

Food Portion Weight Prediction and Nutritional Estimation from Images Using YOLOv8 Segmentation and XGBoost Regression

Ery Setiyawan Jullev Atmadji^{1*}, Freda Adi Ferdana², Husin³, Aji Seto Arifianto⁴

^{1,2,3,4}Information Technology Department, Politeknik Negeri Jember, Jember, East Java, Indonesia
E-mail: ^{1*}ery@polije.ac.id, ²freda_ferdana@gmail.com, ³husin@polije.ac.id, ⁴aji_seto@polije.ac.id

(Received: 26 Mar 2026, revised: 28 Apr 2026, accepted: 1 May 2026)

Abstract

Understanding the nutritional content of food is essential for maintaining balanced dietary habits. However, most existing nutrition information sources rely on fixed portion sizes and do not reflect the actual amount of food consumed. This study proposes an image-based system for estimating food nutrition dynamically using computer vision and machine learning. The system integrates YOLOv8m-seg for food object detection and segmentation with XGBoost regression for food weight prediction. Images uploaded by users through a Telegram chatbot are processed to detect food containers and segment individual food objects. Features extracted from segmentation results, such as object area and dimensions, are then used to estimate the weight of each food item. Nutritional values including calories, fat, carbohydrates, and protein are calculated based on the predicted weights. Experimental results show that the segmentation model achieved an average accuracy of 87.68%, with several food categories reaching 97–98% accuracy. The weight prediction model obtained an MAE of 21.138 g, RMSE of 50.10 g, and R^2 of 0.7838, indicating reasonable predictive performance. The developed system demonstrates the potential of combining object detection, segmentation, and regression models to provide automated nutritional estimation through an accessible chatbot interface, supporting more practical and personalized dietary assessment.

Keywords: Food Detection, YOLOv8, XGBoost, Nutrition Estimation, Computer Vision.

I. INTRODUCTION

Nutrition is vital for human growth and development, since sufficient and balanced nutrient consumption is necessary for general health maintenance. Nutritional imbalances, resulting from either lack or excess [1], can precipitate numerous health issues. Inadequate nutrient consumption correlates with disorders such as anaemia, stunting, and wasting, whereas excessive consumption is linked to non-communicable diseases [2], including hypertension, diabetes, and obesity. In 2022, the World Health Organization (WHO) reported that roughly 390 million persons aged 18 and older globally were underweight, while 2.5 billion were overweight, including 890 million who were classified as obese [3]. Approximately 390 million children and adolescents aged 5–19 years were classified as overweight (including 160 million with obesity), whereas 190 million were identified as underweight according to Body Mass Index (BMI) assessments. Moreover, figures from the United Nations Children's Fund reveal that in Indonesia in 2018, approximately thirty percent of children under five exhibited stunting, and ten percent suffered from wasting, with inadequate dietary patterns highlighted as a significant contributing factor [4].

Suboptimal dietary habits frequently arise from insufficient understanding of the nutritional composition of foods. Inadequate comprehension may lead individuals to consume food without regard for its nutritional worth, heightening the danger of insufficient or excessive nutrient consumption in relation to the body's requirements [5]. Consequently, having at least a fundamental comprehension of nutritional content is vital for facilitating healthy food choices. Despite enhanced public awareness in recent years owing to the increased availability of educational media, obstacles persist in making nutritional information both accessible and practically useful in everyday life [6].

Diverse educational media have been created to improve public awareness of nutrition, including printed booklets featuring food images with nutritional values per 100 grams, social media platforms overseen by nutritionists and dietitians, and mobile applications that offer nutritional information [7]. Nonetheless, the majority of current media provide nutritional information in a fixed and generalised manner, frequently relying on standardised serving sizes instead of actual portions ingested. Thus, the acquired information may be less precise or potentially deceptive when actual portion sizes deviate from

the established guidelines, underscoring the necessity for more adaptive and individualised nutritional assessment systems.

