Journal of
Information Systems Engineering
and Business Intelligence

Vol.8, No.1, April 2022
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

Mask R-CNN and GrabCut Algorithm for an Image-based Calorie Estimation System

Tiara Lestari Subaran[1)*], Transmissia Semiawan[2)], Nurjannah Syakrani[3)]

[1)2)3)] Department of Computer Engineering and Informatics, Bandung State Polytechnic, Indonesia
Jl. Gegerkalong Hilir, Ciwaruga, Bandung
[1)]tiara.lestari.tif417@polban.ac.id, [2)] transmissia@jtk.polban.ac.id, [3)] nurjannahsy@jtk.polban.ac.id

Abstract

Background: A calorie estimation system based on food images uses computer vision technology to recognize and count calories. There are two key processes required in the system: detection and segmentation. Many algorithms can undertake both processes, each algorithm with different levels of accuracy.
Objective: This study aims to improve the accuracy of calorie calculation and segmentation processes using a combination of Mask R-CNN and GrabCut algorithms.
Methods: The segmentation mask generated from Mask R-CNN and GrabCut were combined to create a new mask, then used to calculate the calorie. By considering the image augmentation technique, the accuracy of the calorie calculation and segmentation processes were observed to evaluate the method's performance.
Results: The proposed method could achieve a satisfying result, with an average calculation error value of less than 10% and an F1 score above 90% in all scenarios.
Conclusion: Compared to earlier studies, the combination of Mask R-CNN and GrabCut could obtain a more satisfying result in calculating food calories with different shapes.

Keywords: Augmentation, Calorie Calculation, Detection

Article history: Received 19 September 2021, decision after peer review 14 November 2021, accepted 24 December 2021, available online 28 April 2022

I. INTRODUCTION

Obesity, which often leads to health issues such as diabetes and hypertension [1], is a common problem in today's society. The number of obese people increases every year. Calorie intake—the amount of energy that allows a human body to function properly [2]—if excessive, can lead to obesity. The unused calorie will be stored in the body as fat, which can accumulate over time and increase the body weight [3]. An uncontrolled increase in body weight can lead to overweight and eventually obesity.

Calories in food can be calculated from energy density multiplied by weight [4]. Technologies to weigh calorie has been developing rapidly, such as the one based on a food image. This system utilizes computer vision technology to recognize and count calories as seen on an image. Two key processes required in the system are detection and segmentation [5]. Many algorithms can undertake these processes, with each showing a different level of accuracy [6].

In recent years, studies on this topic have been extensive, proposing different methods to detect and segment food images. Liang and Li [4] used a computer vision-based Faster R-CNN algorithm to detect food and a GrabCut algorithm to segment the food. They created a dataset composed of 19 food categories and proposed the estimation method, which only has an average error value of less than 20% in 15 food categories. In 2019, Yogaswara et al. [7] presented a different approach using the Mask R-CNN algorithm to do an instance-aware semantic segmentation. Unlike Liang and Li, who used food with different shapes, Yogaswara et al. used five different types of food with the same shape, a rectangular prism. The average calorie calculation error was below 20% in all five categories. Meanwhile, Poply and Angel [8] also used Mask R-CNN to detect and segment food in an image. They proposed a method that detects the number of food pixels and multiplies it with the number of calories per square inch. The average calculation error was 5% for a single food and 7% for a whole meal. Poply and Angel then conducted another study using the same calorie estimation method [9] but with a different segmentation process. They used Faster R-

* Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print)
doi: http://dx.doi.org/10.20473/jisebi.8.1.1-10

CNN to generate a bounding box and the RefineNet algorithms to generate a segmentation mask. This segmentation method has an average calculation error of around 10% for a single food and a whole meal.

Other than calorie estimation, research has also covered object segmentation. Wu et al. [10] proposed an improvement of Mask R-CNN by combining it with the GrabCut algorithm to increase the accuracy. The result shows that this combination can increase the accuracy of the segmentation and significantly reduce the time needed for the process.

