FOOD IMAGE DETECTION SYSTEM AND CALORIE CONTENT ESTIMATION USING YOLO TO CONTROL CALORIE INTAKE IN THE BODY

1st Fitroh Romadhon
Department of Electrical Engineering
Universitas Sebelas Maret
Surakarta, Indonesia
fitrohromadhon@student.uns.ac.id

2nd Faisal Rahutomo
Department of Electrical Engineering
Universitas Sebelas Maret
Surakarta, Indonesia
faisal_r@staff.uns.ac.id

3rd Joko Hariyono
Department of Electrical Engineering
Universitas Sebelas Maret
Surakarta, Indonesia
jokohariyono@staff.uns.ac.id

Abstract*— Excess calories in the body can cause obesity and several degenerative diseases such as diabetes mellitus, heart disease, stroke, hypertension, and others. This system is designed to help people maintain a balanced calorie content that enters the body. This system is designed using the YOLO algorithm model to detect the type of food which is then developed using the Python programming language to estimate the calories of the detected food. YOLO uses the principle of feature extraction in images that are processed through filters in the form of arrays to perform detection. Estimated food calories are calculated by multiplying the calories for each food by the amount according to the type of food detected. The calorie value of the food provided is based on the number of calories for each portion of food taken from FatSecret Indonesia. The result of this study is that food detection performance is quite good with average precision, recall, and F1-score values of 0.94, 0.90, and 0.91 respectively when testing the model. However, when tested on Hugging Face, the performance decreased with the average values of precision, recall, and F1-score respectively, namely 0.84, 0.32, and 0.41. This decrease in performance is due to poor CPU usage and a decrease in image quality when uploaded to Hugging Face.

Keywords— calories, degenerative diseases, YOLO, Python, precision, recall, F1-score

I. INTRODUCTION

Excess calories in the body is a bad condition because it can lead to obesity and several degenerative diseases such as diabetes mellitus, heart disease, stroke, hypertension, and others. This problem is increasing rapidly in developing countries such as Indonesia. Based on data obtained from Riskesdas (basic health research), the trend of increasing excess calories in the body in adults is 10.5% in 2007, 14.8% in 2013 and 21.8% in 2018 [1].

The way to prevent excess calories in the body is to regulate a healthy diet by limiting the need for sufficient calories and exercising frequently. A healthy diet can be done by consuming foods that are low in energy, have enough vitamins and minerals, contain lots of fiber, it is advisable to choose foods that contain little fat and carbohydrates [2].

The Regulation of the Minister of Health of the Republic of Indonesia Number 28 of 2019 concerning the recommended nutritional adequacy rate for Indonesian people contains the government's vision to create a healthy Indonesian society by providing the recommended nutritional adequacy rate. The government has advocated for the Indonesian people to live a healthy life by regulating adequate nutritional intake that is needed by the body. This is in line with this research about helping realize the government's vision of creating a healthy Indonesian society [3].

Given these problems, a system design study was created to quickly calculate estimated calories, namely "Food Image Detection System and Estimation of Caloric Content Using Deep Learning to Regulate Calorie Intake in the Body". In summary, the design of this system works using machine learning to detect and estimate the calorie content contained in food.

II. LITERATURE REVIEW

A. Calories

Calories are a measure of the energy in food that humans need to move and even survive. However, excessive calorie intake consumed by the human body can be fatal, for example diabetes and obesity are examples that often occur in today's society. The average man needs 1,600 - 2,650 calories and the average woman needs 1,400 - 2,250 calories [3]. Caloric needs in humans depend on daily activities because this relates to how much energy humans use in daily activities, height, weight, and other factors such as age and gender. Calorie is abbreviated as "cal" [4].

The main cause of obesity is the body's inability to balance the number of calories absorbed into the body with the number of calories released by a person's body. Therefore, this can cause an imbalance of calories in the human body, both lacking and excess calories [4].

B. YOLO

YOLO is an acronym for you only look once which is one of the algorithms from CNN development and one of the deep learning-based object detection models. YOLO was first developed by Joseph Redmond who was later developed by Alexey Bochkovskiy, Chien Yao Wang, and Hong Yuan Mark Liao [5].

