Image Processing Application on Automatic Fruit Detection for Agriculture Industry

Tresna Dewi[1,*] Rusdianasari Rusdianasari [2] RD Kusumanto[3] Siproni Siproni[4]

[1] *Electrical Engineering Department, Politeknik Negeri Sriwijaya*
[2] *Renewable Energy Department, Politeknik Negeri Sriwijaya*
[3] *Electrical Engineering Department, Politeknik Negeri Sriwijaya*
[4] *Mechanical Engineering Department, Politeknik Negeri Sriwijaya*
[*] *Corresponding author. Email: tresna_dewi@polsri.ac.id*

ABSTRACT

The robot brings automation to every sector of human life, including agriculture. Automation in agriculture might be the solution to get a higher quality harvest and less dependency on human farming. The most suitable type of robot for harvesting is an arm robot manipulator. The harvesting robot needs "eye" to "see" the crop/fruit to be harvested. The detection is made possible by using image processing to get the fruit position. The fruit position is the input for a visual servoing robot. The image processing needs to be simple and effective to ensure less computational time to facilitate the limited memory of the available microcontroller. This paper proposes three image processing methods, i.e., image segmentation, edge detection, and blob analysis. The processes were conducted in SCILAB, and three fruit were used as the model, i.e., oranges, grapes, and tomato cherry. The results showed that all the fruit are detected and isolated by the vegetation background.

Keywords: *Blob analysis, digital farming, edge detection, image segmentation, visual servoing.*

1. INTRODUCTION

The robot brings automation to every sector of human life, including agriculture. Automation in agriculture promises a solution for fresh fruit harvesting because a robot can monitor the farm 24/7 without stopping. The seasonal fruit requires the perfect handling at harvest time to ensure its freshness from the picking time until being marketed to the customers. This robot invasion to agriculture is called digital farming, and by applying digital farming, it is expected that the harvested fruit has higher quality and less dependency on human labor [1]-[10]. Less dependency is necessary due to the aging farming and fewer young people participating in the farm.

Arm robot is the most suitable type of robot for harvesting because its end-effector can be customized as needed [7]-[14]. Although the mobile robot also plays some parts in harvesting in some applications, arm robot manipulator is still the best option. The eye is a critical requirement for picking up the crops, and visual servoing is the right method to create a capable eye for the job. Visual servoing is the method of controlling the robot by taking the input from robot vision [8][11]-[12]. Robot vision relies heavily on image processing [15]-[30].

The fruit position is considered the target, and this is detected using image processing [24]-[29]. The problem with image processing is that it requires more memory for graphical computational time. Although current technology has allowed the assembly of cheap robot due to the significant decrement of component prices, the currently available microcontroller is still in limited memory capacity.

As the main controller of the robot, a microcontroller has to share its memory between controlling other parts of the robot such as actuators and sensors and processing the image captured by a camera. Due to this limitation, it is necessary to apply a simple yet effective method that does not require much computational time.

The image processing in harvesting robot has to be conducted online, and this online process has to deal with many noises such as camera motion if eye in hand camera is applied, occlusions if the time-of-flight camera is applied, and vegetation around the fruit that might have the same colors of creating an occlusion. Therefore, an effective method should be able to overcome these problems.

Image segmentation [5][14], blob analysis [10][19][20], and edge detection [16]-[18] are among the standard method used for image processing. This paper proposes applying these methods for fruit detection, searches the fastest processing of those methods to detect and isolate the object, and overcomes the problems of occlusion and vegetation. Three fruit images are processed to show the effectiveness of the proposed methods, and all of them are still located on their trunks. The fruit is oranges, grapes, and tomato cherry. All the image processing is conducted using SCILAB, open-source software that is as effective as the high-end software [31]. These methods have been tested using many objects during robot vision class in the Electrical Engineering Department of Politeknik Negeri Sriwijaya

2. ARM ROBOT MANIPULATOR DESIGN

The arm robot manipulator design for fruit harvesting is shown in fig. 2. This 4-DOF robot is suitable for picking up fruit. The camera can be installed at the end-effector known as an eye in hand and installed elsewhere as a time-of-flight camera. The robot's end-effector can be equipped with scissors or knife to cut the branch and claw or finger-like to grab the fruit and move it to the basket.