Based on the study of Wu et al. [10], the current study proposed the use of Mask R-CNN combined with GrabCut for a calorie estimation system. This study aims to analyze the performance of Mask R-CNN and GrabCut algorithm for segmentation in a calorie estimation system, i.e., the accuracy of the estimation and segmentation. The outline of the paper is as follows. Section 2 presents the proposed calorie estimation system. Section 3 presents the results of this study. Section 4 discusses the results. Section 5 concludes the research.

II. METHODS

The calorie estimation system in this research was developed using the segmentation method proposed by Wu et al. [10], i.e., Mask R-CNN and GrabCut. The complete workflow of the proposed system is depicted in Fig. 1, with a full explanation presented in subsection B.

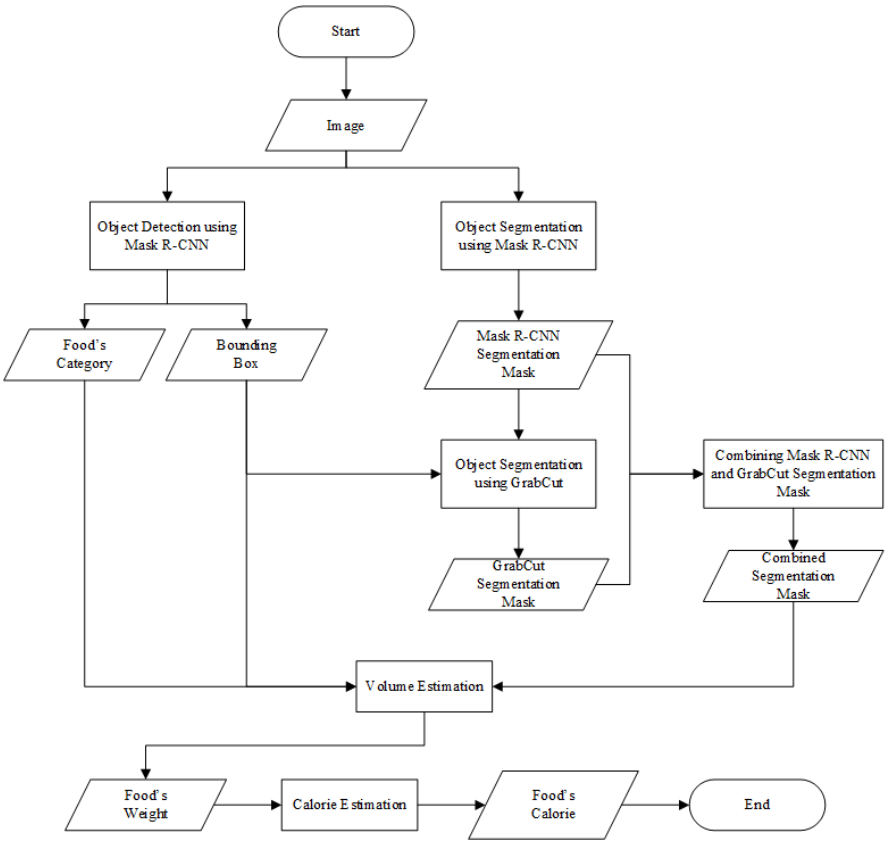

Fig. 1 The proposed system workflow

A. Research Data

The dataset used in this study combines the ECUSTFD dataset [11] and the data we manually collected. ECUSTFD is a dataset created by Liang and Li [11] consisting of 19 food categories with each containing four pieces of information: annotation, mass(g), volume(cm^3), density(g/cm^3), and energy(kcal/g). To complement the ECUSTFD dataset, we also collected images manually by taking a food picture with a calibration tool. The calibration tool—a 500 Rupiah coin with a diameter of 2.7cm—aims to provide a point of comparison to estimate the real food measurement in an image. Examples of the data are shown in Fig. 2.