The advantages of the YOLO algorithm are that it can detect objects accurately in real time, is relatively easy to install compared to other models, is open source so that it can be modified and used for commercial purposes, the documentation is quite complete and the forum is very active, and it has very high generalization capabilities so with just a little data the model can detect well. The YOLO structural model is illustrated in Figure 1 [6].

The YOLO architecture uses the concept of a single-stage object detection method. The idea of object detection with this

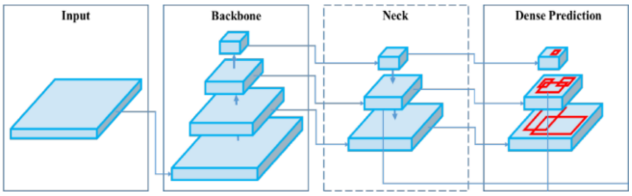

Fig. 1. Yolo Architecture [6]

one stage is to only see the image once. In this one-time image reading stage, there are four block stages, namely input, backbone, neck, and dense prediction. The following describes the four stages of the YOLO architecture [7].

1. Input
   At this stage, the visual image will be resized in pixels according to the resolution of the input layer. The resolution can be resized provided that the pixel value can be divided by 32. In general, the input image that is accepted is a color image or an image with 3 channels.

2. Backbone
   Backbone is the core backbone of the algorithm which refers to the architecture that performs feature extraction. An example of a backbone that is commonly used in the YOLO algorithm is the cross stage partial connection (CSP). CSP divides the input feature map into two parts, namely dense net and CSP dense net.

3. Neck
   This stage aims to provide layers of stages between backbone and dense prediction. With this block, the object detection process can be performed on objects of various sizes.

4. Dense prediction
   This stage is the determination of bounding boxes and the classification of objects detected in the class specified in each bounding box. The way this stage works is by dividing the input image into grid cells and each cell has an anchor box that will make predictions on an image. The prediction results on the anchor box allow for overlapping. This problem is solved by NMS (non max suppression). NMS works by calculating the IoU (intersection over union) between the anchor box and the ground truth. The final result taken is the anchor box with the largest IoU value.

C. *Mean Average Precision (mAP)*

Mean average precision is a matrix used to evaluate an object detection model. The mAP value ranges from 0 to 100. The higher the mAP number, the better the system quality. The way to calculate the mAP value is by calculating the average precision separately for each class, then the average precision value for each class is averaged [7].

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad for\ n\ classes \tag{1}$$

D. *Intersection over Union (IoU)*

Intersection over union is the ratio of the overlapping area between the two bounding boxes detected by the detection and the ground truth bounding box. The threshold value that is often used is 0.5. Therefore, it is often referred to as mAP 0.5 in its accuracy [8].

E. *Confusion Matrix*

The confusion matrix is used as a parameter that can measure the performance of the detection model. This parameter is usually applied to binary classification as well as multi-class classification. In these parameters there are several terms which are explained in the following:

TABLE I. CONFUSION MATRIX

	Predicted (0)	Predicted (1)
Actual (0)	TN	FP
Actual (1)	FN	TP

1. True Positive (TP)
   TP is the number of positive samples that are correctly and accurately predicted. If using a mAP of 0.5, then the prediction results will be considered TP if the prediction results match and the IoU value $\geqslant$ 0.5.

2. True Negative (TN)
   TN is the number of correctly predicted negative samples. TN cannot be used for object detection because it is irrelevant if used.

3. False Positive (FP)
   FP is the number of samples whose original value is negative but is detected as positive. If using a mAP of 0.5, then the prediction results will be considered FP if the prediction results match but the IoU value < 0.5.

4. False Negative (FN)
   FN is the number of samples whose original value is positive but is detected as negative. If using mAP 0.5, then the predicted results will be considered FN if no results are detected and if the detection is appropriate but the IoU value < 0.5.