In the contemporary digital age, there is a growing necessity for systems that can deliver dynamic nutritional assessments based on actual food items ingested. Object detection technology, which pertains to the identification and localisation of items inside an image, presents a viable approach. Numerous object identification techniques have been established, including R-CNN, Fast R-CNN, YOLO, and RetinaNet. Nani's prior research employed a Convolutional Neural Network (CNN) with YOLOv3 to identify five food items, attaining good accuracy in single-object recognition [8]; but, performance deteriorated with an increase in object count due to dataset constraints and differences in object scale. Moreover, comparison analysis conducted by Ramadhansyah [7] revealed that YOLOv8 surpasses previous YOLO iterations, with a mean Average Precision (mAP) of 53.9% and a processing speed of 280 FPS, primarily attributable to its anchor-free detection methodology.

Besides item detection, proper weight estimation is crucial for appropriate nutritional assessment. Research conducted by Konstantakopoulos et al. [9] indicated that XGBoost has robust efficacy in regression tasks for forecasting meal weight. This study uses YOLOv8 to identify five prevalent Indonesian dishes—satay, rendang, soto, fried chicken, and fried rice—and utilises XGBoost to estimate the weight of the discovered food items based on segmentation outcomes. This research seeks to develop an image-based system that integrates advanced object detection and regression techniques to deliver more precise and dynamic nutritional estimates, thereby enhancing technological progress in nutrition education and serving as a novel educational tool for dietary assessment.

II. RESEARCH METHOD

A. Research Flow

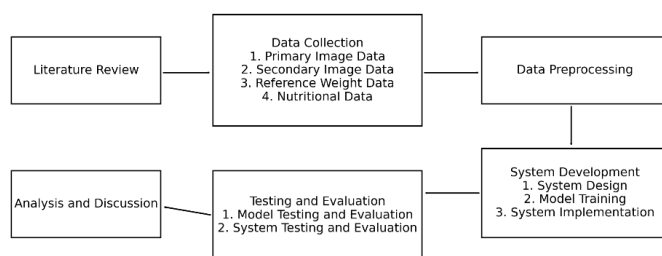


Figure 1. Research Flow

The diagram in Figure 1 depicts the research methodology workflow, commencing with the literature review as the theoretical basis [8]. Subsequently, data collection is conducted, encompassing primary image data, secondary image data, reference weight data, and nutritional data. The gathered data is subsequently subjected to preprocessing prior to advancing to the system development phase, which includes system design, model training, and system implementation [10]. Upon the completion of system development, testing and assessment are conducted on both the model and the entire

system. The process culminates with analysis and debate to interpret the acquired results.

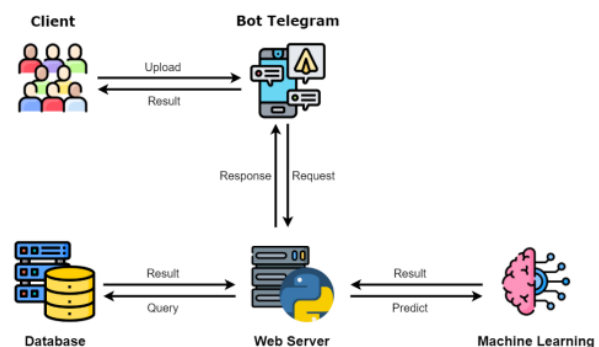


Figure 2. Software Mechanism

Commencing with data collection in the form of image data, the subsequent stage involves preprocessing the gathered data, which encompasses data selection, object annotation utilising bounding boxes and polygon masks for segmentation, and data augmentation to enhance data variability [11]. The dataset is partitioned into training data for model training, validation data for performance monitoring and overfitting prevention, and testing data for assessing model correctness on novel data. Subsequent to the data partitioning procedure, the model training phase is executed utilising the training dataset alongside validation data to guarantee optimal learning and to mitigate overfitting [12]. Figure 2 illustrates the software mechanism for processing the data.

