ROBOMECH Journal

RESEARCH ARTICLE

Open Access

An automated fruit harvesting robot by using deep learning

Yuki Onishi[1*], Takeshi Yoshida[2], Hiroki Kurita[2], Takanori Fukao[3], Hiromu Arihara[4] and Ayako Iwai[4]

Abstract

Automation and labor saving in agriculture have been required recently. However, mechanization and robots for growing fruits have not been advanced. This study proposes a method of detecting fruits and automated harvesting using a robot arm. A highly fast and accurate method with a Single Shot MultiBox Detector is used herein to detect the position of fruit, and a stereo camera is used to detect the three-dimensional position. After calculating the angles of the joints at the detected position by inverse kinematics, the robot arm is moved to the target fruit's position. The robot then harvests the fruit by twisting the hand axis. The experimental results showed that more than 90% of the fruits were detected. Moreover, the robot could harvest a fruit in 16 s.

Keywords: Harvesting fruits, Robot, Manipulation, Deep learning

Background

The agriculture industry has many problems, including the decreasing number of farm workers and increasing cost of fruit harvesting. Saving labor and scale up in agriculture is necessary in solving these problems. In recent years, the automation of agriculture has been advancing for labor saving and large-scale agriculture. However, much of the work in the field of fruit harvesting is manually done. The development of an automated fruit harvesting robot is a viable solution to these problems. The automatic harvesting of fruits by a robot involves two big tasks: (1) fruit detection and localization on trees using computer vision with a sensor and (2) robot arm motion to the position of the detected fruit and fruit harvesting by the end effector without damaging target fruit and its tree.

The fruit detection and localization on trees using computer vision have been investigated in numerous studies, and most of these have been summarized in the review of Gongal et al. [1]. Color, spectral, or thermal cameras have been widely used in these methods. When using spectral camera [2], detecting the fruit shadowed by another fruit as an object is difficult. When a thermal camera is used [3], the fruit is detected based on the temperature difference between the fruit and the background. This method is affected by the fruit size and exposure to direct sunlight. Various different features are used in fruit detection using color camera. Bulanon et al. [4, 5] used luminance and red, green, and blue (RGB) color difference to segment an apple. Rakun et al. [6] used texture analysis to detect an apple. Linker et al. [7] integrated multiple features to improve the accuracy of fruit detection methods. Various image classification methods for fruit detection can also be performed using a color camera. Bulanon et al. [8] used K-mean clustering for apple detection. Linker et al. [7] and Cohen et al. [9] used KNN clustering for apple classification. In addition, Kurtulmus et al. [10] used an Artificial Neural Network for apple classification. Qiang et al. [11] used a Support Vector Machine classification method for apple detection. However, these methods are difficult to use in variable light conditions because the color information cannot be sufficiently acquired. For better accuracy, fruit detection should be performed using multiple features such as color, shape, texture, and reflection to overcome challenges like clustering and variable light conditions.

The present study proposes "fruit detection and localization" and "fruit harvesting by a robot manipulator with a hand which is able to harvest without damaging

*Correspondence: re0069hi@ed.ritsumei.ac.jp
[1] Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1, Noji-higashi, Kusatsu 525-8577, Shiga, Japan
Full list of author information is available at the end of the article

the fruit and its tree" to perform automatic fruit harvesting by a robot. We used a color camera and a Single Shot MultiBox Detector (SSD) [12] to detect the two-dimensional (2D) position of the fruit. The SSD is one of the general object detection methods that use Convolution Neural Network (CNN) [13]. The SSD can comprehensively judge from color and shape. A three-dimensional (3D) position must be obtained to send a command to the robot arm. A stereo camera is used to measure the 3D position of the fruit detected by the SSD. We used inverse kinematics to calculate the route of the robot arm. We moved the robot arm to the fruit position based on inverse kinematics. We used the harvesting robot hand as the end effector. The robot hand harvests a fruit by gripping and rotating it without damaging it and its tree.

Methods

We describe each step in our fruit detection and harvest method in this section.