Fig. 2 A few examples of manually collected food pictures with a calibration tool

In the current experiment, we chose ten food categories: apple, bread, banana, doughnut, grape, orange, tomato, boiled egg, fried tempeh, and fried tofu. Seven out of ten categories were available on the ECUSTFD dataset. Table 1 shows the distribution of the ECUSTFD dataset and the manually collected data on each category, as well as the total images used in this research. This data was used to train the Mask R-CNN architecture, with the distribution ratio of 60% training data, 20% validation data, and 20% testing data.

TABLE 1
RESEARCH DATA

Category	ECUSTFD	Manual
Apple	50	50
Banana	50	50
Boiled Egg	-	100
Bread	50	50
Fried Tempeh	-	100
Fried Tofu	-	100
Doughnut	50	50
Grape	50	50
Orange	50	50
Tomato	50	50
TOTAL	**350**	**650**

B. Calorie Estimation Method

In the studies by Liang and Li [4] and Ege et al. [12], calculating calories from food images can be broken down into five main steps: image acquisition, object detection, object segmentation, volume estimation, and calorie calculation. The current study uses the volume estimation method proposed by Liang and Li [4] by taking the side and top view of the food. In other words, to estimate a volume, the system requires two types of images. Fig. 3 is an example of input images needed by the system to estimate food's volume. After the system receives the input images, the detection and segmentation processes begin.

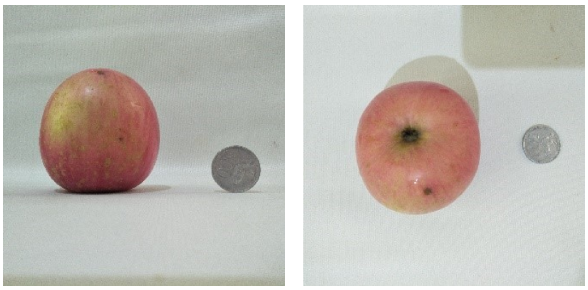

Fig. 3 The side and top view image of an apple with a 500 Rupiah coin as a calibration tool

The detection and the first segmentation process were done by using Mask Region-based Convolutional Neural Network (Mask R-CNN) algorithm. Mask R-CNN is an instance segmentation algorithm developed by He et al. [13]. This algorithm is an enhancement of the object detection algorithm, Faster R-CNN. The enhancement is achieved by adding a new branch in Faster R-CNN to predict the segmentation mask of each object in an image. Therefore, in Mask R-CNN, the detection and segmentation processes will be done in one architecture. For the detection process,

Mask R-CNN will return the bounding boxes for all objects detected in an image and the category of each object. The result of detection using Mask R-CNN can be seen in Fig. 4. The dotted line around the apple is the bounding box that Mask R-CNN will generate. On the top, left corner of each bounding box is the category of the food. The purpose of the bounding box is to locate and determine the object's coordinate. In a volume estimation, the bounding box provides the height and width of the food, and the category determines the food's mass, density, and energy.

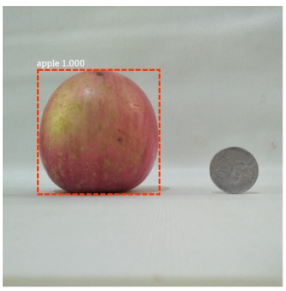

Fig. 4 Detection result using Mask R-CNN

After a detection process was completed, the next step was to segment each object in the image using Mask R-CNN and GrabCut algorithms. The segmentation process using Mask R-CNN generates a segmentation mask in the form of a binary array. This segmentation mask was then used as an input for GrabCut—an interactive segmentation algorithm that uses texture (observed from the colours) and edges (observed from the contrast) to segment an object [14]. Interactive segmentation requires users to select the area or object to be segmented manually. Unlike Mask R-CNN, which can perform the segmentation process automatically after going through some training process, the GrabCut algorithm requires a user's assistance. In this study, the result of Mask R-CNN segmentation was used as an input for the GrabCut algorithm so that the segmentation process was automatic without a user's assistance. The segmentation mask generated by Mask R-CNN and GrabCut was then combined to create a new mask. The result of the segmentation process can be seen in Fig. 5, which shows a segmentation mask for an apple.

