Segmentation and Tracking of Vegetable Plants by Exploiting Vegetable Shape Feature for Precision Spray of Agricultural Robots

Nan Hu[1], Daobilige Su[*1], Shuo Wang[1], Xuechang Wang[1], Huiyu Zhong[1], Zimeng Wang[1], Yongliang Qiao[2], and Yu Tan[1]

[1]College of Engineering, China Agricultural University, Beijing 100083, China
[2]Australian Institute for Machine Learning (AIML), The University of Adelaide, 5005, Australia

Abstract

With the rapid growth of the world population, shortage in labor force and change in the global climate, increasing food demand and food security have become the top priorities that agriculture needs to solve urgently. The rapid development of artificial intelligence and robotics technologies make it possible as a key part of agricultural production. With the increasing deployment of agricultural robots, the traditional manual spray of liquid fertilizer and pesticide is gradually being replaced by agricultural robots. Compared to conventional spray methods which adopt large angle spray nozzles and undifferentiated spray strategy, precision target spray has gained increasing attention as an important concept of precision agriculture, which brings in a more economical and environmentally friendly solution. For robotic precision spray application in vegetable farms, accurate plant phenotyping through instance segmentation and robust plant tracking are of great importance and a prerequisite for the following spray action. Regarding the robust tracking of vegetable plants, to solve the challenging problem of associating vegetables with similar color and texture in consecutive images, in this paper, a novel method of Multiple Object Tracking and Segmentation (MOTS) is proposed for instance segmentation and tracking of multiple vegetable plants. In our approach, contour and blob features are extracted to describe unique feature of each individual vegetable, and associate the same vegetables in different images. By assigning a unique ID for each vegetable, it ensures the robot to spray each vegetable exactly once, while traversing along the farm rows. Comprehensive experiments including ablation studies are conducted, which prove its

---

*Corresponding Author, Email: sudao@cau.edu.cn

superior performance over two State-Of-The-Art (SOTA) MOTS methods. The proposed method achieves a Higher Order Tracking Accuracy (HOTA) score higher than 70 and an Association Precision (AssPr) score higher than 80. The execution speed of the method reaches 29 Frames Per Second (FPS) on a consumer level hardware, which satisfies the real-time operation. Compared to the conventional MOTS methods, the proposed method is able to re-identify objects which have gone out of the camera field of view and re-appear again using the proposed data association strategy, which is important to ensure each vegetable be sprayed only once when the robot travels back and forth. Although the method is tested on lettuce farm, it can be applied to other similar vegetables such as broccoli and canola. Both code and the dataset of this paper is publicly released for the benefit of the community: https://github.com/NanH5837/LettuceMOTS.

Keywords: agricultural robot; precision agriculture; deep learning; precision spray; instance segmentation; multi-object tracking and segmentation; phenotyping

1   Introduction

With the world population growth and climate change, the urgent need for food safety and sustainable production have put forward higher requirements for agriculture. With the shortage of labor force and the limited area of arable land, artificial intelligence and robotics technologies have gained significant attention in agriculture recently. Agricultural robots are increasingly being deployed for tasks including weeding (McCool et al., 2018), crop and weed detection (Bac et al., 2017), and pesticide and fertilizer application (Adamides et al., 2017) *etc.* Application of liquid pesticide and fertilizer is an important process in planting vegetables. Conventional spraying techniques tend to apply liquid pesticide and fertilizer uniformly on vegetable farms, which not only leads to a waste of chemical, but also is not environmentally friendly. In comparison, precision spray of individual plant can effectively resolve the above problem (Chebrolu et al., 2017). Images captured by the vision sensor of the robot can be used to detect vegetables and compute the location of the each vegetable, which guides the robot to apply chemical to each individual vegetable. Accurate detection of vegetables is prerequisite for robotic precision spray. However, only detection of vegetables is usually not enough for robotic precision spray. With only detection results of vegetables, robots usually have to move with a fixed distance every time and spray all detected vegetables at each stop. In this way, robots also have to make sure that there is no overlap between two camera field of views, as well as no vegetable is missed between two consecutive camera filed of views. However, ensuring such a fixed distance is normally difficult, and a vegetable plant is likely to either get sprayed more than once, or be missed by the robot. Another way to handle this problem is to use Global Navigation Satellite System (GNSS) information or Simultaneous Localization and Mapping (SLAM) methods to mark down geo-information of vegetables, so that each plant is assigned to a unique ID. However, GNSS antenna introduces additional cost and is unstable

inside greenhouse. Visual SLAM algorithms are generally not robust in the semi-structured agricultural environment.
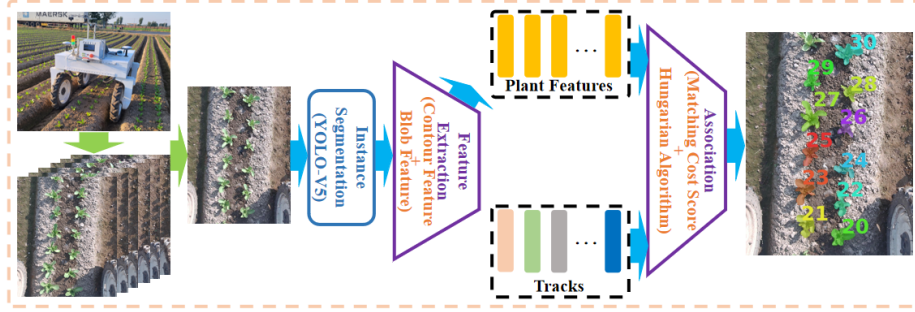

Figure 1: Overview of the proposed method. A downward facing RGB camera is attached in front of the agricultural robot VegeBot, which collects a sequence of images while traveling through a lettuce farm. The proposed method segments and tracks each vegetable instance. Vegetable plants are firstly segmented with YOLOv5 instance segmentation net. Then, shape features consisting of contour and blob characteristics of plants are extracted to tackle the challenging data association problem of plants with similar color and texture. Based on the defined matching cost and Hungarian algorithm, accurate and robust tracking of plants is obtained. The details of the proposed method are described in section 3.

To tackle this challenging problem in a better way, robots have to solve the data association problem for the detected vegetables on the consecutively captured images. When the same vegetable plants on the consecutively captured images are correctly associated to each other and identified as one, unique IDs are assigned to them and the robot can ensure to spray each plant only once. In this way, it forms the classic Multiple Object Tracking (MOT)(Bewley et al., 2016; Zhang et al., 2022) or MOTS problem (Voigtlaender et al., 2019; Gao et al., 2022). Compared to MOT methods(Hu et al., 2022), which use bounding boxes to detect vegetables, MOTS provides instance segmentation image mask, which can be used to infer unique characteristics and phenotype of individual vegetable, such as its size and shape, to determine optimal doze of chemical spray.

In this paper, a novel method of MOTS is proposed for instance segmentation and tracking of multiple vegetable plants for precision spray application of agricultural robots. The overview of the method is shown in Fig. 1. To tackle the challenging problem of associating vegetables with similar color and texture in consecutively captured images, their shape features, which consist of contour and blob features, are extracted to describe a unique feature for each individual vegetable and match the same vegetables in different images. By assigning a unique ID for each vegetable, it ensures the robot to spray each vegetable exactly once, while traversing along the farm rows. Compared to the conventional MOTS methods, the proposed method is able to re-identify objects which have gone

out of the camera field for a long time and re-appear again. With the proposed plant shape feature and data association strategy, the plant which re-occurs into the camera field of view is re-identified and its ID on its first occurrence is recovered. It is common for most of agricultural robots to travel backward when it needs to avoid unexpected obstacles or casually move out for refilling or recharging. Therefore, this is important to ensure each vegetable is sprayed only once, when the robot traverses back and forth and vegetables re-appear in the camera images.

