Measuring Fruit Size via Deep Learning Using a Single Camera

Ying-Tzy Jou[1], Yun-Yun Shih[1], and Chia-Ying Chang[2,*]

[1] Department of Biological Science and Technology, National Pingtung University of Science and Technology, Pingtung, 91201, Taiwan

[2] Bachelor of Program in Scientific Agriculture, National Pingtung University of Science and Technology, Pingtung, 91201, Taiwan

Email: ytjou@mail.npust.edu.tw (Y.T.J), kissmepc@gmail.com (Y.Y.S.), csiefly@mail.npust.edu.tw (C.Y.C.)

*Corresponding author

***Abstract*—This study focused on grading the quality of fruit in packaging plants. Fruit grading requires a body of standard data, which is difficult to achieve in practice. In most studies, relevant data are established solely by the researcher during the course of the research. We mounted cameras at fixed locations on machines to collect image and weight data, allowing farmers to create data feeds without disrupting their work. The system consists of a database of 400 images that were manually labeled and trained on a deep learning network architecture. During the training process, 70% of the images in the database were randomly selected for training, and the other 30% were used for verification to ensure that the training process did not over-learn, as over-learning leads to a decrease in the recognition rate. After the position of the fruit in the image was detected through deep learning, the foreground and background were separated, the information about the fruit was extracted, and the total number of pixels was calculated. Automatic measurement was achieved by converting pixels to millimeters using standards in the environment. The detection rate of the proposed system was over 98%. Using 50 manual measurements of fruit size and automatic detection results for error analysis, the diameter error value was 15.3 mm and the length error value was 14.45 mm.**

***Keywords*—automatic measurement, automatic detection, Intelligent Labor Saving**

I. Introduction

The quality of harvested fruit can be measured through destructive and non-destructive testing. Destructive testing measures levels of Total Soluble Solids (TSS), vitamin C, total sugars, and acidity. The most common non-destructive tests are fruit size and weight. In fruit packaging plants, real-time data for non-destructive testing are required. The current study designed data-based equipment for a traditional fruit weighing and grading machine to collect data such as fruit size and weight for shipping. Size was calculated using real-time identification and standard objects in the environment, while the weight of the fruit was recorded on an electronic scale.

II. Literature Review

Fruit detection methods, including numerical analyses of yield and quality, have been a subject of research since 2005 [1–4]. Most of the detected values are absolute, such as sweetness and acidity, which are measured through destructive testing. The current paper implemented non-destructive testing using deep learning.

In deep learning, there are many different approaches to marking ground truth. Rectangles, irregular-shaped pixels, or polygons can be used. The network architecture and detection results also differ. For example, Yolo v1-v5 [5–9] and SSD [10] use box check marks, Mask-RCNN [11] uses box selection and polygons, and semantic segmentation [12] uses pixel labeling. In the latter example, three marking methods were used to accommodate the irregularity of flower patterns and vines: flower patterns were marked using frame selection, pixels, and outlines, while vines were marked using pixels and outlines.

To increase the accuracy of positioning, semantic segmentation with pixel tags can be based on the pre-trained model Deeplab 3+ [13]. This model is a convolutional neural network specially designed for semantic image segmentation. It is applied in fully convolutional networks, SegNet [14], and U-Net [15]. To speed up the training process, a small amount of information can be added to the pre-trained model. The Cambridge University CamVid dataset [16], which provides pixel-level labels for 32 semantic categories, can also accelerate the process. Object edges can also be optimized, and the Z-axis distance can be used to obtain depth information through radar and binocular and monocular vision to provide 3D positioning parameters. Due to different hardware limitations, achievable error values differ.

The applications of deep learning are diverse. For example, lidar and radar are used in self-driving cars to detect their distance from surrounding objects.

Manuscript received March 19, 2024; revised June 1, 2024; accepted September 30, 2024; published November 18, 2024

Monocular [17, 18] and binocular [19, 20] vision use software logic algorithms to calculate object distance through pinhole imaging. Although the cost of these approaches is less than that of radar, they are easily affected by weather or dirt, and the distance accuracy is poor.

With regard to binocular vision, INTEL D435i has been used in pineapple fields [21] to obtain the center point position of pineapples based on object detection. The Z-axis distance (mm) is determined based on the center point coordinates, and then the X- and Y-axis distances (mm) are obtained through calculation. At a distance of 300–800 mm, the Z-axis error value of this approach is less than 1.12% (−2 to +6 mm), the X-axis error value is less than 1.99%, and the Y-axis error value is less than 1.20%.