Apple and tree

The fruit used in this research is the "Fuji" apple cultivated in the Miyagi Prefectural Agriculture and Horticulture Research Center. However, our method can also be applied to other apple varieties. A pear has a relatively similar shape to an apple; hence, this algorithm is also considered effective for pears. We used herein a joint V-shaped apple tree [14]. The V-shaped tree shape was suitable for mechanization and efficiency, and its fruits can be easily harvested. Figure 1 shows the tree used herein.

Detection and harvest algorithm

The harvest robot was equipped with a stereo camera and a robot arm. Figure 2 presents the detection and harvest algorithm. The algorithm involves three steps:

Fig. 1 Apple tree

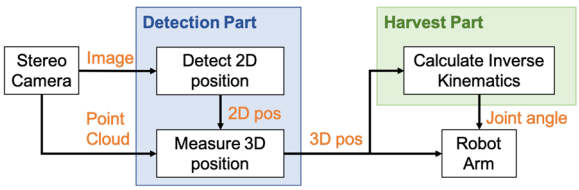

Fig. 2 Flow chart for harvest of apple

detecting the 2D position of the apple, detecting 3D position of the apple, and calculating the inverse kinematics. These steps were divided into the detection and harvest parts. We explain each method in the sections that follow.

Fruit position detection method

The first step of the detection part was detecting the 2D position of the fruit. We received one image from the stereo camera and detected where apples were in the received image. We used the SSD [12] to detect the apple positions.

The SSD is a method based on the CNN [13], which detects objects in an image using a single deep neural network. The other detection methods are Faster R-CNN [15], and You Only Look Once [16], among others. The first step of the SSD is the usage of the VGG net to extract the feature maps. The core of the SSD predicts the category scores and the box offsets for a fixed set of default bounding boxes using small convolutional filters applied to the feature maps. To achieve high detection accuracy, the SSD produces predictions of different scales from feature maps of different scales, and explicitly separates predictions by aspect ratio. These design features lead to simple end-to-end training and high accuracy even on low resolution input images, and improving the speed vs accuracy trade-off. We used the SSD herein because it is superior in speed and accuracy to others. The SSD was 59 FPS with mAP 74.3% on the VOC2007 test on a Nvidia Titan X. Faster R-CNN was 7 FPS with mAP 73.2%. YOLO was 45 FPS with mAP 63.4%. We can detect bounding boxes at the 2D apple positions in the image using the SSD.

For fruits detected by the SSD, we selected a fruit that was nearest the robot arm. We received a point cloud data from the stereo camera and the pixel at the selected 2D apple position. We used the stereo camera to do a 3D reconstruction. The 3D reconstruction by the stereo camera was performed by a triangulation from parallax between the right and left images to obtain the 3D position of the pixel in the image. We can then measure the distance from the stereo camera to the apple.

Table 1 Denavit–Hartenberg parameters for UR3

Link	a_i (m)	α_i (rad)	d_i (m)	θ_i
1	0	$\frac{\pi}{2}$	0.1519	θ_1
2	−0.24365	0	0	θ_2
3	−0.21325	0	0	θ_3
4	0	$\frac{\pi}{2}$	0.11235	θ_4
5	0	$-\frac{\pi}{2}$	0.08535	θ_5
6	0	0	0.08190	θ_6

Table 2 UR3 specifications

Weight capacity	3 (kg)
Reach	500 (mm)
Degree of freedom	6
Weight	11 (kg)
Repeatability	± 0.1 (mm)

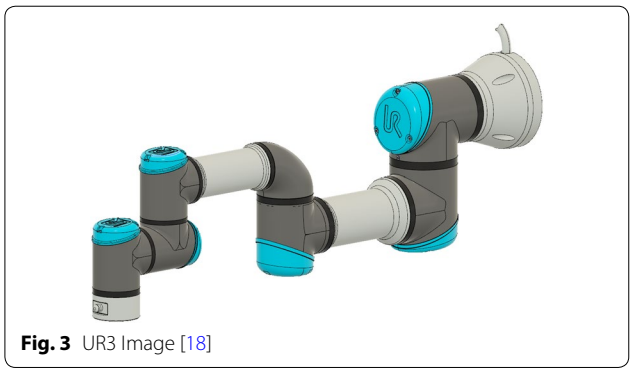

Fig. 3 UR3 Image [18]

