Counting pear flower buds using worm's-eye view images

Takumi Hanakawa*, Masashi Nishiyama*‡, Yoshio Iwai*‡, Eiji Morimoto† and Shintaro Nakatani*,
*Graduate School of Sustainability Science, Tottori University, Japan
†Faculty of Agriculture, Tottori University, Japan
‡Cross-informatics Research Center, Tottori University

***Abstract*—We propose a method of counting pear flower buds from sequential images acquired from a worm's-eye view by a driven camera system. When only flower buds are detected at each time point, the problem of duplicate counting of the same flower buds may occur. Thus, in addition to the detection at each time point, our method identifies flower buds through keypoint matching for sequential image pairs and then counts the flower buds. Experimental results show that our method has less error in counting flower buds compared with using only a detector.**

***Index Terms*—Pear flower buds, counting, worm's-eye view, detection, keypoint matching**

I. Introduction

The social background of agriculture faces the problems of an increase in physical labor, a decrease in the number of agricultural workers, and an increase in the amount of work. Smart agriculture, in which advanced information technology is introduced into daily operations, is expected to overcome these problems. In smart agriculture, crop growth information is acquired from driving devices equipped with cameras and analyzed using artificial intelligence technologies. High-yielding and high-quality conditions are expected to be found by comparing the growth information currently acquired with that accumulated previously. Additionally, smart agriculture is expected to optimize crop production control, reduce physical labor, secure new farmers, and reduce labor.

In Tottori Prefecture, the introduction of smart agriculture is being promoted to farms targeting pears, a specialty of the prefecture. We here consider the acquisition of crop growth information on pear trees as an initial step in introducing smart agriculture. Specifically, information on the growth of pear trees includes information on flower buds, fruit, branches, and leaves. For example, the number of pears in summer [1], [2] has been counted for smart agriculture. In contrast, we consider how to count the number of pear flower buds in early spring for smart agriculture in the present study. Obtaining the number of pear flower buds in early spring and comparing it with the number of fruit shipped at the beginning of fall can help farmers manage production. If a large volume of this growth information is accumulated, there is the possibility to predict the yield of pears accurately in the future.

We here propose a method of counting flower buds from sequential images of pear trees acquired using a camera system driven on the ground. Our method finds the regions of pear flower buds from worm's-eye view images using detectors and then counts the number of pear flower buds through keypoint matching of the image pairs. In our experiments, we used a single-shot multibox detector (SSD) [3] and a you-only-look-once (YOLO) detector [4]. Furthermore, we used SuperPoint [5] and SuperGlue [6] for keypoint matching. The experiments confirmed that the use of our method reduces the error in counting flower buds compared with the simple method of using only detectors.

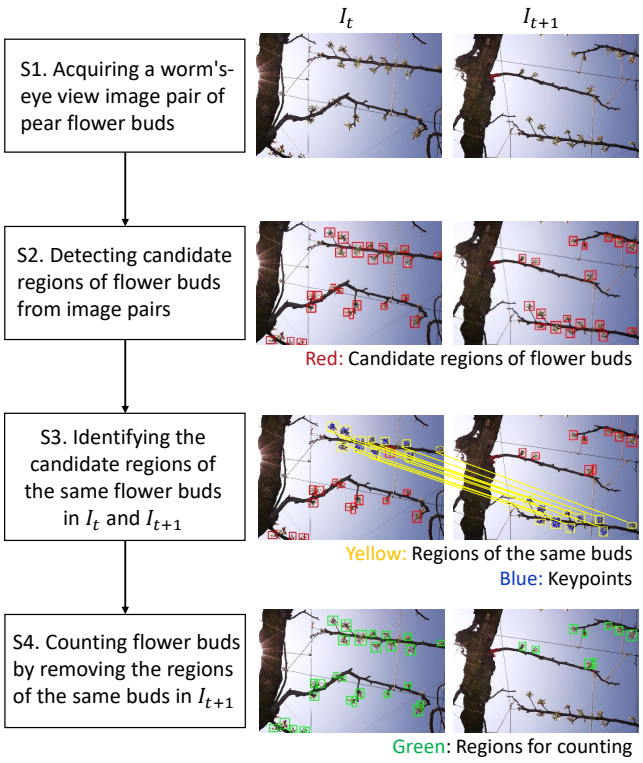

Fig. 1. Overview of our method.

