Tomato Flower Detection and Counting in Greenhouses Using Faster Region-Based Convolutional Neural Network

Umme Fawzia Rahim and Hiroshi Mineno
Graduate School of Integrated Science and Technology, Shizuoka University, Japan
Email: fawzia@minelab.jp, mineno@inf.shizuoka.ac.jp

***Abstract*—To optimize fruit production and improve profitability cultivators remove excess flowers and fruitlets from plants and trees in the early growing season. The proportion of the flowers to be removed is determined by the flower intensity, i.e., the total number of flowers present in a row in the greenhouse. Several automated computer vision methods have been presented to estimate flower intensity, but their overall performance is still far from satisfactory. With the aim of designing a method for flower detection which is robust to occlusions and to changes in lighting conditions and camera position, this study presents a technique in which a pre-trained Faster Region-based Convolutional Neural Network (Faster R-CNN) is fine-tuned, followed by a color-based thresholding process to detect and count tomato flowers in greenhouses. Experimental results on a dataset composed of greenhouse tomato flower images acquired under different conditions, demonstrate significantly high performance, with precision and recall of 96.02% and 93.09%, respectively. The flower count from the proposed technique is comparable with the number counted manually with an error of – 4 to 3 flowers per image.**

***Index Terms*—agricultural engineering, computer vision, deep learning, faster R-CNN, flower detection and counting**

I. Introduction

Flower intensity has a major effect on fruit yield and quality of fruits [1], [2]. Along with other factors such as climate, flower intensity is especially critical to guide thinning, which is the process of removing excess flowers and fruitlets in the early growing season. Proper thinning increases fruit market value, since it affects fruit size, color, skin performance, firmness, soluble solids, sugar and acid content.

Although flower intensity estimation is significant for crop production, there has been relatively limited advancement so far in automating flower counting. Currently, this activity is typically performed manually. However, manual counting is tedious, labor-intensive, and prone to errors and uncertainties. Machine vision systems using different types of image sensors and image processing techniques can improve the efficiency of manual counting and minimize labor cost. Flowers generally have very distinct color and texture from the background. Several studies used traditional image processing methods such as color and shape analysis to segment flower pixels [3]-[5]. Flower intensity was calculated using morphological operations on the segmented flower pixels [5] or exploring the correlation of flower pixel percentage [3], [4]. However, those methods have their applicability hindered especially by change in illumination, background clutter and occlusion by leaves, stems or other flowers. In addition, most existing methods estimate flower numbers from flower pixel percentage instead of counting individual flowers. Such techniques require adjustment of parameter whenever changes in flower density (high/low) or in camera position (distance and angle) occur.

Inspired by successful studies using deep Convolutional Neural Networks (CNNs) in challenging computer vision and object detection tasks, we propose a robust method to detect and count tomato flowers in variant greenhouse conditions using a state-of-the-art object detector called Faster Region-based Convolutional Neural Network (Faster R-CNN) [6]. In our approach, a pre-trained Faster R-CNN is adopted through transfer learning and is further tuned to become particularly sensitive to tomato flowers. Finally, thresholding according to color and size features is applied to each identified flower region to eliminate misclassifications and very small faraway flowers that we do not seek yet.

II. Related Work

Many computer vision methods for automatic identification of flowers in image have been proposed. In a work aimed to estimate flowering in an apple orchard, the researchers used simple color thresholding in order to segment the white apple flowers from the background [7]. The images were acquired at night using artificial lighting so lighting conditions were invariant and good for the detection. However, when images are captured at day, lighting conditions become a challenge. In a study on estimating the intensity of lesquerella flower, the images were transformed to HSI color space to perform the segmentation [4]. The model estimated flower counts with root mean squared errors that ranged from 159 to 194 flowers. Although the researchers used Monte Carlo approach to minimize uncertainty in HSI parameters used

Manuscript received May 16, 2020; revised October 12, 2020.

doi: 10.18178/joig.8.4.107-113

for image segmentation, but it is still not robust to variance in natural illumination conditions. Flower intensity was estimated from flower pixel percentage; therefore, parameters must be readjusted whenever changes in flower density or in camera position occur. Similarly, in a research of detecting and counting of yellow tomato flowers in greenhouses, the researchers used the HSV color space for the segmentation of flower from background, followed by some simple morphological operations [8]. However, this approach has its applicability hindered especially by the presence of yellow objects (e.g., plastic, ribbon) in the greenhouses (see Fig. 1).

