Smart Agricultural Technology 5 (2023) 100271

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

RoseTracker: A system for automated rose growth monitoring

Risa Shinoda [a,c,*,1], Ko Motoki [b,1], Kensho Hara [c], Hirokatsu Kataoka [c], Ryohei Nakano [b], Tetsuya Nakazaki [b], Ryozo Noguchi [a]

[a] Graduate School of Agriculture, Kyoto University, Kyoto-city, 606-8502, Kyoto, Japan
[b] Graduate School of Agriculture, Kyoto University, Kizugawa-city, 619-0218, Kyoto, Japan
[c] Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tsukuba-city, 305-8560, Ibaraki, Japan

ARTICLE INFO

Editor: Spyros Fountas

Dataset link: https://github.com/dahlian00/RoseBlooming-Dataset

Keywords:
Rose
Deep learning
Object detection
Tracking
Dataset

ABSTRACT

In cut-flower cultivation, production planning is an important task because demand fluctuates throughout the year. For precise cultivation planning, understanding the cultivation status is necessary by the growing stage. However, manually counting all the roses in the greenhouse to determine the cultivation status is difficult without incurring considerable time and labor. Some studies have engaged in detecting the number of flowers, but these studies used close-up images and could not count flowers without omissions or overlapping in an entire farm. In addition, limited datasets for object detection based on cut-flower blooming stages are available. In this study, we propose the RoseBlooming dataset and an efficient rose-monitoring system called RoseTracker to bridge the gap between computer vision techniques and the horticulture cultivation industry. The RoseBlooming dataset is the innovative dataset of labeled images for cut flowers at the growing stage. RoseTracker can detect small roses from various angles while moving the camera, reduces detection omissions, and achieves an F1 score of 0.950, thereby outperforming conventional models. For application, we used overhead images captured under actual growing conditions. RoseTracker and the RoseBlooming dataset contribute to constructing the rose-growth monitoring system in high demand worldwide.

1. Introduction

Production planning is an important task in cut-flower cultivation. Because cut flowers require freshness, producing them according to peak demand periods is valuable; peak demand periods fluctuate throughout the year depending on events such as celebrations. Several studies have examined environmental conditions, and cut-flower growth [18,21]. As indicators for such environmental controls, it is necessary to accurately assess current growth conditions. However, manually counting all the roses in a greenhouse requires considerable time and labor. An easy manner of obtaining rose cultivation status in greenhouses is in demand.

With the development of computer vision technology, several studies have been conducted on cut flowers using computer vision for flower detection [20,2,4]. However, these studies are few and detected only blooming flowers just before the harvest period, excluding those in the budding stage, which are unsuitable for the purpose of yield prediction. With the expanding scope of research on fruit flowers, more research has been conducted. For buds detection [7], kiwifruit buds were detected in blooming flowers. The primary aim of that study was robotic pollination; thus, they used close-up images. Previous studies have detected the number of flowers in a relatively large area [22,9] for crop management decisions. However, understanding the growth status of an entire farm remains almost impossible using a single photograph. In addition, even if multiple images are taken of a farm and combined, some areas will be omitted or duplicated.

The main problem in applying existing research to harvest forecasting is that they are unsuitable for a growth stage acquisition system of an entire farm. Using close-up images leads to high accuracy, but also to an inability to count them in an entire farm. To detect a wide range of areas, detection from videos is desirable; in particular, the concept of object tracking from videos is well suited for detecting objects without omissions or overlaps. Several studies have combined object detection and tracking on farms [5,14]. These existing studies have been conducted for relatively large fruits, not cut flowers.

* Corresponding author at: Graduate School of Agriculture, Kyoto University, Kyoto-city, 606-8502, Kyoto, Japan.
E-mail address: shinoda.lisa.47z@st.kyoto-u.ac.jp (R. Shinoda).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.atech.2023.100271
Received 7 April 2023; Received in revised form 31 May 2023; Accepted 11 June 2023
Available online 16 June 2023

Another problem is that, datasets on cut flowers by growth stage are limited in number. There are several publicly available flower datasets to classify the flower species [11,19,27]. However, there are a limited number of publicly available datasets for counting cutting flowers from a relatively large area. Also, accurate yield prediction requires a dataset with early growth stages such as buds, not only flowers. Therefore, actually performing stage-specific flower detection on farms is hindered. Furthermore, annotations must be conducted manually, which is time- and labor-intensive.

To overcome these issues, we introduce the RoseBlooming dataset, and the rose growth-status gain model called RoseTracker. RoseBlooming dataset images are taken from overhead using selfpropelled sprayers. Since overhead images can be captured using selfpropelled sprayers and drones, the proposed system and dataset are easily deployed in other greenhouses. This dataset contains two maturity level roses, including buds and flowers. We release this RoseBlooming dataset publicly available; the RoseBlooming dataset is an important dataset that will promote further research on stage-specific flower detection. By using this RoseBlooming dataset, we introduce RoseTracker, which counts roses depending on maturity levels by video. RoseTracker uses tracking techniques that can be used not only for object counting but also for more detailed object detection. Rosebuds can be hidden by overgrown leaves at certain angles, which makes it difficult to detect them from a single image. We believe that if we can detect objects from various angles while moving the camera and tracking the same object, we can overcome omissions in detecting small objects such as buds and flowers.

Our main contributions are as follows:

- We release the RoseBlooming dataset that allows for stage-specific flower detection. The dataset, consisting of overhead images, contains two rose cultivars and was filmed over a period of months. This is an innovative dataset that can be used to construct a flower-growth monitoring system and predict the flower yield, which is in demand in the horticultural field.
- We propose a new detection model called RoseTracker that combines object detection, object tracking, and the original regression model. This model detects roses using videos to enable the detection of each flower and bud from various angles —— even those hidden by leaves.