II. Our method

A. Overview

We describe a method of counting flower buds adopting a detector and keypoint matching. Figure 1 shows an overview of our method. First, sequential image pairs are acquired in step S1. An image pair comprises an image I_t and image I_{t+1} acquired consecutively in a time series. The camera system

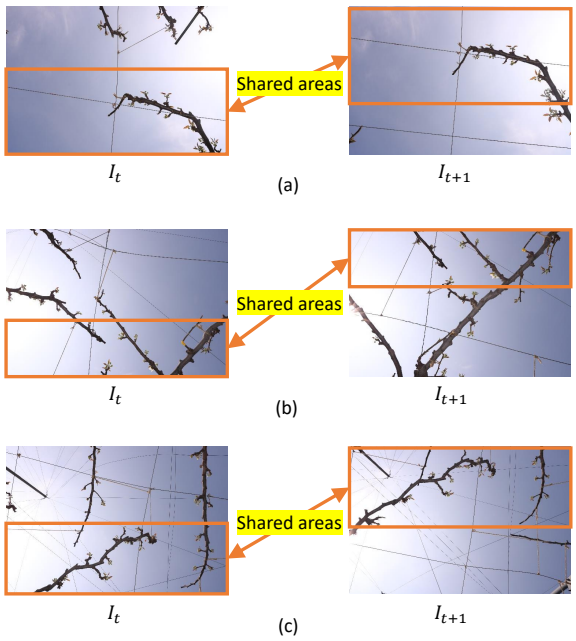

Fig. 2. Examples of the same flower buds between images. These buds appear in the shared areas of the image pairs.

is driven on the ground and acquires sequential images from the ground by looking up at the branches of a pear tree. In step S2, our method detects candidate regions of flower buds from I_t and I_{t+1} using a detector. However, it is noted that conducting only step S2 generates duplicate-counting problems. The ground of a pear farm is bumpy, and even if the time interval between image acquisitions is constant, the distance interval is not always constant. Thus, camera fields of view often overlap between time points. Examples of this overlapping are shown in Fig. 2. In the following, we refer to the overlapping parts of an image pair of I_t and I_{t+1} as the shared areas. If only the detection of flower buds for I_t and I_{t+1} is adopted, there can be no determination of whether the detected flower buds are the same, which results in the duplicate counting of flower buds at the two time points. Flower buds being the same means that they appear in the two shared areas of the image pair. We here consider solving the problem of duplicate counting in steps S3 and S4 of our method. In step S3, our method identifies the candidate regions of the same flower buds detected from the shared areas in I_t and I_{t+1}. Our method extracts keypoints within the candidate regions to identify the candidate regions. If the keypoints extracted from the candidate regions of I_t correspond to the keypoints extracted from the candidate region of I_{t+1}, our method determines that they are identical flower buds. In step S4, our method excludes the same flower buds from I_{t+1} through keypoint matching of the candidate regions. After this exclusion process, the remaining candidate regions in the image pair I_t and I_{t+1} are counted as the number of flower buds.

B. Detection of flower buds

Object detection is a method of finding specific object regions in images. Recent technological advances in object detection have led to the use of deep learning techniques, such as in [3], [4]. These techniques have the advantage of detecting objects more accurately than methods without deep learning techniques.