Time of Flight (TOF) is a method that uses light reflection to calculate the distance of an object. It uses a light-emitting diode or a laser diode to emit infrared light. When the infrared light is reflected by the object, the distance of the object is obtained by multiplying the speed of light by the time difference. TOF technology can be combined with optical fiber and monocular vision to more accurately detect the distance of an object. Research on this approach was published in a 2020 symposium [22].

III. Materials and Methods

The system flow chart implemented in this study is shown in Fig. 1. The network camera has high-definition resolution, in which the image size is 1280 × 720. We placed the fruit on an automatic rotating disk (i.e., turntable) to obtain images from different angles and then took eight or nine images of each fruit to build an image database. We used a digital scale with an accuracy of 0.02 g to establish the weight of 50 fruits and manually measured the length and diameter of each fruit to analyze the error value of automatic measurement. Because the turntable is an object with a fixed length, width, and height, we regarded this as a standard object with a diameter of 146 mm and a height of 35 mm. We used this standard to obtain the ratio between the actual size in millimeters and the pixels in the picture (mm/pixel). The length and diameter of the fruit were selected by the deep learning frame and multiplied by the ratio to estimate the true size of the fruit. The estimated fruit size and weight were then sent to the database.

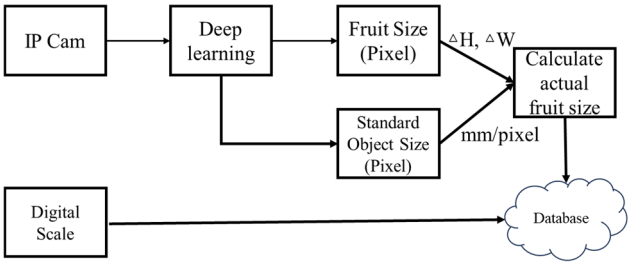


Fig. 1. System flow chart.

A. Image Labeling

We placed the fruits on the turntable and made a mark every 45 degrees from A to H. We took eight images of 50 fruits, resulting in a total of 400 samples. We used MATLAB Image Label tool, as shown in Fig. 2, using a rectangular shape and two category labels (guava, turntable).

For the label ‘guava’, the four sides of the frame had to fit around the periphery of the fruit and could not include the fruit stem. The four sides of the turntable label fit around the periphery of the turntable, completely framing it.

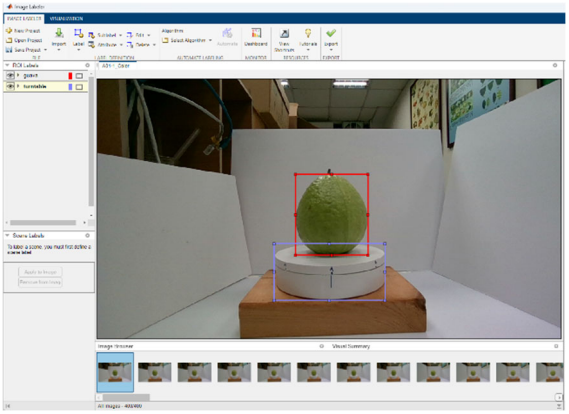


Fig. 2. Two categories of image labeling tools in Matlab.

B. Deep Learning Network

We used the YOLO v4-coco [19] network architecture, which is a real-time system and pre-trained model. In packaging plants, great importance is attached to processing time. Therefore, we selected this architecture for its efficiency. As our image database held less than a thousand images, we needed a pre-trained YOLO network architecture. We thus used the COCO database to pre-train YOLO v4-COCO to obtain the initial network weight. After training with the 400 images in our database, this network architecture could increase the resolution of the system. In order to avoid over-learning during the training process, we randomly selected 70% of the image database (280 images) for training. The remaining 30% (120 images) was used to verify the network architecture after training and was therefore called the test group.

We set the maximum number of epochs to 80 and the learning rate as shown in Fig. 3.

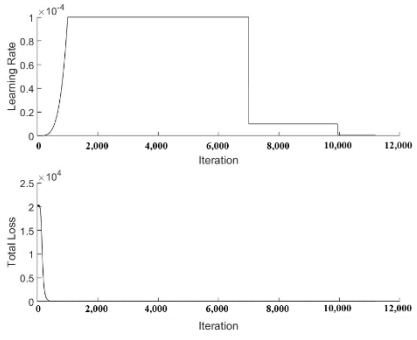


Fig. 3. Learning rate and changes in total loss.