More advanced computer vision methods that utilized machine learning strategies have been employed for fruit identification. In [9], Hung *et al.* proposed a multi-class image segmentation for agrovision which classifies image pixels into leaves, almonds, trunk, ground and sky. Their approach combines sparse autoencoders for feature extraction, logistic regression for label associations and conditional random fields to model correlations between pixels. In study [10], the researchers utilized Support Vector Machine (SVM) classifier to recognize apples which use features extracted from different shape descriptors and color spaces as input. However, these techniques are still limited by background clutter and occlusions. Therefore, rather than relying on simple hand-engineered features, deep CNNs can be used to learn hierarchical features and to develop more robust methods for flower and fruit detection [11], [12]. Recently, in a research on detecting apple flowers in orchard, the flower features were extracted using a CNN [13]. The methodology is composed of generating region proposals using the simple linear iterative clustering (SLIC) superpixel algorithm, feature extraction from proposed regions using CNN and finally classification of each proposed region as flower or non-flower using SVM. Compared to their method Faster R-CNN [6] provide a uniform architecture in which both region proposal and classification modules can be fine-tuned for a specific task such as tomato flower detection. With the goal of developing a technique for tomato flower detection and counting which is robust to occlusions, changes in illuminations, variance in camera position and applicable for complex greenhouse environment that include background clutter such as presence of yellow plastics and ribbons, we therefore propose a novel approach in which a pre-trained Faster R-CNN is fine-tuned, followed by color thresholding to become especially delicate to tomato flowers.

Figure 1. Examples of labeled images. To avoid misclassification, images were labeled into two classes: Flower and yellow object.

III. Proposed Method

The overall method for tomato flower detection and counting can be divided into three major steps: (1) Flower region identification using Faster R-CNN (2) Extraction of identified region and thresholding and (3) Noise removal and flower number estimation (Fig. 2).

A. Flower Region Identification Using Faster R-CNN

The Faster R-CNN object detection system consists of two modules: 1) a Region Proposal Network (RPN) and 2) a classification module. The region proposal module is used for detection of Region of Interests (RoIs) where the object(s) of interest could reside within the image. The region classification module then classifies the individual regions whether it belongs to an object class of or not and regresses a bounding box around the object; the classifier could be a binary or multiclass classifier. During training, a 3-channel color image (BGR) of arbitrary size (within constraints of GPU memory), with annotated bounding boxes around each flower and yellow object is feed into the network. Depending on the choice of the CNN the image data is propagated through a number of convolutional layers. In this paper we used the convolutional layers of ResNet50 model [14]. The output from the convolutional layers is a high dimensional feature map, that is forward propagated into the RPN layer. The RPN is composed of two sibling fully connected layers, a bounding box-regression layer and a bounding box-classification layer. The RPN generates up to a predefined number of regions which may contain objects. Next, using the features extracted by the CNN and the bounding boxes with relevant objects, Region of Interest (RoI) Pooling is applied to extract those features which would correspond to the relevant objects. Finally, feature map for each proposal are propagated through subsequent fully connected layers (the R-CNN module), ending once again with two sibling fully connected layers with a better region classification result and associated finer object bounding box. Training is performed end-to-end using Stochastic Gradient Descent (SGD), allowing for the convolutional layers to be shared between the RPN and the R-CNN modules. At test time, the network returns Np = 300 bounding box detections per image (as in [6]) with probability score for each class. A threshold is applied to the output, followed by Non-Maximum Suppression (NMS) to remove overlapping detections. Fig. 3 shows a test time illustration of the Faster R-CNN network, with a sample tomato flower image from greenhouse. For a detailed overview of the network architecture and other implementation details the reader is referred to the original paper [6]. For further processing the only objects classified as flower were taken into consideration, although the network was trained on multiple class (flower and yellow object). Yellow objects were disregarded since the objective of the study is to count the flower number in the image.