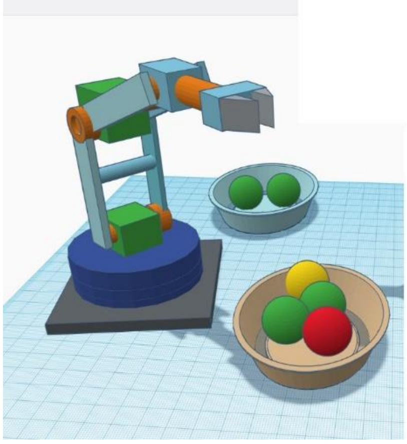
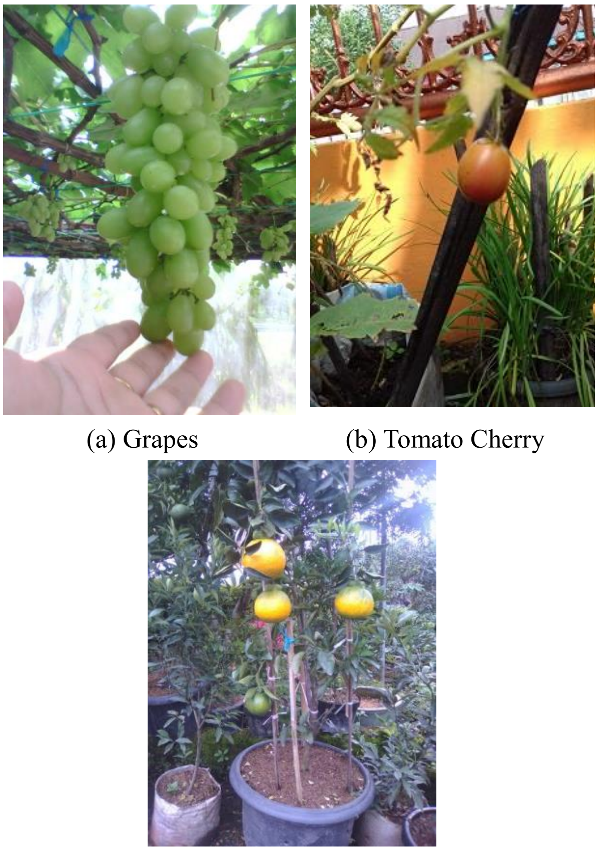


Figure 1 The designed arm robot manipulator

3. IMAGE PROCESSING METHOD

The image processing methods applied in this study are image segmentation, edge detection, and blob analysis. The original images are captured with a camera and processed using SCILAB for simulating fruit detection and isolating the fruit from noises such as vegetation and other occlusions.

The images to be processed are grapes, a tomato, and oranges, as shown in figure 2. The processes of image processing presented in this paper are given in the block diagram shown in fig. 3.



(a) Grapes (b) Tomato Cherry

(c) Oranges

Figure 2 Raw images

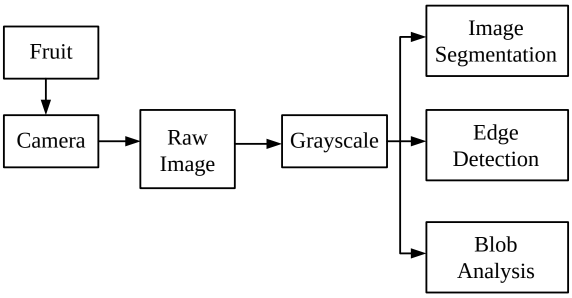


Figure 3 Steps of Image processing

3.1 Image segmentation

Image segmentation partitions an image captured by the camera into multiple segments or sets of pixels in order to define the representation of an image into a set of boundaries such as curves and lines in images. All the assigned pixels are labeled to find the pixels with the same characteristics, such as color, intensity, and texture. The image segmentation can also merely be divided between the foreground region of interest and the background.