C. Pixel-to-Distance Conversion

Because packaging plants requires immediate, low-cost, and easy-to-install equipment, we did not consider using higher unit price methods such as binocular systems or TOF to obtain more accurate Z-axis distances. Rather, we simply used a color camera to capture a standard object (i.e., a turntable with a diameter of 146 mm and a height of 35 mm) placed in front of the screen. We used the turntable width (*turntable_w*) selected by the frame and the actual turntable diameter (*turntable_diameter*) to calculate the ratio of pixels to the actual size (*conversion ratio*) in each picture, as follows:

$$conversion\_rate = \frac{mm}{pixel} = \frac{turntable\_diameter}{turntable\_w} \quad (1)$$

This allowed us to obtain four parameters (x, y, ΔW, ΔH) for each picture. The starting point in the frame was x and y, and the ΔW and ΔH were the width and length of the frame. We used ΔW and ΔH to multiply the conversion ratio to obtain the estimated fruit size as follows:

$$Fruit_{size(Length)} = \Delta H \times conversion\_rate \quad (2)$$

$$Fruit_{size(Diameter)} = \Delta W \times conversion\_rate \quad (3)$$

IV. Result and Discussion

Using the deep learning architecture of YOLO v4, the recognition rate of both fruits and turntables was 100%. The maximum error between the manual measurement and deep-learning prediction of fruit diameter was 15.3 mm, and the maximum error in length was 14.5 mm.

A. Deep Learning Recognition Rate

The two categories were classified after 11,191 iterations, which took 4.84 hours, with the following results: total loss (6.6×10^{-2}), box loss (2.2×10^{-2}), object loss (4.1×10^{-2}), and class loss (3.1×10^{-3}). The results are shown in Fig. 4.

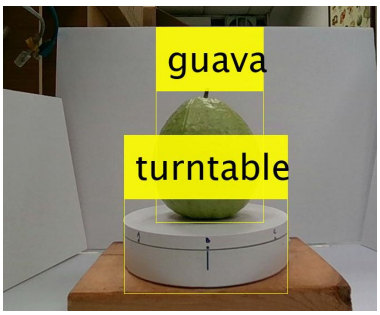
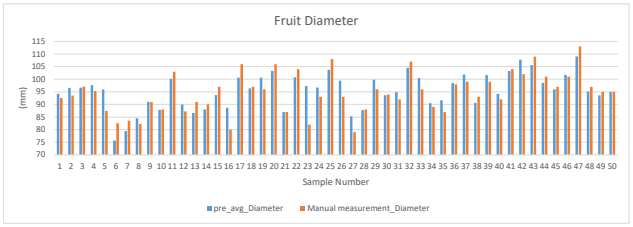


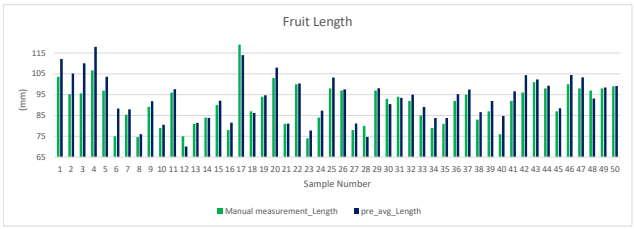
Fig. 4. Image of detected fruit and turntable.

B. Actual Size Error Value

To calculate the predicted fruit size, images were taken of each fruit from eight different angles. In each image, the pixel sizes of the fruit and the turntable were detected. The conversion ratio value was calculated through the pixel width of the turntable in each image and its known width of 146 mm (Eq. (1)). The predicted fruit diameter and length in each image could then be calculated using Eqs. (2) and (3). We averaged the predicted fruit size for each fruit and calculated the standard deviation of the eight pieces of data for each fruit. A total of 100 standard deviation values were all less than 12.5. The maximum diameter error of the average predicted value of each fruit was 15.3 mm, and the maximum length error was 14.5mm, as shown in Fig. 5 and Table 1. In Fig. 5(a), the blue bars were the average predicted fruit diameter value of each fruit, and the orange bars were the diameter value measured manually using a vernier caliper. As shown in Fig. 5(b), the black bars were the average predicted fruit length value of each fruit, and the green bars were the diameter value measured manually.



(a)



(b)

Fig. 5. Prediction and manual measurement of 50 fruits.

In Table 1, the manual measurement values were obtained by measuring the diameter and length of each fruit once. The predicted size values were obtained using Eqs. (1) to (3). The △ values were obtained by subtracting the predicted size value from the manually measured value.

To figure axis labels, use words rather than symbols. Do not label axes only with units. Do not label axes with a ratio of quantities and units.