The contributions of this paper are summarized as follows:

1) Firstly, a novel MOTS method is proposed for tracking and segmentation of vegetable plants for robotic precision spray. Based on the instance segmentation of each vegetable plant, shape information of each plant is extracted and matched for tracking individual plants. Specifically, the contour information of a plant represented by Fourier Descriptor (FD) and blob information of a plant represented by parameters of the fitted ellipse are used to uniquely identify the plant. Shape features of all tracks are stored, and during data association, not only the plants in the current active tracks are searched, but also plants which have gone out of the camera field of view but geographically close the plants in the current camera field of view are searched. As a result, the proposed method can effectively re-identify these re-occurred plants, and recover their previous IDs.

2) A lettuce multi-object tracking and segmentation dataset, LettuceMOTS, is constructed and publicly released. It contains 12 sequences, 1308 RGB images with corresponding annotated labels, 314 object instances and 17562 masks. Based on the LettuceMOTS dataset, comprehensive experiments including ablation studies are conducted, which show the superior performance of the proposed method over two SOTA MOTS methods.

3) The implementation of the proposed method is also publicly released for the benefit of the community.

The rest of the paper is organised as follows. In section 2, the related work of plant segmentation in agriculture and MOTS methods are discussed. In section 3, the details of the proposed method are illustrated. In section 4, the details of data acquisition and structure of dataset are provided. In section 5, experimental validation of the proposed method, comparison against two SOTA MOTS methods, and ablation studies are presented. section 6 presents conclusions and a discussion about further work.

2   Related Work

Two main research fields related to the proposed method are accurate segmentation of plants and efficient tracking of them. Therefore, related work in terms of plant segmentation and MOTS is presented in this section.

4

2.1 Plant Segmentation

Plant segmentation requires the precise separation of plant from the background. Early work utilizes hand-crafted feature for crop segmentation (Song and Yang, 2015). However, hand-crafted feature needs to be designed and adjusted according to the specific application and situation, and is affected by factors such as illumination change. In recent years, the emergence and application of Deep Neural Network (DNN) have triggered fundamental changes in the field of computer vision. This is because the more advanced and representative features are extracted by a large number of convolutional layers and pooling operations. The perception ability of robots has also been greatly improved with the continuous advancement of DNN. Recently, many methods based on deep learning have been proposed for plant segmentation and achieved impressive results (Bargoti and Underwood, 2017; Milioto et al., 2018).

Bargoti and Underwood (2017) deployed a ground vehicle to collect images in an apple orchard. Apple segmentation is conducted by utilizing multiscale multilayered perceptrons and Convolutional Neural Network (CNN), and the number of apples is counted with Watershed Segmentation and Cyclic Hough Transforms. The results show that the combination of Watershed Segmentation and CNN achieves the best counting performance, and the square correlation coefficient is 0.826. Milioto et al. (2018) proposed a method for semantic segmentation of sugar beet utilizing vegetation indexes. The results show that it achieves image processing speed of 20Hz on a variety of robotic systems. Khan et al. (2020) proposed CED-Net, a semantic segmentation approach to classify plant. This method is based on a cascaded encoder-decoder network, and outperforms other segmentation architectures at the time on four public agricultural datasets. Bai et al. (2022) deployed a multi-network model to solve the problem of cucumber segmentation and detection in multiple scenarios. They first utilized the improved U-Net (Ronneberger et al., 2015) method to perform pixel-level segmentation of cucumbers, and then performed the further detection with the object detection algorithm.

These methods can accurately segment plants and locate them with pixel-level accuracy. However, they do not solve the problem of tracking the same plant on consecutive frames. Traditionally, agricultural robots have to move a fixed distance, segment and spray all plants in the current camera field of view, and move to the next stop. It has to ensure either two adjacent camera field of views have minimum overlapping region and do not contain any same plant, so as to achieve the purpose of neither repeating nor missing any plant. However, it brings in extra harsh requirement of precise robot navigation, which is normally difficult to achieve for most robotic platforms in the challenging farm environment. This problem can be solved by MOT or MOTS technology, which assigns an ID to each plant for continuous tracking, and sprays each plant only once. With every plant being tracked, the precise navigation requirement is effectively released, and navigation becomes uncoupled with perception to most extent.

2.2 Multiple Object Tracking and Segmentation

The main goal of MOT is to detect and associate the same object in an image sequence (Luo et al., 2021). Currently, these methods are divided into two categories, single-stage MOT methods (Zhang et al., 2021; Liang et al., 2022) and two-stage MOT methods (Bewley et al., 2016; Wojke et al., 2017; Zhang et al., 2022).

SORT proposed by Bewley et al. (2016) is a simple and fast tracking system. It predicts the position of targets in the current frame through the Kalman filter (Kalman, 1960), and matches them with the Hungarian algorithm (Kuhn, 1955). Wojke et al. (2017) proposed DeepSort based on Sort, which integrates the appearance model to obtain the feature embedding of the object. It further solves the tracking failure problem caused by occlusion. The downside of the method is that it handles detection and feature extraction tasks separately, which slows down the its processing. Wang et al. (2020) presented the first MOT system that placed object detection and feature embedding in the same task network, and achieved near real-time running speed. By utilizing two homogeneous branches to perform detection and feature extraction tasks separately, Zhang et al. (2021) overcame unfairness of the operation of the two tasks and achieved high detection and tracking accuracy.

MOTS extends the perception accuracy of MOT further, by replacing the bounding box to pixel-wise instance segmentation. Voigtlaender et al. (2019) first came up with the concept of MOTS and proposed a baseline method named TrackR-CNN. It extends the Mask R-CNN (He et al., 2017) with three-dimensional convolution to combine contextual information and deploys association head to extract instance embedding for data association. Xu et al. (2020) proposed PointTrack, which performed the tracking-by-instance segmentation paradigm. It first obtains high-quality instance segmentation results with spatial embedding (Neven et al., 2019), and then extracts instance features from the segmentation results through an unordered 2D point cloud. Based on PointTrack, Gao et al. (2022) deployed SENet (Hu et al., 2017) as an instance segmentation network, and utilized IDNet to extract object embedding for lightweight and high efficiency. These methods show accurate and robust results in tracking cars and pedestrians of large variance in color and texture. However, to track plants in farms, which have similar color and texture, color and texture embedding is prone to failure. Furthermore, these methods discard objects which have gone out of the camera field of view after a long time, and assign new IDs to them if they re-occur again. For robotic precision spray application, this means repeated spray for the same plant.

Specifically for agricultural application, several MOTS methods have been successfully applied. de Jong et al. (2022) presented a MOTS dataset containing apple instances using wearable cameras and drone recordings. Experimental results of two open-source methods show that tracking apples with similar color and texture is challenging. Qiang et al. (2022) proposed a tracking method for leafy plants. They first apply a weakly supervised instance segmentation of leafy vegetables through semi-supervised learning. Mask Intersection over Union (IoU)

and bipartite graph matching are then used for data association and tracking. However, this method only uses the mask IoU as a position feature, which is more similar to MOT, and does not take full advantage of the information obtained by the pixel level instance mask.

3 Method

3.1 Feature Exaction

The overview of the proposed method is shown in Fig. 2. It adopts the famous YOLOv5 architecture to obtain instance segmentation of vegetables. YOLOv5 is preferred among various other instance segmentation nets due to its high accuracy and lightweight, which is important for the real time requirement for robotic operation. The output size of the net is a matrix of $N \times H \times W$, where N represents the number of objects, H and W are the height and width of the input image, respectively. The mask value of the plant is 1, and that of the background is 0.

In agricultural scenarios, tracking of plants is generally difficult when conventional color and texture features are used (Hu et al., 2022; de Jong et al., 2022). Our previous work (Hu et al., 2022) presented a location information based feature extraction method based on the geometric relationship between the target plant and its neighboring plant. The method overcomes the challenging tracking problem of plants with similar appearance, but it requires presence of multiple plants on an image to extract such location feature. The proposed method pushes perception accuracy further to the pixel level, gaining more useful information such as plant shapes, which can be used to differentiate different plants.