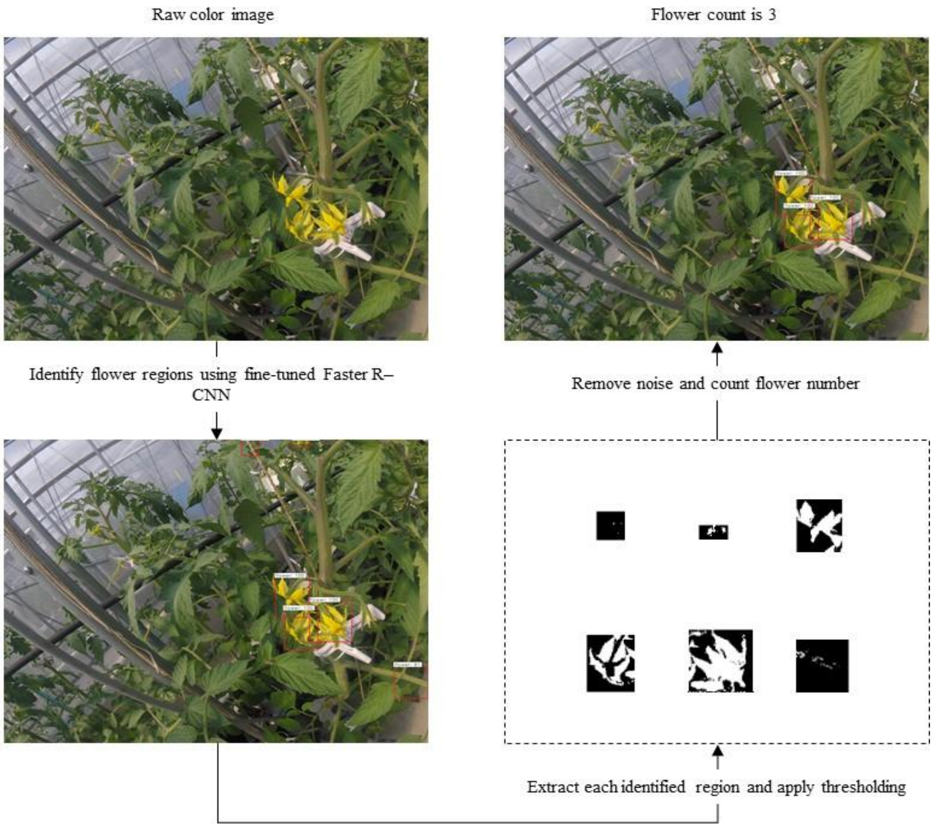

Figure 2. Flowchart of tomato flower detection and counting method, demonstrating the output of each step.