F. *Precision*

Precision is a measurement standard that is used to calculate the performance accuracy of the results of object detection. The precision value can be obtained from the calculation of the true positive value divided by the sum of the true positive and false positive values. For more details described in the following equation [9]:

$$Precision = TP/(TP+FP) \tag{2}$$

G. *Recall*

Recall is a measurement value to find out the number of ground truth objects that can be detected by the system. The recall calculation can be done by comparing the quotient of the number of true positives with the sum of the true positives and false negatives by dividing the true positive values by the ground truth. For more details described in the following equation [9]:

$$Recall = TP/(TP+FN) \tag{3}$$

H. *F1-score*

F1-score is the average comparison value of precision and recall calculations that have been weighted. F1-score has a maximum value of 1.0 and a minimum value of 0. The model can be said to be good in terms of precision and recall if it has an F1-score value of 1 or close to 1. However, if the F1-score has a value of 0 or close to 0 then the model bad to say. How to find the F1-score value is explained in the following equation [9]:

$$F1 = 2\times(Precision\times Recall)/((Precision+Recall)) \tag{4}$$

I. *Python*

Python is a versatile programming language that has a high level of capability and capability by focusing on very clear code syntax. Python was first created in 1991 by a programmer from the Netherlands which was then developed today. The development of Python can be said to be quite rapid by adding various kinds of syntax for its use so that it has a very large and comprehensive functional library to date. The Python programming language can also be said to be a language that is quite easy to understand in terms of writing scripts compared to other programming languages [10].

III. RESEARCH METHODOLOGY

A. *Research Methods*

In this study, several methods were used to collect data used as information in the preparation of this study. The research method diagram is presented as shown in Figure 2.

B. *Calorie Data Retrieval*

Food image calorie data retrieval in this study uses the assumption that Indonesian food portions are taken from information on calorie data from FatSecret Indonesia https://www.fatsecret.co.id/kalori-gizi/. All calorie data per serving of the types of food that will be used for this study are shown in Table 3 which will be taken from FatSecret Indonesia.

C. *Image Data Retrieval*

Image data is taken manually by downloading images on the internet. The total number of image data taken is 560 images. There are 50 image data for each class consisting of apples, fried chicken, oranges, white rice, cakes, bananas, fried tofu, omelet, boiled eggs, and fried tempeh and 20 image data for each class ½ apple, ½ orange, and ½ boiled egg.

D. *Image Annotation*

At this stage an annotation or class name labeling is carried out on each object in the image with the appropriate bounding box. This image annotation process produces a file with the extension "txt" or "text document". Inside the file there is some information about the image that has been annotated, such as the class name, the x and y center coordinates of the bounding box, and the length and width of the bounding box.

E. *Dataset Augmentation*

The image augmentation stage is the stage after the preprocessing is done. This augmentation process aims to increase the dataset with a varied amount of data. Image augmentation process using tools in Roboflow. In this study

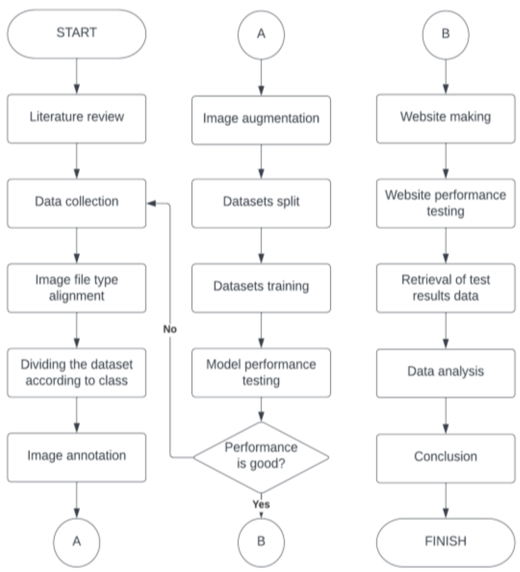

Fig. 2. Research Methods Flowchart

TABLE II. FOOD LIST

No.	Food name	Name in model
1.	Apple	Apel
2.	Half Apple	Apel Set
3.	Fried chicken	Ayam Goreng
4.	Orange	Jeruk
5.	Half Orange	Jeruk Set
6.	Rice	Nasi Putih
7.	Potato Cakes	Perkedel Kentang
8.	Banana	Pisang
9.	Fried tofu	Pisang Set
10.	Omelet	Telur Dadar
11.	Boiled eggs	Telur Rebus
12.	Half Boiled Egg	Telur Rebus Set
13.	Fried Tempe	Tempe Goreng

using 7 ways of augmentation, namely flip, saturation, exposure, blur, noise, cutout, and mosaic.