The proposed method uses FD to extract plant contour information based on instance segmentation mask of each plant. The FD is a shape description method based on the Fourier transform of the shape contour and can represent the shape information in the frequency domain. In addition, blob feature of each plant is extracted by fitting an ellipse to its image mask. The blob feature is represented by the ratio of the major and minor axes, denoted as R, and center rotation angle, denoted as θ. θ angle is normalized to keep R and θ similar in magnitude. The combination of FD of contour information and ellipse parameters of blob information serves as the shape information of the plant. It allows maximum information to be obtained with a small size number description dimension, which ensures less memory consumption. It is important for the shape information descriptor to be less in size, since descriptors of all plants are stored for tracking re-occurred plants.

As shown in Fig. 2. Firstly, contour of the plant is extracted based on its instance segmentation mask, using Suzuki85 border following algorithm (Suzuki and Abe, 1985). The contour descriptor FD, and blob descriptor R and θ are computed based on this. The extracted contour in t frame is represented by its image coordinates as follows:
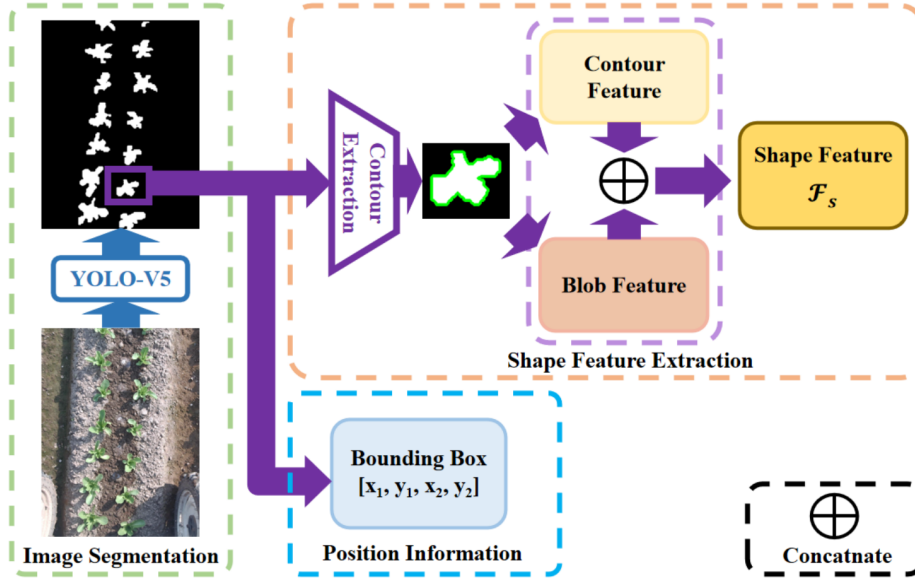
Figure 2: Details of feature extraction. YOLOv5 instance segmentation yields masks and bounding box coordinates of all plants in the image. Based on the plant mask, the plant contour is extracted. Then FD is applied to the plant contour to obtain contour feature, and an ellipse is fitted to the plant contour to obtain blob feature. The shape feature of the plant is obtained by combining the contour and blob features.

$$\mathcal{C}_n^t = \{(x_n, y_n)|n = 0, 1, \cdots, N - 1\}. \tag{1}$$

The FD derived in the form of centroid distance is used in this paper, which can better describe the shape features of the object (Zhang and Lu, 2003). The centroid of the contour point is first calculated as follows,

$$x_c = \frac{1}{N}\sum_{n=0}^{N-1} x_n, y_c = \frac{1}{N}\sum_{n=0}^{N-1} y_n, \qquad n = 0, 1, \ldots, N-1, \tag{2}$$

where x_c and y_c are the X and Y coordinates of the contour centroid. The distance from the contour point to the centroid is computed as follows,

$$r_n = \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2}, \qquad n = 0, 1, \ldots, N-1. \tag{3}$$

Then, a discrete Fourier transform for the centroid distance r_n is applied as follows,

$$\Gamma_k = \frac{1}{N}\sum_{n=0}^{N-1} r_n e^{\frac{-j2\pi kn}{N}}, \qquad k = 0, 1, \ldots, N-1. \tag{4}$$

With a set of obtained Γ_ks composed of complex numbers, the contour descriptor FD of the plant is computed by carrying out translation, rotation, and scale invariance operation as follows,

$$\bar{\Gamma}_i = \frac{\|\Gamma_k\|}{\|\Gamma_1\|}, \qquad i = k - 2, k = 2, 3, \ldots, N - 1. \tag{5}$$

Note the first element Γ_0 is not used. Since the size of $\bar{\Gamma}_i$ is not fixed, the first I elements of it is selected to represent its contour descriptor vector as follows,

$$\hat{\Gamma}_I = \begin{bmatrix} \bar{\Gamma}_0 \\ \vdots \\ \bar{\Gamma}_{I-1} \end{bmatrix}, \tag{6}$$

In the baseline form of the proposed method, the first 5 elements, $i.e.$ $I = 5$, are selected to make a balance between performance and speed.

To extract the blob feature of the plant, an ellipse formulated below is fitted to plant mask,

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \tag{7}$$

where A, B, C, D, E, and F are the six parameters of the ellipse. When the nth contour point (x_n, y_n) of the plant mask is considered, the corresponding fitting error E_n is,

$$E_n = \begin{bmatrix} x_n^2 & x_n y_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix}. \tag{8}$$

Then the sum of squared errors of all contour points, E_S, is used to represent ellipse fitting error,

$$E_S = \sum_{n=0}^{N-1} E_n{}^2, \qquad n = 0, 1, \ldots, N - 1. \tag{9}$$

When, E_S is minimized, we have the parameters of the optimum ellipse fitting.

The algebraic distance algorithm (Fitzgibbon and Fisher, 1995) is used to minimize the objective function E_S. Based on the obtained ellipse, the ratio R of the long and short axes, and rotation angle θ of the ellipse are used to construct the blob feature.

Finally, the shape feature of the plant is the combination of the contour feature $\hat{\Gamma}_I$ in eq. (6) and the blob feature represented by R and θ of the ellipse. It is formulated as follows,

$$\mathcal{F}_s = \begin{bmatrix} \hat{\Gamma}_I \\ R \\ \theta \end{bmatrix}, \tag{10}$$

9

In addition to the proposed shape information of the plant, the position information of the plant in the image frame is also utilized, as many MOTS methods do. The position information is specifically represented by parameters of the coordinates of the bounding box containing the plant mask, which is denoted as follows,

$$\mathbf{B} = \begin{bmatrix} B_1^x & B_1^y & B_2^x & B_2^y \end{bmatrix}, \tag{11}$$

where B_1^x, B_1^y are the horizontal and vertical coordinates of the upper left corner of the bounding box, and B_2^x, B_2^y are the horizontal and vertical coordinates of the lower right corner of the bounding box.

During tracking process, the bounding box of a plant in the current frame is predicted by Kalman filter first. When data association is successfully carried out, the bounding box is updated accordingly. The shape feature of the successfully tracked plant is updated to that in the current frame. Note that since the plant at top or bottom of the image does not appear completely, they are discarded to maintain the performance of tracking process.

3.2 Data Association

Data association is the process of matching the objects in two frames, and it is critical for tracking plants. It mainly includes two steps, which are calculating the matching cost between different objects and using the bipartite graph matching to associate objects according to the matching cost. The data association process is summarized in Fig. 3. In the figure, the plant shape features are extracted by plant instances in the current image frame. Track refers to plants in the previously captured images frames, which has been successfully assigned unique IDs. After the data association process, a plant in the current frame either is assigned to a track if is successfully match to it, or initialized a new track if it is not matched to any previously constructed track.

