponding to the same ground truth, the Non-Maximum Suppression (NMS) was set to 0.7. During training, augmentation methods were employed to enhance the robustness of the model, including brightness with a random factor between 0.8 and 1.5, contrast with a random factor between 0.6 and 1.3, saturation with a random factor between 0.8 and 1.4, lighting through adding colour jittering with a random degree sampled from normal distribution with a standard deviation of 0.7, and horizontal flipping with a probability of 50%. The default anchor sizes of 32 and 512 were removed as the peduncle nodes in collected images were typically around 200 pixels in height and 140 pixels in width.

*2.2.2.3. Peduncle-node detection.* The trained keypoint detector has the capability to localise multiple peduncle nodes in the image. However, considering the harvesting robot can only harvest one tomato truss at a time in real application, when multiple peduncles were detected, the most prominent node was selected as the target. This node was also used for comparison with the annotated ground truth during evaluation. The final peduncle node was determined based on the confidence value corresponding to the detection (Fig. 5a).

### 2.2.3. Pose estimation in 3D

*2.2.3.1. 2D keypoints extension.* After a peduncle node was detected, its keypoints were projected into 3D by using the spatial information from the aligned point cloud. However, the collected point clouds are usually sparse, the projection of only the four keypoints might fail due to missing corresponding points in the point cloud. To deal with this, the 2D keypoints were extended with circular masks (Fig. 5b) to include more pixels and all pixels inside the masks were projected to the point cloud (Fig. 6a and b).

The use of a fixed circular mask radius for all peduncle nodes is arbitrary, as the diameter of the peduncle usually varies from node to node and is affected by different camera shooting distances. In some cases, a fixed radius can produce a suitable circular mask, but it can be too large for some nodes, extending beyond the range of the peduncle pixels and resulting in the background pixels being segmented, which can significantly affect the 3D pose estimation. According to Equation (1), to avoid segmenting extra pixels from the background, the radius $r_m$ of the circular masks was determined by applying a weighting factor $w$ to the diagonal $d$ of the predicted bounding box of the peduncle node.

$$r_m = w \bullet d\ [pixel] \tag{1}$$

This ensured the radius of the circular mask was always determined relative to the size of the peduncle node. $w$ was set to 0.03 in the experiments through a preliminary test, resulting in an as large as possible radius without extending beyond the boundaries of the node/stem width. A higher ratio often resulted in background points being segmented.

*2.2.3.2. Point cloud projection.* The colour and depth images captured by the camera were aligned. Using the depth data, the 3D coordinates, $X_i$, $Y_i$, $Z_i$, were determined for all pixels $p_i$ that had valid depth data. Thus, the relation between the image coordinates, $p_i = \{x_i, y_i\}$, and the 3D coordinates was known. Fig. 6a displays an example of a point cloud where the pixels within the keypoint masks in the colour image were projected into the point cloud. However, it can be seen in Fig. 6b and c that the raw point clusters were influenced by the outliers in the background. This caused the centroid of some clusters to deviate, see Fig. 6c for an example. Outlier points were removed in the next step.

*2.2.3.3. Outliers removal.* The outliers were determined based on the density of the surrounding points. A point was considered an outlier using Equation (2). For each point $i$, the number of neighbours within a 3D sphere of radius $r_f$ was counted. If there were fewer than the minimum required number of neighbours $n_{min}$, the point was handled as an outlier and removed. Consider $n_p$ as the total number of points and $d_{ij}$ as the distance from neighbouring point $j$ to point $i$. The function $\rho$ equals 1 when $d_{ij} < r_f$, otherwise it equals 0. During the experiments, the radius $r_f = 0.005\ [m]$ and the threshold $n_{min} = 10$ were used. These values were determined by a simple pilot experiment and by manually checking the result of noise removal.

$$Outlier_i = \begin{cases} 1, \sum_{j=1}^{n_p} \rho(d_{ij} < r_f) < n_{min} \\ 0, \text{otherwise} \end{cases} \tag{2}$$

## 2.3. Evaluation

A test set containing 145 RGB-D images was used to evaluate the performance of the algorithm from 3 perspectives: (1) object detection,


(a) (b)

**Fig. 5.** (a) the detection with the highest confidence value was determined as the final target. (b) the predicted keypoints of the selected target were extended to circular masks to include more pixels to be projected in the point cloud.

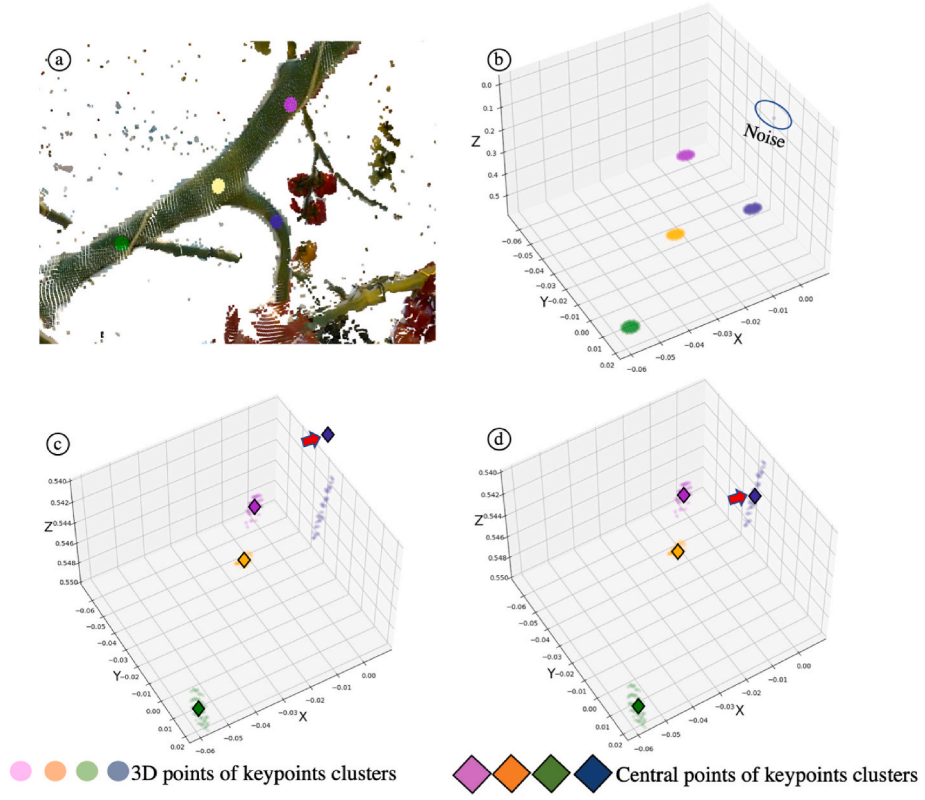


**Fig. 6.** From 2D keypoints to 3D pose. a) show the masked keypoints in the colour image projected into the 3D point cloud, with each keypoint marked in a different colour. b) show the 3D points belonging to the keypoint clusters, containing outliers. c) show a zoom-in of b) with the centroids of each noisy clusters marked by diamonds. d) shows the centroids of the four clusters after removal of outliers. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

(2) keypoint detection, and (3) pose estimation. For each aspect, the overall performance of the model was evaluated on the entire test set (with images from all viewpoints), while its performance at a specific viewpoint was evaluated on images from the corresponding viewpoint. In evaluations (1) and (2), only the RGB images were used, and in evaluation (3), the aligned point clouds resulting from the depth images were used. The configurations of the thresholds for NMS and IoU in the testing phase were identical to those in the training phase. But the maximum-detection-per-image parameter was adjusted to 1 corresponding to the detection with the highest confidence. In other words, the evaluation was done for the most prominent peduncle node in the image.