The first step is by clustering in pattern recognition by partitioning a set of pattern vectors into subsets or clusters. The method used in this study is the thresholding method, where the raw image is converted into a grayscale image and then into a binary image within balanced histogram thresholding. The thresholding method replaces each pixel that has image intensity (Ii,j) lower than the thresholding value (Ii,j < T)and white if the pixel has the image intensity more significant than the thresholding value (Ii,j > T).

The histogram method requires the least computation time of the measurement-space clustering techniques since it only requires one pass through the data, compared to the iterative technique. Histogram mode seeking is measuring the space of the clustering process by assuming that homogeneous objects in the image become the cluster in management space, i.e., on the histogram. The image segmentation is conducted by mapping the clusters back to the image domain, where the maximally connected components of the cluster labels constitute the image segments.

3.2. Edge Detection

The edge detection methods work by assuming that the edges occur in a discontinuity in the intensity function that is very steep in the image. Edge is defined as a very contrasting intensity value compared to pixels in the neighborhood. There are four steps in edge detection, i.e., image smoothing, enhancement, detection, and edge localization. Object detection using edge detection has three methods, that are Sobel, Prewitt, and finally, Canny edge detection.

The Sobel edge detection can be achieved by differentiating the intensity value of all pixels in an image and finding the maximum derivative point. A gradient is defined as the vector where its composition measures how fast the pixel value changes within the x and y-direction. The gradient can be calculated by

$$\frac{f(x,y)}{x} = \Delta x = \frac{f(x+dx,y)-f(x,y)}{dx} \qquad (1)$$

$$\frac{f(x,y)}{x} = \Delta y = \frac{f(x,y+dy)-f(x,y)}{dy} \qquad (2)$$

where dx and dy are the distances along the x and y direction in sequences.

The operator/kernel functions are to calculate the gradient of image intensity at each point. The maximum of magnitude gradient and direction of the gradient is given by

$$|G| = \sqrt{Gx^2 + Gy^2} \qquad (3)$$

$$\theta = \tan^{-1}\frac{Gy}{Gx} \qquad (4)$$

where G is the gradient magnitude, and θ is the gradient direction. The value of *Gx* and *Gy* is given in fig 4.

-1	0	1
-2	0	2
-1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

Figure 4 The kernel for Sobel Edge detection

The Prewitt edge detection is similar to Sobel. However, its filter is faster compared to Sobel. The kernel of Prewitt edge detection is suitable for an image with high contrast and less noise. Fig 5 shows the kernel of Prewitt edge detection, and the calculation of G and θ is also given in (3) and (4).

-1	0	1
-1	0	1
-1	0	1

Gx

1	1	1
0	0	0
-1	-1	-1

Gy

Figure 5 The kernel for Prewitt Edge detection

Canny edge detection is considered the complex optimal edge detector. This method has to go through several steps: noise reduction, finding the intensity gradient of the image, finding the non-maximum value, hysteresis thresholding, geometric formulation of the Canny edge detector, and finding the variational geometric formulation of the Haralick-Canny edge detector. The Canny edge detection requires a longer processing time due to its multistage algorithm detecting various image sides. Fig 6 shows the kernel of Canny edge detection, and the calculation of G and θ is also given in (3) and (4).

-1	0	1
-2	0	2
-1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

Figure 6 The kernel for Canny Edge detection

3.3 Blob Analysis

Blob (Binary Large Object) analysis is the technique to isolate a group of pixels representing some properties such as brightness and color and compare to surrounding regions within an image. All the points in a blob consider similar to each other. The first step of blob analysis is blob extraction. The extraction is conducted by comparing the blob with a template to match the local features such as points, lines, edge, and the detected region. The next step is blob classification by determining the blob characteristics and comparing the features of each blob with the features of the type of object sought.