In order to re-identify the re-occurred plant and recover its original ID, information of all tracks is stored in the proposed method, as opposed to only keeping active tracks in conventional MOTS methods. However, objects of the current image frame are not matched against all tracks stored in the memory, but only matched to active tracks and their geographical neighbours, since there is no sudden jump for camera field of view. By effectively reducing the number of matching candidates, the execution time and chance of incorrect matching can be efficiently minimized. Since the IDs of tracks are initialized in numerical order while the robot traverses through the farm, geographically neighbouring plants have their IDs close to each other. There, the search scope of the tracks can be restricted within the range defined as follows,

$$RNG = [ID_{min} - s, ID_{max} + s], \tag{12}$$

where ID_{min} and ID_{max} are the minimum and maximum values of the successfully tracked object IDs in the previous frame, respectively. The variable s controls the search scope and it is related to the maximum number of new objects that can potentially show up in the next frames. It is set to 6 in the
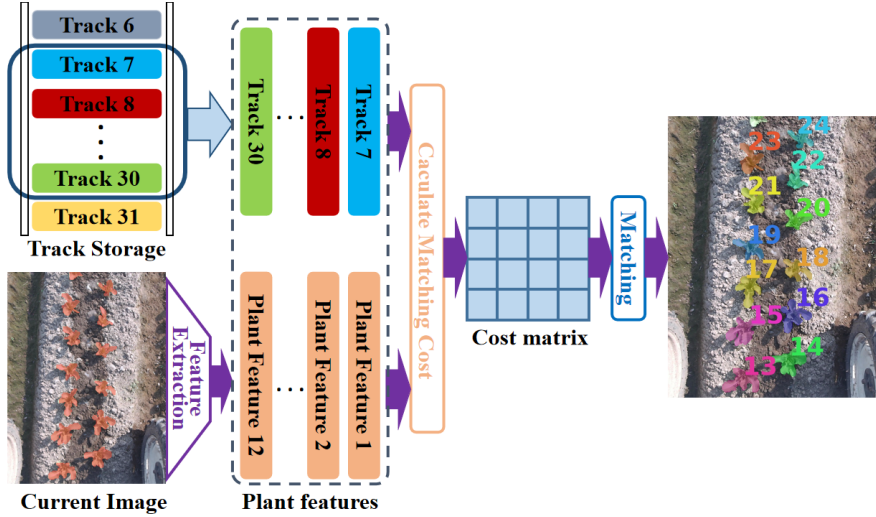
Figure 3: Data association process. Shape features of plants in the current image frame are extracted. Meanwhile, tracks which are geographically close to the current active tracks, which are successfully matched to plants in the previous frame, are selected for match candidates. Matching cost is computed between plant features and tracks, and the matching cost matrix is constructed. Hungarian algorithm is used to resolve the optimum data association problem based on the matching cost matrix.

implementation according to the speed of the robot, which ensures sufficient number of neighbouring plants are considered, and filters out plants that are too far away.

To tackle the challenging problem of matching plants with similar appearance, establishing an effective matching cost is key to data association. The proposed matching cost consists of a position matching cost and a shape matching cost.

Firstly, for position matching cost, Generalized Intersection over Union (GIoU) is adopted to calculate the position relationship between the two bounding boxes of plants as follows,

$$GIoU = IoU - \frac{|Area_C - Area_{Union}|}{|Area_C|}, \tag{13}$$

where $Area_C$ is the area of the smallest rectangle containing two coordinate boxes, and $Area_{Union}$ is the area of the union of two bounding boxes. Since the range of IoU is between 0 and 1, the position similarity based on IoU is less distinguishable than GIoU adopted by the proposed method. The value of GIoU goes to -1 when the distance between two boxes is infinite, and goes to 1 otherwise. Then, the calculation of the position matching cost can be defined as follows,

$$\delta_p = -G(B_i^{t-1}, B_j^t), \tag{14}$$

where δ_p is the matching cost of positions, G represents the GIoU in eq. (13) between the two bounding boxes. When calculating the position matching cost, Kalman filter is used to obtain the predicted position of the plant bounding box in the current frame. B_i^{t-1} and B_j^t represent the bounding box position predicted by Kalman filter of the plant in the $t-1$ frame, and the bounding box position of the detected plant in the t frame, respectively. However, when a track is out of camera field of view, B_i^{t-1} represents the position of the bounding box in the last frame before the plant disappears.

For the shape matching cost, the cosine distance is utilized to indicate the cost score between two shape features, which is defined as follows,

$$\begin{aligned}
\delta_s &= D(\mathcal{F}_{s}^{t-1}{}_i, \mathcal{F}_{sj}^t) \\
&= 1 - \frac{\mathcal{F}_{s}^{t-1}{}_i \mathcal{F}_{sj}^t}{\|\mathcal{F}_{s}^{t-1}{}_i\|_2 \|\mathcal{F}_{sj}^t\|_2},
\end{aligned} \tag{15}$$

where D refers to the cosine distance between the two plant shape features. $\mathcal{F}_{s_i}^{t-1}$ and $\mathcal{F}_{s_j}^t$ represent the shape feature vectors of two plants in $t-1$ frame and t frame, as defined in eq. (10). Lower shape matching cost means two plants in the consecutive frames are more likely to be the same plant. When the two features are exactly the same, the cosine distance reaches 0.

When the robot traverses on farm, the plant that disappears from the camera field of view is not tracked, and only plants detected in the current frame are tracked. For plants being actively tracked, both shape and position similarities are effective for matching, while only the shape similarity is effective for re-identifying the re-occurred plants. The overall matching cost combining both shape and position matching cost is formulated as follows,

$$\delta = \delta_s(1 + \alpha \delta_p), \tag{16}$$

where δ refers to the overall matching cost, and scalar α controls the influence of the position matching cost. For currently active tracks, the position information is more effective for matching, so α is set to be 1. For tracks that are currently inactive, their shape features are more effective and α is set to be 0.

In addition, thresholds are applied to both position and overall matching costs to further filter out false positive matches whose position and overall matching costs are way too large. Threshold operation is formulated as follows,

$$\delta = \begin{cases} \delta & \delta < T_{all} \ and \ \delta_p < T_p \\ \infty & otherwise \end{cases}, \tag{17}$$

where T_{all} and T_p are maximum thresholds for overall and position matching costs, respectively. Based on empirical results, T_{all} and T_p in the implementation are set to 0.1 and 0.4, respectively.

Finally, based on the overall matching cost δ, Hungarian algorithm (Kuhn, 1955) is utilized for data association of multiple plants.

4   LettuceMOTS Dataset

In addition to the proposed MOTS method, this paper presents and publicly releases a challenging lettuce MOTS dataset, LettuceMOTS, captured by an agricultural robot. This section describes the agricultural robot and its sensor used in data acquisition and the structure of the dataset.

4.1   Data Acquisition

All images in LettuceMOTS were collected from a lettuce farm in Tongzhou District, Beijing, China, in September to October 2022 as shown in Fig. 4(a). The distance between two adjacent lettuces in the same row is about 0.15 m to 0.25 m, and the distance between adjacent rows is about 0.25 m to 0.3 m. The maximum weed density is close to 10 per square meter due to regular weeding.

(a)   (b)

Figure 4: Details of data acquisition. Fig. 4(a) shows the lettuce farm where the data collection took place. Fig. 4(b) shows the setup of the data acquisition process based on VegeBot.

Images are collected by VegeBot, a four-wheel-steer and four-wheel-drive agricultural robot designed and manufactured by China Agricultural University to perform autonomous operation in vegetable farms. The key parameters related to VegeBot are listed in table 1. As is shown in Fig. 4(b), the VegeBot is equipped with two vision sensors, and a RTK-GPS sensor for GNSS based global localization. An Intel RealSense D435i depth camera with IMU sensor is mounted in front of the robot tilted downward for localization and autonomous navigation. A RGB monocular camera is mounted on top front location of the robot facing straight downward for segmenting and tracking vegetables for precision spray application.

The VegeBot can adopt two motion control modes: vision-based motion autonomous navigation and manual drive with a remote controller. In order to ensure the quality of the acquired images, the robot is remotely control during data acquisition process. Since the velocities and steering angle of the robotic four wheels can be controlled independently, the robot can move in a variety of

motion modes. At the time of data collection, Ackerman steering is adopted for the robot to travel straight forward or backward along the farm lanes.

Table 1: Key parameters of VegeBot

Parameter	Value
Length	1.2 m
Width	1.1 m
Height	1.1 m
Weight	approx. 350 kg
Max Load	200 kg
Max Speed	0.8 m/s

The RGB camera used in this paper is installed approximately 1.4 m away from the ground. The camera has a 1/2.8-inch SONY IMX317 CMOS sensor. Its maximum resolution and pixel size are 3840 × 2160 and 1.62 μm × 1.62 μm, respectively. The camera is able to capture images at 30 FPS when the resolution is set to be 3840 × 2160 or 120 FPS when the resolution is set to be 1920 × 1080. The lens of the camera has a field of view (FOV) of 100 degrees and the f-number of 2.7.