### 2.3.1. Evaluation of object detection

The objective of this experiment was to evaluate the algorithm's performance in detecting the peduncle nodes. A commonly used metric for object detection, the Average Precision (AP) and F1-socre, was applied to evaluate the performance. To judge if the detections were correct, a criterion of Intersection-over-Union (IoU) was employed to calculate the similarity between the predicted bounding box ($B_p$) and the ground-truth box ($B_{gt}$). It is by dividing the area of the intersection of the two bounding boxes by their union area:

$$IoU(B_P, B_{gt}) = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{3}$$

The IoU ranges from 0 to 1, where a higher value represents a better match. Based on the calculated IoU, it is determined if a detection is a true positive (TP) or a false positive (FP), and if there are false negatives (FN). Specifically, if $IoU(B_P, B_{gt}) > \theta$, where $\theta$ is a predefined threshold, then the prediction is TP. A detection is FP if the $IoU(B_p, B_{gt}) \leq \theta$ or the annotated object is already detected by another prediction with a higher IoU value. If an annotated object is not detected, in other words, the IoU with all the predictions is lower than $\theta$, then it is an FN. Based on the number of TP, FP, and FN, the value of precision and recall under $\theta$ can be calculated. The value of precision represents what fraction of the detections were correct out of all the detections while recall reflects what fraction of the targets were correctly detected.

$$Precision^{\theta} = \frac{TP^{\theta}}{TP^{\theta} + FP^{\theta}} \tag{4}$$

$$Recall^{\theta} = \frac{TP^{\theta}}{TP^{\theta} + FN^{\theta}} \tag{5}$$

Based on the value of precision and recall, the AP and F1-score can be calculated. According to Equation (6), AP is determined as the area under the interpolated precision-recall curve, where $R_1, R_2, \ldots, R_n$ are the recall levels, and interpolated precision $P_{interp}(R)$ is the maximum precision found at any recall level $\geq R$. F1-score can be calculated using Equation (7), which computes the harmonic mean of the precision and recall giving a more intuitive evaluation for the performance. Both the evaluations were conducted using three *IOU* thresholds $\theta$ at 0.5, 0.75, and 0.5:0.95.

$$AP@\theta = \sum_{i=1}^{n} \left(R_i^{\theta} - R_{i-1}^{\theta}\right) P_{interp}^{\theta} \tag{6}$$

$$F1 - score@\theta = 2 \bullet \frac{Precision^{\theta} \bullet Recall^{\theta}}{Precision^{\theta} + Recall^{\theta}} \tag{7}$$

### 2.3.2. Evaluation of keypoint detection

This experiment aimed to evaluate the performance of the keypoint detector in detecting pre-defined keypoints of the peduncle nodes. Four keypoints were expected to be detected for each node. The detection

performance was evaluated using the Percentage of Detected Joints (PDJ) metric, also known as Percentage of Correct Keypoints (PCKh) in some studies (Andriluka et al., 2014; X. Li et al., 2019; Russello et al., 2022). Equation (8) was used to calculate the PDJ. A predicted keypoint was considered correctly detected if the Euclidean distance ($d_i$) between it and the ground truth was smaller than a certain fraction $f$ of the diagonal $d$ of the predicted bounding box. Accounting for the object diagonal allowed the method to handle nodes of arbitrary sizes. The term $n$ referred to the number of keypoints on a node. And $\sigma(x) = 1$ if $x \leq 0$, otherwise $= 0$.

$$PDJ@f = \frac{1}{n}\sum_{i=1}^{n}\sigma(d_i - f \bullet d) \bullet 100 \quad [\%] \tag{8}$$

In this analysis, the PDJ values were calculated for three fractions $f$: 0.05, 0.1, and 0.2. The smaller the value of $f$, the closer the predicted points need to be to the ground truth keypoints to be considered correct. The analysis was conducted from two perspectives: (1) detection-wise and (2) keypoint-wise. The detection-wise evaluation calculated the number of detections that achieved different PDJ values. The PDJ of a detection was 100% all 4 keypoints were correctly detected, and 0% if none of the keypoints were detected correctly (Equation (8)). The keypoint-wise evaluation calculated success rate per keypoint type, with a higher ratio indicating a higher successful detection rate.

#### 2.3.3. Evaluation of pose estimation

Using the 3D keypoints generated in section 2.2.3., the 3D pose of the peduncle node was estimated. The 3D pose was defined to include the node position, orientation, and relative angles. However, measuring the complete ground truth was challenging, so only the relative angles $\alpha_{up}$ and $\alpha_{down}$ were evaluated. The angels were calculated using Equation (9):

$$\alpha = \arccos\frac{\overrightarrow{A} \bullet \overrightarrow{B}}{\left|\overrightarrow{A}\right| \bullet \left|\overrightarrow{B}\right|} \tag{9}$$

for any two given vectors $\overrightarrow{A}$ and $\overrightarrow{B}$ in 3D space, their included angle $\alpha$ equals their dot product divided by their magnitudes. For 4 keypoints 'U', 'N', 'D', and 'P', the $\alpha_{up}$ was calculated using vectors $\overrightarrow{NU}$ and $\overrightarrow{NP}$, while $\alpha_{down}$ was calculated by $\overrightarrow{ND}$ and $\overrightarrow{NP}$.

Two primary sources of error were considered in the estimated angles: (1) pose estimation in 2D, resulting from incorrect localisation of keypoints, (2) pose estimation in 3D, including 3D keypoints projection, noise filtering, and pose estimation. The errors resulting from limitations in the sensor's capability were not considered, such as insufficient depth measurements leading to bad quality of generated point cloud, as these would not reflect the performance of the algorithm. After removing the data where 3D information was not adequately captured, 134 test samples remained for analysis.

To gain a deeper understanding of the source of errors, the accuracy of three types of predicted angles was evaluated: (1) $\widehat{\alpha}^{\text{anno}}$ is the predicted angle based on the annotated keypoints, so with perfect keypoint detection, which reflect errors caused in 3D, (2) $\widehat{\alpha}^{\text{kp}}$ is the predicted angle based on the predicted keypoints, which includes errors in keypoint detection and 3D calculation, and (3) $\widehat{\alpha}^{\text{4kp}}$ represents the predicted angles only when all 4 keypoints were correctly detected, which reflects the optimal performance that the proposed method can achieve. To compare to, $\alpha$ represents the ground-truth angle manually measured in the greenhouse.