Upon completion of the training phase, the model is assessed using the testing data to determine its efficacy in detecting and segmenting food objects in novel photos. Should the evaluation results be good or optimal, the model may be preserved for deployment into the system. Should the evaluation findings prove unsatisfactory or suboptimal, the model must be reassessed and retrained using enhancements to the dataset or training parameters [13].

This study included manual annotation via CVAT to achieve enhanced precision. The annotated data were subsequently exported in COCO format and transformed into CSV format for model training purposes. The subsequent phase entailed data preprocessing through the selection of pertinent features, including object area, height, and width. The dataset was later partitioned into three subsets: training data, validation data, and testing data [14].

Subsequent to the data partitioning, hyperparameter optimisation was performed to ascertain the ideal parameter configuration for the trained model. The model was subsequently trained using the training data and assessed with the testing data. If the evaluation findings were inadequate, enhancements were implemented to the dataset or hyperparameters were recalibrated until suitable performance was attained [15].

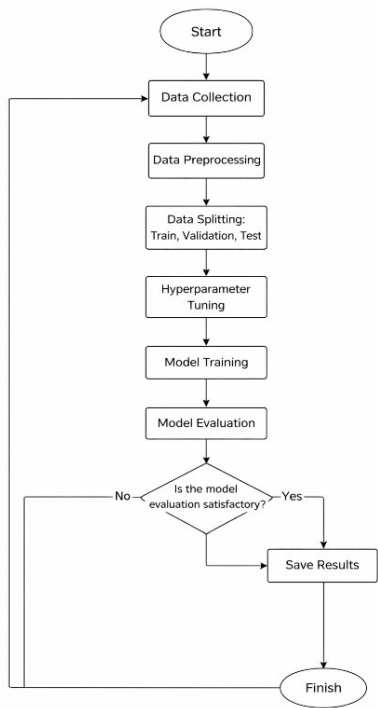


Figure 3. Learning Flowchart

Figure 3 depicts the process of training an XGBoost model to forecast meal weight utilising features extracted from food image annotations. This study involved manual annotations utilising CVAT to achieve enhanced precision. The annotated data were subsequently exported in COCO format and transformed into CSV format for model training. The subsequent phase entailed data preprocessing through the selection of pertinent features, including object area, height, and width. The dataset was later partitioned into three subsets: training, validation, and testing. Following the data partitioning, hyperparameter optimisation was performed to ascertain the ideal parameter configuration for the model. The model was subsequently trained on the training data and assessed with the test data. If the evaluation findings were inadequate, enhancements to the data or additional hyperparameter tuning were conducted iteratively until satisfactory performance was attained. The results were ultimately preserved for utilisation in the established system.

III. ANALYSIS

A. Preprocessing Image Data

This phase encompasses annotation, normalisation, augmentation, dataset partitioning, and ultimately dataset exportation. The execution of each phase is as follows. a. Annotation refers to the process of labelling that occurs subsequent to data gathering. It is conducted to convey information regarding the object's location inside the image and to designate class labels. Annotation is employed to delineate the position and form of food objects within each photograph. A segmentation (polygon) annotation result for a food object is illustrated in the Figure 4.

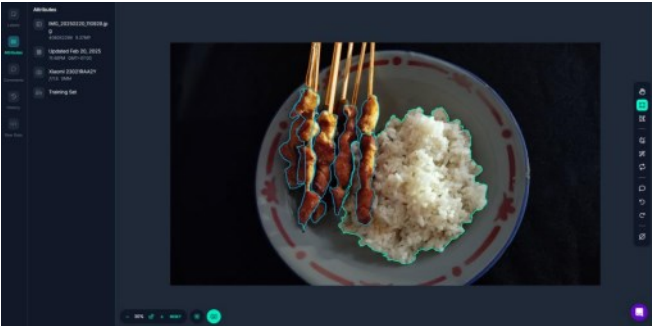


Figure 4. Segmentation Using YoLo

The segmentation annotation method generates an output format that adheres to the YOLO segmentation standard. This format is utilised to retain data regarding the object class and the object's shape within the image. The resultant structure comprises the object class index succeeded by the coordinates of points (x, y) that delineate a polygon representing the object boundaries. The coordinate values are expressed in relative terms about the image resolution, therefore ranging from 0.0 to 1.0. This format pertains to the official Ultralytics documentation accessible at <https://docs.ultralytics.com/datasets/segment/>. The example of the annotation result is as follows (Figure 5):