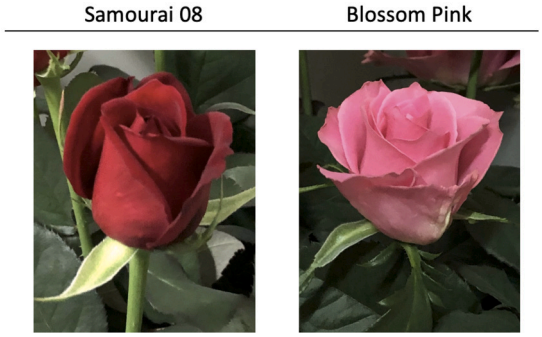

Fig. 1. The example images of two cultivars of roses. The left image is 'Samourai 08', and the right image is 'Blossom Pink' rose. This growth stage is classified in the *rose_large* category.

Fig. 2. The data acquisition system.

2. Related work

Flower Detection Palacios et al. [13] estimated the number of flowers per grapevine. They used SegNet for segmentation and predicted the number of flowers using a regression model. A normalized root mean squared error of 23.7% was achieved. These images were captured at night using an artificial illumination system. For daytime detection, Sun et al. [22] used a CNN to roughly locate a flower object and then used the difference in color between the flower and background to refine the network. This study focused on apple, peach, and pear blossoms, and the F1 score was 0.777 – 0.89. This method assumes that the background and flowers have different colors, which is inapplicable to bud detection, where the background is often the same green color. For bud detection, Li et al. [7] detected kiwifruit buds and flowers with an average precision of 96.66% and 98.57%, respectively. However, this experiment was intended for robotic pollination; thus, close-up images were used, which are unsuitable for monitoring the growth of an entire farm.

Object Tracking Xiong, Ge, and From [26] used YOLOv4 [3] and DeepSORT [25] for strawberry harvesting robots and improved the entire strawberry cultivation process. For object counting, Tan et al. [23] combined YOLOv4 and a tracking method to count seedlings. They achieved high accuracy without 57.3% of the test videos having counting errors. Their method focused on processing speed and did not intend to examine the extent to which double counts or occlusions were included in the predicted number of counted objects. Several studies have focused on evaluating crop counts in detail. Itakura et al. [5] used YOLOv2 [16] combined with a classical Kalman filter to track pears and apples, achieving F1 scores of 0.972 and 0.929, respectively. With a more advanced object tracking model, Parico and Ahamed [14] used YOLOv4 and DeepSORT [24] to detect pears. These two studies were conducted using images taken relatively close to the object, and counting fruit in an entire farm is difficult. In addition, the target object was a mature fruit, which is considered easier to detect than rose buds.

Flower Datasets Flower datasets are limited in number. Several datasets are available for classifying flower types. The Oxford University group released the Oxford-17 [11], and Oxford-102 [12]. Other well-known datasets include the Jena Flower 30 [19], and HFD100 [27] datasets. The HFD100 has more images — up to 10,738. However, these datasets are intended for classification problems and exclude buds. Therefore, our dataset, which includes buds and flowers obtained from actual farms, significantly contributes to the study of image recognition in horticulture.

3. Materials and methods

This study consisted of four steps. (1) Data acquisition on rose farms and annotation of the data according to growth stage to create the

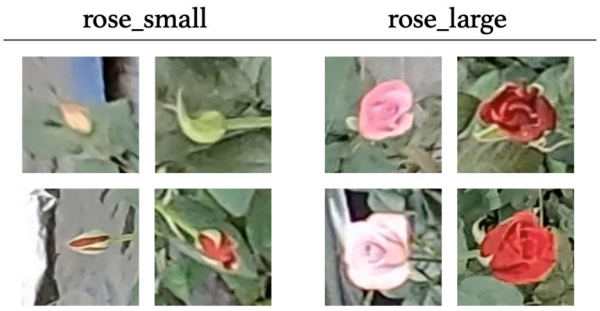

Fig. 3. The flower was designated as *rose_small* while it was a bud through to the stage where the petals are parallel with the central axis of the flower; the flower was designated as *rose_large* once the petals exceeded the parallel point.

Table 1
The data acquisition period and the number of images.

	Training	Validation	Test
Number of images	312	106	101
Period	4/1-5/2	5/3-5/12	5/13-5/18

RoseBloomig dataset. (2) Building a detection model using YOLOv5 [6] and assessing the results. (3) Using an object tracking model with the YOLOv5 weight to count all roses within a target area. In addition to SORT [1], a regression model was combined, and the RoseTracker model was created. (4) Conducting an ablation study for evaluation purposes.

3.1. RoseBlooming dataset

Data acquisition Data acquisition was conducted in a rose greenhouse at the Kizu Experimental Farm of Kyoto University in Kizugawa, Japan. The target cultivars were *Rosa hybrida* hort. 'Samourai 08' and 'Blossom Pink' roses (Fig. 1). The planting density in the greenhouse was eight plants per m^2, and the branches were trimmed by cut-up arching. An action camera (Gopro MAX, GoPro, Inc.) was attached to a self-propelled sprayer (Grinmate, Grintec Co., Ltd.) at the height of approximately 2.5 m from the growing bench and moved in parallel to take videos of two rows of roses in the greenhouse (Fig. 2). Further, video was filmed in the area of two rows of growing benches ($3\ m \times 20\ m$) once every 1–2 days in the late afternoon by manually turning on the power of the system. The resolution of the video was 1920×1440. The videos were taken from April 1, 2021, to May 18, 2021.

Dataset Construction The data were divided into three datasets — training, validation, and test data — with an approximate ratio of 6:2:2 (Table 1). Note that the flowering period is under 3 days; thus, the training and the test dataset differ in appearance. To compare the accuracy of rose flower tracking, we used the video taken on May 16, 2021, contained in the test data. For annotation, we used the VOTT [10] tool provided by Microsoft. The developmental stages of flowering branches were visually classified and annotated into two stages (Fig. 3). The flower was designated as *rose_small* while it was a bud through to the stage where the petals are parallel with the central axis of the flower; the flower was designated as *rose_large* once the petals exceeded the parallel point. An example image from the RoseBlooming dataset is shown in Fig. 4. As shown in Table 1, the RoseBlooming dataset contains 519 images (312 training, 106 validations, and 101 test). The video is taken over two months; therefore, it contains the images under various weather conditions. Fig. 5 shows the number of bounding boxes per image. Fig. 6 shows the total number of bounding boxes per category. As shown in Fig. 5, most of the images contain several bounding boxes, therefore, this dataset contains over 7,000 bounding boxes.