Two evaluation metrics, Mean Absolute Error (MAE) and Mean Relative Error (MRE), were used to evaluate the accuracy of the estimated angles. The MAE is calculated as the mean of the absolute errors between the predicted angle, $\widehat{\alpha}_t$, and the ground-truth angle, $\alpha_t$, for peduncle $t$, as shown in Equation (10). The MRE was calculated as the mean of the relative errors obtained by dividing the absolute errors with the corresponding ground-truth angle, as shown in Equation (11). Here, $T$ denotes the total number of detections.

$$\text{MAE} = \frac{1}{T}\sum_{i=0}^{t}\left|\widehat{\alpha}_t - \alpha_t\right| [^\circ] \tag{10}$$

$$MRE = \frac{1}{T}\sum_{i=0}^{t}\frac{\left|\widehat{\alpha}_t - \alpha_t\right|}{\alpha_t} \quad [\%] \tag{11}$$

## 3. Results

The results of the evaluation in terms of object detection, keypoint detection, and 3D angle estimation were displayed in sections 3.1, section 3.2, and section 3.3 respectively.

### 3.1. Node detection

Table 2 presents the overall AP and F1-score across all viewpoints for different IoU thresholds (0.5, 0.75, and 0.5:0.95). The method achieved outstanding results for both metrics on $IoU_{0.5}$, with AP@0.5=0.96 and F1-score@0.5=0.98, indicating the model can successfully detect most of the peduncles under a relatively tolerant intersection-over-union criterion. However, both metrics experienced a significant decrease for a stricter threshold ($IoU_{0.75}$), suggesting that the bounding-box predictions of the complete peduncle were not very accurate with a mismatch to the ground-truth bounding box. It should be noted that the peduncle node, as we defined it, is poorly represented by a bounding box, so it is not unexpected that the accuracy of the bounding-box prediction is not so high. The bounding-box predictions are not use for further operations, but instead use the keypoint detections discussed in the next subsection.

According to Table 3, the algorithm accurately detected nodes from all 5 viewpoints, with an AP@0.5 greater than 0.91 and an F1-score@0.5 greater than 0.95 for all viewpoints, reaching 1.0 for three of the five viewpoints. This suggests the algorithm can deal with the variations caused by different viewpoints. Specifically, Viewpoints 1, 2, and 3 achieved perfect node detection, while Viewpoints 4 and 5 had lower accuracy (refer to Fig. 7 for visual samples cropped from the raw images). This is likely because the lateral views can introduce more shape variations in the peduncle node, making it more difficult for the algorithm to accurately detect. When looking at $IoU_{0.75}$ and $IoU_{0.5:0.95}$ thresholds for both the metrics, larger differences can be seen. Viewpoint 2 exhibited the highest values across all metrics, indicating that the top view is the optimal viewpoint for tasks such as harvesting. Viewpoint 3 exhibited a significantly lower value. One possible reason could be the impact of lighting on the images, as this viewpoint is more susceptible to lighting variations, which affects the algorithm's ability to detect the bounding-box of the peduncle nodes accurately. To provide a clearer illustration, Fig. 8 displays a sample with detection results from Viewpoint 3.

### 3.2. Keypoint detection

The evaluation of detected keypoints was performed on three fractions ($f$) of 0.05, 0.1, and 0.2. Table 4 displays the count of detections achieving different PDJ values, a higher PDJ value indicates more keypoints were correctly detected. The PDJ@0.2 and PDJ@0.1 display a similar distribution, with the most of detections achieving a PDJ of

**Table 2**
Overall results of AP and F1-score across all viewpoints.

| Metrics | $IoU_{0.5}$ | $IoU_{0.75}$ | $IoU_{0.5:0.95}$ |
|---|---|---|---|
| AP | 0.96 | 0.37 | 0.45 |
| F1-score | 0.98 | 0.53 | 0.54 |

**Table 3**
The AP and F1-score obtained from 5 different viewpoints.

| Viewpoint | AP | | | F1-score | | |
|---|---|---|---|---|---|---|
| | $IoU_{0.5}$ | $IoU_{0.75}$ | $IoU_{0.5:0.95}$ | $IoU_{0.5}$ | $IoU_{0.75}$ | $IoU_{0.5:0.95}$ |
| View1 | 1.0 | 0.33 | 0.48 | 1.00 | 0.47 | 0.54 |
| View2 | 1.0 | 0.86 | 0.67 | 1.00 | 0.89 | 0.71 |
| View3 | 1.0 | 0.08 | 0.30 | 1.00 | 0.16 | 0.37 |
| View4 | 0.94 | 0.47 | 0.46 | 0.95 | 0.58 | 0.52 |
| View5 | 0.91 | 0.17 | 0.35 | 0.95 | 0.21 | 0.38 |

100% (124 detections for PDJ@0.2 and 68 detections for PDJ@0.1), while the smallest number of detections achieved a PDJ smaller than 50% (3 detections for PDJ@0.2 and 9 detections for PDJ@0.1). In contrast, for PDJ@0.05, only 19 detections achieved PDJ = 100%, while a large number of detections (45) achieved a PDJ smaller than 50%. The significant shift in the ratio when reducing $f$ from 0.2 to 0.05 is that the smaller fraction $f$ requires the detected keypoints be closer to the ground truth to be considered correct, which resulted in most of the detected keypoints considered correct under $f = 0.2$ being considered as failed under a more stringent criterion ($f = 0.05$).

The results of PDJ@0.2 and PDJ@0.1 indicate the feasibility of the model, which essentially located the keypoints on the right locations. But some detections are not accurate enough in specific keypoints or scenarios to fulfil a more stringent criterion. To investigate the sources of error, further analysis was carried out by calculating the PDJ for each individual keypoint.

As shown in Table 5, the 4 keypoints 'U', 'N', 'D' and 'P' achieved PDJ@0.2 at 91.03%, 97.93%, 91.72%, and 96.55% respectively, suggesting the model was capable of accurately localising the keypoints and achieving a success rate above 90% for all the keypoints. The corresponding PDJ values decreased at varying levels by 60%, 8.96%, 57.93%, and 35.17% when reducing $f$ from 0.2 to 0.05. The slight reduction of keypoint 'N' indicates it could be consistently localised close to the ground truth, and hence, its results were not largely affected even under a stricter criterion. In contrast, the significant decrease of keypoints 'U' and 'D' at PDJ@0.05 suggests these two keypoints usually deviated largely from the ground truth, and thus were no longer considered correct when $f$ was decreased to 0.05. The keypoint 'N' achieved the highest PDJ in all three fractions, implying the most accurate detection, while the keypoint 'U' and 'D' showed significantly lower success rates, and their gap with the keypoint 'N' became even larger when decreasing $f$.