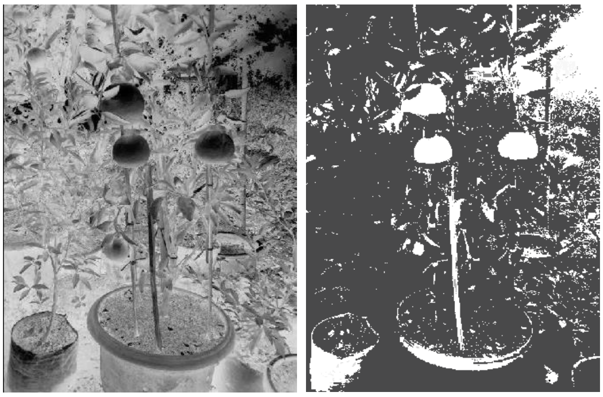
The last step of blob analysis is drawing the bounding box around a blob. The rectangle is drawn by finding the minimum x value, maximum x value, minimum y value, and maximum y value. The bounding box is also known as the ROI (Region of Interest).

4. RESULT AND DISCUSSION

The raw images in fig. 1 are captured by the camera and processed to detect the fruit and isolate them from their backgrounds. The image processing results are fruit position, and based on this position, the robot end-effector can reach the target without being confused by the vegetation. Fig. 7-9 shows the separated target from the vegetation background.

4.1 Image Segmentation

The first method is by using image segmentation to detect oranges, shown in fig. 2. The image processing in this study uses the histogram method by simply setting the proper thresholding method to isolate the background with the oranges to be detected, as shown in fig. 2a. The raw image is converted to an inverted image in fig. 7a, and fig. 7b shows the detected oranges different from their backgrounds vegetation.

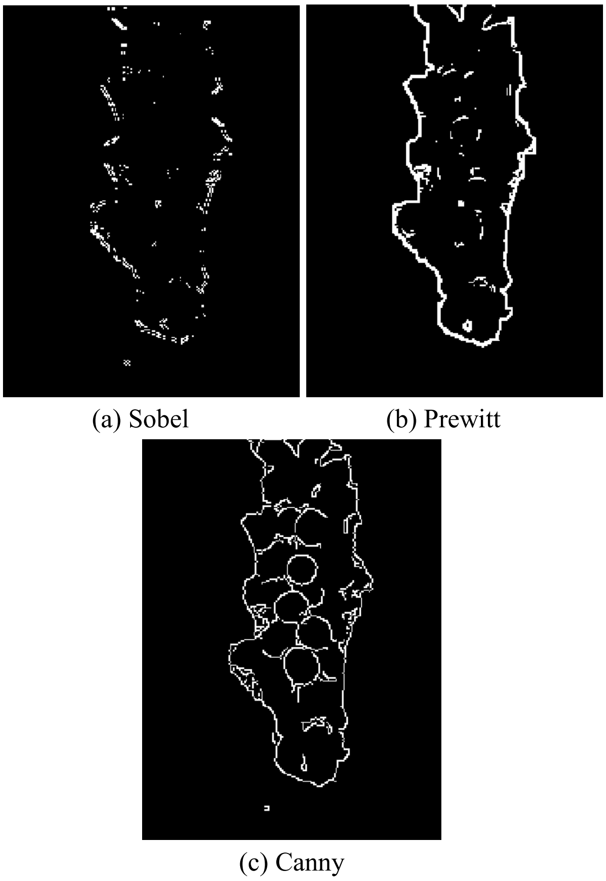


(a) Inverted image (b) The detected orange

Figure 7 Image segmentation

4.2 Edge Detection

The raw image of grapes in its trunk is shown in fig. 8. The first process is Sobel edge detection, shown in fig. 8a where noises are still visible. During the convolution algorithm, noise is also included in the process. Therefore, the result is still noisy, and the shape of the grapes is not visible enough.