3.2. Proposed method

An overall view of RoseTracker is shown in Fig. 7. Using the weights of the object detection model created from the RoseBlooming dataset, the object tracking model SORT is used to detect roses from the video by growth stage and obtain TrackIDs. Then merge the TrackIDs using the regression model to obtain the final output.

Object detection Given the system is to be used for daily management of crop management use, fast speed object detections are needed. We adopted the YOLOv5 model, which is one of the fastest operating object detection models. The object detection model performs two tasks: positional regression, which determines the location of an object in an image, and class classification, which infers the object. A model that simultaneously performs these two tasks is called a one-shot detector, and YOLO is one of the most representative methods. YOLOv5 is a model of the YOLO system developed by Jocher et al. [6]. It was developed by the same company that developed the PyTorch [15] version of YOLOv3 [17]. YOLOv5 has four models: s, m, l, and x, depending on the detection accuracy and computational load, with s being the fastest and lightest model. We chose the YOLOv5-s model because of its fast operation, simple implementation, and less consumption of memory.

The main parameters were as follows: the epoch was set to 4000, batch size to 19, learning rate to 0.01, and training image size were 1280×960 pixels. For comparison with YOLOv5, comparative experiments were conducted using YOLOv4 [3]. For a fair comparison, the batch size, learning rate, and epoch of YOLOv4 were kept the same as those of YOLOv5. For data augmentation process, we follow the official implementation on both models [3,6]. The YOLOv4 and YOLOv5 evaluations included precision, recall, and average precision (AP).

Object tracking SORT [1] is one of the most famous object tracking models that use the Kalman filter to predict positions. In SORT, each object is approximated as the inter-frame displacement of each object using a linear constant velocity model. The state of each target is described as:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{1}$$

where u and v represent the horizontal and vertical pixel locations of the center of the target, respectively, and s and r represent the scale and aspect ratio of the bounding box, respectively. $\dot{u}$, $\dot{v}$, $\dot{s}$ indicate the amount of change in each value per time. The aspect ratio is considered constant. If no detection is associated with the target, its state is simply predicted using the linear velocity model without a correction method.

In our dataset, roses were considered to move in a constant-velocity linear motion from the bottom to the top of the screen. Because SORT expects the target to move in various manners, we changed some parameters from the default values as follows:

max_age: the number of frames in which an unmatched tracker exists. We increased this value to 100 because targets moved from the bottom of the screen to the top; thus, they did not disappear in the middle of the screen (default is 1).

iou_threshold: the minimum value of the associated intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from existing targets. This value was lowered to 0.1 as the SORT prediction was simple in this study owing to the nearly constant-velocity linear motion of the rose (default is 0.3).

Videos taken on the same dates as the validation data were used to adjust these parameters. Some models, such as DeepSORT, use the information on object features to reassign IDs when tracking is off. However, SORT, which does not use features, was employed in this dataset because of the dense concentration of similar-looking buds and flowers.

Merging track IDs In addition to SORT, a custom algorithm was applied to prevent incorrect ID switching. We took advantage of the fact that roses do not change the direction of their movement or framed-out

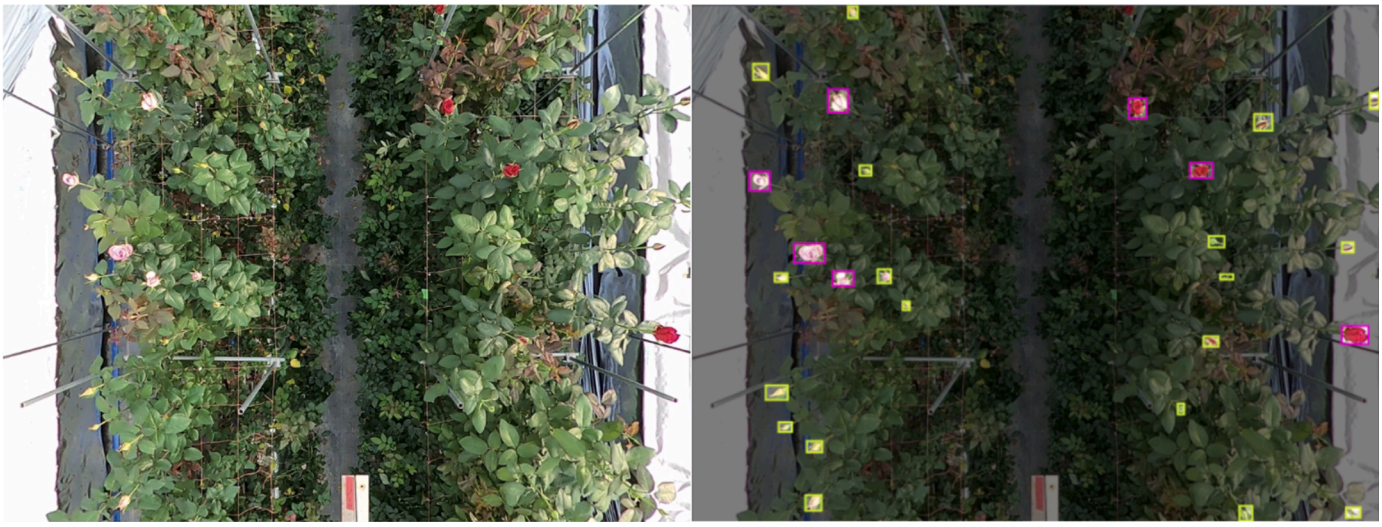

Fig. 4. (Left) An example image from RoseBlooming Dataset; (Right) The corresponding annotated image: Pink bounding boxes indicate *rose_large*; yellow bounding boxes indicate *rose_small*.