As described in section 2.1.2, keypoints 'N' and 'P' were determined

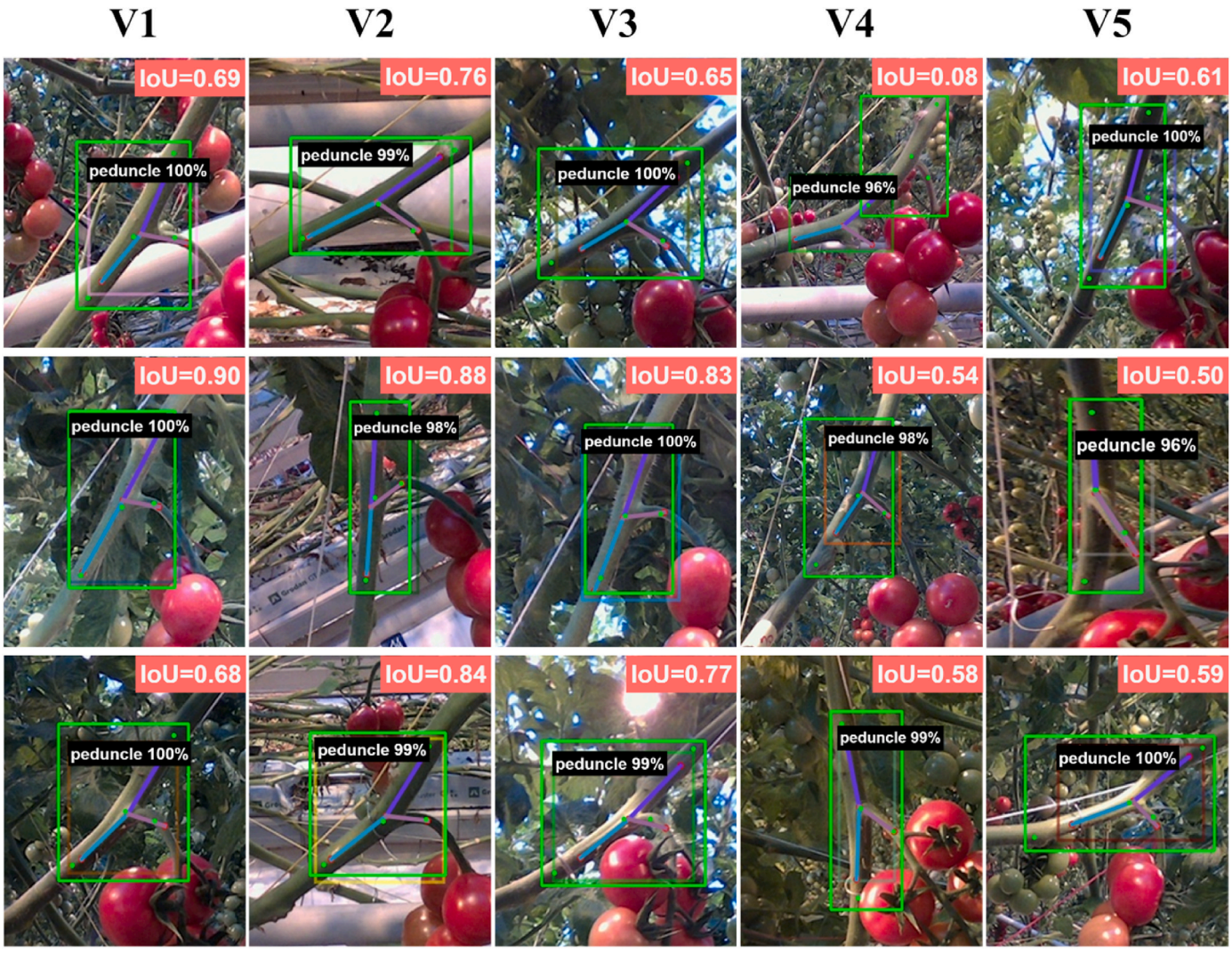


**Fig. 7.** Illustration of node detection results for 3 peduncle nodes (rows) from 5 viewpoints (columns). All images are cropped from the raw images to give a better illustration. The keypoints and bounding box in green colour indicate the ground truth. The IoU value between the predicted and ground truth bounding boxes of the detection is calculated and plotted in the pink box. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)


**Fig. 8.** Illustration of the raw image used for 2D pose estimation, captured from viewpoint V3. The model faces challenges including accurately identifying peduncle nodes among similar plant parts (petiole nodes), distinguishing between foreground and background, and dealing with changing light. The keypoints and bounding box in green colour indicate the ground truth. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

**Table 4**
The number of images that achieved different PDJ values are counted at three fractions, 0.05, 0.1, and 0.02 independently.

| Metrics | <50% | 50% (2/4) | 75% (3/4) | 100% (4/4) |
|---|---|---|---|---|
| PDJ@0.05 | 31.03%(45) | 37.35%(44) | 25.52%(37) | 13.10%(19) |
| PDJ@0.1 | 6.21%(9) | 23.45%(34) | 23.45%(34) | 46.90%(68) |
| PDJ@0.2 | 2.07%(3) | 2.07%(3) | 10.34%(15) | 85.52%(124) |

**Table 5**
The PDJ of different keypoints for the entire testset with 145 images under 3 fractions of 0.05, 0.1, and 0.2.

| Metrics | U | N | D | P |
|---|---|---|---|---|
| PDJ@0.05 | 31.03% | 88.97% | 33.79% | 61.38% |
| PDJ@0.1 | 57.93% | 97.93% | 60.69% | 92.41% |
| PDJ@0.2 | 91.03% | 97.93% | 91.72% | 96.55% |

based on unique visual features ('N' linking the peduncle and the main stem; 'P' normally bending), while keypoints 'U' and 'D' were annotated based on the Euclidean distance. Based on the results, it is suggested that the visual features are more effective for the model to learn and recognise compared to the distance features.

Fig. 9 displays some samples in which the localisations of the keypoints 'U' or 'D' were considered incorrect due to large deviations from the ground truth. However, all these localisations were found to be correctly assigned to the corresponding plant parts. Based on this, it is suggested that the results can be further improved if the system is tasked with accurately locating the different landmarks without necessarily requiring proximity to the ground truth.

Table 6 presents keypoints detection results from different viewpoints, with an average PDJ@0.2 of over 89.47r all views and two views reaching above 97.37%. View1 and View2 outperformed the other views, making them ideal for keypoints detection in practical applications. Keypoints 'N' and 'P' consistently showed significantly higher PDJ values compared to 'U' and 'D' across all views. Notably, View5 displays lower PDJ for 'P'. This is because in the greenhouse used in this study, the peduncles tend to grow towards the same direction, which is roughly View5, leading to significant shape variations and challenging keypoints detection. The reduced detection accuracy of 'P' can largely impact angle estimation, as its precise 3D position is crucial for estimating both angles, $\alpha_{up}$ and $\alpha_{down}$.

### 3.3. 3D pose estimation

Based on the results presented in Fig. 10, $\hat{a}^{anno}$ achieved a MAE of 8° and 7° for $\alpha_{up}$ and $\alpha_{down}$, respectively, with a MRE of 12% and 6%. These values show that even when all keypoints are accurately localised, errors can still occur due to inaccuracies in the 3D pose estimation, relating to, for instance, noise in the depth data. These errors serve as a minimum level of error that can be achieved by the full algorithm. When including the 2D pose estimation and evaluating those cases where all 4 keypoints were correctly estimated, it can be seen that $\hat{a}^{4kp}$ shows a slightly higher error with MAE of 8° and 7° for $\alpha_{up}$ and $\alpha_{down}$, respectively, and an MRE of 13% and 6%. This increase in error results from small inaccuracies in predicting the keypoint locations. Including all cases of 2D pose estimation, the error for $\hat{a}^{kp}$ increases further to an MAE of 11° and 10° for $\alpha_{up}$ and $\alpha_{down}$, respectively, and an MRE of 17% and 8%. The larger errors in $\hat{a}^{kp}$ result from some additional inaccuracy in the 2D keypoint estimation. The figure suggests that the largest error in 3D pose estimation stems from the final step in the method.