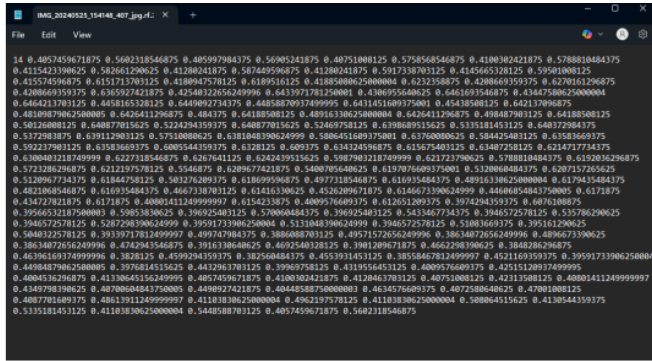


Figure 5. Annotation Results

In contrast to the segmentation employed for annotating food objects, the annotation format for object detection data utilises bounding boxes labelled with the class name "foods." This annotation pertains to photographs of food containers or serving dishes. A bounding box annotation result for a food container object is illustrated in the Figure 6.

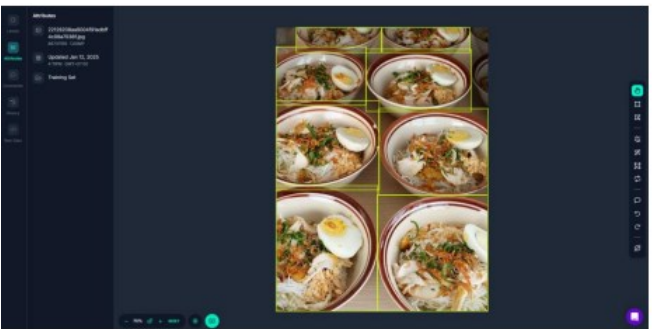


Figure 6. Detection Object Using Bounding Box

B. Object Detection Using YOLOv8m for Training

The food container object detection model was trained utilising the YOLO framework with the dataset obtained during the image preparation phase. The variant employed in this training phase was YOLOv8m, specifically tailored for object detection. The training focused on equipping the model to identify and locate food container objects, which then acts as a basis for partitioning meal portions, allowing for the individual segmentation of each portion. The training procedure encompassed the optimisation of hyperparameters, including the number of epochs, image dimensions, and batch size. The model exhibited superior efficacy in identifying food container items according to the training outcomes. The validation metrics, comprising 184 pictures and 401 objects inside a single class, yielded a mAP50 (Mean Average Precision at IoU > 0.5) of 0.986 and a mAP50–95 of 0.966. These statistics demonstrate that the model can identify items with exceptional precision and a minimal localisation error rate.

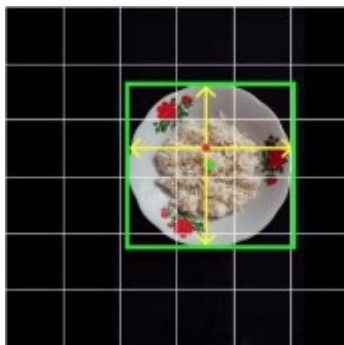


Figure 7. Center of Bounding Box

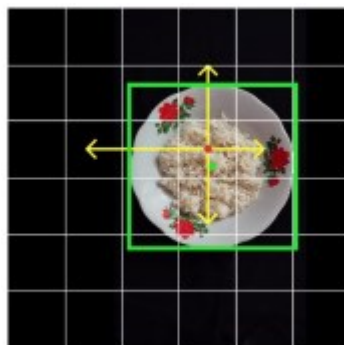


Figure 8. Center to Bounding Box

The training phase commenced with the input of annotated photos including bounding box labels into the model. The backbone processed these images, comprising convolutional layers that extract visual elements including edges, textures, patterns, colours, and other pertinent qualities. Every convolutional layer generates feature maps. The feature maps are subsequently transmitted to the neck, which serves to refine and amalgamate multi-scale characteristics extracted by the backbone. The enhanced feature maps are further processed in the brain. At this level, the model executes anchor-free center point prediction on grids with objects, as depicted in Figure 7,