1. Flip

Flip augmentation process is done in two ways, namely flip horizontally and vertically. Flip horizontally is to reverse the position of the image to the horizontal line, while flip vertically is to change the position of the image to the vertical line.

2. Saturation

Saturation augmentation is done by changing the saturation value in the image, which means reducing and increasing the level of color intensity in the image. In this study, the image saturation value was changed by -25% and +25%.

3. Exposure

Exposure augmentation is done by changing the exposure value in the image, which means reducing and adding light intensity to the image. In this study, changing the image exposure value was -25% and +25%.

4. Blur

Blur augmentation is carried out with the aim of providing several very diverse conditions regarding the level of clarity in shooting later in the test. In this research, we changed the blur value by 2.5 px.

TABLE III. FOOD CALORIE DATA

No.	Food name	Portion (Weight)	Calories (cal)
1.	Apple	1 piece (150 g)	72
2.	Half Apple	½ piece (75 g)	36
3.	Fried chicken	1 piece (86 g)	239
4.	Orange	1 piece (140 g)	62
5.	Half Orange	½ piece (70 g)	31
6.	Rice	1 piece (158 g)	204
7.	Potato Cakes	1 piece (15 g)	21
8.	Banana	1 piece (118 g)	105
9.	Fried tofu	1 piece (13 g)	35
10.	Omelet	1 piece (61 g)	93
11.	Boiled eggs	1 piece (50 g)	77
12.	Half Boiled Egg	½ piece (25 g)	39
13.	Fried Tempe	1 piece (15 g)	34

5. Noise

Noise augmentation is done by adding unwanted color dots in an image. This noise is also done with the aim of providing several very diverse conditions later during testing. In this research, the addition of noise is 5% of the pixels.

6. Cutout

Cutout or gridmask augmentation is done by adding several black squares to an image. This is done to force the system to study the image by not displaying the object completely. In this study, 5 boxes were cut out with a size of 15% each.

7. Mosaic

Augmentasi mosaic dilakukan guna menambah variasi dataset by combining several images into one to provide conditions for testing objects that have many and different classes in one frame/image.

F. Dataset Split

The next step is to connect Google Colaboratory with Google Drive storage to synchronize between projects on Google Colaboratory and storage on Google Drive in the email used. This is done to access data that contains data on Google Drive which will later be split and trained, such as image data and data with the "yaml" extension that were previously uploaded to Google Drive storage.

Before conducting dataset training using the YOLO v8 model, a dataset split was performed to divide the dataset into training and validation data. The dataset split process was carried out by 80% for training data and 20% for validation data, resulting in 1344 training image data and 336 validation images from a total of 1680 datasets. The split data is then placed in the data folder.

G. Dataset Training

The next stage is to conduct image data training using Google Colaboratory. The data training process uses the YOLO model with the version "yolov8m.pt". Model training is carried out to train each image data to recognize the specified objects. Model training in this study uses the value of epochs = 100 with batch = 8 and adjusts the image size to 640x640 as shown in Table 4.5. The results of the training model use "yolov8m.pt" in the form of a file "best.pt" which is a training result file from the YOLO v8 model.

IV. RESULT AND DISCUSSION

A. Calorie Estimation Calculation

Calculation of estimated calories in the detected food is done by setting the number of calories for each food in the program code. Calculation of total estimated food calories uses the principle of multiplying the number of calories in each type of food by the number of objects detected as that type of food. For example is in Figure 8 there are white rice objects which have 204 cal calories, fried chicken 239 cal, and fried tempeh 34 cal, so the results of the total calories detected are the total number of each calorie of the object detected which is 477 cal.

If there are cases of stacked food objects, the detection system cannot classify objects properly. However, if the food objects in the image are only stacked or slightly covered by other objects, the system can detect food objects properly. This is influenced by the number of datasets being trained and the augmentation stage can also play a good role in this case, especially in the cutout type augmentation which removes some of the food image so that the system can detect it properly.