Compared to $\alpha_{down}$, $\alpha_{up}$ showed a slightly larger MAE but a signifi-

**Table 6**
The PDJ obtained by the algorithm at different viewpoints under fraction equals 0.2.

| Viewpoints | U | N | D | P | Mean |
|---|---|---|---|---|---|
| View1 | 100% | 100% | 94.74% | 100% | 98.68% |
| View2 | 94.74% | 100% | 94.74% | 100% | 97.37% |
| View3 | 73.68% | 100% | 94.74% | 100% | 92.11% |
| View4 | 89.47% | 94.74% | 78.95% | 94.74% | 89.47% |
| View5 | 94.74% | 94.74% | 89.47% | 89.47% | 92.11% |
| Mean | 90.53% | 97.89% | 90.53% | 96.84% | |

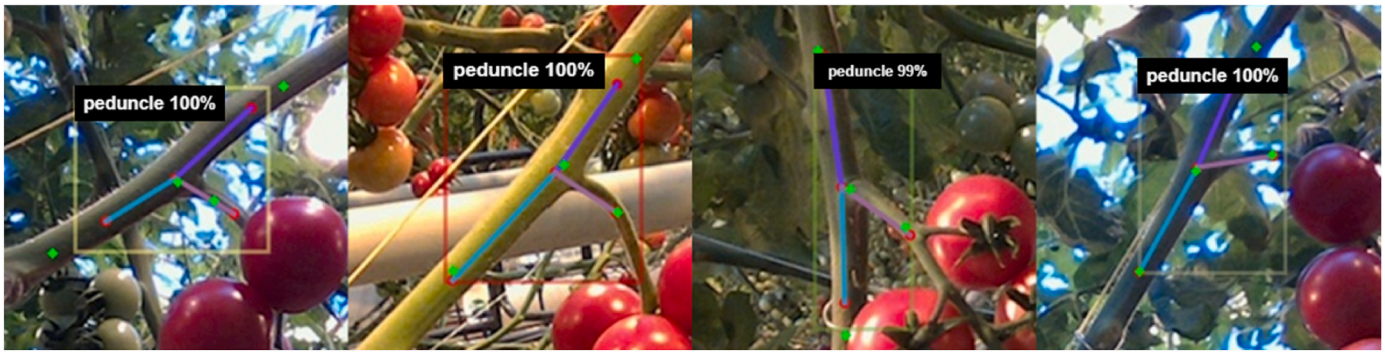


**Fig. 9.** Illustration of some detections failed to localise the keypoints 'U' and 'D' due to their larger deviations from the ground truth. The keypoints and bounding box in green colour indicate the ground truth. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

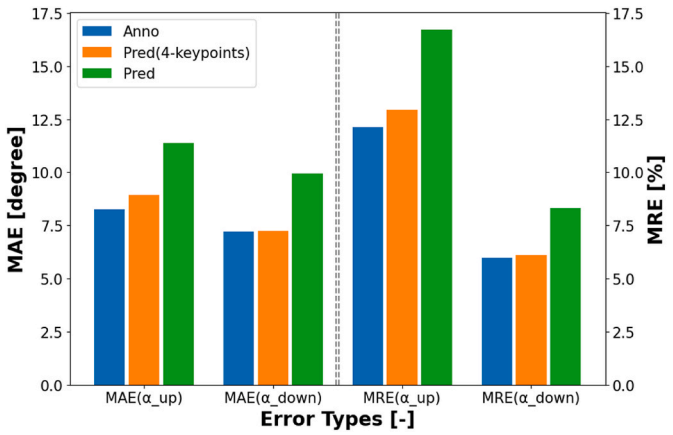


**Fig. 10.** Three kinds of angles were evaluated on MAE and MRE. The 'Anno' represents the errors contained in $\hat{a}^{anno}$, the 'Pred' represents the errors contained in $\hat{a}^{kp}$, while 'Pred(4-keypoints)' represents the errors contained in $\hat{a}^{4kp}$ only when 4 keypoints are correctly detected under $f = 0.2$.

cantly larger MRE. This is because the actual angle of $\alpha_{up}$ is generally much smaller, and as a result, the MRE can be amplified when the MAE is similar to that $\alpha_{down}$.

It is important to note that not all 134 detections were considered in the evaluation of $\hat{a}^{kp}$, as there were cases where $\alpha_{up}$ or $\alpha_{down}$ were incomputable. For example, keypoints localised on the background due to incorrect keypoints detection usually lack 3D information, which can make the angles incomputable. In this evaluation, only one case was excluded for this reason, so a total of 133/134 cases were included. For $\hat{a}^{4kp}$, 115 detections were used.

Table 7 displays the results of angle estimation acquired from different viewpoints. View1 showed the lowest errors, followed by View2 and View3, which had a vertical deviation from View1, lower than View4. However, View5 exhibited significantly higher errors, mainly due to the low detection accuracy of 'P' at this view (refer to section 3.2), suggesting that this view should be avoided in real applications.

## 4. Discussion

### 4.1. General evaluation of proposed method

The proposed method demonstrated a high success rate in detecting the peduncle nodes. Although the nodes were detected well, the exact bounding-box coordinates often did not accurately match the ground truth. This is mainly because keypoints 'U' and 'D' were annotated based on distance and not based on a clear image feature, which the model struggled to effectively learn. This was further evident in the evaluation of the keypoint detection. While the method displayed high accuracy across all types of keypoints, 'U' and 'D' showed significantly lower PDJ compared to 'N' and 'P'. Since the proposed method directly performs 3D pose estimation based on the detected keypoints, the accuracy of the 3D pose estimation relied on the accuracy of the 2D pose detection.

**Table 7**
Errors achieved by $\hat{a}^{kp}$ in angle estimation at different viewpoints. The values were kept to integers considering the precision of manual measurements in real environments.

| Viewpoint | MAE (up) | MAE (down) | MRE (up) | MRE (down) |
|---|---|---|---|---|
| View1 | 9° | 6° | 7% | 5% |
| View2 | 11° | 9° | 10% | 8% |
| View3 | 9° | 7° | 8% | 6% |
| View4 | 12° | 11° | 9% | 9% |
| View5 | 21° | 23° | 18% | 19% |

The proposed method demonstrated capability in handling variations caused by different viewpoints with respect to the peduncle node. The node detection results showed an AP ranging between 0.94 and 1.00, while the F1 score ranged between 0.95 and 1.00, indicating high detection performance for all viewpoints, with View 1–3 showing the best performance. Similarly, the results of the keypoint detection showed a high consistency over viewpoints, with PDJ between 89.47% and 98.68%, with View 1 and 2 resulting in the best performance and View 4 showing the lowest performance. The accuracy of the pose estimation showed more variation between viewpoints, with errors ranging between 6° and 12° for Views 1–4 and errors between 21° and 23° for View 5. Overall, View 1 and 2 showed to provide the best view on the node. While the node and keypoint detection for View 5 were good in general, it did show a lower PDJ for the node point 'P'. As the 3D location of this point is crucial to estimate the two pose angles, this explains the lower pose-estimation accuracy for this view. View 5 is the rightwards view, which relates to the view where the tomato truss is closer to the camera than the node point 'P', possibly partially blocking a good view on the node.