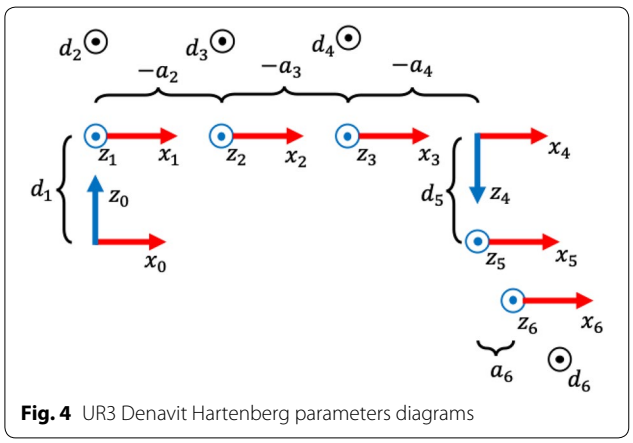

Fig. 4 UR3 Denavit Hartenberg parameters diagrams

Fruit harvesting method by the robot arm

Position $\boldsymbol{p}$ and posture $\boldsymbol{R}$ of the hand must be moved to as specified harvest the fruit using the robot hand attached to the robot arm. In the case of a vertically articulated robot arm, the position and posture of the hand $(\boldsymbol{p}, \boldsymbol{R})$ are determined by the angles $\boldsymbol{q}$ of each joint. Therefore, the relationship between the joint coordinate system representing the joint angle of the robot arm and the hand coordinate system representing the position and posture of the hand must be clarified.

The problem of determining the angles $\boldsymbol{q}$ of each joint from the hand position $\boldsymbol{p}$ and posture $\boldsymbol{R}$ is called an inverse kinematics problem [17]. The inverse kinematics problem aims to find a nonlinear function $\boldsymbol{f}^{-1}$ for the equation Eq. (1) is determined by the robot arm mechanism and configuration.

$$\boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{p}, \boldsymbol{R}). \tag{1}$$

Inverse kinematics model

We considered that the inverse kinematic problem of the robot arm had six links. We used UR3 made by UNIVERSAL ROBOTS as the robot arm. UR3 has six degrees of freedom; thus, arbitrary position and posture can be expressed as long as they are within the operating range. Table 1 shows the Denavit–Hartenberg parameter of UR3. Table 2 presents the UR3 specification. Figure 3 displays the UR3 used herein. The Denavit–Hartenberg parameters in UR3 are described in Fig. 4.

We obtain the angles $\boldsymbol{q} = \theta_i (i = 1, 2, \ldots, 6)$ of each joint when we are given the position $\boldsymbol{p}(p_x, p_y, p_z)$ and posture $\boldsymbol{R}(\phi, \theta, \psi)$ of the hand for Eq. (1). The rotation matrix $\boldsymbol{R}$ is expressed as

$$\boldsymbol{R}(\phi, \theta, \psi) = \begin{bmatrix} C_\phi C_\theta & C_\phi S_\theta S_\psi - S_\theta C\psi & C_\phi S_\theta C_\psi + S_\theta S_\psi \\ S_\phi C_\theta & S_\phi S_\theta S_\psi + C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\theta S_\psi \\ -S_\theta & C_\theta S_\psi & C_\theta C_\psi \end{bmatrix}, \tag{2}$$

where we used the abbreviations of $S_x = \sin x$, and $C_x = \cos x$.