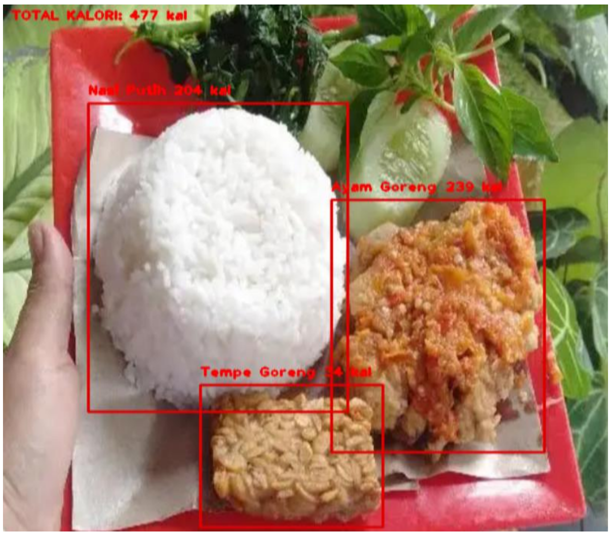

Fig. 3. Model Testing Result

B. Model Performance Testing

Model performance testing is carried out to test the accuracy of the performance of the model resulting from training for detection. The test was carried out using 100 sample images taken from image data that are different from the training data. Testing this model was carried out using the Pycharm IDE. Example of model testing result is shown in Figure 3.

The results of model testing are illustrated in Table 4 using the TP, FP, and FN values. In the confusion matrix table there are still some tests that are not quite right in the process of detecting objects. For example, in the omelet detection test, there are still many wrong object detections. For example, if the actual value is an omelet object but the system detects it is not an omelet or does not detect it and if the actual value is not an omelet but the system detects an omelet. These are respectively referred to as false negatives (FN) and false positives (FP). However, there are also several classes whose detection results are quite good, for example, apple. Even

TABLE IV. CONFUSION MATRIX MODEL TESTING

Prediksi \ Aktual	Apel	Apel Set	Ayam Goreng	Jeruk	Jeruk Set	Nasi Putih	Perkedel	Pisang	Tahu Goreng	Telur Dadar	Telur Rebus	Telur Set	Tempe Goreng	Back Ground
Apel	17													
Apel Set		10												
Ayam Goreng			37							2				4
Jeruk				49										1
Jeruk Set					8									1
Nasi Putih						9								
Perkedel			1				78						1	
Pisang								89			1			2
Tahu Goreng							1		107	1				1
Telur Dadar						2			2	15				1
Telur Rebus											40	2		2
Telur Set												10		
Tempe Goreng			5										44	1
Back Ground			7	2		2	9		6	1			6	

TABLE V. MODEL TESTING PERFORMANCE

	TP	FP	FN	Precision	Recall	F1-Score
Apple	17	0	0	1	1	1
Half Apple	10	0	0	1	1	1
Fried Chicken	37	6	13	0.86	0.74	0.79
Orange	49	1	2	0.98	0.96	0.97
Half Orange	8	1	0	0.89	1	0.94
Rice	9	0	4	1	0.69	0.82
Potato Cakes	78	2	10	0.97	0.89	0.93
Banana	89	3	0	0.98	1	0.98
Fried Tofu	107	3	8	0.97	0.93	0.95
Omelet	15	5	4	0.75	0.79	0.77
Boiled Egg	40	4	1	0.91	0.97	0.94
Half Boiled Egg	10	0	2	1	0.83	0.91
Fried Tempe	44	6	7	0.88	0.86	0.87

though there is still one FP value because the orange object is detected as an apple.

The TP, FP, and FN values can be further developed to determine the precision, recall, and F1-score values as shown in Table 5. These values are obtained by calculating each precision, recall, and F1-score equation.

In Table 5 it can be concluded that the average class has obtained a fairly good performance with an average value of precision, recall, and F1-score respectively, namely 0.94 or 94%, 0.90 or 90%, and 0.91 or 91%. Data from the test results for object detection with the highest F1-score value were obtained by apple and half apple data with an F1-score value of 1 or 100%, while the lowest F1-score value was obtained by the omelet test with a value of 0.77 or 77%. This is likely to occur because in the apple dataset there are many apple objects in an annotated image so that they can be data for better training, whereas in the majority of the omelette dataset there is only one omelette object in an image so there is only one annotation which causes a lack of data for the training process. Another thing that is possible to happen is because the apple testing factor is not too diverse so that the model can detect objects easily, while the omelet has test data that has too many conditions so the test results are not good enough.