(a) Sobel (b) Prewitt

(c) Canny

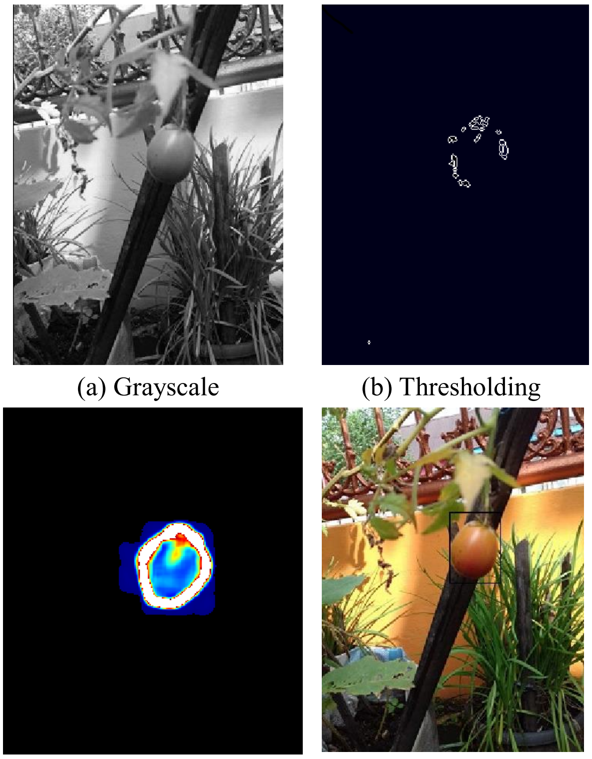
Figure 8 Edge detection for Grapes

The second step is Prewitt edge detection, where the noise still exists as points, but the result is better than Sobel edge detection. The noise is visible due to the size of the kernel filter, and coefficients are fixed and not adjustable to the image input. The result of Prewitt edge detection is shown in fig. 8b.

The last edge detection is Canny. Canny edge detection is based on a Laplacian algorithm that is not sensitive to noise. The output filter image is tiny compared to the gradient-based algorithm (Sobel and Prewitt edge detection). The result of Canny edge detection is shown in fig. 8c, where the shape of grape is visible, this detection is suitable for detecting and tracking the fruit detection.

4.3 Blob analysis

Fig. 8 shows the blob analysis from the raw image of tomato cherry shown in fig. 2c. The first step is converting the raw RGB image into grayscale, as shown in fig. 9a. The gray level image is inverted and segmented using a manually selected threshold, and the result is shown in fig. 9b. where the tomato cherry has been separated from its background.


(a) Grayscale

(b) Thresholding

(c) The detected blob

(d) Bounding box

Figure 9. Blob detection

Fig. 9b is achieved by a cumulated histogram of the most significant detected blob. Fig. 9c is the detected blob achieved by taking the enormous blob and leaving out the other connected area more minor than this minimum size.

Fig. 9c shows the bounding box by taking the x minimum, x maximum, y minimum, and y maximum. Moreover, by blob detection, the position of tomato cherry is detected, and this detection can be used as the input for visual servoing in controlling the arm robot used for fruit harvesting.

The result of image processing in fig. 7-9 show that the fruit is successfully isolated from the vegetation, and its position is known. The detected position of fruit is the input to visual servoing to move the robot in cutting and grabbing the fruit during harvest time.

5. CONCLUSION

The harvest time of seasonal fruit requires special treatment to ensure that the picking fruit is ripe enough to endure the transport time from the farm to the market. Digital farming is an excellent solution to get a higher quality of fruit because the robot is available 24/7 for harvesting. This paper presents the image processing for fruit detection and isolates it from the vegetation. The fruit used as an example is the common fruit found in farming, such as oranges, grapes, and tomato cherry. The results show that the object is detected and relative requires less computational time and is simple. This simplicity and fastness in detecting the object are essential to ensure the available microcontroller in the market can conduct the processing.

AUTHORS’ CONTRIBUTIONS

T. Dewi designed, directed the project, and wrote the paper, Rusdianasari and RD Kusumanto were responsible for data analysis and paper writing, and Siproni designed the mechanical system and was responsible for data collections.

ACKNOWLEDGMENTS

The authors would like to thank Politeknik Negeri Sriwijaya for the support in accomplishing this project and paper by Award Number 3627/PL6.4.2/SK/2021.

REFERENCES

[1] M. F. Stoelen et al., "Low-Cost Robotics for Horticulture: A Case Study on Automated Sugar Pea Harvesting", 10th European Conference on Precision Agriculture (ECPA), Tel Aviv, Israel, 2015, DOI: 10.3920/978-90-8686-814-8 34.