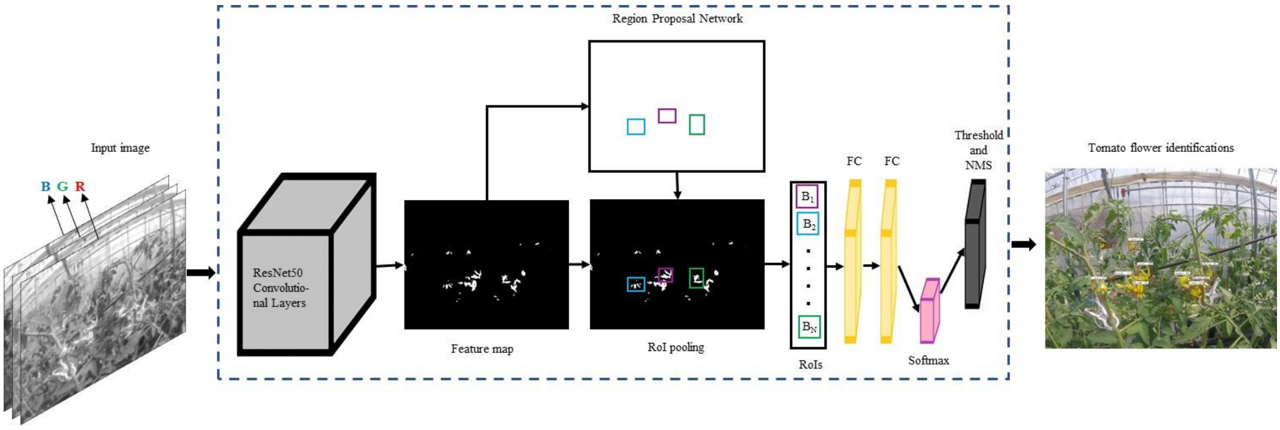

Figure 3. Illustration of test time of the Faster R-CNN. A 3-channel input image is propagated through a set of convolutional layers, from which Region of Interests (RoIs) are calculated. B_k denotes a bounding box of the K- th RoI. N represents the number of proposals and is set as 300. Each proposed box is propagated through two fully connected layers, which return their class probability and regresses to adjust more finer bounding box around individual objects. An IoU (Intersection over Union) threshold of 0.4 is applied to the output, followed by Non-Maximum Suppression (NMS) to remove duplicate results.

B. *Extraction of Identified Region and Thresholding*

In this step, all the image regions classified as flower by Faster R-CNN were extracted from input color image. The regions were first transferred into Hue, Saturation, and Value (HSV) color space and then thresholding was performed over H, S and V channels. Boundary conditions were based on six parameters, including the minimum and maximum hue, minimum and maximum saturation, and minimum and maximum value. The parameter values were derived from manual segmentation of 250 images of tomato flowers. After some trial and error procedures, the minimum and maximum hue were chosen 0.10 and 0.17 respectively, minimum and maximum saturation were chosen 0.43 and 1.0 respectively, and minimum and maximum value were chosen 0.43 and 1.0 respectively, since it comprises more than 95 % of the samples. The thresholding process produced a binary image with 1 indicating pixels that fell within the boundary criteria and 0 otherwise.

C. *Noise Removal and Flower Number Estimation*

Since non-flowers can be classified as flowers in several images, it is necessary to remove them before counting the flowers. False detections and noisy objects (very small faraway flower that we do not seek yet) were eliminated based on the fact that tomato flowers are yellow color and noisy objects are very small in size. Taking binary images produced by thresholding of all the object regions, the number of non-zero pixels for each region was counted and objects containing less than 300 pixels were removed. The detected flower objects left in the list after elimination were counted and stored as the total number of tomato flowers in the image.

IV. Faster R-CNN Fine-Tuning

We adapted Faster R-CNN through transfer learning and further tuned the model for tomato flower identification using labeled images from our training set.

A. *Dataset*

Images of tomato flowers were acquired using two smartphone cameras under natural daylight illumination in the greenhouse of Faculty of Agriculture of Shizuoka University, Shizuoka, Japan. The dataset is composed of a total of 1445 images, captured in RGB color space with resolution of 2720×2040 and 2324×1310 pixels from different angles, from random distances and at different heights. For performance evaluation and training purposes, the entire dataset was labeled using Python’s labelImg graphical image annotation tool and each image itself was categorized by size of flowers and presence or absence of occlusions in the image. A statistic of entire dataset is presented in Table I. Yellow objects used in greenhouses (see Fig. 1) are difficult to distinguish from yellow tomato flowers. In order to avoid misclassification and to let the network learn the pattern differences between yellow tomato flower and yellow object, images were labeled into two classes: flower and yellow object. Each visible flower and yellow plastic in the image were tagged by rectangle bounding boxes. Flowers on plant rows behind the main plant row in the image are more often very small and hazy. These flowers were disregarded and were not labeled (Fig. 1). Finally, the labelled dataset of tomato flowers was split into training, validation and testing sets composed of 1040, 145 and 260 images, respectively. The split was done such that each set contained images of different categories to minimize biased results.