Data collection is carried out at three different growth stages of the lettuce. Views of the lettuce farm and typical captured images at three periods are shown in Fig. 5. The robot travels at a mostly constant velocity within a range between 0.35 m/s to 0.4 m/s. The robot moves both forward and backward, and there are vegetables, which re-occur after they have gone out of camera field of view for a long time, as a result. This makes our dataset more challenging than existing MOTS datasets. The camera captures images with a resolution of 1920×1080 at 15 FPS, and images were cropped into the resolution of 810×1080 to remove irrelevant areas. Images are collected in natural light, and the exposure time of the camera is automatically determined.

4.2 Dataset Structure

The format and structure of LettuceMOTS follow the famous KITTI MOTS format (Geiger et al., 2012, 2013). The annotation tool $CVAT$(https://github.com/opencv/cvat) is used to label the captured images. $CVAT$ is developed and open sourced by Intel, and its tracking, interpolation and fine tuning labeling method reduces manual labeling time significantly. The resulting annotation file of TXT format has the following format,

$$instance = \{frame_{id}, object_{id}, category, image_{height}, image_{width}, RLE\}, \quad (18)$$

where $frame_{id}$ is the ID number of the frame, $object_{id}$ is the ID number of the object, and $category_{id}$ is the ID number of the category of object. $category$ is 1 for vegetable. $image_{height}$ and $image_{width}$ are the height and width of the image

14

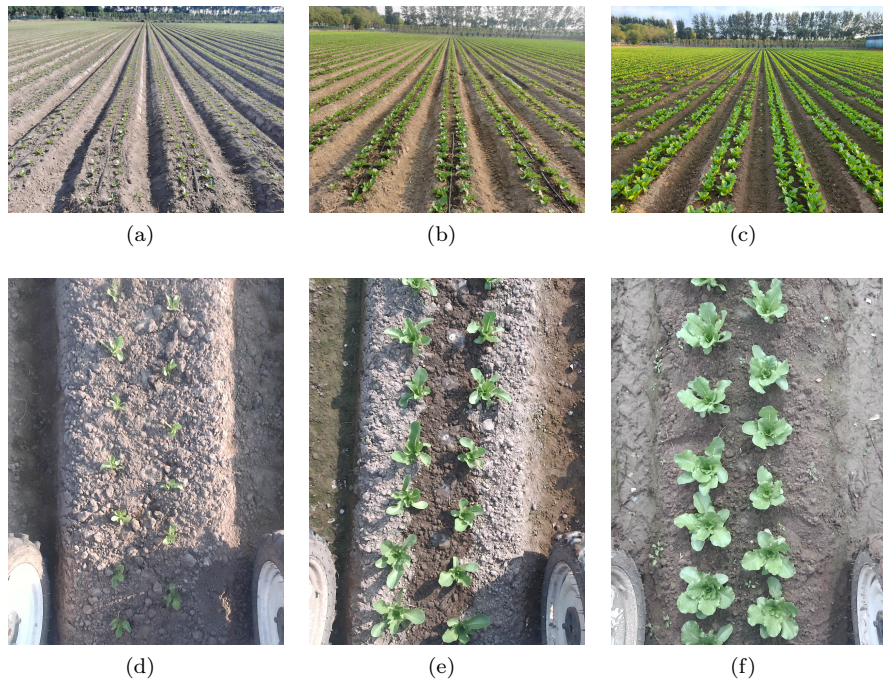|		
(a)	(b)	(c)
(d)	(e)	(f)

Figure 5: Overview of the lettuce farm and typical captured images at 3 different growth stages of lettuces. Lettuces are at the growth stage of seeding, rosette, cupping and head in subfigures (a) and (d), (b) and (e), and (c) and (f), respectively.

in which the plant is located. *RLE* is a string of numeric encoding containing the plant mask information.
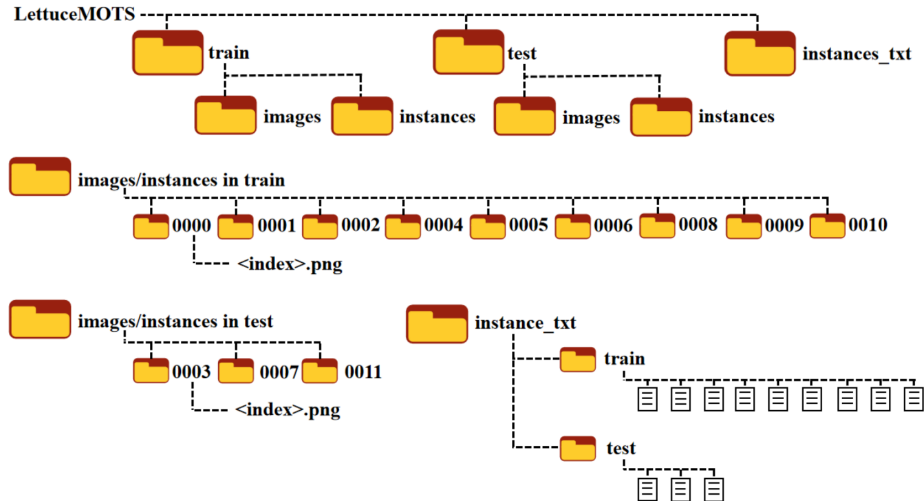

Figure 6: Structure of LettuceMOTS dataset. The images and instances folders under train and test folders contain the captured RGB images and their corresponding instance segmentation masks for different image sequences. The instance_txt folder contains MOTS annotation for plant segmentation and tracking for different image sequences.

The file structure of LettuceMOTS is shown in Fig. 6. It contains a total of 1308 frames, 314 objects, and 17562 manual annotated masks. The 12 sequences of the dataset are numbered from 0000 to 0011, with each growth period containing four sequences. Among them, three of them serve as the training set, and the other one serves as the test set. Detailed information about LettuceMOTS can be found in table 2.

5 Experimental and Results

5.1 Implementations Details

As mentioned in section 3.1, YOLOv5(Ultralytics, 2022) is chosen to segment images. For segmentation of vegetables, the YOLOv5m model is selected to make a balance between accuracy and efficiency. It is fine tuned on the training set of the LettuceMOTS based on the pre-trained model based on COCO dataset provided by its author. SGD is used as the optimizer during training and the initial learning rate is set to be $1e^{-2}$. All other training parameters follow default parameters provided by YOLOv5.

Table 2: Summary of the proposed LettuceMOTS dataset.

Dataset sequences	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	Total
Length(Frame)	88	50	48	245	117	51	47	244	96	54	45	223	1308
Plant Number	23	23	21	40	22	22	21	37	21	23	24	37	314
Re-occurred Plants	9	/	/	26	10	/	/	24	8	/	/	23	/
Mask Number	1243	678	571	3304	1444	669	577	3341	1252	733	677	3073	17562
Robot Motion	F	F	F	B-F	F	F	F	B-F	F	F	F	B-F	/
Growth Stage	Seedling stage				Rosette stage				cupping and head stage				/
Weather	Sunny				Sunny				Cloudy				/
Light intensity	Strong				Strong				Weak				/
Date	Afternoon, Sept. 21, 2022				Afternoon, Sept. 29, 2022				Morning, Oct. 5, 2022				/

[1] F and $B-F$ in the Robot Motion refers to the robot motion of moving forward continuously, and the robot motion containing both forward and backward, respectively.

17

For comparison purpose, two open source SOTA MOTS methods TrackR-CNN and PointTrack(The following open source implementations are used in the experiment. TrackR-CNN: `https://github.com/VisualComputingInstitute/TrackR-CNN` and PointTrack: `https://github.com/detectRecog/PointTrack`) are tested. They are fine tuned on the training set of LettuceMOTS based on the pre-trained model provided by the author. For PointTrack, learning rates of segmentation networks and tracker are set to $5e^{-6}$ and $2e^{-3}$, respectively. The learning rate of TrackR-CNN is set to be $5e^{-7}$. All other hyper parameters follow the default implementation. All three methods, including the proposed method, are trained for 100 epochs to ensure fairness. The training and inference of all methods are conducted on a computer with a NVIDIA GeForce RTX 2080Ti GPU and a Intel® Core™ i7-10700K CPU.