and calculates the distances from the center point to the four edges of the bounding box, as demonstrated in Figure 8. The head produces object class predictions and confidence scores for every detection. The forecasts from the model are subsequently juxtaposed with the actual labels, as depicted in Figure 7, to compute the loss value. This loss is utilised to adjust the model weights, hence enhancing detection performance with each iteration. During the training process, model efficacy is assessed with validation data to avert overfitting. The training procedure employed twin T4 GPUs from Kaggle Notebook to enhance computational speed. The outcomes of the model training are illustrated in Figure 9.

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Mask(P)	R	mAP50	mAP50-95
all	682	3116	0.912	0.819	0.874	0.793	0.878	0.82	0.855	0.734
Ayam Goreng	85	163	0.929	0.718	0.883	0.717	0.874	0.712	0.823	0.607
Bumby sate	99	221	0.698	0.439	0.504	0.336	0.683	0.498	0.55	0.381
Hortimun	48	106	0.563	0.255	0.382	0.158	0.512	0.297	0.265	0.14
Nasi Goreng	114	312	0.956	0.908	0.968	0.92	0.944	0.92	0.961	0.982
Nasi Goreng	76	92	0.959	0.75	0.832	0.775	0.954	0.783	0.845	0.759
Rendang	90	90	0.935	0.963	0.985	0.932	0.925	0.967	0.985	0.986
Sate	85	90	0.993	0.967	0.986	0.929	0.999	0.978	0.991	0.983
Seperempat Telur Rebus	182	542	0.923	0.774	0.888	0.874	0.906	0.78	0.839	0.533
Soto	34	61	0.885	0.723	0.87	0.819	0.88	0.754	0.87	0.799
Telur Ceplok	82	106	0.986	0.915	0.95	0.921	0.895	0.915	0.945	0.985
Telur Rebus	185	188	0.996	0.981	0.995	0.971	0.938	0.926	0.949	0.953
Tempo goreng	33	77	0.967	0.961	0.987	0.952	0.943	0.961	0.987	0.968
Tusuk sate	68	68	0.965	0.983	0.975	0.961	0.96	0.983	0.975	0.949
Tusuk sate	38	34	0.982	0.971	0.981	0.954	0.975	0.971	0.981	0.957
Tusuk sate	33	99	1	0.948	0.978	0.955	0.982	0.949	0.978	0.953
Tusuk sate	83	164	0.945	0.982	0.953	0.878	0.921	0.896	0.94	0.853
Tusuk sate	99	791	0.896	0.757	0.865	0.835	0.877	0.856	0.847	0.29

Figure 9. Result Model Yolov8m-seg

The testing phase aimed to assess the efficacy of the YOLOv8m-seg model in segmenting food articles within containers from photographs submitted by users via the Telegram Bot. Initially, the researcher assessed the model using individual food items within a single container for each category analysed. Subsequently, research was conducted by amalgamating several food items within a single container.

Table 1. Testing Scenario

Segmentation Image	Food Detection	Accuracy
	Fried Chicken	95%
	Fried Rice	97%

	Satay	93%
	Satay with sauce	Undetected
	Rice	94%
	Boiled Egg	80%

The test findings (Table 2) indicate that the model excelled at identifying the majority of food things. Items including fried rice, fried tofu, fried tempe, half-boiled egg, cucumber, and tomato attained exceptional segmentation accuracy (97–98%). Additional items, such as fried chicken, shredded chicken, rendang, unsauced satay, rice, quarter-boiled eggs, and fried eggs, were also categorised with high precision (92–96%). The model performed optimally on foods exhibiting distinct shapes and substantial contrast with the background. Nonetheless, performance diminished for whole cooked eggs (80%) and soto (79%) due to their colours resembling the background or their boundaries being indistinct. Satay with sauce was undetected, presumably due to occlusion or insufficient training data. The overall average accuracy attained was 87.68%, signifying commendable general performance.