The Denavit–Hartenberg notation [17] is the relationship between links i and $i+1$. The homogeneous transformation matrix of the Denavit–Hartenberg notation is

$${}^{n-1}\boldsymbol{T}_n = \begin{bmatrix} C_{\theta_n} & -S_{\theta_n} C_{\alpha_n} & S_{\theta_n} S_{\alpha_n} & r_n C_{\theta_n} \\ S_{\theta_n} & C_{\theta_n} C_{\alpha_n} & -C_{\theta_n} S_{\alpha_n} & r_n S_{\theta_n} \\ 0 & S_{\alpha_n} & C_{\alpha_n} & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where we used the abbreviation of $S_x = \sin x$, and $C_x = \cos x$.

We can obtain Eq. (4) from the relationship between the robot arm Denavit–Hartenberg notation ${}^0\boldsymbol{T}_6$ and the hand position $\boldsymbol{p}$ and posture $\boldsymbol{R}$

$$
{}^{0}\boldsymbol{T}_{6}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ 0\ 0\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & p_x \\ R_{21} & R_{22} & R_{23} & p_y \\ R_{31} & R_{32} & R_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}
$$

With Eq. (4), the angle θ_i of each joint of the robot arm can be obtained as follows, but first, θ_1 is presented as

$$
\begin{aligned}
A_1 &= \arctan\left(\frac{p_y - d_6 R_{23}}{p_x - d_6 R_{13}}\right), \\
B_1 &= \arccos\left(\frac{d_4}{\sqrt{(p_x - d_6 R_{13})^2 + (p_y - d_6 R_{23})^2}}\right), \\
\theta_1 &= A_1 \pm B_1 + \frac{\pi}{2}.
\end{aligned} \tag{5}
$$

θ_5 is denoted as follows

$$
\begin{aligned}
A_5 &= p_x \sin\theta_1 - p_y \cos\theta_1 - d_4, \\
\theta_5 &= \pm \arccos\left(\frac{A_5}{d_6}\right).
\end{aligned} \tag{6}
$$

where $\sin\theta_5 \neq 0$, θ_6 is

$$
\begin{aligned}
A_6 &= (R_{12} - R_{11})\sin\theta_1 + (R_{22} - R_{21})\cos\theta_1, \\
\theta_6 &= \frac{\pi}{4} - \arctan\left(\frac{\pm\sqrt{2\sin^2\theta_5 - A_6^2}}{A_6}\right).
\end{aligned} \tag{7}
$$

If $\theta_{234} = \theta_2 + \theta_3 + \theta_4$, θ_{234} is denoted as

$$
\begin{aligned}
A_{234} &= \cos\theta_5 \cos\theta_6, \\
B_{234} &= \sin\theta_6, \\
C_{234} &= R_{11}\cos\theta_1 + R_{21}\sin\theta_1, \\
D_{234} &= R_{31}, \\
\theta_{234} &= \arctan\left(\frac{A_{234}D_{234} - B_{234}C_{234}}{A_{234}C_{234} + B_{234}D_{234}}\right).
\end{aligned} \tag{8}
$$

θ_3 is

$$
\begin{aligned}
A_3 &= p_x \cos\theta_1 + p_y \sin\theta_1 + d_6 \cos\theta_{234}\sin\theta_5 - d_5 \sin\theta_{234}, \\
B_3 &= p_z - d_1 + d_6 \sin\theta_{234}\sin\theta_5 + d_5 \cos\theta_{234}, \\
\theta_3 &= \arccos\left(\frac{{A_3}^2 + {B_3}^2 - {a_2}^2 - {a_3}^2}{2a_2a_3}\right).
\end{aligned} \tag{9}
$$

θ_2 is

Fig. 5 Example of apple image

Table 3 SSD learning parameters

Architecture	Caffe
Net	VGG-16
Image (trainval)	200 images (1081 apples)
Image (test)	50 images (259 apples)
Base learning rate	0.0001
Batch size	4
Learning times	10,000 steps

$$
\begin{aligned}
A_2 &= a_3\cos\theta_3 + a_2, \\
B_2 &= a_3\sin\theta_3, \\
C_2 &= p_z - d_1 + d_6\sin\theta_{234}\sin\theta_5 + d_5\cos\theta_{234}, \\
\theta_2 &= \arctan\left(\frac{A_2}{B_2}\right) - \arctan\left(\pm\frac{\sqrt{A_2^2 + B_2^2 - C_2^2}}{C_2}\right).
\end{aligned} \tag{10}
$$