### 4.2. Future improvements

To evaluate the algorithm's performance in a realistic scenario, all test data were collected from a natural greenhouse. However, the complex environment posed challenges for object annotation and ground-truth measurement, as it was impossible to cover all the nodes in the background. To find a solution, only the main node in each image was considered, matching the prediction that gained the maximum confidence value. Focusing only on the most confident prediction ignored other predictions based by the method. However, for the use of robotic harvesting, that is justified, as the robot will harvest the trusses one by one anyway. Occasionally, a node in the background was detected, which was evaluated as a FP. A similar problem was seen in the experiment of Rong et al. (2022), where the models were trained to detect only the fruits and peduncles of tomato plants in the foreground, but targets in the background were still likely to be detected because both had quite similar traits. In future work, depth information can be integrated as input for the pose-estimation method to ignore the nodes in the background. This would force the model to only focus on objects in a certain depth range.

As in normal in commercial production, the leaves surrounding the peduncle node were removed before the fruit ripened. Consequently, situations where the peduncle node was occluded by the leaves were not considered. The deep-keypoint-detection method can inherently handle a certain level of occlusions and estimate the object's pose based on a subset of the visible keypoints. However, if the nodes are significantly occluded, the current method may encounter some limitations. In such cases, utilising the information from multi-views is a popular solution, it makes information that is not perceivable from a single view become available, and reduces perception uncertainty (Hemming et al., 2014; Rapado-Rincón et al., 2023; Shi et al., 2019; van Henten et al., 2003). To increase efficiency, next-best-viewpoint (NBV) methods can be used that propose camera poses to maximise the information gain (Burusa et al., 2022; Mendoza et al., 2020; Zeng et al., 2020). As the results showed, some viewpoints are more favourable for a good peduncle-node detection than others. In future work, a method that integrates NBV planning with the keypoint-based pose estimator can be proposed to change camera poses that maximise the view on the peduncle.

Due to the lack of visual features, the keypoints 'U' and 'D' were determined based on pixel distance, which was expected to be learned by the model. However, as shown in Table 5, these two keypoints presented significantly lower PDJ values compared to keypoints 'N' and 'P'. Fig. 9 also suggests that keypoints 'U' and 'D' were likely considered incorrect due to their large deviations from the ground truth, even though they were localised on the correct plant part and could be used to

calculate the angles. So, it is considered that the traditional PDJ metric may not the most suitable way to evaluate these two keypoints. On the other hand, keypoints are possible not the best possible representation for these points. Instead, the DNN could be adapted to predict a vector up the main stem and a vector down the main stem instead of two keypoints.

In section 2.2.3, circular masks with a radius of $r_m$ (Equation (1)) were used to expand the keypoints to include more pixels. The value of $r_m$ was self-adjusted based on the diagonal ($d$) of the predicted bounding box and a constant weighting factor ($w$) to avoid segmenting the pixels in the background. This approach was found to significantly reduce errors in 3D pose estimation compared to using a single pixel. However, beyond a certain range (0–0.05), increasing the value of $w$ caused a negative effect, as pixels in the background were added. The sensitivity of the model to $w$ will reduce its robustness. This problem is considered can be solved by employing a better strategy for expanding keypoints including 3D distance information, or by growing only inside the mask of the node.

In Fig. 10, the errors of the estimated angles did not reduce to very low even when all keypoints were accurately located ($\widehat{a}^{anno}$ and $\widehat{a}^{4kp}$), suggesting potential errors in the ground truth measurements. The measurement of $a$ was subjective as three measurers manually located keypoints and measured angles with a protractor. The large standard deviations in $\alpha_{up}$ and $\alpha_{down}$ (section 2.1.3) also highlight significant deviations in measurements. To improve accuracy and reduce time consumption, using $\widehat{a}^{anno}$ directly as the ground truth is recommended in future experiments.

The analyses of the influence of the viewpoint with respect to the node suggests that some viewpoints are better than others. Particular V1 (canonical view) and V2 (the higher view) resulted in higher detection and pose-estimation performance. In future work, active-vision methods should be explored to actively guide the robot to the right viewpoint to make the most accurate detection.

### 4.3. Comparison with related work

Boogaard et al. (2020) utilised YOLO v3 (Redmon & Farhadi, 2018) to detect cucumber internodes using multi-view imaging. The comparable results were achieved if it comes down to the node detection, although a direct comparison is not possible due to differences in experimental conditions, tasks, and datasets. It is worth noting that their experiments were conducted in a controlled environment, while ours occurred in a commercial greenhouse, posing more challenges for node detection. They investigated the benefits of multiple viewpoints in object detection, which is what can be explored for keypoints detection method in the future work. While their work only detected the nodes, the method proposed in this study also estimated the pose of the peduncle node.

The keypoint detection results obtained in this study are consistent with Zhang et al. (2022), who used a DL-based keypoint detector for tomato bunches. Both studies found that keypoints 'U' and 'D' had significantly lower PDJ values compared to 'N' and 'P'. While our study used a stricter evaluation criterion based on the weighted size of the node, they used the weighted size of the entire tomato bunch, allowing for greater deviation in their predicted keypoints to still be considered correct. Nevertheless, our method demonstrated higher PDJ values for 'N' and 'P', confirming its effectiveness in keypoints detection. In comparison to the method of Kim et al. (2023), the proposed method in this study achieved better results with an average PDJ@0.2 = 94.31% across all types of keypoints, outperforming their results of PDJ@0.5 = 92.9%, even under a more stringent criterion. However, it must be noted that a one-on-one comparison of results is not possible due to differences in targets, criteria, and datasets. In addition, compared to their work, which used zoomed-in images with only one tomato bunch per image, the images used in this research were originally captured by the camera, containing many tomato plants in the background and changing light, making the task more challenging (refer to Fig. 8). On the other hand, their method included also the location of the individual tomatoes in the truss, which makes their task more challenging. Similar to Kim et al.'s (2023) method, the proposed method also has the capability to detect the pose of multiple peduncle nodes simultaneously. However, only the most prominent nodes were used in the evaluation, as a robot would harvest trusses one by one.

The proposed method for 3D pose estimation was compared with Luo et al. (2022), who used Mask R–CNN for target detection in RGB images and estimating peduncle orientations by mapping the detections to the aligned point clouds. Direct comparison of the results is not possible due to differences in evaluation metrics and targets. Their method focused solely on peduncle orientation, ignoring relative angles. Furthermore, The proposed method estimated the pose by processing only the keypoint clusters, making it more efficient and simpler than their method, which processed the entire point cloud. However, their method, by using more complete point cloud can be less sensitive to 2D object detection accuracy and point cloud quality. Zhang et al. (2023) evaluated the accuracy of their predicted 3D keypoints compared to human annotations in the point cloud, but they did not evaluate their method with real-world measurements as this work did. By doing so, valuable insights into the error sources in different steps in the pipeline were allowed to be acquired in this research. The contribution to the 3D pose error of the 2D keypoint detection and the integration of the 3D data could be shown, as displayed in Fig. 10. Although their method to predict 3D keypoints directly from the image data using a deep neural network is promising, it requires the availability of sufficient amount training data. In future work, this end-to-end approach will be investigated and compare the performance of both approaches as a function of the amount of available training data.

To the best of knowledge, there are no other studies investigating the impact of different viewing angles on the accuracy of pose detection. According to the results, for this crop, views 1 and 2 provide a higher performance, which is valuable input for future work on robotic harvesting of tomatoes.