TABLE I. Statistics of the Entire Dataset

Category	Description	Images
All flowers	All types of flowers	1445
Large flowers	Flowers of size 2000 pixels or more	548 (38%)
Large flowers with occlusion	Flowers of size 2000 pixels or more and partially hidden by leaves, stem or other flowers	305 (21%)
Small flowers	Flowers of size less than 2000 pixels	302 (21%)
Small flowers with occlusion	Flowers of size less than 2000 pixels and partially hidden by leaves, stem or other flowers	290 (20%)

B. *Transfer Learning*

Like other deep learning techniques potentiality of Faster R-CNN is often stymied by its heavy reliance on large amounts of high-quality data which are well-labeled. Transfer learning allows researchers to overreach the need for lots of new data. A CNN that has already been trained on a task for which plenty of labeled training data is available will be able to handle a new task with far less labelled examples. The ImageNet dataset, which contains 1000 object categories and 1.2 million images is often used as a base. Using pre-trained CNN features on ImageNet dataset, state-of-the-art results have been obtained on a variety of image processing tasks such as image classification and image captioning [15]. In this study, we used ResNet50 base network trained on ImageNet dataset.

C. *Data Augmentation*

Data augmentation is a common way to amplify variability in the training data by artificially expanding the dataset utilizing label-preserving changes. The process increases the networks ability to generalize and decreases overfitting. This study utilized horizontal flip, vertical flip and image rotation (90^0) procedures to expand shape variability in the dataset. Augmentation was done by expanding the data during each training epoch to avoid pre-computing the wide range of random augmentations.

D. *Parameters and Implementation Details*

During both training and test images were re-scaled to 717×600 pixels. For anchors, we used smaller scales with box areas of 64^2, 128^2 and 256^2 pixels since tomato flowers are small. Most image analysis tasks were performed using OpenCV 3.4.1. We used the neural-network library Keras (Tensorflow backend) with Cuda compilation tools (release 9.0, V9.0.176) for fine-tuning Faster R-CNN. Both training and testing were done on GPU (GeForce GTX 1080) having 8GB RAM and Intel core i5 processor with Python 3.6.8.

V. Flower Detection and Counting Result

The proposed method in this study was tested on a total of 260 images consists of 65 images from each category presented in Table I. The purpose of the analysis was to evaluate the performance of the model according to size of flowers and occlusions in the image. Precision and Recall of flower detection and histogram of flower counting error were used for analysis and visualization of the results. The detection results showed high precision (> 94%) for all four categories of images (Table II). However, the recall for the images with small flowers (without and with occlusions) was low.

TABLE II. DETECTION RESULT

Metric	Image Category				Average
	Large flowers	Large flowers with occlusion	Small flowers	Small flowers with occlusion	
Precision	94.97 %	95.33 %	95.96 %	97.85 %	96.02 %
Recall	98.95 %	94.23 %	89.08 %	90.11 %	93.09 %