C. *Website Performance Testing*

Website testing is carried out to test the performance of object detection on websites that have been created using the Hugging Face platform. On the website testing is done with data that is different from the training data. However, to compare the performance of the model with the web, a test was carried out using the same data as the previous model test data. The image test data totals 100 test data images.

The testing process is carried out by inputting images one by one then analyzing the results by recording the TP, FP, and FN values. In the image output there is also the total caloric value of the object detected in the image. As an example, in Figure 4 there are objects of white rice 204 cal, fried tempeh 34 cal, and fried chicken. However, in testing Figure 10, the fried chicken object was not detected by the system so that the total calories detected were 238 cal.

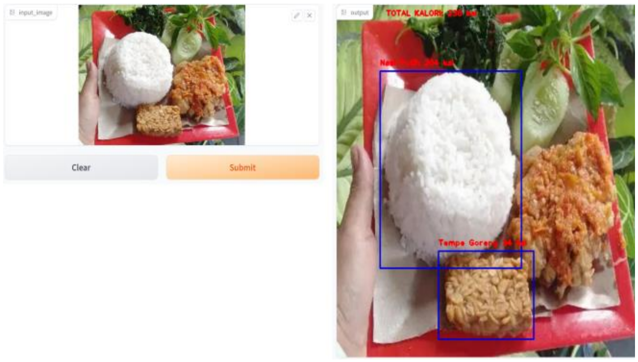

Fig. 4. Hugging Face Test Results

After carrying out the testing process on the website, the TP, FP, and FN values were obtained from all the test image results. The test result data is displayed in the confusion matrix table as shown in Table 6. From the confusion matrix data it can be analyzed that the results are different from the confusion matrix table during model testing even though testing uses the same image test data. The confusion matrix data testing on the web has many objects that are not detected and the prediction results have many errors.

If you look at Table 7 regarding the calculation of precision, recall, and F1-score values, it can be analyzed that the performance of the test results on this implementation is quite poor with an average value of precision, recall, and F1-score, respectively 0.84 or 84%, 0.32 or 32%, and 0.41 or 41%. The system can be said to be quite good if it has an F1-score close to 1 or at least greater than 0.80. In this test, the best F1-score was obtained by testing boiled eggs with a value of 0.88 or 88%, while the worst score was found in fried tofu, which was 0.06 or 6%.

The reason for the poor test results is that there are several factors that might be the reasons. The most important factor is the difference between GPU and CPU when processing object detection. In testing the training results model on Pycharm

TABLE VI. CONFUSION MATRIX HUGGING FACE TESTING

Prediksi \ Aktual	Apel	Apel Set	Ayam Goreng	Jeruk	Jeruk Set	Nasi Putih	Perkedel	Pisang	Tahu Goreng	Telur Dadar	Telur Rebus	Telur Set	Tempe Goreng	Back Ground
Apel	9						9		2					
Apel Set		3												
Ayam Goreng			8											
Jeruk				9										
Jeruk Set					1									
Nasi Putih						7								
Perkedel							40		3				1	
Pisang							1	37		2				1
Tahu Goreng									4					
Telur Dadar										3				
Telur Rebus						1					37	5		
Telur Set												3		
Tempe Goreng			1										5	
Back Ground	8	7	41	42	7	5	38	52	106	14	4	4	45	

TABLE VII. CONFUSION MATRIX HUGGING FACE TESTING

	TP	FP	FN	Precision	Recall	F1-Score
Apple	9	11	8	0.45	0.53	0.49
Half Apple	3	0	7	1	0.3	0.46
Fried Chicken	8	0	42	1	0.16	0.27
Orange	9	0	42	1	0.18	0.30
Half Orange	1	0	7	1	0.12	0.22
Rice	7	0	6	1	0.54	0.70
Potato Cake	40	4	48	0.91	0.45	0.61
Banana	37	4	52	0.90	0.41	0.57
Fried Tofu	4	3	111	0.57	0.03	0.06
Omelet	3	5	16	0.37	0.16	0.22
Boiled Egg	37	6	4	0.86	0.90	0.88
Half Boiled Egg	3	0	9	1	0.25	0.40
Fried Tempe	5	1	46	0.83	0.10	0.17

IDE using a Laptop GPU with NVIDIA GeForce RTX 3050 specifications, while in testing the model on the Hugging Face web using the default CPU with 2 basic vCPU specifications 16 GB RAM. The second possible factor is that when uploading the training result file to Hugging Face in the form of best.pt it experiences degradation so that the performance of the training result file decreases not according to the original. Another possible factor is the use of Hugging Face which is free, not premium or paid, thus allowing restrictions on the use of systems or functions in Hugging Face.