Even with deep-learning-based detection methods, there are issues that we must consider. If the detection target is small compared with the image size, the detection accuracy is low. As an example, we consider the case that an SSD [3] is used as a detector. The SSD300, one of the SSD implementations, internally reduces the size of input images I_t and I_{t+1} to 300×300 pixels. In this case, the bounding box of the candidate flower bud region in the resized input images in SSD300 is small. Specifically, the median size of the resized bounding box of a flower bud is 9×13 pixels. Thus, the resized bounding box is small, making it difficult to detect candidate regions of flower buds.

In this paper, to improve the accuracy of detecting flower bud candidate regions, we add a method of dividing the input images I_t and I_{t+1} into equal parcels and applying them to SSD. These images are divided equally into B parcels of $I_t^b(b=1,\ldots,B)$, and $I_{t+1}^b(b=1,\ldots,B)$. Inside SSD300, input images I_t^b and I_{t+1}^b are resized to 300×300 pixels. The bounding box of flower buds after the resizing of I_t^b and I_{t+1}^b is approximately B times as large as that of the original I_t and I_{t+1}. By dividing I_t and I_{t+1} equally, the bounding box of the candidate region after resizing is enlarged, which mitigates the issue of candidate regions not being detected.

C. Keypoint matching of candidate regions of flower buds

Keypoint matching is a method of extracting local descriptors representing features around keypoints such as corner points from an image pair and identifying the same local descriptors for the images. There are several methods for keypoint matching. In this paper, we use SuperPoint [5] for extracting local descriptors and SuperGlue [6] for identifying the local descriptors.

SuperPoint is a method of extracting the local descriptors of keypoints from images with a deep neural network. Within SuperPoint, the input image is resized to 1600×1067 pixels. The median size of the bounding box of the candidate flower bud region is then 48×48 pixels. The image size after resizing is sufficient to extract the local descriptors. SuperPoint uses I_t and I_{t+1} as the original input images. After extracting local descriptors from I_t and I_{t+1} adopting SuperPoint, the same local descriptors for these images are identified adopting SuperGlue. SuperGlue is a deep neural-based method that performs identification by finding the corresponding local descriptors in an image pair and rejecting the non-corresponding local descriptors.

We suppose that the identification results obtained adopting SuperGlue are used directly. In this case, candidate and non-candidate regions are occasionally misidentified. We make an

(a) Camera system (c) Path from a bird's-eye view

(b) Pear trees planted on a pear farm
(Red: Path of the camera system)

Fig. 3. We used a camera system driven along a path in the pear farm for image acquisition.

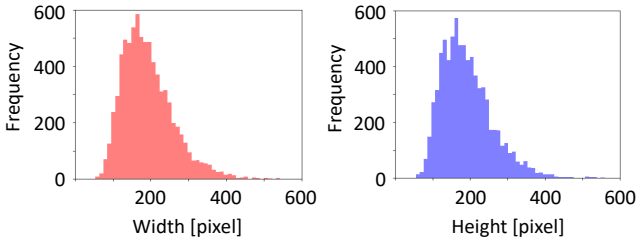

Fig. 4. Histograms of the width and height of the bounding boxes of the pear flower buds (manually labeled).

TABLE I
ACCURACY IN COUNTING PEAR FLOWER BUDS USING EACH OF THE METHODS C_1, C_2, O_1 AND O_2. WE PRESENT THE PREDICTED NUMBER OF PEAR FLOWER BUDS AND THE ERROR RELATIVE TO THE CORRECT VALUE.

Method	Prediction	Error
C_1 (SSD)	1006	231
C_2 (YOLO)	1082	307
O_1 (SSD+SuperPoint+SuperGlue)	**755**	**20**
O_2 (YOLO+SuperPoint+SuperGlue)	832	57

improvement that identifies only within the detected candidate regions to avoid this issue.