θ_4 is

$$
\theta_4 = \theta_{234} - \theta_2 - \theta_3. \tag{11}
$$

We can calculate the angles $\boldsymbol{q}$ of each joint from the hand position $\boldsymbol{p}$ and posture $\boldsymbol{R}$ by inverse kinematics.

Results and discussion

Fruit position detection

This describes the result of the fruit position detection.

The images taken at Miyagi Prefectural Agriculture and Horticulture Research Center were used for learning and testing. Shooting was performed to look at the fruit from below considering the minimized occlusion by the leaves, branches and other fruits. Figure 5 depicts the image taken by this method. We used the learning parameters shown in Table 3.

Fig. 6 Example of test image1

Fig. 7 Example of test image2

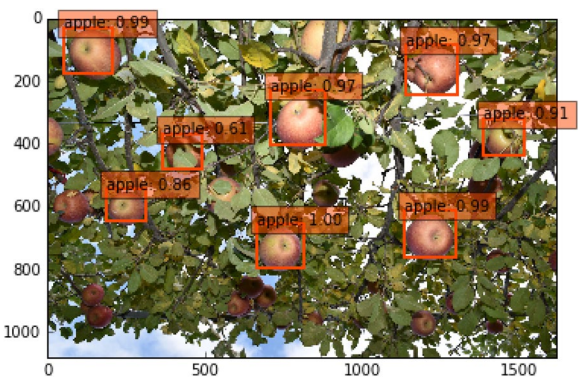

Fig. 8 Result of detection1

We tested whether fruits can be detected using unlearned images taken in the orchard using the learned model. We surrounded the area where the possibility of fruit was 60% or more with a red frame. We detected the presence of an apple to be tested from 30 images with 169 apples in total. Figures 6 and 7 depict the tested

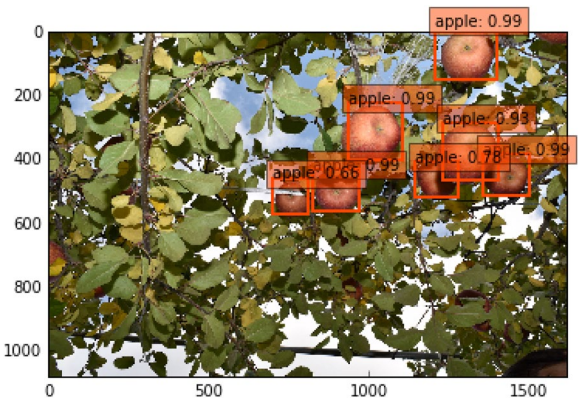

Fig. 9 Result of detection2

Table 4 Result of the apple position detection

Total	169
Detected apples	156
Undetected apples	13
Falsely detected apples	0
Precision	100%
Recall	92.31%

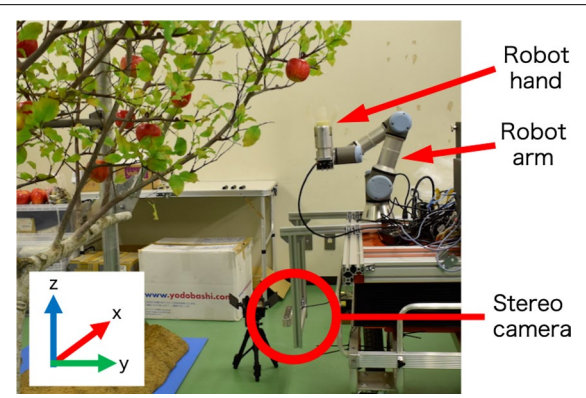

Fig. 10 Harvest robot

images. Figures 8 and 9 show the test image result. The model can detect even if the fruits are partially occluded by other fruits and leaves. However, the fruits at the edge of the image and those far from the camera could not be detected. The edge of the image could not be detected because the fruits were cut off in the image. The fruits far from the camera could not be detected because they had become smaller in the image. However, this was not a problem herein because these fruits were out of reach of the robot arm. Table 4 presents this test result.