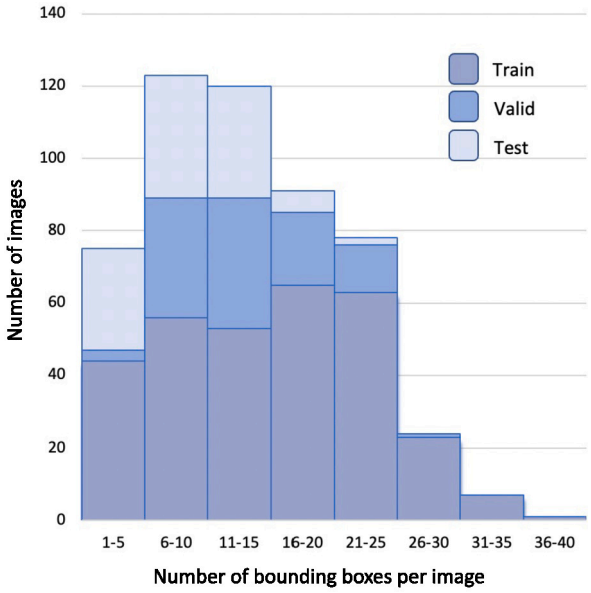

Fig. 5. The number of bounding boxes per image in RoseBlooming dataset.

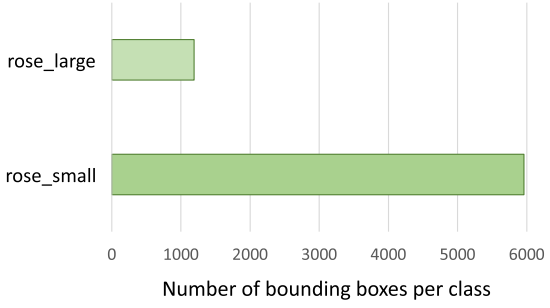

Fig. 6. The number of bounding boxes per category in RoseBlooming dataset.

along mid-frame in the video data. The y-axis represented the camera's direction of motion, and consequently, the y-coordinate increased as the rose moved forward. The x-axis represents the direction perpendicular to the y-axis. Assuming that the rose followed a constant velocity linear motion along the y-axis, the trackID should not disappear in the middle of the shooting range of the camera. Then, linear interpolation was used to interpolate and extrapolate the (x,y)-coordinates of broken tracking IDs from the (x,y)-coordinates and frame numbers of the detection results.

With these operations, we now have the (x,y)-coordinate with frame number, assuming unbroken tracking for each trackID. If two trackIDs had (x,y)-coordinates that were less than a threshold apart at any frame number, they were considered the same trackID. This threshold was determined by examining the validation data and set to 15 pixels. Compared with the execution time of SORT, the execution time of this process is short, and even when implemented as an application, the execution time should not have a significant impact. Fig. 8 illustrates this merging track IDs method as a flowchart.

3.3. Evaluation

Ablation study We conducted an ablation study to verify the effectiveness of the RoseTracker model. RoseTracker consists of three parts: object detection, object tracking, and improvement of the track ID with regression models. In the ablation study, i) YOLOv5 alone, ii) YOLOv5 combined with SORT, and iii) YOLOv5 combined with SORT and a regression model (RoseTracker) were compared. Because YOLOv5 supports object detection by image, we collected overhead images for each section of the greenhouse and counted the detected roses to ensure that there was no overlap. Therefore, precision and recall were based on counting and not the bounding-box areas. The combined YOLOv5 and SORT models and the RoseTracker were tested for accuracy on an overhead video taken of the greenhouse, and counting correctness was evaluated. Although tracking IDs were automatically assigned to each rose in the video, we checked the video to see if any IDs were switched or if the IDs were mistakenly assigned for detailed evaluation.

Tracking Evaluation In this study, four indicators were used to evaluate the counting accuracy. i) Correct detection (*Correct*): This is the number of correctly counted roses. YOLOv5 uses eight images to detect the target greenhouse area with no overlap, and the correct count for YOLOv5 alone is the sum of the number of correct detections for the eight images. For video, the count increases by one if a unique tracking number is assigned to each rose and never switched to another track number. ii) Double count (*Double*): This is determined when the tracking ID is switched in the video. Because this does not occur in YOLOv5 alone, that only uses images, it was set to 0. iii) *Omission*: This indicates the number of undetected roses. iv) Wrong detection (*Wrong*): The number of objects detected as roses that are not roses. In addition,

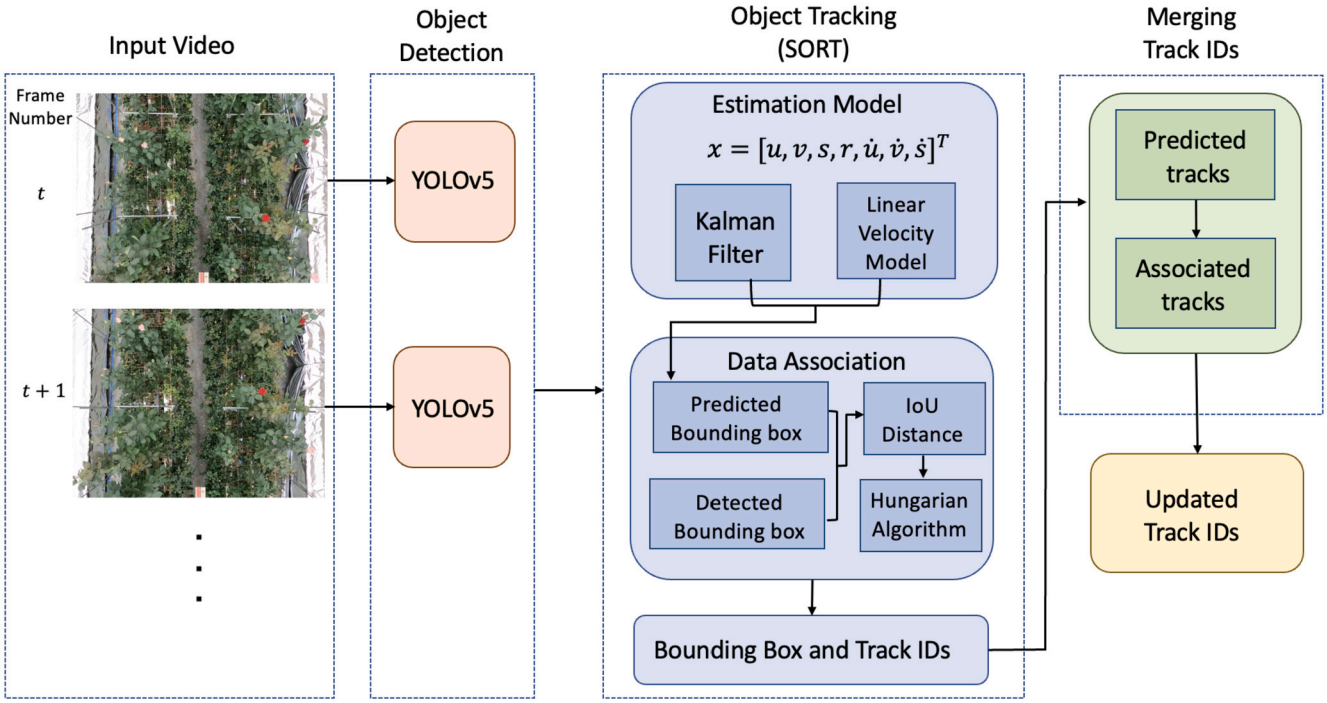

Fig. 7. An overview of the RoseTracker model.