III. EXPERIMENTS

A. Dataset

We used sequential images acquired on a pear farm of Fukube-Cho, Tottori, Tottori Prefecture, Japan. Figure 3(a) shows our camera system. This camera system comprises a camera (Canon EOS8000D), camera stabilizer (DJI Ronin-SC), and dolly. We considered using a drone to acquire images, but strong spring winds in Fukube-Cho made it impossible to acquire the images stably, and we thus used a camera system that looked upward at the branches from the ground. Figure 3(b) shows the pear trees planted on the pear farm and the paths of the camera system. Pear trees were planted at equal spatial intervals. We acquired the sequential images of the pear flower buds along the path indicated by the red arrow. Figure 3(c) shows the path of the camera system from a bird's-eye view. We used a path that was passable for the camera system. The sequential images were acquired from 10:28 AM to 4:39 PM on April 3, 2020. Pear leaves grow quickly and the images were thus acquired on a single day so that the additional growth of leaves would not obscure flower buds. The number of worm's-eye view images was 550, and the image size was 6000×4000 pixels. A total of 7684 bounding boxes of flower buds were labeled manually. We present histograms in Fig. 4 to visualize the frequency of the height and width of the bounding boxes. The median width and height of the bounding box were 181 and 180 pixels, respectively.

B. Basic performance

We evaluated whether our method is effective in preventing the duplicate counting of flower buds. The methods of evaluation are described below.

- C_1 (SSD): Comparative method 1. We used only the SSD [3].
- C_2 (YOLO): Comparative method 2. We used only the YOLO detector [4].
- O_1 (SSD+SuperPoint+SuperGlue): Our method 1. We used the SSD [3] to find flower bud regions in step S2. We then used SuperPoint [5] and SuperGlue [6] to identify the same flower buds in step S3.
- O_2 (YOLO+SuperPoint+SuperGlue): Our method 2. We used the YOLO detector [4] to find flower bud regions in step S2. We then used SuperPoint [5] and SuperGlue [6] to identify the same flower buds in step S3.

Four-hundred images were used in detector training and 100 images in detector validation. The number of image divisions described in Sec. II-B was set at B=8. The number of epochs in training of the SSD was set at 50 and the reliability at 0.4. The number of epochs in training of the YOLO detector was set at 90 and the reliability at 0.3. We selected YOLOv5x[1] from several YOLO models. In SuperPoint, the maximum number of keypoints was set at 3072, and the threshold for keypoint extraction was set at 0.0001.

We describe an index for evaluating the accuracy of counting flower buds. In obtaining the correct number of flower buds, we counted the buds manually to ensure no overlaps of flower buds between sequential images. We predicted the number of flower buds using each method. The absolute difference between the correct number of flower buds and the predicted number of flower buds was taken as the error.

Table I gives the accuracy in counting pear flower buds using each of the methods C_1, C_2, O_1, and O_2. We evaluated the accuracy using 25 image pairs not used to train or validate