Table 5 ZED specification

Output resolution	3840 × 1080
Frames per second	30
Depth range	0.5–20 (m)
Base line	120 (mm)

Fig. 11 Apple tree model

Harvesting robot

Figure 10 displays the harvesting robot used herein. We conducted fruit harvesting using this robot with a stereo camera installed at approximately 0.5 (m) below the base of the robot arm such that the fruit tree is looked up from directly below. If the distance to the target fruit is too long and the robot arm cannot reach the target, the table lift on which all equipment rides goes up and down, moving to the distance where the arm can reach.

We use UR3 (UNIVERSAL ROBOTS) as the robot arm. Table 2 shows the robot repeatability is ± 0.1 (mm). The robot palm diameter was 5 cm; hence, even if an error occurs, it can be suppressed by the robot hand. We used ZED (STEREO LABS) as the stereo camera, with specifications shown in Table 5.

Fruit automated harvest

We describe the automated apple harvesting in this section. Figure 11 illustrates the experimented tree and a model of the apple tree at the Miyagi Prefectural Agriculture and Horticultural Research Center. These trees were joint V-shaped trees [14] like those in the Miyagi Prefectural Agricultural and Horticultural Research Center. Conducting the experiment during apple harvest time was difficult; hence, we experimented with a tree model.

The results of the automated fruit harvesting experiments are presented herein along with the detection unit of the harvesting robot. First, we detected the 2D fruit position. Figure 12 shows the fruit detection result by the

Fig. 12 Detection of two-dimensional position

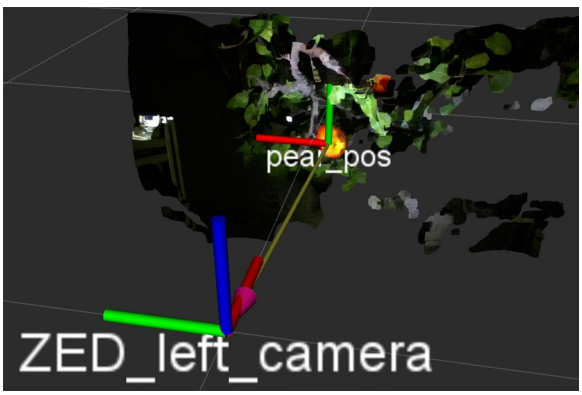

Fig. 13 Detection of three-dimensional position

Fig. 14 Approching target apple

SSD. We used a learning model that can detect more than 90% of the fruits used (fruit position detection section). We surrounded the area where the possibility of fruit was 60% or more, with a red frame. The robot was able to detect the apples the same as the real ones; hence, it seemed enough for the experiment.

Fig. 15 Harvesting target apple

Fig. 16 Grasping target apple

Second, we measured the 3D fruit position. Figure 13 depicts the 3D position of the center point of the frame detected by the SSD. The 3D reconstruction of the parts other than the apples themselves was inadequate, but in this experiment it is unnecessary except for the bottom surface of the apple. Sufficient results were obtained because we were able to capture the bottom of the apple.

Next, we will describe the harvesting part of the harvesting robot. To insert the robot hand from the underside for fruit harvesting, the robot was first moved 10 (cm) below the target fruit (Fig. 14). The arm then rose below the fruit (Fig. 15). The robot hand then grasped the fruit and harvesting it by twisting from the peduncle by rotating for four times (Fig. 16).

The harvest time for each fruit was approximately 16 s. Detecting the fruit position and calculating the joint angle at that position took approximately 2 s. Fruit harvesting took approximately 14 s. Harvesting consumed much time because the hand rotated for several times. By reconsidering these points, speedup is possible.