CONCLUSION

In general, the way to design a system for detecting and estimating the calorie content of food starts from dataset search, dataset annotation, dataset augmentation, dataset training, and model implementation. The dataset used is image data in "JPG" format. Dataset annotations are labeling of objects by giving bounding boxes to objects in the image. Dataset augmentation is done to increase the variety of data. Dataset training is carried out to train data so that it can detect the model. Model implementation is carried out to apply the model that has been trained so that it can be used to detect and estimate calories in food.

The performance of the object detection model training in food is good enough to detect objects in food. Detection performance can be seen by calculating the precision, recall, and F1-score values of the model test results. In the test results, the model has an average precision, recall, and F1-score of 0.94, 0.90, and 0.91, respectively. This means that object detection in the training model has a performance of 91%. The higher the F1-score or close to 1.0, the better the detection performance. This proves to be true like the existing theory.

The performance of the results of system implementation on websites using Hugging Face gets a pretty bad value. This can be seen through the calculation of the value of precision, recall, and F1-score from the results of testing the implementation of the model. The precision, recall, and F1-score values generated in the Hugging Face test have relatively low average values, respectively 0.84, 0.32, and 0.41. This means that object detection in the implementation of the model has a performance of 41%. The lower the F1-score or close to 0, the worse or inaccurate the results of the detection performance will be. This proves to be true like the existing theory. Some of the factors that cause low performance on Hugging Face are the use of low-quality CPU so that the processing of detection is not good. Another factor is the degradation of file quality when uploading on the Hugging Face platform and the performance limitation of Hugging Face due to using a free or non-premium account.

REFERENCES

[1] Director General of Disease Prevention and Control of the Ministry of Health of the Republic of Indonesia, "Guidelines for the Implementation of the Archipelago Movement to Reduce Obesity Rates (GENTAS)," 2017.

[2] Khairullah and L. A, "Implementation of Dynamic Programming to Determine Diet Programs for Overweight (Obesity) Sufferers," *Media Infotama*, vol. 15, no. 1, pp. 37-43, 1 February 2019.

[3] Minister of Health of the Republic of Indonesia, "Regulation of the Minister of Health of Indonesia number 28 of 2019 concerning the recommended nutritional adequacy rate for Indonesian people," 2019.

[4] R. E. Jayaputra, "Estimation of Calories in Food through Food Imagery using the SSD (Single Shot Detector) Model on a Mobile Device," *Thesis*, 26 October 2020.

[5] P. Hidayatullah, Buku Sakti Deep Learning, 3rd ed., Cimahi, West Java: Stunning Vision AI Academy, 2023.

[6] A. Dahlan, D. Ariateja and A. Arghanie, "The Automatic Weapon Detection System uses Smart CCTV-based Deep Learning," *Sistem Cerdas*, vol. 4, no. 2, pp. 126-141, 2021.

[7] Raharja and Algonz, "Machine Learning: Pengertian, Cara Kerja, dan 3 Metodenya!," 24 February 2022.

[8] R. Padilla, S. Netto and E. d. SIlva, "A Survey on Performance Metrics for Object-Detection Algorithms," *Conference Paper*, 24 July 2020.

[9] M. Maulidah, W. Gata, R. Aulianita and I. C. Agustyaningrum, "Decision Tree Classification Algorithm for Book Recommendations based on Book Category," *Jurnal Ilmiah Ekonomi dan Bisnis*, vol. 13, no. 2, pp. 89-96, December 2020.

[10] A. N. Syahrudin and T. Kurniawan, "Input and Output in the Python Programming Language," *Jurnal Dasar Pemrograman Python STMIK*, pp. 1-7, June 2018.