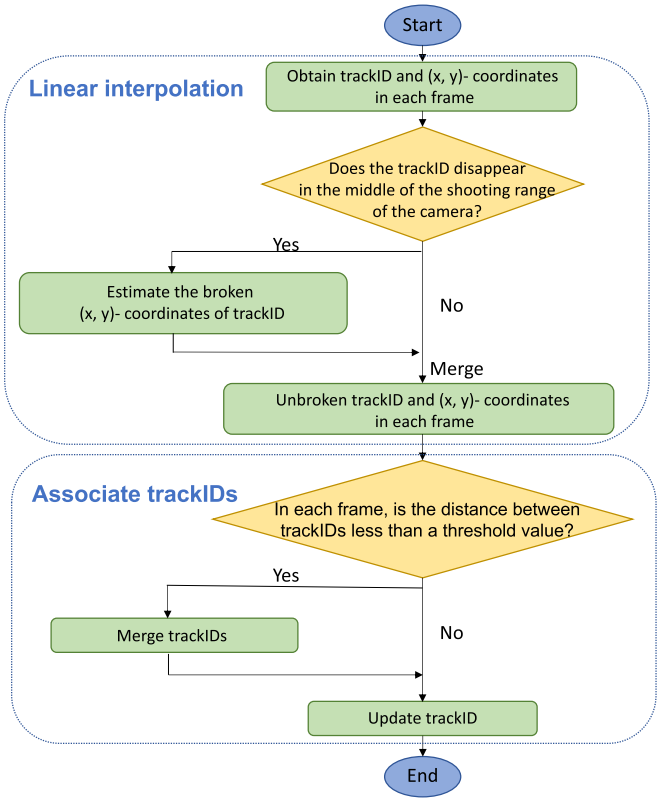

Fig. 8. Flowchart of the merging track IDs part. A threshold value is set to 15 pixels.

the ablation study calculated the precision, recall, F1, and AP (Average Precision) score using the following equations:

$$Precision = \frac{TP}{TP + FP} = \frac{Correct}{Correct + Wrong + Double} \quad (2)$$

Table 2
The results for object detection. Small and large represent *rose_small* and *rose_large*, respectively.

Model	Precision	Recall	AP	
			small	large
YOLOv5-s	**0.79**	**0.72**	**0.601**	**0.938**
YOLOv4	0.70	0.69	0.596	0.937

$$Recall = \frac{TP}{TP + FN} = \frac{Correct}{Correct + Double + Omission} \quad (3)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

$$AP = \int_0^1 p(r)dr \quad (5)$$

4. Results and discussion

4.1. Object detection

Table 2 shows the detection results of YOLOv4 and YOLOv5. Examples of detection results are shown in Fig. 9. The results for each class showed that all indicators for *rose_small* were lower than those for *rose_large*. In YOLOv5, the AP value was 0.601 for *rose_small*, and 0.938 for *rose_large*. The lower result for *rose_small* may be owed to the smaller rose sizes as the flowers had not yet blossomed, making their detection difficult. This also allows small roses to easily hide among leaves. Furthermore, *rose_small* includes green buds, increasing the difficulty in distinguishing their boundary from that of the leaves, whereas flowers corresponding to *rose_large* have a prominent petal color. YOLOv5 outperformed YOLOv4 in all categories. The storage size of the YOLOv5 model was 14.2 MB, which was much smaller than that of YOLOv4's 244.3 MB; thus, YOLOv5 was used for RoseTracker based on the model's accuracy and lightness. Fig. 10 presents the precision-recall curve of YOLOv5 on the validation dataset and the loss curve during training.

Table 3
The results from the ablation study for object counting.

Class	Model	Correct	Double Count	Omission	Wrong Detection	Precision	Recall	F1
all classes	YOLOv5	74	0	28	1	**0.987**	0.725	0.836
	YOLOv5 + SORT	84	12	6	2	0.857	0.824	0.840
	RoseTracker	95	1	6	2	0.969	**0.931**	**0.950**
rose_small	YOLOv5	40	0	26	0	**1.000**	0.606	0.755
	YOLOv5 + SORT	55	5	6	2	0.887	0.833	0.859
	RoseTracker	60	0	6	2	0.968	**0.909**	**0.938**
rose_large	YOLOv5	34	0	2	1	0.971	0.944	0.958
	YOLOv5 + SORT	29	7	0	0	0.806	0.806	0.806
	RoseTracker	35	1	0	0	**0.972**	**0.972**	**0.972**