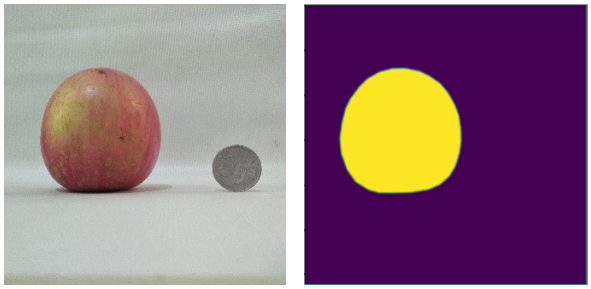

Fig. 5 Segmentation result using both Mask R-CNN and GrabCut

After both detection and segmentation processes were completed, the next step was to estimate the volume of the food in the image by using a calibration tool. Using the method proposed by Liang and Li [4], the scale factor was calculated using (1) to determine the side view's scale factor and (2) to determine the top view's scale factor.

$$\alpha_S = \frac{d}{(W_S+H_S)/2} \quad (1)$$

$$\alpha_T = \frac{d}{(W_T+H_T)/2} \quad (2)$$

where,
α_S : side view's scale factor
α_T : top view's scale factor
d : calibration tool diameter
W_S, H_S : width and height of calibration tool's bounding box from side view image
W_T, H_T : width and height of calibration tool's bounding box from top view image

The volume of each food will then be calculated based on the food's shape. Liang and Li [4] divided the food into three shapes, ellipsoid, column (rectangular prism), and irregular. Each shape has a different formula used to calculate volume, (3) calculate the volume for ellipsoid-shaped food, (4) calculate the volume for column-shaped (rectangular prism) food, (5) calculate the volume for irregularly shaped food, and (6) helps to calculate the sum of the number of pixels in the top view's mask.

$$v_e = \frac{\pi}{4} \times \sum_{k=1}^{H_S} (L_S^k)^2 \times \alpha_S^3 \quad (3)$$

$$v_c = (s_T \times \alpha_T^2) \times (H_S \times \alpha_S) \quad (4)$$

$$v_i = (s_T \times \alpha_T^2) \times \sum_{k=1}^{H_S} (\frac{L_S^k}{L_S^{MAX}})^2 \times \alpha_S \quad (5)$$

$$s_T = \sum_{k=1}^{H_T} L_T^k \quad (6)$$

where,

v_e : volume for ellipsoid-shaped food
v_c : volume for column-shaped (rectangular prism) food
v_i : volume for irregularly-shaped food
s_T : sum of the number of pixels in row k from the top view's mask
π : a mathematic constant with the value of 3.14
L_S^k : the number of pixels in row k from the side view's mask
L_T^k : the number of pixels in row k from the top view's mask
L_S^{MAX} : the maximum number of pixels in one row from the side view's mask

The volume will then be used to calculate the calorie of each food. Two pieces of information are needed at this stage: food's weight and energy density (kcal/g). The weight was obtained from the volume by multiplying it with the density of the food (g/cm^3) as defined in (7).

$$Weight\ (gr) = Volume\ (cm^3) \times Density\ (gr/cm^3) \quad (7)$$

$$Calorie\ (kcal) = Weight\ (gr) \times Energy\ Density\ (kcal/gr) \quad (8)$$

The density was collected manually by weighing and calculating the volume of each food used in this experiment. Based on the study by Stumbo and Weiss [15], food density can be estimated by collecting a few samples of each food and counting the average density. Equation (7) calculates the food's weight, then used to calculate the calorie using (8). To do so, we need information about the energy density obtained from FoodData Central (FDC)—an integrated data system that provides nutritional information about food. These data were collected by the US Department of Agriculture (USDA) and accessible via https://fdc.nal.usda.gov/download-datasets.html.