5.2 Evaluation Metrics

The segmentation and tracking performance of the three methods are evaluated separately. The performance of instance segmentation is measured by Average Precision (AP), which is widely used by many classic datasets, *e.g.* COCO dataset (Lin et al., 2014). This metrics adopts AP, AP_{50} and AP_{75} to show accuracy of segmentation. AP_{50} and AP_{75} are the AP at IoU of 0.5 and 0.75, respectively. AP is the average of ten AP_{IoU}s, with IoU ranging from 0.5 to 0.95 and an increase of 0.05 every step.

The evaluation of MOTS is relatively more complex than segmentation. In this paper, HOTA proposed by Luiten et al. (2021) is utilized for the evaluation of tracking tasks, which balances the performance of segmentation and tracking. HOTA can better reflect the human's visual perception for MOTS evaluation. It is calculated by Detection Accuracy Score (DetA) and Association Accuracy Score (AssA) as follows,

$$HOTA = \sqrt{DetA \cdot AssA},\tag{19}$$

where DetA and AssA represent comprehensive metrics of segmentation accuracy and association accuracy. The association metrics are defined as follows,

$$AssA = \frac{AssRe \cdot AssPr}{AssRe + AssPr - AssRe \cdot AssPr},\tag{20}$$

where Association Recall (AssRe) reflects how good predicted trajectories cover ground truth trajectories, while AssPr reflects the ability of predicted trajectories to continuously track the same ground truth trajectories. The detailed description of DetA, AssA, AssRe and AssPr can be found in the original work (Luiten et al., 2021), which is omitted here for the brevity of the paper.

In this paper, we compute the above mentioned MOTS evaluation metrics using the KITTI MOTS official kit(`https://github.com/JonathonLuiten/TrackEval`).

5.3 Validation Results

5.3.1 Results and Comparison

Firstly, we evaluate the segmentation results of the three MOTS methods with AP metrics, and results are shown in table 3. It can be seen from the table that PointTrack has the highest AP score for segment accuracy. When IoU is 0.5 and 0.75, YOLOv5 has the best segmentation performance among three methods. TrackR-CNN does not get the highest score, but achieves good and balanced result.

Table 3: Segmentation performance of the proposed method and comparison to two state-of-the-art MOTS methods.

Dataset	Method	$AP \uparrow$	$AP_{50} \uparrow$	$AP_{75} \uparrow$
0003	TrackR-CNN	0.592	0.967	0.770
	PointTrack	**0.662**	0.852	**0.851**
	Ours(YOLOv5)	0.595	**0.983**	0.796
0007	TrackR-CNN	0.720	0.957	0.919
	PointTrack	**0.824**	0.940	0.940
	Ours(YOLOv5)	0.757	**0.979**	**0.961**
0011	TrackR-CNN	0.805	0.958	0.938
	PointTrack	**0.852**	0.968	0.954
	Ours(YOLOv5)	0.843	**0.977**	**0.955**

[1] Symbols $\uparrow$ after the evaluation metrics indicate the value of it is the higher the better. The bold numbers show the best performing method.

Then, tracking performance of the proposed method is compared against the two SOTA MOTS methods with the test set of LettuceMOTS using the MOTS metrics mentioned in section 5.2. Results are shown in table 4. It can be seen that our method yields superior performance than the other two methods in general. Specifically, our method gets the highest scores on all three test sets in terms of HOTA, AssA and AssPr. It yields low DetA scores, since the proposed method discards plants on top or bottom of the captured images that do not appear completely. As mentioned in section 3.1, we impose such constraint to minimize the false positive data association, especially for re-identifying re-occurred plants, since the plants with incomplete appearance show quite different shape feature. PointTrack yields the lowest AssPr because the same ID is assigned to multiple objects, and hence is not successful in tracking the same plant. Since PointTrack also introduces offset, position, and IoU as clues during data association, ID switch rarely occurs and high AssRe scores are achieved. TrackR-CNN shows frequent ID switches when tracking the same object, and thus yields lower AssRe scores.

Qualitative examples of three MOTS methods are shown in Fig. 7. We

Table 4: Performance of the proposed method and comparison to two SOTA MOTS methods.

Dataset	Method	HOTA(%) ↑	DetA(%) ↑	AssA(%) ↑	AssRe(%) ↑	AssPr(%) ↑
0003	TrackR-CNN	50.016	**77.757**	32.332	57.965	39.485
	PointTrack	45.381	74.369	27.854	62.372	31.683
	Ours	**71.989**	70.827	**73.561**	**74.792**	**84.318**
0007	TrackR-CNN	59.918	84.195	42.709	55.134	63.978
	PointTrack	60.261	**88.537**	41.066	**85.325**	42.636
	Ours	**72.083**	71.419	**72.843**	73.565	**89.998**
0011	TrackR-CNN	62.042	88.480	43.543	61.389	63.052
	PointTrack	58.374	**91.481**	37.447	**76.977**	41.083
	Ours	**70.095**	68.868	**71.433**	71.597	**95.167**

[1] Symbols ↑ after the evaluation metrics indicate the value of it is the higher the better. The bold numbers show the best performing method.

discuss the tracking ability of three methods with their performance on test set 0011 as an example. The purple arrows above images represent forward and backward directions of the robot. The number in the upper right corner of each plant is assigned ID of that plant. 30 frames are skipped between every two neighbouring rows of images. For each method, the two images at left and right side are captured in the same positions when the robot travels forward and backward. As can be seen from Fig. 7, TrackR-CNN and PointTrack yield false positive association of a plant which just goes out of camera field of view to a newly appeared plant. This is because these methods utilize instance embedding to associate objects. In our case, since vegetables are quite similar to each other in terms of color and texture, methods based on instance embedding are prone to false positive data association. In addition, these two methods do not track plants which have gone out of camera field of view for a long time. As a result, they tend to assign new IDs when these plants appear again as the robot travels backward. It is reflected by plant IDs in the right columns of images of these two methods when the robot travels backward tend to be larger than those of the same plants in the left column of the images. We also can see that, PointTrack tends to mistakenly assign a previously assigned ID, which belongs to a plant previously appeared but has just gone out the camera field of view, to a newly appeared plant. Since PointTrack matches objects mostly based on color and texture and plants have quite similar color and texture, PointTrack mistakenly believes that the newly appeared plant is the plant which just has disappeared.