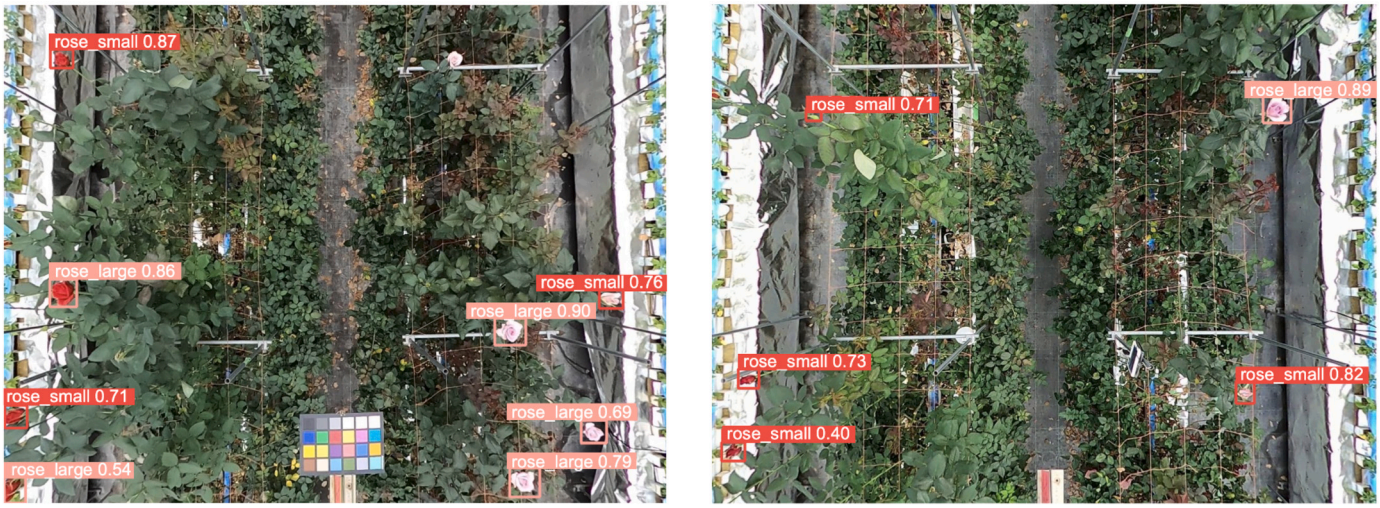

Fig. 9. Example images of YOLOv5 detections on test dataset. The class name and the probability value of belonging to the class are displayed.

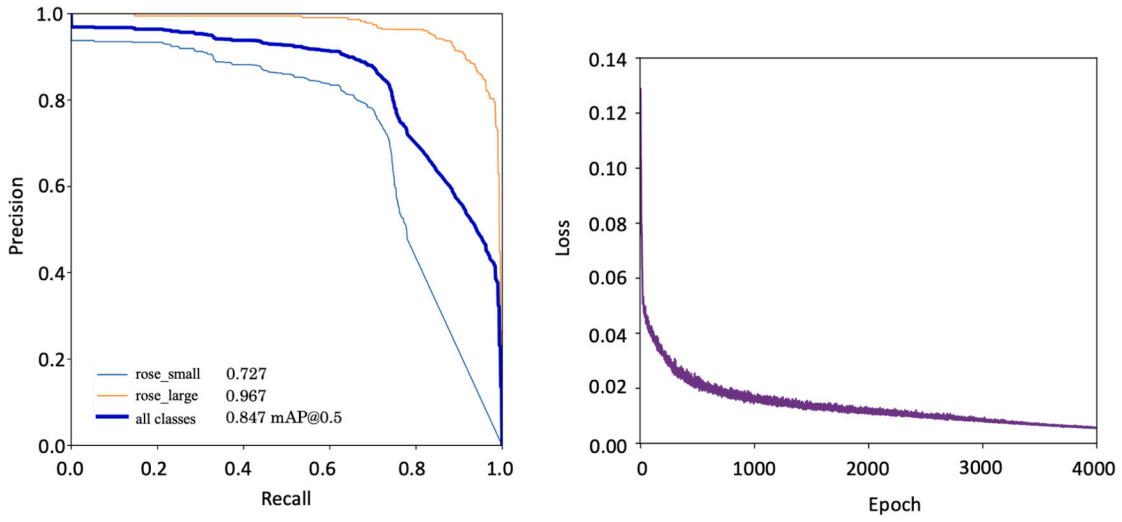

Fig. 10. (Left) The Precision-Recall curve of YOLOv5 on the validation dataset; (Right) Loss curve of YOLOv5 on the training dataset.