In a separate testing scenario utilising XGBoost, the assessment concentrated on forecasting food weight based on segmentation features generated by YOLOv8m-seg. The identical photos from the prior test were utilised, and the exact weights of the meals had been measured in advance. The anticipated weights were juxtaposed with the actual values to compute errors utilising MAE and RMSE. Certain foods, like

soto and egg-based products, were omitted as their nutritional qualities are assessed by portion or unit rather than weight estimation.

Table 2. Object Test Result

Food Object	Real Weight (gram)	Weight Prediction (gram)	Error (gram)
Fried Chicken	36.9	54.21	+17.31
Fried Rice	383.9	227.93	- 155.97
Chicken Suwir	40.9	41.73	+ 0.83
Rendang	45.3	49.53	+ 4.23
Satay	7.2	8.35	+ 1.15
Rice	105.9	126.28	+ 20.38
Fried Tofu	46.9	51.45	+ 4.55
Fried Tempe	24.6	28.73	+ 4.13
Cucumbar	7.9	9.93	+ 2.03
Tomato	11.9	12.70	+ 0.8

Using segmentation features taken from images uploaded by users through the Telegram Bot, the XGBoost model was tested to see how well it could predict the weight of each food item. Three regression metrics were used to evaluate the results: MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and R^2 (R-Squared).

MAE (Mean Absolute Error) is calculated using the following Equation 1:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

$$MAE = \frac{17.31 + 155.97 + 0.83 + 4.23 + 1.15 + 20.38 + 4.55 + 4.13 + 2.03 + 0.8}{10} = \frac{211.38}{10} = 21.138$$

RMSE (Root Mean Squared Error) is calculated using the following Equation 2:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of samples :

$$RMSE = \sqrt{\frac{17.31^2 + 155.97^2 + 0.83^2 + 4.23^2 + 1.15^2 + 20.38^2 + 4.55^2 + 4.13^2 + 2.03^2 + 0.8^2}{10}} = \sqrt{\frac{299.64 + 24326.64 + 0.69 + 17.89 + 13225 + 415.34 + 20.70}{10}}$$

$$RMSE = \sqrt{\frac{25510.046}{10}} = \sqrt{2551.04046} \approx 50.10$$

In R^2 (R-Squared) :

$$R^2 = 1 - \frac{25104.046}{116132.164} = 1 - 0.2162 = 0.7838$$

The model had an MAE of 21.138 grams, an RMSE of about 50.10 grams, and a R^2 of 0.7838, according to the calculations. The model did a good job of predicting the weights of shredded chicken, rendang, satay, fried tofu, fried tempeh, cucumber, and tomato, with errors of less than 5 grams. This means that the anticipated weights were extremely near to the actual weights. But there were certain meal groups where the predictions were off by a lot, like fried rice, rice, and fried chicken. The difference between the projected and actual weights was -155.97 grams for fried rice, +20.83 grams for rice, and +17.31 grams for fried chicken. The big mistake in predicting fried rice is probably because of differences in density and the types of ingredients utilised, especially how much oil is used when cooking. So, even though the segmentation features may look like the training data, they don't entirely show the real weight because the materials are different.

The forecast error for rice is also rather large because rice might have different densities, moisture levels, and serving styles, including soft rice, dry rice, or newly cooked rice. These aspects cannot be fully represented by visual cues such as segmented object area, resulting in diminished prediction accuracy. The main reason for the mistake in predicting the weight of fried chicken is that the object is not shaped regularly, thus the segmented area may not entirely cover the whole chicken part or may have constricted areas. This irregularity makes it harder to accurately segment and estimate weight. The researcher finds that the model still has limits when forecasting products with irregular shapes, variable ingredient composition, moisture content, and density based on these data. Along with employing regression metrics like MAE, RMSE, and R^2 , we also employed a dataset split of 80:20, with 80% for training and 20% for testing, to get a better idea of how well the model worked.