C. Image Augmentation

Image augmentation is a technique to artificially create images to increase the amount of training data [16]. Machines can learn new data types by using augmentation techniques on training datasets. Some examples of augmentation techniques are crop, rotate, and shear. Image augmentation was used to improve the performance of deep neural networks with relatively little training data. The imgaug library (https://github.com/aleju/imgaug) used in this study has been developed to perform image augmentation in a machine learning experiment. The hue and saturation augmentation techniques augment the images before being used to train the Mask R-CNN model.

D. Evaluation Method

The proposed method's performance was evaluated both in terms of calorie estimation and segmentation process. To evaluate the accuracy of the calorie estimation, the predicted calorie was compared to the ground truth calorie to find out the error value of each calculation, which was calculated using (9) following Yogaswara et al. [7]. According to the FDA (Food and Drug Administration), the US Health Department's acceptable error limit of calorie value displayed on packaged foods is 20%. Hence, the benchmark is less than 20%.

$$Error = Ground\ Truth - Prediction\ \frac{100}{Ground\ Truth} \quad (9)$$

Meanwhile, the segmentation process can be evaluated using the Intersection over Union (IoU) metric, which shows the difference between the ground truth and the predicted bounding box [17]. But instead of using the bounding box, the IoU value for segmentation process was calculated by comparing the ground truth and the predicted segmentation mask formulated in (10) [18]. This approach was chosen to accurately compare the pixels difference between the ground truth and the predicted mask area.

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \quad (10)$$

Fig. 6 section A shows the detection process' IoU, and section B shows examples of how the position of the bounding box will affect the IoU. A larger IoU indicates that the predicted data is closer to the ground truth (more accurate). This calculation result was then used to determine the precision and recall scores. The former shows the percentage of all objects detected correctly in an image, while the latter shows the percentage of all objects detected in an image.

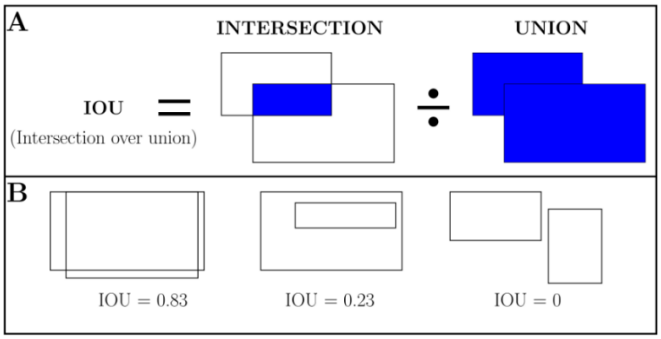

Fig. 6 Intersection over Union (IoU) exemplarily for bounding boxes [17]

The precision and recall scores were calculated by grouping the IoU calculation into three categories [19]. True Positive (TP) is when a system gives correct predictions for positive data. False Positive (FP) is when a system gives wrong predictions for positive data. False Negative (FN) is when a system gives wrong predictions for negative data. The categorization follows a minimum threshold. For a minimum threshold value of 0.5, the categorization for each IoU is as shown in Table 2 below.

TABLE 2
RESULT CATEGORIZATION

Scenario	Category
IoU > 0.5	TP
IoU < 0.5	FP
IoU > 0.5 but incorrect classification	FN

After determining the category of each result, the precision score was calculated using (11) and the recall score was calculated using (12).

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

F1 score metric can show the balance between precision and recall. It can measure the accuracy of a system against the dataset by combining the precision and recall values [20]. Equation (13) was used to calculate the F1 score obtained by the system.

$$F1\ score = 2 \times \left(\frac{precision \times recall}{precision + recall}\right) \quad (13)$$

III. RESULTS

In this study, the experiment was carried out in six scenarios, as shown in Table 3. Each scenario had a different value of epochs and learning rate, as well as the status of augmentation when training the machine learning model. For scenarios that used augmentation, the hue and saturation augmentation technique from imgaug library [12] was used in model training. First, food images were inputted into the calorie estimation system. The output given by the system was recorded: the food volume, weight, and calorie. These pieces of information were used to calculate the system error by comparing it with the ground truth data. Table 4 shows the error calculation result of each scenario. Another output to be noted was the combined segmentation mask generated by the system. This was subsequently used to calculate the IoU of each object detected in the image to show the system performance. Table 5 shows the value of precision, recall, and F1 score obtained from each scenario.