Color figures will be appearing only in online publication. All figures will be black and white graphs in print publication.

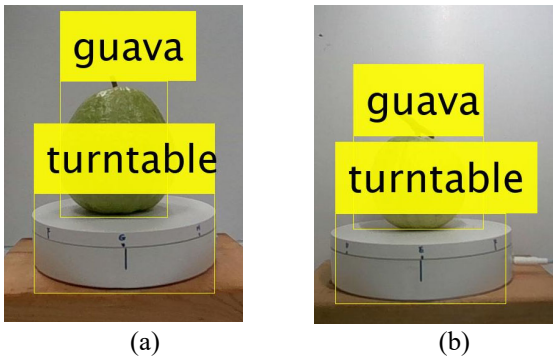
TABLE I. Forecast and Actual Value Error Table

Sample #	Manual measurement (mm)		Predicted size (mm)		△manual-predicted (mm)	
	Diameter	Length	Diameter	Length	Diameter	Length
1	92.5	103.6	94.25	112.12	−1.75	−8.52
2	93.5	95.2	96.53	105.15	−3.03	−9.95
3	97.1	95.6	96.63	110.05	0.47	**−14.45**
4	95.3	106.7	97.67	117.98	−2.37	−11.28
5	87.4	96.9	95.96	103.61	−8.56	−6.71
6	82.5	75	75.70	88.34	6.80	−13.34
7	83.6	85.4	79.41	87.91	4.19	−2.51
8	82.3	74.6	84.49	75.99	−2.19	−1.39
9	91	89.2	91.11	91.82	−0.11	−2.62
10	88	79	87.86	80.52	0.14	−1.52
11	103	96	100.12	97.64	2.88	−1.64
12	87.2	75	89.95	70.08	−2.75	4.92

13	91	81	86.58	81.48	4.42	−0.48
14	90	84	88.04	83.84	1.96	0.16
15	97	90	93.71	92.05	3.29	−2.05
16	80	78	88.69	81.52	−8.69	−3.52
17	106	119	100.61	114.03	5.39	4.97
18	97	87	96.35	86.20	0.65	0.80
19	96	94	100.67	94.66	−4.67	−0.66
20	106	103	103.30	107.97	2.70	−4.97
21	87	81	86.95	81.07	0.05	−0.07
22	104	100	100.76	100.39	3.24	−0.39
23	82	74	97.29	77.83	**−15.29**	−3.83
24	93	84	96.71	87.29	−3.71	−3.29
25	108	98	103.76	103.26	4.24	−5.26
26	93	97	99.43	97.58	−6.43	−0.58
27	79	78	85.26	81.21	−6.26	−3.21
28	88	80	87.73	74.66	0.27	5.34
29	96	97	99.77	98.06	−3.77	−1.06
30	94	93	93.64	90.48	0.36	2.52
31	92	94	94.87	93.51	−2.87	0.49
32	107	92	104.49	95.00	2.51	−3.00
33	96	85	100.46	89.07	−4.46	−4.07
34	89	79	90.46	73.05	−1.46	5.95
35	87	81	91.59	83.77	−4.59	−2.77
36	98	92	98.46	95.21	−0.46	−3.21
37	99	95	101.89	97.47	−2.89	−2.47
38	93	83	90.60	86.65	2.40	−3.65
39	99	87	101.61	91.95	−2.61	−4.95
40	92	76	94.25	84.82	−2.25	−8.82
41	104	92	103.24	96.60	0.76	−4.60
42	102	96	107.72	104.39	−5.72	−8.39
43	109	101	105.60	102.29	3.40	−1.29
44	101	98	98.54	99.37	2.46	−1.37
45	97	87	96.00	88.48	1.00	−1.48
46	101	100	101.71	104.46	−0.71	−4.46
47	113	98	109.05	103.31	3.95	−5.31
48	97	97	95.09	93.21	1.91	3.79
49	95	98	93.59	98.45	1.41	−0.45
50	95	99	95.00	99.16	0.00	−0.16

The predicted height of the third sample was measured manually, and its standard deviations were all less than 5.99. This is because when deep learning automatically selected the fruit, the frame fit incorrectly over the top and bottom of the fruit, as shown in Fig. 6(a). The calculated length therefore exceeded estimates.

The predicted height of the 23rd sample was measured manually, and its standard deviation was less than 6.54. This is because when deep learning automatically selected the turntable, the frame fit incorrectly over the left and right sides of the turntable, as shown in Fig. 6(b). Therefore, the wrong conversion ratio value was calculated.