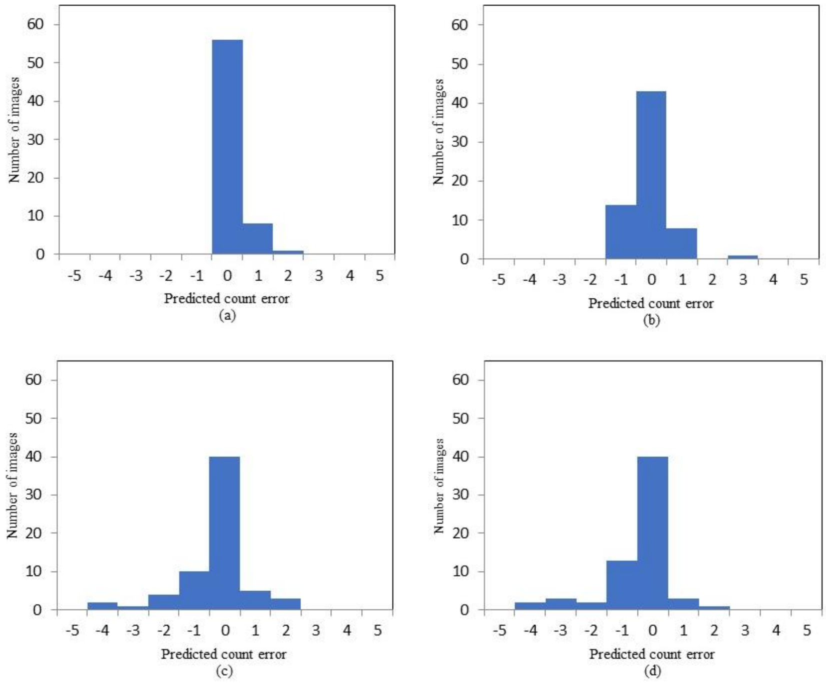

Figure 4. Histogram of predicted count error for large flowers (a), large flowers with occlusions (b), small flowers (c) and small flowers with occlusions (d).

Overestimation

Split flower
Manual count: 2
Predicted count: 4

Blur flower
Manual count: 4
Predicted count: 5

Bunch of faraway flowers
Manual count: 3
Predicted count: 4

Tomato with flower petals
Manual count: 3
Predicted count: 4

Yellow ribbon (False classification)
Manual count: 8
Predicted count: 9

Underestimation

Overlapped flowers
Manual count: 11
Predicted count: 9

Small flowers
Manual count: 12
Predicted count: 8

Subtle flower
Manual count: 5
Predicted count: 4

Figure 5. Example images of the detection and counting result.

The method caused overestimation or underestimation when comparing the number of predicted flowers and the number of flowers counted manually. The error of the predicted count was between 0 and 2 for images with large flowers, between -1 and 3 for images with large flowers with occlusions, and between -4 and 2 for images with small flowers and small flowers with occlusions (Fig. 4), showing that more images with small flowers were underestimated than images with large flowers. This is due to small flowers are more likely to be hidden and subtle which are difficult to distinguish from the background.

Overestimation was triggered mostly for images with large flowers since large flowers are more prone to split by leaves or stems and duplicate detections (Fig. 5). Another main reason of overestimation is the classification error of the Faster- RCNN. The misclassified flowers caused by the plants (e.g., leaves and stems) were eliminated by thresholding, but the misclassified flowers caused by the yellow objects (e.g., yellow ribbon) were not removed (Fig. 5). Some blur flowers, bunch of faraway flowers and green tomatoes with few petals at its stem also contribute to overestimation (Fig. 5). The underestimation was caused by occlusions in image, or small flowers that were shown in the images but were not identified correctly by the Faster-RCNN (Fig. 5).

VI. Conclusion and Future Work

In this study, we proposed a novel approach for tomato flower detection, which is based on Faster-RCNN that represent the state-of-the-art deep learning technique for object detection. Experimental results on the dataset composed of flower images acquired under different conditions, showed significantly high performances for all the cases under consideration. Analysis performed on four different categories of datasets demonstrated that the proposed model allows highly accurate flower identification even in scenarios of different flower sizes and occlusions, with precision and recall of 96.02% and 93.09%, respectively.

In the future, the study will be extended to develop a real-time system for automatic recording of flowering date to monitor the flowering in tomato plants and estimate production. We will also focus on classification between buds and flowers for accurate estimation of production.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Umme Fawzia Rahim conducted the research, implementation, experiments, analysis the results and wrote the initial draft of the manuscript; Hiroshi Mineno provided administrative, technical and supervisory support with the interpretation of the reported experiments and results; all authors had approved the final version of the manuscript.