TABLE 3
EXPERIMENTAL SCENARIOS

Scenario	Epoch	Learning Rate	Augmentation
A	30	0.001	No
B	30	0.002	No
C	30	0.002	Yes
D	100	0.001	No
E	100	0.002	No
F	100	0.002	Yes

TABLE 4
ERROR CALCULATION RESULT

Scenario	Error (%)		
	Volume (cm^3)	Weight (g)	Calorie (kcal)
A	9.25	9.06	9.08
B	7.75	7.68	7.64
C	5.47	5.38	5.43
D	8.55	8.48	8.31
E	7.51	7.53	7.31
F	7.83	8.07	7.59

TABLE 5
CALCULATION OF PRECISION, RECALL, AND F1 SCORE

Scenario	Precision	Recall	F1 Score
A	1.00	0.88	0.93
B	0.97	0.87	0.91
C	1.00	0.93	0.96
D	1.00	0.90	0.95
E	1.00	0.88	0.93
F	1.00	0.95	0.97

IV. DISCUSSION

The proposed method used in this study obtained an average calculation error value of less than 10% and an F1 score above 90% for all scenarios. The error calculation obtained by Yogaswara et al. [7] was also less than 10% but only in five categories with the same shape. Meanwhile, Liang and Li's [4] calculation error was less than 20% in 14 out of 19 categories. This study shows that Mask R-CNN and GrabCut can achieve a calculation error of less than 10% with various food shapes. Additionally, the accuracy of the F1 score does not completely affect the value of the calorie calculation error generated by the system. Table 4 shows that the lowest error is obtained by scenario C, while the largest F1 score is obtained by scenario F. This can happen because, in calculating the volume, the number of pixels in each line of the segmentation mask affects the final volume calculation. Fig. 7 shows an example of a banana's prediction results, considered food with an irregular shape. To calculate the volume of food with an irregular shape, the number of pixels on each line of the segmentation mask greatly affects the final calculation result. As shown in Fig. 7, the segmentation mask generated does not cover the entire object. Therefore, the volume calculation is

affected by the loss of several pixel lines in the segmentation mask. This is why segmentation accuracy does not completely affect the calculation error.

Fig. 7 Prediction result of a banana

The analyses also cover the relation between the value of epoch and learning rate and the accuracy of the segmentation process. Table 6 and Fig. 8 show the training and validation losses in each scenario. Two epoch values of 30 and 100 were used in this study. The epoch value of 30 was used in scenarios A, B, and C, while the epoch value of 100 was used in scenarios D, E, and F. As shown in Fig. 8, the scenarios using epoch value of 100 (D, E, F) has a smaller average train loss and validation loss compared to scenarios using epoch value of 30 (A, B, C). From these results, it can be concluded that the epoch value can affect the accuracy of the system because the smaller the loss value, the higher the accuracy is.

TABLE 6
TRAIN LOSS AND VALIDATION LOSS OF EACH SCENARIO

Scenario	Train Loss	Validation Loss
A	0.13918	0.18796
B	0.15510	0.21574
C	0.16588	0.19544
D	0.07800	0.15718
E	0.08079	0.16363
F	0.09496	0.16063