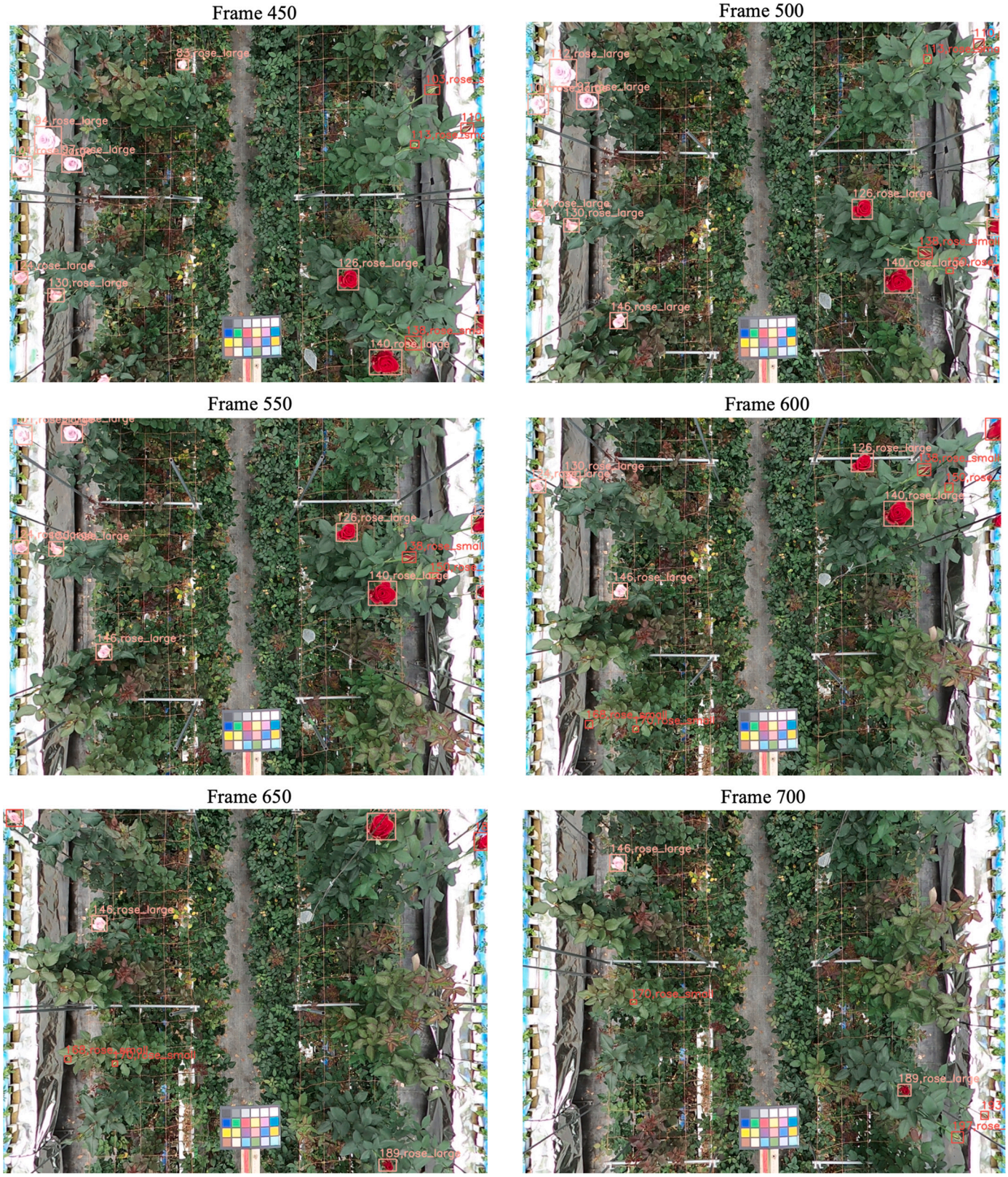

Fig. 11. Images of RoseTracker tracking results. Each bounding box has a tracking number and a class name. If the ID number switches during tracking, the class with the highest number of detections is selected.

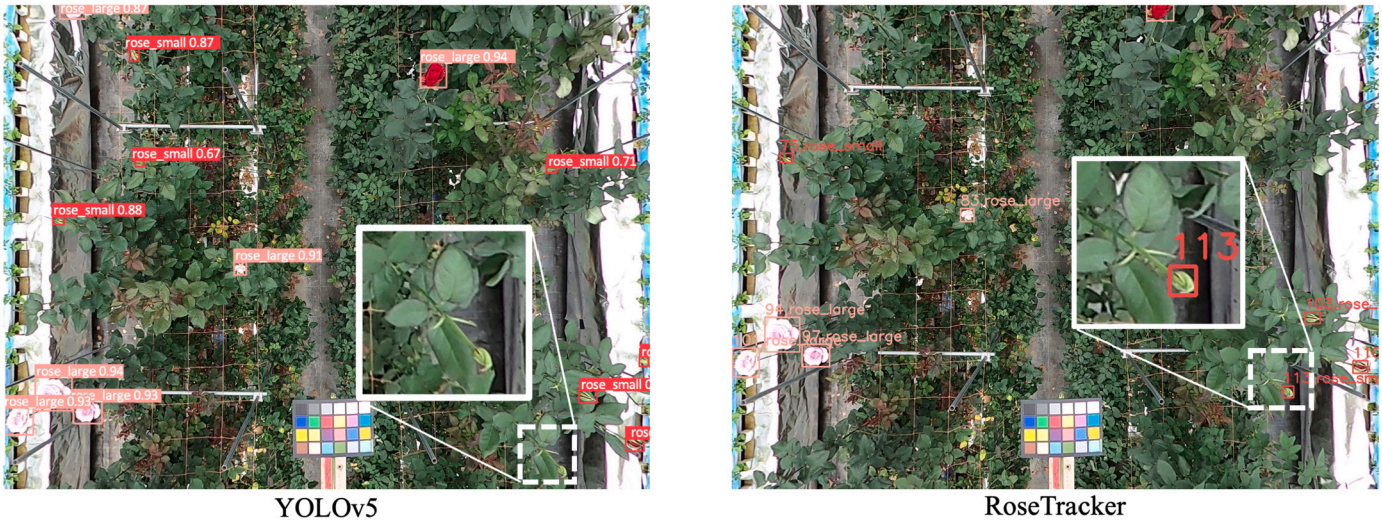

Fig. 12. Example of RoseTracker preventing omission. (Left) An example image of YOLOv5 detection; (Right) An example image of RoseTracker detection and tracking. RoseTracker detects and tracks from different angles in the video, lowering the Omission value.

Fig. 13. Example of RoseTracker preventing omission of the hidden rose flower by occlusion. (Left) Omission of rose flower due to the occlusion by another branch in the front; (Right) Detection of the same flower in another picture taken from a different angle. Red arrows indicate the rose flower, which is occluded in the left picture. Yellow dotted arrows indicate the rose flower in front, which was not also detected in the left picture due to the overlapping with the behind flower.

4.2. Rose counting system

The 'all classes' row in Table 3 shows the object tracking results of two class categories. From the F1 value, we can conclude that our model, RoseTracker, is the most effective with an F1 value of 0.950, which is higher than 0.836 achieved by YOLOv5. Using RoseTracker, we could reduce omissions from 28 to 6 compared to using YOLOv5 alone. The object tracking technique using video enables the detection of buds and flowers obscured by leaves by considering various angles. In addition, RoseTracker using the regression equation, reduced duplications compared with SORT alone. Examples of RoseTracker tracking results are shown in Fig. 11. For example, in Frame 450, track ID 150 (the lower right of the picture) in Frame 500 could not detect. However, in the Frame 500, 550, and 600 track ID 150 buds can be detected and assigned ID properly. This shows the effectiveness of the videos, which enable the detection of each flower and bud from various angles. Fig. 12 shows an example case where RoseTracker prevented omission. This indicates that the regression model using the tracking IDs obtained in SORT allowed for more detailed tracking using frame numbers in the video and coordinates, even without measuring the camera speed. In Fig. 13, RoseTracker also prevents the omission of the hidden rose flower by occlusion. In Frame 119, two rose are hidden by the leaves, but they can detect in Frame 70. The object tracking model performed an incorrect detection, but this can be attributed to the object detection being performed on many images spliced from the video. Despite the incorrect detection, our method increased the number of correct detections by 21 by reducing omissions compared with that of image detection.