Acknowledgment

This work was partially supported by JST, PRESTO Grant Number JPMJPR15O5, Japan.

References

[1] H. Link, "Significance of flower and fruit thinning on fruit quality," *Plant Growth Regul.*, vol. 31, pp. 17-26, 1992.

[2] C. Forshey, "Chemical fruit thinning of apples," *New York's Food and Life Sciences Bulletin*, 1986.

[3] F. J. Adamsen, T. A. Coffelt, J. M. Nelson, E. M. Barnes, and R. C. Rice, "Method for using images from a color digital camera to estimate flower number," *Crop Sci.*, vol. 40, no. 3, pp. 704-709, 2000.

[4] K. R. Thorp and D. A. Dierig, "Color image segmentation approach to monitor flowering in lesquerella," *Ind. Crops Prod.*, vol. 34, no. 1, pp. 1150-1159, 2011.

[5] B. V. Biradar and S. P. Shrikhande, "Flower detection and counting using morphological and segmentation technique," vol. 6, no. 3, pp. 2498-2501, 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. 28th International Conference on Neural Information Processing Systems*, 2015, pp. 91-99.

[7] M. Hočevar, B. Širok, T. Godeša, and M. Stopar, "Flowering estimation in apple orchards by image analysis," *Precis. Agric.*, vol. 15, no. 4, pp. 466-478, Aug. 2014.

[8] D. Oppenheim, Y. Edan, and G. Shani, "Detecting tomato flowers in greenhouses using computer vision," *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 11, no. 1, pp. 104-109, 2017.

[9] C. Hung, J. Nieto, Z. Taylor, J. Underwood, and S. Sukkarieh, "Orchard fruit segmentation using multi-spectral feature learning," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5314-5320.

[10] W. Ji, D. Zhao, F. Cheng, B. Xu, Y. Zhang, and J. Wang, “Automatic recognition vision system guided for apple harvesting robot,” *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1186-1195, 2012.
[11] R. Xu, C. Li, A. H. Paterson, Y. Jiang, S. Sun, and J. S. Robertson, “Aerial images and convolutional neural network for cotton bloom detection,” *Front. Plant Sc*i., vol. 8, pp. 1-17, 2018.
[12] S. Bargoti and J. Underwood, “Deep fruit detection in orchards,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3626-3633.
[13] P. A. Dias, A. Tabb, and H. Medeiros, “Apple flower detection using deep convolutional networks,” *Comput. Ind.*, vol. 99, pp. 17-28, 2018.
[14] G. A. Hembury, V. V. Borovkov, J. M. Lintuluoto, and Y. Inoue, “Deep residual learning for image recognition kaiming,” *Chem. Lett.*, vol. 32, no. 5, pp. 428-429, 2003.
[15] M. Huh, P. Agrawal, and A. A. Efros, “What makes ImageNet good for transfer learning?” arXiv Prepr. arXiv1608.08614, 2016.

Umme Fawiza Rahim received a B.Sc. in Computer Science and Engineering from Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 2012. Currently, she is pursuing her M.Sc. in Computer Science at Shizuoka University, Shizuoka Prefecture, Japan. Her current research involves Deep learning and Computer vision for uses in precision agricultural applications.

Hiroshi Mineno received his B.E. and M.E. degrees from Shizuoka University in 1997 and 1999, respectively. In 2006, he received his Ph.D. degree from the information science and electrical engineering of Kyushu University. Between 1999 and 2002, he was a researcher of the NTT Service Integration Laboratories. Currently, he is a Professor in the Department of Computer Science of Shizuoka University. His research interests include intelligent IoT system as well as heterogeneous network convergence. He is also a member of IEEE, ACM, IEICE and IPSJ.