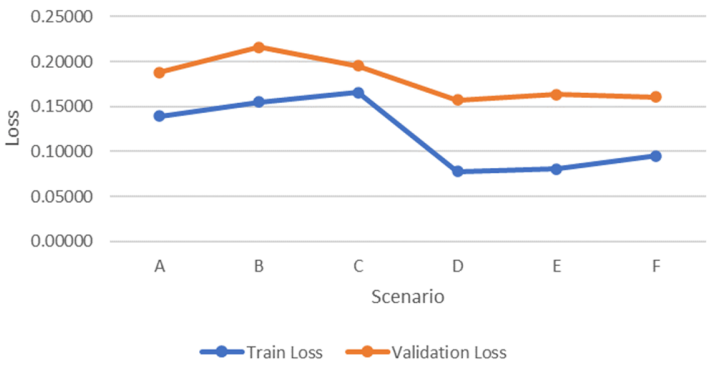

Fig. 8 Graphic of train loss and validation loss obtained by each scenario

Regarding learning rate, there are two learning rate values used in this experiment, 0.001 and 0.002. Scenarios A and D use the learning rate value of 0.001, while the other scenarios use the learning rate value of 0.002. The collected

data in Table 6 shows that a smaller learning rate value results in a smaller loss. In the epoch value 30's scenarios (A, B, C), the lowest average loss value was in scenario A. Then, in epoch value 100's scenarios (D, E, F), the smallest average loss value was in scenario D. If the loss value is compared to the precision and recall values obtained for each scenario, excluding the scenarios using augmentation (C and F), scenarios A and D yield the best result.

A large precision value indicates that the prediction done by the system is accurate and generates an IoU value above the minimum threshold. While a large recall value indicates how well the system can detect objects inside an image, greater values mean that the system successfully detects more objects. The results shown in Table 5 show that scenarios A and D generate higher precision and recall values than scenarios B and E.

Other than epoch and learning rate value, another factor that can affect the system accuracy is image augmentation. Scenarios C and F, where augmentation techniques were used, yield the largest precision and recall values in their respective epoch groups. This shows that augmentation techniques can increase the accuracy and recognition of the system. To see the model's accuracy in each scenario, F1 scores were calculated. In Table 5, the results of the F1 score calculation for each scenario shows that the largest F1 score was in scenario F, which uses image augmentation.

V. CONCLUSIONS

Mask R-CNN and GrabCut algorithms were used in an image-based calorie estimation system in this research. The proposed method achieved a satisfying result with an average calorie calculation error of less than 10% for all food categories in the dataset and an F1 score above 90% for all experimental scenarios. Scenarios with epochs 100, a learning rate of 0.002, and a hue and saturation augmentation technique show the highest F1 score of 0.97. In this case, future studies will benefit from changing the backbone used in Mask R-CNN and using a simpler volume calculation method.

The calorie calculation system built using Mask R-CNN, and GrabCut took a long time to predict one input image. The time required to process images and calculate calories was approximately three minutes. This was caused by the many layers in the Mask R-CNN architecture. Therefore, the backbone can be changed to reduce the processing time in future studies. In this study, ResNet-101 was used as the backbone, further studies can be conducted using ResNet-50, which has fewer layers.

Further studies can also be conducted for a simpler volume calculation method. The method used in this study requires two pictures of food taken from the top and side to calculate the volume of the food. With two images, the time required by the system also doubles. Future studies should aim to find and compare other volume calculation methods that can calculate the volume from side images and handle stacked foods, processed foods, and liquid foods' images.

Author Contributions: *Tiara Lestari Subaran*: conceptualization, methodology, software, formal analysis, investigation, visualization, data curation, writing - original draft. *Transmissia Semiawan*: supervision, methodology, validation, writing – review & editing. *Nurjannah Syakrani*: supervision, validation, formal analysis, writing – review & editing.

Funding: This research received no specific grant from any funding agency.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] S. M. Fruh, "Obesity: Risk factors, complications, and strategies for sustainable long-term weight management," *Journal of the American Association of Nurse Practitioners*, vol. 29, pp. S3–S14, 2017, doi: 10.1002/2327-6924.12510.