In comparison, the proposed method yields superior data association performance, thanks to its extraction of vegetable shape feature which makes vegetable plants more differentiable with each other. In addition, limiting the searching range of tracks to those which are geographically close to current active tracks, *i.e.* tracks associated to plants in the previous camera image, also contributes to reducing false positive matches and increasing data association accuracy significantly.
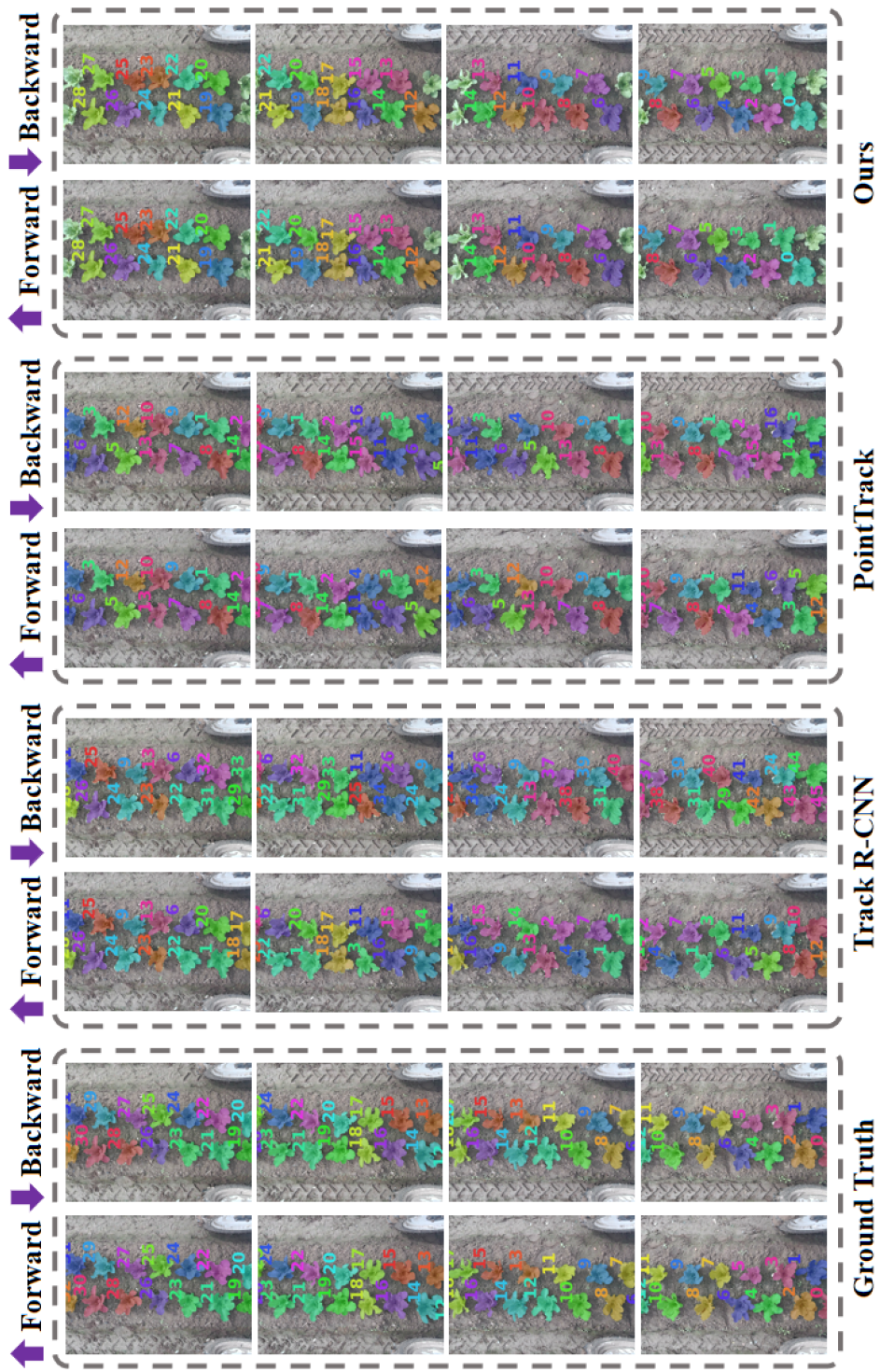
Figure 7: Tracking performance of three methods on test set 0011.

The inference speed of the three methods are shown in table 5. Our method yields the fastest running speed of approximately 29 FPS. In terms of tracking speed alone, our method yields the highest processing speed exceeding 140 FPS, since it is not a learning-based method. It can be run only on a CPU, which is easier to deploy on the robot.

Table 5: The inference speed of the proposed method and comparison to two SOTA MOTS methods.

Dataset	Method	Time(ms) ↓			FPS ↑		
		Segment	Tracking	Total	Segment	Tracking	Total
	TrackR-CNN	581.67	20.49	602.16	1.72	48.83	1.66
0003	PointTrack	69.39	44.90	114.29	14.41	22.27	8.75
	Ours	**27.80**	**6.45**	**34.16**	**35.97**	**155.22**	**29.29**
	TrackR-CNN	543.20	20.82	564.02	1.84	48.02	1.77
0007	PointTrack	61.48	36.89	98.36	16.27	27.11	10.16
	Ours	**27.50**	**6.60**	**34.02**	**36.35**	**151.56**	**29.41**
	TrackR-CNN	487.04	20.18	507.22	2.05	49.61	1.97
0011	PointTrack	67.26	40.36	107.62	14.87	24.78	9.29
	Ours	**28.07**	**7.00**	**34.93**	**35.64**	**143.05**	**28.62**

[1] Symbols ↑ and ↓ after the evaluation metrics indicate the value of it is the higher the better or the lower the better, respectively. The bold numbers show the best performing method.

5.3.2 Ablation Studies

Finally, in order to validate the effectiveness of the proposed contour feature in terms of FD of plant contour and blob feature in terms of R and θ values of the fitted ellipse, we carry out an ablation study of the feature used. Specifically, performance of the proposed method using only contour feature and only blob feature is compared to using both of them, *i.e.* the baseline of the proposed method.

The results are shown in table 6. It can be seen that the baseline of the proposed method combining both contour and blob features yields the best performance compared to using only contour or blob feature in all three test sets of different growth stages. This validates that both contour and blob features are critical and effectively contribute to the performance gain brought by the proposed method.

Next, we study the influence of the length of FD vector of the contour feature to the performance of tracking plants, by varying the length of FD vector. Specifically, different lengths of FD vector of 1, 3, 7, 9 are tested and compared with the baseline approach with the length of FD vector of 5. Results are shown in table 7, where FD_i indicates FD with the length of i. It can be seen that the performance of the method does not increase when the length of FD vector is larger than 5. Therefore, the baseline configuration of the proposed method

Table 6: Performance of the method using different plant shape features.

Dataset	Feature	HOTA(%) ↑	DetA(%) ↑	AssA(%) ↑	AssRe(%) ↑	AssPr(%) ↑
0003	Contour	64.816	**70.827**	59.699	63.673	72.945
	Blob	65.325	**70.827**	60.668	63.956	75.993
	Baseline	**71.989**	**70.827**	**73.561**	**74.792**	**84.318**
0007	Contour	70.285	**71.419**	69.247	70.331	87.920
	Blob	64.633	**71.419**	58.579	62.295	77.420
	Baseline	**72.083**	**71.419**	**72.843**	**73.565**	**89.998**
0011	Contour	66.485	**68.868**	64.278	66.303	87.968
	Blob	32.185	**68.868**	15.126	26.055	27.901
	Baseline	**70.095**	**68.868**	**71.433**	**71.597**	**95.167**

[1] Symbols ↑ after the evaluation metrics indicate the value of it is the higher the better. The bold numbers show the best performing method.

adopts FD vector length of 5 to balance between accuracy and speed.

Table 7: Performance of the method with different lengths of FD vector for plant contour feature.

Dataset	Feature	HOTA(%) ↑	AssA(%) ↑	AssRe(%) ↑	AssPr(%) ↑
0003	FD_1	56.982	45.970	50.302	66.101
	FD_3	69.360	68.186	70.079	80.894
	Baseline (FD_5)	**71.989**	**73.561**	**74.792**	**84.318**
	FD_7	**71.989**	**73.561**	**74.792**	**84.318**
	FD_9	**71.989**	**73.561**	**74.792**	**84.318**
0007	FD_1	56.252	44.340	49.359	65.862
	FD_3	71.914	72.509	73.256	**90.229**
	Baseline (FD_5)	**72.083**	**72.843**	**73.565**	89.998
	FD_7	**72.083**	**72.843**	**73.565**	89.998
	FD_9	71.656	71.983	72.873	89.210
0011	FD_1	50.500	37.125	44.436	61.218
	FD_3	69.960	71.157	71.322	**95.167**
	Baseline (FD_5)	**70.095**	**71.433**	**71.597**	**95.167**
	FD_7	**70.095**	**71.433**	**71.597**	**95.167**
	FD_9	**70.095**	**71.433**	**71.597**	**95.167**

[1] Symbols ↑ after the evaluation metrics indicate the value of it is the higher the better. The bold numbers show the best performing method.
[2] FD_i refers to taking the first i element of FD descriptors. For example, FD_5 takes the first 5 elements of FD descriptors *etc.*

Another interesting question to be answered is whether we can achieve the same tracking performance by increasing the length of FD vector of the contour feature and using such contour feature alone, *i.e.* without using the blob feature.