[2] T. Bakker et al., "Robotic weeding of a maize field based on navigation data of the tractor that performed the seeding," IFAC Proceeding Volumes, vol. 43, no. 26, pp. 157-159, 2010, DOI: 10.3182/20101206-3-JP-3009. 00027

[3] S. Amatya et al., "Detection of cherry tree branches with full foliage in planar architecture for automated sweet-cherry harvesting," Biosystems Engineering, vol. 146, pp. 3-15, 2016, DOI: 10.1016/ j.biosystemseng.2015.10.003

[4] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, and X. Zou, "Recognition and localization methods for vision-based fruit picking robots : A review," Frontier in Plant Science, vol. 11, Article 510, pp. 1-17, 2020. DOI: 10.3389/fpls.2020.00510

[5] T. Dewi, P. Risma, and Y. Oktarina, "Fruit Sorting Robot based on Color and Size for an Agricultural Product Packaging System," Bulletin of Electrical Engineering, and Informatics (BEEI), vol. 9, no. 4, pp. 1438-1445, 2020, DOI: 10.11591/eei.v9i4.2353.

[6] G. Sambasivam and G. D. Opiyo, "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks," Egypt. Informatics Journal, In Press, 2020, DOI: 10.1016/j.eij.2020.02.007

[7] T. Dewi, P. Risma, Y. Oktarina, and M. Nawawi, “Tomato Harvesting Arm Robot Manipulator; a Pilot Project,” Journal of Physics: Conference Series, 1500, p 012003, Proc. 3rd FIRST, Palembang: Indonesia, 2020. https://doi.org/10.1088/1742-6596/1500/1/012003

[8] T. Dewi, C. Anggraini, P. Risma, Y. Oktarina, and Muslikhin, “Motion Control Analysis of Two Collaborative Arm Robots in Fruit Packaging System,” SINERGI Vol. 25, No. 2, pp. 217-226, 2021. http://doi.org/10.22441/sinergi.2021.2.013

[9] T. Dewi, P. Risma, Y. Oktarina, and S. Muslimin, “Visual Servoing Design and Control for Agriculture Robot; a Review,” Proc. 2019 ICECOS, 2-4 Oct. 2018, Pangkal Pinang: Indonesia, 2018, pp. 57-62. https://doi.org/10.1109/ICECOS.2018.8605209

[10] T. Dewi, Z. Mulya, P. Risma, and Y. Oktarina, “BLOB Analysis of an Automatic Vision Guided System for a Fruit Picking and Placing Robot,” International Journal of Computational Vision and Robotics, Vol. 11, No 3, pp. 315-326, 2021. https://doi.org/10.1504/IJCVR.2021.115161

[11] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, "Inverse Kinematic Analysis of 4 DOF Pick and Place Arm Robot Manipulator using Fuzzy Logic Controller," International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 2, pp. 1376-1386, 2019. DOI: 10.11591/ijece. v10i2.pp1376-1386.

[12] Y. Oktarina, F. Septiarini, T. Dewi, P. Risma, and M. Nawawi, "Fuzzy-PID Controller Design of 4 DOF Industrial Arm Robot Manipulator," Computer Engineering and Application Journal, vol. 8, no. 2, pp. 123-136, 2019, DOI: 10.18495/COMENGAP.v8i2. 300

[13] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, “Inverse Kinematic Analysis of 4 DOF Pick and Place Arm Robot Manipulator using Fuzzy Logic Controller,” International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 2, pp. 1376-1386, 2019. http://doi.org/10.11591/ijece.v10i2.pp1376-1386

[14] T. Dewi, Rusdianasari, RD Kusumanto, Siproni, F. Septiarini, and M. Muhajir, Autonomous Visual Servoing for Alternately Working Arm Robots, Kinetik: Game Technology, Information System, Computer Network, Computing Electronics, and Control, Vol. 6, No. 3, pp. 177-186, 2021. https://doi.org/10.22219/kinetik.v6i3.1285

[15] C. Wang, Y. Tang, X. Zou, W. SiTu, and W. Feng, ``A Robust Fruit Image Segmentation Algorithm against Varying Illumination for Vision System of Fruit Harvesting Robot", *Optik*, Vol. 131, pp. 626-631, 2017. http://dx.doi.org/10.1016/j.ijleo.2016.11.177.