The proposed method demonstrated good performance in 3D pose estimation of peduncle nodes, making it suitable for acquiring comprehensive information for robotic harvesting. However, defining precise accuracy requirements for robotic harvesting poses challenges due to limited research and variations in crops, harvesting mechanisms, and other factors. For example, the design of grippers and cutters in a robotic harvester can tolerate some level of error, reducing the accuracy requirements for the vision system significantly (Zhang et al., 2022).

The impact of post-harvest peduncle node structure on keypoint detection was not considered in this project. This is because, to minimise plant infection, tomato trusses are typically detached as close as possible to the main stem, between the 'N' and 'P' keypoints. Consequently, the residual peduncle length is shorter than the annotated distance from 'N' to 'P', and thus does not affect keypoint detection. If this becomes a concern in other application scenarios, the issue can be addressed by including post-harvest nodes in the training set.

A test set comprising 145 images with aligned point clouds was used to evaluate the proposed method. The size of the test set is similar to Kim et al. (2023), who used 136 images for evaluation, but smaller than that of Zhang et al. (2023), with 400 images. The test set used in this study, however, consist of representative set of images from an actual commercial tomato greenhouse, including cluttered scenarios, variation in the appearance and pose of the peduncle nodes, different viewing angles and natural variations in illumination. In addition, and different from related work, the manual reference measurements performed in the greenhouse were included to evaluate the accuracy of the method in the real world.

## 5. Conclusions

In this study, an innovative method was proposed to estimate the 3D pose of tomato peduncle nodes from RGB-D images. The method provided complete 3D pose information, including the position, orientation, and relative angles to the main stem, which are needed for robotic operations. Meanwhile, this method is expected to benefit other agricultural operations using robotics, such as phenotyping and monitoring, with the output of comprehensive information. After being analysed from the three aspects of (1) object detection, (2) keypoint detection, and (3) pose estimation, the feasibility of the proposed algorithm was shown. For each evaluation, the impact of viewpoint angles was investigated, indicating the model can handle a level of variations caused by view change. The evaluation resulted in the following conclusions.

(1) For object detection, the proposed algorithm achieved outstanding results in both AP@0.5 and F1-score@0.5, suggesting its good capability in node detection. However, there was a significant drop in both metrics for the IoU threshold of 0.75, indicating that most of the bounding-box predictions do not accurately match the ground truth.
(2) Keypoint detection resulted in above 90% of PDJ@0.2 for all keypoints, demonstrating the model's capability to accurately localise the keypoints. However, there is a significant drop in PDJ@0.05 for 'U' and 'D' keypoints, indicating these two keypoints are localised with less accuracy. Nevertheless, as discussed earlier, some predicted keypoints considered incorrect can still be localised on the correct plant parts.
(3) For pose estimation, the errors of $\widehat{a}^{4kp}$ displayed a significant decrease compared to $\widehat{a}^{kp}$, indicating a large proportion of the errors in 3D pose estimation were due to incorrect keypoint detection in 2D. However, even if all the keypoints were accurately detected, there was still a level of error, suggesting the measurement of $a$ might not be very accurate. Therefore, in future experiments, the $\widehat{a}^{anno}$ could be directly used as the ground-truth, which can save significant time in actual angle measurement.
(4) The methods achieved high AP@0.5 (>0.91) in object detection and PDJ@0.2 (>0.89) in keypoint detection for all viewpoints, indicating robustness to variations caused by view changes. View1 (canonical view) and view2 (higher view) yielded the best results and are recommended for practical applications.

While the proposed method was tested for the 3D pose estimation of tomato peduncles, the method has potential to be applied to other similar crops, such as cucumber, zucchini, and bell pepper and for other plant parts, such as the pedicel (e.g., for deleafing) and fruit clusters (e.g., for monitoring tasks). Besides robotic applications, the method can also provide valuable information for plant monitoring and plant phenotyping.

## Funding

This research is funded by the Netherlands Organisation for Scientific Research (NWO) project Cognitive Robots for Flexible Agro-Food Technology (FlexCRAFT), grant P17-01.

## CRediT authorship contribution statement

**Jianchao Ci:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft. **Xin Wang:** Conceptualization, Data curation, Supervision, Writing – review & editing. **David Rapado-Rincón:** Conceptualization, Software, Writing – review & editing. **Akshay K. Burusa:** Conceptualization, Writing – review & editing. **Gert Kootstra:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

We thank the members of the FlexCRAFT project for engaging in fruitful discussions and providing valuable feedback to this work. Special thanks to Vereijken Kwekerijen for welcoming us to their greenhouse.

## References

Afonso, M. V., Barth, R., & Chauhan, A. (2019). Deep learning based plant part detection in Greenhouse settings. In *12th EFITA International Conference: Digitizing agriculture* (pp. 48–53). https://research.wur.nl/en/publications/deep-learning-based-plant-part-detection-in-greenhouse-settings.

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer vision and pattern Recognition* (pp. 3686–3693).

Bac, C. W., van Henten, E. J., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics, 31*(6), 888–911.

Bargoti, S., & Underwood, J. (2017). Deep fruit detection in orchards. *Proceedings - IEEE International Conference on Robotics and Automation*, 3626–3633. https://doi.org/10.1109/ICRA.2017.7989417

Benavides, M., Cantón-Garbín, M., Sánchez-Molina, J. A., & Rodríguez, F. (2020). Automatic tomato and peduncle location system based on computer vision for use in robotized harvesting. *Applied Sciences, 10*(17). https://doi.org/10.3390/app10175887

Boogaard, F. P., Rongen, K. S. A. H., & Kootstra, G. W. (2020). Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging. *Biosystems Engineering, 192*, 117–132. https://doi.org/10.1016/j.biosystemseng.2020.01.023

Brooks, J. (2019). *COCO annotator*. https://github.com/jsbroks/coco-annotator/.

Burusa, A. K., van Henten, E. J., & Kootstra, G. (2022). Attention-driven active vision for efficient reconstruction of plants and targeted plant parts. http://arxiv.org/abs/2206.10274.

Burusa, A. K., van Henten, E. J., & Kootstra, G. (2023). *Gradient-based local next-best-view planning for improved perception of targeted plant nodes*. ArXiv Preprint ArXiv: 2311.16759.

Eizentals, P., & Oka, K. (2016). 3D pose estimation of green pepper fruit for automated harvesting. *Computers and Electronics in Agriculture, 128*, 127–140.

Halstead, M., McCool, C., Denman, S., Perez, T., & Fookes, C. (2018). Fruit quantity and ripeness estimation using a robotic vision system. *IEEE Robotics and Automation Letters, 3*(4), 2995–3002. https://doi.org/10.1109/LRA.2018.2849514

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hemming, J., Ruizendaal, J., Hofstee, J. W., & van Henten, E. J. (2014). Fruit detectability analysis for different camera positions in sweet-pepper. *Sensors, 14*(4), 6032–6044.