Table 3 also shows results by class. As shown, our method is particularly effective for *rose_small*. The number of omissions in *rose_small* was 6 for our proposed method, which is significantly less than 26 of YOLOv5 alone. These counting system improvements indicate that the proposed method that combines SORT and a regression model has considerable power for providing a more accurate monitoring system of cut flowers such as rose buds on a farm.

In addition, object counting on video is more effective for actual farm applications than using YOLOv5 alone on images. The method using only YOLOv5 requires combining the images to obtain a picture

of the entire greenhouse, which involves considerable labor. An image must be captured directly above each section to ensure that the images do not overlap. From a filming perspective, object tracking using video can be easily applied in the cut-flower production industry.

5. Conclusion

In this paper, we provided the RoseBlooming dataset, which is an innovative cut-flower dataset annotated by growth stages. The RoseBlooming dataset will encourage the application of computer vision technology in the cut-flower industry, which has limited available datasets. We also proposed RoseTracker, which combines YOLOv5, SORT, and a regression model to obtain an accurate growth status. We significantly reduced the number of omissions, which indicates that the proposed object tracking technique using video enables the detection of buds and flowers obscured by leaves. Because the manner of taking video from above is easy, we believe that this method can be applied to other greenhouses and contributes to the construction of automatic growth monitoring and yield prediction systems. Such a growth monitoring system could be used to improve the efficiency of environmental control to adjust the harvest time and quality of cut flowers. The RoseBlooming dataset and the RoseTracker model bridge the gap between the horticultural field and image recognition.

CRediT authorship contribution statement

Risa Shinoda: Formal analysis, Funding acquisition, Investigation, Methodology, Writing – original draft. **Ko Motoki:** Conceptualization, Data curation, Methodology, Writing – review & editing. **Kensho Hara:** Methodology, Writing – review & editing. **Hirokatsu Kataoka:** Methodology, Writing – review & editing. **Ryohei Nakano:** Resources, Supervision. **Tetsuya Nakazaki:** Resources, Supervision. **Ryozo Noguchi:** Project administration, Supervision.

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.

Data availability

We publish the RoseBlooming dataset used in this study. The annotation data is released in COCO [8] format. The link of RoseBlooming dataset is https://github.com/dahlian00/RoseBlooming-Dataset.

Funding

This work was supported by JST SPRING, Grant Number JPMJSP2110.

References

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468.
[2] B.V. Biradar, S.P. Shrikhande, Flower detection and counting using morphological and segmentation technique, Int. J. Comput. Sci. Inf. Technol. 6 (3) (2015) 2498–2501.
[3] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv, 2020.
[4] Z. Cheng, F. Zhang, Flower end-to-end detection based on YOLOv4 using a mobile device, Wirel. Commun. Mob. Comput. 2020 (2020) 1–9.
[5] K. Itakura, Y. Narita, S. Noaki, F. Hosoi, Automatic pear and apple detection by videos using deep learning and a Kalman filter, OSA Contin. 4 (5) (2021) 1688–1695.
[6] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, V.A. Lorna, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, M.T. Minh, 2022. ultralytics/yolov5: v6.1.
[7] G. Li, R. Suo, G. Zhao, C. Gao, L. Fu, F. Shi, J. Dhupia, R. Li, Y. Cui, Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination, Comput. Electron. Agric. 193 (2022) 106641.
[8] T.-Y. Lin, M. Maire, S.J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: European Conference on Computer Vision (ECCV), 2014, pp. 740–755.
[9] H. Mann, A. Iosifidis, J. Jepsen, J. Welker, M. Loonen, T. Høye, Automatic flower detection and phenology monitoring using time-lapse cameras and deep learning, Remote Sens. Ecol. Conserv. (2022).
[10] Microsoft, VoTT (Visual Object Tagging Tool), version 2.2.0, Available online https://github.com/microsoft/VoTT, 2021.
[11] M.-E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2006, pp. 1447–1454.
[12] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing, 2008, pp. 722–729.
[13] F. Palacios, G. Bueno, J. Salido, M.P. Diago, I. Hernández, J. Tardaguila, Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions, Comput. Electron. Agric. 178 (2020) 105796.
[14] A.I.B. Parico, T. Ahamed, Real time pear fruit detection and counting using YOLOv4 models and deep SORT, Sensors 21 (14) (2021).
[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019.
[16] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, arXiv, 2016.
[17] J. Redmon, A. Farhadi, YOLOv3: an Incremental Improvement, arXiv, 2018.
[18] A. Rezazadeh, R.L. Harkess, T. Telmadarrehei, The effect of light intensity and temperature on flowering and morphology of potted red firespike, Horticulturae 4 (4) (2018).
[19] M. Seeland, M. Rzanny, N. Alaqraa, J. Wäldchen, P. Mäder, Plant species classification using flower images—a comparative study of local feature representations, PLoS ONE 12 (2) (2017) 1–29.
[20] P.K. Sethy, B. Routray, S.K. Behera, Detection and counting of marigold flower using image processing technique, Adv. Comput. Commun. Control (2019) 87–93.
[21] L. Shi, Z. Wang, W. Kim, Effect of drought stress on shoot growth and physiological response in the cut rose 'charming black' at different developmental stages, Hortic. Env. Biotechnol. 60 (2018).
[22] K. Sun, X. Wang, S. Liu, C. Liu, Apple, peach, and pear flower detection using semantic segmentation network and shape constraint level set, Comput. Electron. Agric. 185 (2021).
[23] C. Tan, C. Li, D. He, H. Song, Towards real-time tracking and counting of seedlings with a one-stage detector and optical flow, Comput. Electron. Agric. 193 (2022) 106683.
[24] N. Wojke, A. Bewley, Deep cosine metric learning for person re-identification, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 748–756.
[25] N. Wojke, A. Bewley, D. Paulus, Simple Online and Realtime Tracking with a Deep Association Metric, arXiv, 2017.
[26] Y. Xiong, Y. Ge, P.J. From, An improved obstacle separation method using deep learning for object detection and tracking in a hybrid visual control loop for fruit picking in clusters, Comput. Electron. Agric. 191 (2021) 106508.
[27] Y. Zheng, T. Zhang, Y. Fu, A Large-Scale Hyperspectral Dataset for Flower Classification, vol. 236, 2022, p. 107647.