Conclusions

In this study, we performed automatic fruit harvesting through the method of fruit position detection and harvesting using a robot manipulator with a harvesting hand that does not damage the fruit and its tree. Using the SSD, we showed that the fruit position of 90% or more can be detected in 2 s. The proposed fruit harvesting algorithm also showed that one fruit can be harvested in approximately 16 s.

The fruit harvesting algorithm proposed herein is expected to be applicable even if it is a near species of apple. Moreover, if one learns again with the target fruit, harvesting fruits, such as pears is highly possible.

Abbreviations

SSD: Single Shot MultiBox Detector; CNN: Convolution Neural Network.

Acknowledgements

This research was supported by grants from the Project of the Bio-oriented Technology Research Advancement Institution, NARO (the research project for the future agricultural production utilizing artificial intelligence).

Authors' contributions

YO conducted all research and experiments. TY and TF conducted a research concept, participated in design adjustment, and drafted a paper draft assistant. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1, Noji-higashi, Kusatsu 525-8577, Shiga, Japan. [2] Research Organization of Science and Technology, Ritsumeikan University, 1-1-1, Noji-higashi, Kusatsu 525-8577, Shiga, Japan. [3] Department of Electrical and Electronic Engineering, Ritsumeikan University, 1-1-1, Noji-higashi, Kusatsu 525-8577, Shiga, Japan. [4] DENSO Corporation, 1-1, Showa-cho, Kariya 448-8661, Aichi, Japan.

Received: 5 January 2019 Accepted: 10 October 2019
Published online: 01 November 2019

References

1. Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K (2015) Sensors and systems for fruit detection and localization: a review. Comput Electron Agric 116:8–19
2. Okamoto H, Lee WS (2009) Green citrus detection using hyperspectral imaging. Comput Electron Agric 66(2):201–208
3. Stajnko D, Lakota M, Hočevar M (2004) Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. Comput Electron Agric 42(1):31–42
4. Bulanon DM, Kataoka T, Ota Y, Hiroma T (2002) Ae-automation and emerging technologies: a segmentation algorithm for the automatic recognition of fuji apples at harvest. Biosyst Eng 83(4):405–412
5. Bulanon DM, Kataoka T (2010) Fruit detection system and an end effector for robotic harvesting of fuji apples. Agric Eng Int CIGR J 12(1):203–210
6. Rakun J, Stajnko D, Zazula D (2011) Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. Comput Electron Agric 76(1):80–88
7. Linker R, Cohen O, Naor A (2012) Determination of the number of green apples in rgb images recorded in orchards. Comput Electron Agric 81:45–57

8. Bulanon DM, Kataoka T, Okamoto H, Hata S-i (2004) Development of a real-time machine vision system for the apple harvesting robot. In: SICE 2004 annual conference. vol 1, IEEE, New York, pp 595–598
9. Cohen O, Linker R, Naor A (2010) Estimation of the number of apples in color images recorded in orchards. In: International conference on computer and computing technologies in agriculture. Springer, Berlin, pp 630–642
10. Kurtulmus F, Lee WS, Vardar A (2014) Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. Precis Agric 15(1):57–79
11. Qiang L, Jianrong C, Bin L, Lie D, Yajing Z (2014) Identification of fruit and branch in natural scenes for citrus harvesting robot using machine vision and support vector machine. Int J Agric Biol Eng 7(2):115–121
12. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, Berlin, pp 21–37
13. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in neural information processing systems 25. Curran Associates Inc., pp 1097–1105
14. Shinnosuke K (2017) Integration of the tree form and machinery in Japanese. Farming Mech 3189:5–9
15. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems 28. Curran Associates Inc., pp 91–99
16. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 779–788
17. Slotine J-JE, Asada H (1992) Robot analysis and control, 1st edn. Wiley, New York
18. Universal Robot Support. https://www.universal-robots.com/download/. Accessed 23 Oct 2019

Publisher's Note