(a) (b)

Fig. 6. Images of two fruits with larger error values: (a) frame selection did not touch the top and bottom of the fruit; (b) frame selection did not touch the left and right sides of the turntable.

V. Conclusion

In this paper, we used low-cost and simple-to-install equipment. Only a lens and an electronic scale are necessary for the proposed approach to measure fruit quality in packaging plants. We took 8 images of 50 guavas, resulting in a total of 400 images in the database. We also manually measured the length and diameter of the fruit to serve as the ground truth dataset. The recognition rate was 100%, with a maximum error value of 15 mm for fruit diameter and length and a maximum standard deviation of 12.5. In the samples with the largest error, the frame selection was floating; i.e., the object did not touch the edges of the frame. In future work, we plan to add more samples to the database, including images taken in different environments to increase the robustness of identification.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ying-Tzy Jou and Chia-Ying Chang conducted the research; Yun-Yun Shih analyzed the data; Chia-Ying Chang wrote the paper and program coding; all authors have approved the final version.

Funding

This work was supported by the National Science and Technology Council (NSTC) of Taiwan (NSTC 113-2313-B-020-010).

References

[1] K. Thaipong, and U. Boonprakob, "Genetic and environmental variance components in guava fruit qualities," *Scientia Horticulturae*, vol. 104, no. 1, pp. 37–47, 2005.

[2] P. Kumar, J. P. Tiwari, and Raj Kumar, "Effect of N, P & K on fruiting, yield and fruit quality in guava cv. Pant Prabhat," *J. Hortic. Sci*, vol. 3, no. 1, pp. 43–47, 2008.

[3] V. Rawat, Y. K. Tomar, and J. M. S. Rawat, "Influence of foliar application of micronutrients on the fruit quality of guava cv. Lucknow-49," *Journal of Hill Agriculture*, vol. 1, no. 1, pp. 75–78, 2010.

[4] D. M. Kadam1, P. Kaushik, and R. Kumar, "Evaluation of guava products quality," *International Journal of Food Science and Nutrition Engineering*, vol. 2, no. 1, pp. 7–11, 2012.

[5] J. Redmon, *et al*., "You Only Look Once: Unified, Real-Time Object Detection," In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] J. Redmon, and A. Farhadi, "YOLO9000: Better, Faster, Stronger," In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] J. Redmon, and A. Farhadi, "YOLOv3: An Incremental Improvement." *Computer Vision and Pattern Recognition*, vol. 1804. Berlin/Heidelberg, Germany: Springer, 2018.

[8] A. Bochkovskiy, C.-Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv Print, arXiv: 2004.10934, 2020. doi: 10.48550/arXiv.2004.10934

[9] X. K. Zhu, *et al*., "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," arXiv Print, arXiv: 2108.11539, 2021. doi: 10.48550/arXiv.2108.11539

[10] W. Liu, *et al.*, "SSD: Single shot multibox detector," *Computer Vision—ECCV*. vol. 9905, 2016.

[11] K. He, *et al.*, "Mask R-CNN," arXiv Print, ArXiv: 1703.06870 [Cs], 2018. doi: 10.48550/arXiv.1703.06870

[12] R. Kemker, *et al.*, "High-resolution multispectral dataset for semantic segmentation," CoRR, arXiv:1703.01918, 2017. doi: 10.48550/arXiv.1703.01918

[13] L. C. Chen, *et al.*, "Encoder-decoder with atrous separable convolution for semantic image segmentation," ECCV, 2018.

[14] V. Badrinarayanan, *et al.*, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 39, no. 12, pp. 2481–2495, 2017.

[15] O. Ronneberger, *et al.*, "U-Net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, p. 9351, 2015.

[16] G. J. Brostow, *et al.*, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[17] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," In *Proc*. *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2017, pp. 6602–6611.

[18] Y. Kuznietsov, J. Stuckler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," In *Proc*. *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2017, pp. 6647–6655.

[19] J. Sun, N.N. Zheng, and H.Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp.787–800, 2003.

[20] T. Kanade, and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 920–932, 1994.

[21] C. Y. Chang, C. S. Kuan, H. Y. Tseng, P. H. Lee, S. H. Tsai, S. J. Chen, "Using deep learning to identify maturity and 3d distance in pineapple fields," *Scientific reports*, vol. 12, no. 1, p. 8749, 2022.

[22] C. Y. Chang, S. J. Chou, L. J. Lee, T. S. Liao, "Design image module of laser rangefinder with dual branch fiber light guide," *Automation*, 2020.