[2] J. L. Hargrove, "Does the history of food energy units suggest a solution to 'calorie confusion'?," *Nutrition Journal*, vol. 6, pp. 1–11, 2007, doi: 10.1186/1475-2891-6-44.

[3] D. Park, J. Lee, J. Lee, and K. Lee, "Deep learning based food instance segmentation using synthetic data," *2021 18th International Conference on Ubiquitous Robots, UR 2021*, pp. 499–505, 2021, doi: 10.1109/UR52253.2021.9494704.

[4] Y. Liang and J. Li, "Deep Learning-based Food Calorie Estimation Method in Dietary Assessment," *arXiv*, no. Jianhua Li, 2017.

[5] K. Okamoto and K. Yanai, "An automatic calorie estimation system of food images on a smartphone," *MADiMa 2016 - Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, co-located with ACM Multimedia 2016*, pp. 63–70, 2016, doi: 10.1145/2986035.2986040.

[6] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, "Application of Deep Learning in Food: A Review," *Comprehensive Reviews in Food Science and Food Safety*, vol. 18, no. 6, pp. 1793–1811, 2019, doi: 10.1111/1541-4337.12492.

[7] R. D. Yogaswara, E. M. Yuniarno, and A. D. Wibawa, "Instance-Aware Semantic Segmentation for Food Calorie Estimation using Mask R-CNN," *2019 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, no. August, pp. 416–421, 2019, doi: 10.1109/ISITIA.2019.8937129.

[8] P. Poply and A. J. Angel, "An Instance Segmentation approach to Food Calorie Estimation using Mask R-CNN," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, Oct. 2020, pp. 73–78. doi: 10.1145/3432291.3432295.
[9] P. Poply and J. A. Arul Jothi, "Refined image segmentation for calorie estimation of multiple-dish food items," in *Proceedings - IEEE 2021 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS 2021*, Feb. 2021, pp. 682–687. doi: 10.1109/ICCCIS51004.2021.9397169.
[10] X. Wu, S. Wen, and Y. ai Xie, *Improvement of Mask-RCNN Object Segmentation Algorithm*, vol. 11740 LNAI, no. October. Springer International Publishing, 2019. doi: 10.1007/978-3-030-27526-6_51.
[11] Y. Liang and J. Li, "Computer Vision-based Food Calorie Estimation: Dataset, Method, and Experiment," *arXiv*, 2017.
[12] T. Ege, W. Shimoda, and K. Yanai, "A New Large-scale Food Image Segmentation Dataset and Its Application to Food Calorie Estimation Based on Grains of Rice," *MADiMa '19: Proceedings of the 5th International Workshop on Multimedia Assisted Dietary Management*, no. October, 2019.
[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.
[14] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut' - Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004, doi: 10.1145/1015706.1015720.
[15] P. J. Stumbo and R. Weiss, "Using Database Values to Determine Food Density," *Journal of Food Composition and Analysis*, vol. 24, no. 8, pp. 1174–1176, 2011, doi: 10.1016/j.jfca.2011.04.008.
[16] J. Yang, Y. Zhao, J. C. W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2016-Novem, pp. 5079–5082, 2016, doi: 10.1109/IGARSS.2016.7730324.
[17] J. Salau and J. Krieter, "Instance Segmentation with Mask R-CNN Applied to Loose-Housed Dairy Cows in A Multi-Camera Setting," *Animals*, vol. 10, no. 12, pp. 1–19, 2020, doi: 10.3390/ani10122402.
[18] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary IoU: Improving Object-Centric Image Segmentation Evaluation," pp. 15334–15342, 2021.
[19] K. E. Koech, "Confusion Matrix for Object Detection." https://towardsdatascience.com/confusion-matrix-and-object-detection-f0cbcb634157 (accessed Sep. 18, 2021).
[20] V. M. Zolotarev, "Thresholding Classifiers to Maximize F1 Score Zachary," *Journal of Applied Spectroscopy*, vol. 7, no. 5, pp. 503–506, 2014.