Therefore, we test tracking performance of the method using only contour feature of different lengths of FD vector, and compare them with the baseline approach using the FD length of 5 and blob feature. The results are shown in table 8, where FD_i w/o blob denotes the proposed method using the contour feature with FD vector of length i and without using blob feature. It can be seen from the results that the baseline approach of using both contour and blob feature still yields the best performance. Although increasing the length of FD vector increases the tracking performance, the performance gain stops when length of vector reaches a certain number, which is not as good as the baseline approach in general.

Table 8: Comparison of tracking performance of different numbers of FDs with baseline combined feature in the proposed method.

Dataset	Feature	HOTA(%) ↑	AssA(%) ↑	AssRe(%) ↑	AssPr(%) ↑
0003	FD_1 w/o blob	33.389	15.818	24.912	26.390
	FD_3 w/o blob	52.820	48.413	52.409	69.232
	FD_5 w/o blob	64.816	59.699	63.673	72.495
	FD_7 w/o blob	66.699	63.272	66.515	76.267
	FD_9 w/o blob	66.699	63.272	66.515	76.267
	Baseline (FD_5 and blob)	**71.989**	**73.561**	**74.792**	**84.318**
0007	FD_1 w/o blob	26.712	10.025	18.818	19.118
	FD_3 w/o blob	68.008	64.840	66.240	85.860
	FD_5 w/o blob	70.285	69.247	70.331	87.920
	FD_7 w/o blob	71.656	71.983	72.873	89.210
	FD_9 w/o blob	71.656	71.983	72.873	89.210
	Baseline (FD_5 and blob)	**72.083**	**72.843**	**73.565**	**89.998**
0011	FD_1 w/o blob	21.064	6.462	17.438	9.513
	FD_3 w/o blob	56.244	46.016	51.241	69.247
	FD_5 w/o blob	66.485	64.278	66.303	87.968
	FD_7 w/o blob	**70.095**	**71.433**	**71.597**	**95.167**
	FD_9 w/o blob	**70.095**	**71.433**	**71.597**	**95.167**
	Baseline (FD_5 and blob)	**70.095**	**71.433**	**71.597**	**95.167**

[1] Symbols ↑ after the evaluation metrics indicate the value of it is the higher the better. The bold numbers show the best performing method.

[2] FD_i stands for taking the first few descriptors, for example, FD_5 is taking the first five descriptors as features *etc.*

6   Conclusions

To solve the challenging problem of associating vegetables with similar color and texture in consecutive images, in this paper, we propose a novel MOTS method for segmenting and tracking of vegetables for robotic precision spray application in agriculture. The proposed method exploits shape feature of plants consisting of their contour and blob features, rather than conventional color and texture features, and yields superior tracking performance over conventional MOTS methods in the challenging data association problem of vegetable plants with

similar color and texture. In addition, the proposed method stores all constructed tracks, and searches within tracks which are all geographically close to vegetables in the current camera field of view during every data association step. Such a tracking strategy enables it to be able to re-identify re-occurred plants again, which is important to avoid spraying these plants more than once when the robot traverses back and forth. Comprehensive experiments and ablation studies are conducted to validate the superior performance of the proposed method, as well as various property of it. Furthermore, the dataset and implementation of the method are publicly released. Potential future work includes applying the proposed method to visual SLAM and building an object level SLAM system for robust localization and mapping in vegetable farms.

References

Adamides, G., Katsanos, C., Constantinou, I., Christou, G., Xenos, M., Hadzila-cos, T., and Edan, Y. (2017). Design and development of a semi-autonomous agricultural vineyard sprayer: Human–robot interaction aspects. *Journal of Field Robotics*, 34(8):1407–1426.

Bac, C. W., Hemming, J., van Tuijl, B. A. J., Barth, R., Wais, E., and van Henten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, 34(6):1123–1139.

Bai, Y., Guo, Y., Zhang, Q., Cao, B., and Zhang, B. (2022). Multi-network fusion algorithm with transfer learning for green cucumber segmentation and recognition under complex natural environment. *Computers and Electronics in Agriculture*, 194:106789.

Bargoti, S. and Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039–1060.

Bewley, A., Ge, Z., Ott, L., Ramos, F. T., and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP 2016)*, pages 3464–3468.

Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., and Stachniss, C. (2017). Agricultural robot dataset for plant classification, local-ization and mapping on sugar beet fields. *International Journal of Robotics Research*, 36(10):1045–1052.

de Jong, S., Baja, H., Tamminga, K., and Valente, J. (2022). Apple mots: Detection, segmentation and tracking of homogeneous objects using mots. *IEEE Robotics and Automation Letters*, 7:11418–11425.

Fitzgibbon, A. W. and Fisher, R. B. (1995). A buyer's guide to conic fitting. In *British Machine Vision Conference*, pages 513–522.

Gao, Y., Xu, H., Zheng, Y., Li, J., and Gao, X. (2022). An object point set inductive tracker for multi-object tracking and segmentation. *IEEE Transactions on Image Processing*, 31:6083–6096.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 32:1231–1237.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397.

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2017). Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2011–2023.

Hu, N., Su, D., Wang, S., Nyamsuren, P., Qiao, Y., Jiang, Y., and Cai, Y. (2022). Lettucetrack: Detection and tracking of lettuce for robotic precision spray in agriculture. *Frontiers in Plant Science*, 13.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45.

Khan, A., Ilyas, T., Umraiz, M., Mannan, Z. I., and Kim, H. (2020). Ced-net: Crops and weeds segmentation for smart farming using a small cascaded encoder-decoder architecture. *Electronics*, 9:1602.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics*, 2:83–97.

Liang, C., Zhang, Z., Lu, Y., Zhou, X., Li, B., Ye, X., and Zou, J. (2022). Rethinking the competition between detection and reid in multiobject tracking. *IEEE Transactions on Image Processing*, 31:3182–3196.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, volume 8693, pages 740–745.

Luiten, J., Osep, A., Dendorfer, P., Torr, P. H. S., Geiger, A., Leal-Taixé, L., and Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129:548–578.

Luo, W., Xing, J., and Milan, A. (2021). Multiple object tracking: A literature review. *Artificial Intlligence*, 293:103448.

McCool, C., Beattie, J., Firn, J., Lehnert, C., Kulk, J., Bawden, O., Russell, R., and Perez, T. (2018). Efficacy of Mechanical Weeding Tools: a study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters*, 3(2):1184–1190.

Milioto, A., Lottes, P., and Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *2018 IEEE International Conference on Robotics and Automation (ICRA 2018)*, pages 2229–2235.

Neven, D., Brabandere, B. D., Proesmans, M., and Gool, L. V. (2019). Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 8829–8837.

Qiang, Z., Shi, J., and Shi, F. (2022). Phenotype tracking of leafy greens based on weakly supervised instance segmentation and data association. *Agronomy*, 12.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *2015 Medical Image Computing and Computer-Assisted Intervention(MICCAI 2015)*, pages 234–241.

Song, X. and Yang, L. (2015). The study of adaptive multi threshold segmentation method for apple fruit based on the fractal characteristics. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 168–171.

Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30:32–46.

Ultralytics (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. `https://github.com/ultralytics/yolov5.com`. Accessed: 7th May, 2023.

Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., and Leibe, B. (2019). Mots: Multi-object tracking and segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pages 7934–7943.

Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking. In *2020 European Conference on Computer Vision(ECCV 2020)*, pages 107–122.

Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP 2017)*, pages 3645–3649.

Xu, Z., Zhang, W., Tan, X., Yang, W., Huang, H., Wen, S., Ding, E., and Huang, L. (2020). Segment as points for efficient online multi-object tracking and segmentation. In *2020 European Conference on Computer Vision (ECCV 2020)*, pages 264–281.

Zhang, D. and Lu, G. (2003). A comparative study of curvature scale space and fourier descriptors for shape-based image retrieval. *Journal of Visual Communication and Image Representation*, 14(1):39–57.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., and Wang, X. (2022). Bytetrack: Multi-object tracking by associating every detection box. In *2022 European Conference on Computer Vision(ECCV 2022)*, pages 1–21.

Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2021). Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087.