C. Implementation

In the system implementation of this research, Python was used to develop all processes within the system. These processes include system integration with the Telegram Bot, data processing, object detection, object segmentation, weight prediction, and nutritional estimation. To ensure the system can be managed effectively, the components were organized into a directory structure based on their respective functions.

The system executes steps that involve getting the highest-resolution image from the Telegram Bot API. The image undergoes processing involving object detection, segmentation, food weight prediction, and nutritional estimate based on the anticipated weight of the food items. Upon completion of all operations, the system transmits the results

to the user via the Telegram Bot, presenting photos and text that encompass the identified and segmented objects, together with the estimated nutritional information derived from the expected meal weight. The system is implemented using a Telegram chatbot, as shown in the following Figure 10.

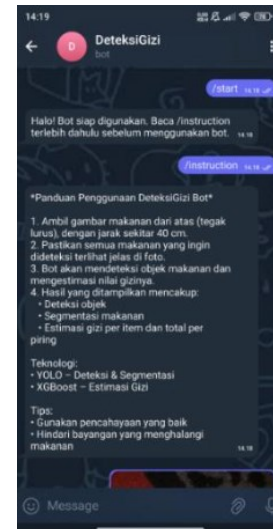


Figure 10. Manual on Telegram

Figure 10 illustrates the usage instructions of the application implemented through a chatbot-based interface. By utilizing a bot model, the system provides interactive guidance that helps users understand how to operate the application effectively. This approach is expected to assist users in accessing the system features more easily, allowing them to interact with the application in a simple and intuitive manner.

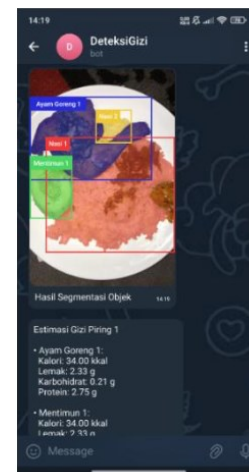


Figure 11. Object Detection

Figure 11 presents the result of the object detection process used to identify individual food items in the image and estimate the calorie content of each detected object. Through this process, the system first detects and segments each food item within the container, such as fried chicken, rice, and vegetables. After the objects are successfully identified, the system calculates the estimated calorie content based on the

predicted weight of each food item. This result is then displayed to the user through the Telegram chatbot interface, providing both visual detection results and nutritional information for each detected object.



Figure 12. Object Detection Result

The detection results are presented in detail to the user through the Telegram chatbot interface. As shown in Figure 12, the system provides a detailed nutritional estimation for each detected food object, including calorie content, fat, carbohydrates, and protein. These values are calculated based on the predicted weight of each food item obtained from the model. In addition to individual nutritional information, the system also summarizes the total nutritional values for all detected food items in the container. This detailed output allows users to easily understand the nutritional composition of their meal.

IV. CONCLUSION

Based on the design, development, and testing processes carried out, this research successfully built an integrated food nutrition estimation system by leveraging a combination of YOLOv8 and XGBoost methods. Food object detection and segmentation were performed using two separate models, namely YOLOv8m for detecting food containers with an average accuracy of 88.1% and YOLOv8m-seg for food object segmentation with an accuracy of 91.92%. Both models were trained through a series of stages including data collection, object annotation covering class and location information, model training, and evaluation using new data. The best performance was achieved on images containing one to two objects within a single frame, while accuracy tended to decrease under conditions involving more than two objects or objects with low contrast against the background.