[1]https://github.com/ultralytics/yolov5

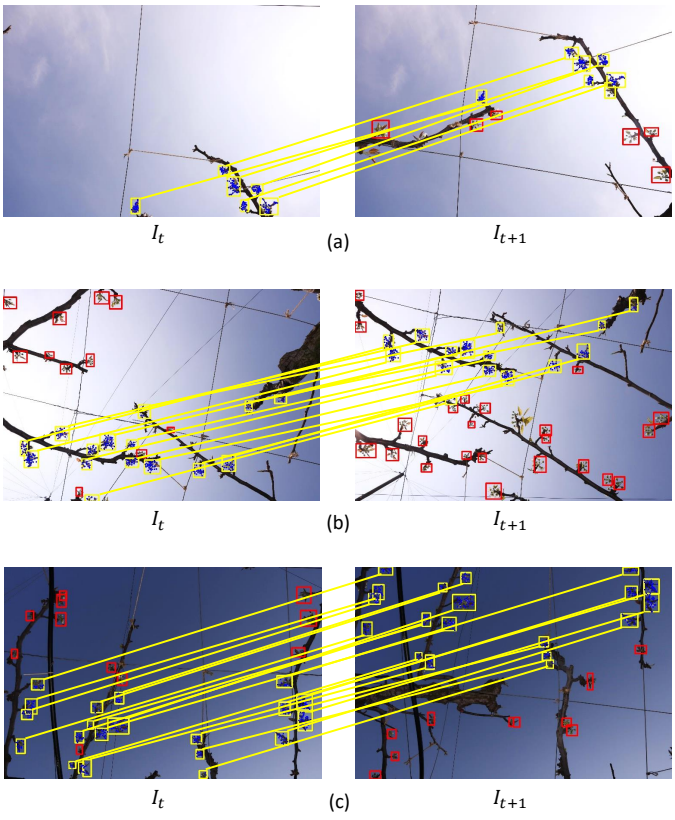

Fig. 5. Visualization of identifying the same flower buds in step S3 of our method O_1. (Red: candidate regions of pear flower buds, Yellow: regions of the same flower buds, Blue: keypoints).

the detector of candidate flower bud regions. The number of flower buds was counted manually for the 25 image pairs, resulting in a correct number of 775 flower buds. We compared the accuracy between methods with the same base detector in the following. The error for C_1 was 231, and the error for O_1 was 20. The error for C_2 was 307, and the error for O_2 was 57. The errors of O_1 and O_2 were less than those of C_1 and C_2. Furthermore, the error for O_1 was smaller than that for O_2. These results confirm that our method has less error than the simple method of using only detectors.

C. Visualization of identifying the same flower buds

We visualize the results of identifying the same flower buds in sequential images in step S3 of our method O_1 in Fig. 5. The accuracy of O_1 used in this figure is that of O_1 in Table I. In Fig. 5(a), I_t is the image acquired immediately after the branch entered the field of view. In I_{t+1}, in addition to branches and flower buds in I_t, other branches and flower buds come into view. We see that the candidate regions are successfully identified because the yellow boxes correspond between the shared areas in the images. In Fig. 5(b), I_t is an image of branches and flower buds acquired from directly below the pear tree. In the sequential images, many flower buds are in the field of view. We see that even when the number of flower buds in image pair (b) is larger than that in image pair (a), the candidate regions are still successfully identified. In Fig. 5(c), the sequential images were acquired in the evening, and the sky is thus darker than the sky in the images in (a) and (b). We see that candidate regions are successfully identified even in the evening when the sun was setting.

TABLE II
ACCURACY WITH AND WITHOUT DIVIDING THE IMAGE EQUALLY INTO PARCELS IN OUR METHOD O_1.

Image division	Prediction	Error
Without ($B = 1$)	133	642
With ($B = 8$)	**755**	**20**

D. Effect of dividing the image equally into parcels

We conducted an experiment to confirm whether dividing the image equally into parcels as described in Sec. II-B is effective for our method. Table II shows the accuracy of our method O_1 with image division ($B = 8$) and without image division ($B = 1$). The error of O_1 with image division is less than that of O_1 without image division. We confirm that detection using our method with image division is effective for sequential images of worm's-eye view acquired by the camera system.

IV. Conclusions

We proposed a method of detecting candidate regions of pear flower buds from sequential images of worm's-eye view using a detector. Candidate regions of the same flower buds are identified through keypoint matching, and the flower buds are counted. Experiments confirmed that our method prevents duplicate counting and reduces the error in counting pear flower buds. In the experiments, false negatives in detecting candidate flower bud regions were more frequent than false positives and keypoint matching errors. In future work, we intend to develop a method of reducing false negatives for flower buds with a small bounding box. This work was partially supported by Research Center for Sustainable Science, Tottori University.

References

[1] A. I. B. Parico and T. Ahamed, "Real time pear fruit detection and counting using yolov4 models and deep sort," *Sensors*, vol. 21, no. 14, 4803, 2021.

[2] K. Itakura, Y. Narita, S. Noaki, and F. Hosoi, "Automatic pear and apple detection by videos using deep learning and a kalman filter," *OSA Continuum*, vol. 4, no. 5, pp. 1688–1695, 2021.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–33712.

[6] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue:learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4937–4946.