Ji, W., Zhao, D., Cheng, F., Xu, B., Zhang, Y., & Wang, J. (2012). Automatic recognition vision system guided for apple harvesting robot. *Computers & Electrical Engineering, 38*(5), 1186–1195.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture, 147*, 70–90. https://doi.org/10.1016/J.COMPAG.2018.02.016

Kang, H., & Chen, C. (2020). Fruit detection, segmentation and 3D visualisation of environments in apple orchards. *Computers and Electronics in Agriculture, 171*, Article 105302. https://doi.org/10.1016/J.COMPAG.2020.105302

Kang, H., Zhou, H., Wang, X., & Chen, C. (2020). Real-time fruit recognition and grasping estimation for robotic apple harvesting. *Sensors, 20*(19), 1–15. https://doi.org/10.3390/s20195670

Kim, T., Lee, D.-H., Kim, K.-C., & Kim, Y.-J. (2023). 2D pose estimation of multiple tomato fruit-bearing systems for robotic harvesting. *Computers and Electronics in Agriculture, 211*, Article 108004. https://doi.org/10.1016/j.compag.2023.108004

Kootstra, G., Wang, X., Blok, P. M., Hemming, J., & Van Henten, E. (2021). Selective harvesting robotics: Current research, trends, and future directions. *Current Robotics Reports, 2*, 95–104.

Lehnert, C., Sa, I., McCool, C., Upcroft, B., & Perez, T. (2016). Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE International Conference on robotics and automation (ICRA)* (pp. 2428–2434).

Lehnert, C., Tsai, D., Eriksson, A., & McCool, C. (2019). 3d move to see: Multi-perspective visual servoing towards the next best view within unstructured and occluded environments. In *2019 IEEE/RSJ International Conference on Intelligent robots and systems (IROS)* (pp. 3890–3897).

Li, X., Cai, C., Zhang, R., Ju, L., & He, J. (2019). Deep cascaded convolutional models for cattle pose estimation. *Computers and Electronics in Agriculture, 164*. https://doi.org/10.1016/j.compag.2019.104885

Li, H., Zhu, Q., Huang, M., Guo, Y., & Qin, J. (2018). Pose estimation of sweet pepper through symmetry axis detection. *Sensors, 18*(9), 3083.

Liang, C., Xiong, J., Zheng, Z., Zhong, Z., Li, Z., Chen, S., & Yang, Z. (2020). A visual detection method for nighttime litchi fruits and fruiting stems. *Computers and Electronics in Agriculture, 169*, Article 105192.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on computer vision and pattern Recognition* (pp. 2117–2125).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on computer vision* (pp. 740–755).

Lin, G., Tang, Y., Zou, X., Xiong, J., & Li, J. (2019). Guava detection and pose estimation using a low-cost RGB-D sensor in the field. *Sensors, 19*(2). https://doi.org/10.3390/s19020428

Luo, L., Tang, Y., Zou, X., Ye, M., Feng, W., & Li, G. (2016). Vision-based extraction of spatial information in grape clusters for harvesting robots. *Biosystems Engineering, 151*, 90–104. https://doi.org/10.1016/j.biosystemseng.2016.08.026

Luo, L., Yin, W., Ning, Z., Wang, J., Wei, H., Chen, W., & Lu, Q. (2022). In-field pose estimation of grape clusters with combined point cloud segmentation and geometric analysis. *Computers and Electronics in Agriculture, 200*. https://doi.org/10.1016/j.compag.2022.107197

Mathis, A., Biasi, T., Mert, Y., Rogers, B., Bethge, M., & Mathis, M. W. (2020). Imagenet performance correlates with pose estimation robustness and generalization on out-of-domain data. In *International Conference on Machine learning 2020 Workshop on uncertainty and robustness in deep learning*.

Mendoza, M., Vasquez-Gomez, J. I., Taud, H., Sucar, L. E., & Reta, C. (2020). Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters, 133*, 224–231. https://doi.org/10.1016/j.patrec.2020.02.024

Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature Methods, 16*(1), 117–125. https://doi.org/10.1038/s41592-018-0234-5

Rapado-Rincón, D., van Henten, E. J., & Kootstra, G. (2023). Development and evaluation of automated localisation and reconstruction of all fruits on tomato plants in a greenhouse based on multi-view perception and 3D multi-object tracking. *Biosystems Engineering, 231*, 78–91. https://doi.org/10.1016/j.biosystemseng.2023.06.003

Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement*. ArXiv Preprint ArXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems, 28*.

Rong, J., Dai, G., & Wang, P. (2022). A peduncle detection method of tomato for autonomous harvesting. *Complex and Intelligent Systems, 8*(4), 2955–2969. https://doi.org/10.1007/s40747-021-00522-7

Russello, H., van der Tol, R., & Kootstra, G. (2022). T-LEAP: Occlusion-robust pose estimation of walking cows using temporal information. *Computers and Electronics in Agriculture, 192*. https://doi.org/10.1016/j.compag.2021.106559

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors, 16*(8). https://doi.org/10.3390/s16081222

Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., & Perez, T. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. *IEEE Robotics and Automation Letters, 2*(2), 765–772.

Santos, T. T., de Souza, L. L., dos Santos, A. A., & Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Computers and Electronics in Agriculture, 170*, Article 105247. https://doi.org/10.1016/J.COMPAG.2020.105247

Shi, W., van de Zedde, R., Jiang, H., & Kootstra, G. (2019). Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering, 187*, 81–95. https://doi.org/10.1016/j.biosystemseng.2019.08.014

van Henten, E. J., van Tuijl, B. A. J., Hemming, J., Kornet, J. G., Bontsema, J., & van Os, E. A. (2003). Field test of an autonomous cucumber picking robot. *Biosystems Engineering, 86*(3), 305–313.

Virlet, N., Sabermanesh, K., Sadeghi-Tehran, P., & Hawkesford, M. J. (2016). Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Functional Plant Biology, 44*(1), 143–153.

Vit, A., Shani, G., & Bar-Hillel, A. (2020). Length phenotyping with interest point detection. *Computers and Electronics in Agriculture, 176*, Article 105629. https://doi.org/10.1016/j.compag.2020.105629

Wagner, N., Kirk, R., Hanheide, M., & Cielniak, G. (2021). Efficient and robust orientation estimation of strawberries for fruit picking applications. In *2021 IEEE International Conference on robotics and automation (ICRA)* (pp. 13857–13863).

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Yoshida, T., Fukao, T., & Hasegawa, T. (2018). Fast detection of tomato peduncle using point cloud with a harvesting robot. *Journal of Robotics and Mechatronics, 30*(2), 180–186. https://doi.org/10.20965/jrm.2018.p0180

Yoshida, T., Fukao, T., & Hasegawa, T. (2020). Cutting point detection using a robot with point clouds for tomato harvesting. *Journal of Robotics and Mechatronics, 32*(2), 437–444. https://doi.org/10.20965/jrm.2020.p0437

Zeng, R., Zhao, W., & Liu, Y. J. (2020). PC-NBV: A point cloud based deep network for efficient next best view planning. *IEEE International Conference on Intelligent Robots and Systems*, 7050–7057. https://doi.org/10.1109/IROS45743.2020.9340916

Zhang, F., Gao, J., Song, C., Zhou, H., Zou, K., Xie, J., Yuan, T., & Zhang, J. (2023). TPMv2: An end-to-end tomato pose method based on 3D key points detection. *Computers and Electronics in Agriculture, 210*. https://doi.org/10.1016/j.compag.2023.107878

Zhang, F., Gao, J., Zhou, H., Zhang, J., Zou, K., & Yuan, T. (2022). Three-dimensional pose detection method based on keypoints detection network for tomato bunch. *Computers and Electronics in Agriculture, 195*. https://doi.org/10.1016/j.compag.2022.106824