[16] R. K. Singh, R. B. Singh, and S. C. V. Shekhar, “A Comparative Study of Edge Detection Techniques”, vol. 100 no. 19, 2014, pp. 5–8.

[17] N. M. Syahrian. P. Risma, and T. Dewi, “Vision-Based Pipe Monitoring Robot For Crack Detection Using Canny Edge Detection Method as an Image Processing Technique”. *Kinetik*, vol 2, no.4, pp. 243–250. https://doi.org/10.22219/kinetik.v2i4.243

[18] H. M. Qul’am, T. Dewi, P. Risma, Y. Oktarina, and D. Permatasari, Edge Detection for Online Image Processing of a Vision Guide Pick and Place Robot, 2019 International Conference on Electrical Engineering and Computer Science (ICECOS), pp. 102-106, 2019. DOI: 10.1109/ICECOS47637.2019.8984522

[19] D. K. Ramanpreet Kaur, “Adaptive Grass-Fire Blob Detection Algorithm (AGFBDA) for the Image Matrices,”,2017, pp. 125–128.

[20] M. D. Yusuf, RD. Kusumanto, Y. Oktarina, T. Dewi, and P. Risma, “BLOB Analysis for Fruit Recognation and Detection,” Computer Engineering and Application Journal, vol 7, no 1, 2018, pp. 25-36.

[21] Y. Al Ohali, "Computer Vision Based Date Fruit Grading System : Design and Implementation," Journal of King Saud University - Computer and Information Sciences, vol. 23, no. 1, pp. 29–36, 2011, DOI: 10.1016/j.jksuci.2010.03.003

[22] M. Khojastehnazhand, M. Omid, and A. Tabatabaeefar, "Development of a Lemon Sorting System Based on Color and Size," African Journal Plant Science, vol. 4, no. 4, pp. 122–127, April 2010.

[23] J. Jhawar, "Orange Sorting by Applying Pattern Recognition on Colour Image," Procedia Computer Science, vol. 78, pp. 691–697, December 2016, DOI: 10.1016/ j.procs.2016.02.118.

[24] J. Clement, N. Novas, J. A. Gazquez, and F. Manzano-Agugliaro, "High Speed Intelligent

Classifier of Tomatoes by Colour, Size and Weight," Journal of Agricultural Research, vol. 10, no. 2, pp. 314–325, 2012, DOI: 10.5424/ sjar/2012102-368-11

[25] R. Mahendran, J. Gc, and K. Alagusundaram, "Application of Computer Vision Technique on Sorting and Grading of Fruits and Vegetables," Journal of Food Processing and Technology, pp. 1-7, 2012. DOI: 10.4172/2157-7110.S1-001.

[26] U. O. Dorj, M. Lee, and S. Yun, "An Yield Estimation in Citrus Orchards Via Fruit Detection and Counting Using Image Processing," Computers and Electronics in Agriculture, vol. 140, pp. 103-112, 2017, DOI: 10.1016/j.compag.2017.05.019

[27] L. F. S. Pereira, et al., "Predicting The Ripening of Papaya Fruit With Digital Imaging and Random Forests," Computers and Electronics in Agriculture, vol. 145, pp. 76-82, 2018, DOI: 10.1016/j.compag.2017.12.029

[28] Muslikhin, J. Horng, S. Yang and M. Wang, "Object Localization and Depth Estimation for Eye-in-Hand Manipulator Using Mono Camera," in IEEE Access, vol. 8, pp. 121765-121779, 2020, DOI: 10.1109/ACCESS.2020. 3006843.

[29] G. Tonguc and A. K. Yakut, "Fruit Grading Using Digital Image Processing Techniques," Journal Agricultural Machinery Science, vol. 5, no. 1, pp. 93-101, 2009.

[30] T. Dewi, Amperawan, P. Risma, Y. Oktarina, and D. A. Yudha, "Finger Cue for Mobile Robot Motion Control," Computer Engineering and Application Journal, vol. 9, no. 1, pp. 39-48, 2020, DOI: 10.18495/ COMENGAPP.v9i1.319

[31] https://www.scilab.org/, accessed on October 12, 2017