The segmentation results from YOLOv8m-seg were subsequently utilized as the primary features in the XGBoost model to predict food weight, which then served as the basis for automatic nutritional content estimation. The XGBoost model yielded a MAE of 21.138 grams, an RMSE of 50.10 grams, and an R^2 value of 0.7838, demonstrating strong predictive performance on food items such as shredded

chicken, rendang, satay, fried tofu, fried tempeh, cucumber, and tomato with an error margin of less than 5 grams. However, significant prediction errors were observed for fried rice, plain rice, and fried chicken, attributed to their irregular shapes and varying densities that could not be adequately represented through visual segmentation features alone. Overall, the system was successfully integrated and validated through black-box testing, in which all key features performed as expected and were fully accessible via a Telegram Bot.

REFERENCES

- [1] R. Aslam, S. R. Sharma, J. Kaur, A. S. Panayampadan, and O. I. Dar, "A systematic account of food adulteration and recent trends in the non-destructive analysis of food fraud detection," *J. Food Meas. Charact.*, vol. 17, no. 3, pp. 3094–3114, Jun. 2023, doi: 10.1007/s11694-023-01846-3.
- [2] G. C. Utami, C. R. Widiawati, and P. Subarkah, "Detection of Indonesian Food to Estimate Nutritional Information Using YOLOv5," *Teknika*, vol. 12, no. 2, pp. 158–165, Jun. 2023, doi: 10.34148/teknika.v12i2.636.
- [3] G. I. Nugraha *et al.*, "The urgency in proposing the optimal obesity cutoff value in Indonesian population: A narrative review," *Medicine (Baltimore)*, vol. 101, no. 49, p. e32256, Dec. 2022, doi: 10.1097/MD.00000000000032256.
- [4] S. Oktaviani, M. Mizutani, R. Nishide, and S. Tanimura, "Factors associated with overweight/obesity of children aged 6–12 years in Indonesia," *BMC Pediatr.*, vol. 23, no. 1, p. 484, Sep. 2023, doi: 10.1186/s12887-023-04321-6.
- [5] "Enhancing Nutritional Predictive Models: Addressing Class Imbalance with Machine Learning," *J. Technol. Humanit.*, vol. 6, no. 1, pp. 22–32, Jun. 2025, doi: 10.53797/jthkks.v6i1.3a.2025.
- [6] R. Ramadhansyah, S. Simatupang, and R. Abdillah, "Development of a YOLO-Based Artificial Intelligence (AI) System for Early Detection of Stunting Risk in Children in 3T Regions of North Sumatra," *J. Comput. Netw. Archit. High Perform. Comput.*.
- [7] A. P. Saputri, A. Taqwa, and S. Soim, "Analisis Deteksi Objek Citra Digital Menggunakan Algoritma YOLO Dan Cnn Dengan Arsitektur Repvgg Pada Sistem Pendeteksian dan Pengenalan Ekspresi Wajah".
- [8] P. Nani, S. Das, and S. Dey, "Enhancing object recognition: a comprehensive analysis of CNN based deep learning models considering lighting conditions and perspectives," *Evol. Intell.*, vol. 18, no. 4, p. 72, Aug. 2025, doi: 10.1007/s12065-025-01061-7.
- [9] R. Afrinanda, W. Tawa Bagus, and L. Efrizoni, "Comparison of Machine Learning Algorithm Models in Bitcoin Price Sentiment Analysis," *Indones. J. Comput. Sci.*, vol. 12, no. 2, pp. 502–513, Apr. 2023, doi: 10.33022/ijcs.v12i2.3180.

- [10] N. Sulistianingsih and G. H. Martono, "Analysis of the Effectiveness of Traditional and Ensemble Machine Learning Models for Mushroom Classification".
- [11] Asrul Abdullah, M. Iwan, and S. Rama Dani, "Applying Tree Based Model for Crop Recommendation System Based on Soil Parameters And Weather Conditions," *JITK J. Ilmu Pengetah. Dan Teknol. Komput.*, vol. 11, no. 1, pp. 228–235, Aug. 2025, doi: 10.33480/jitk.v11i1.6476.
- [12] M. Sakmar, N. T. Kadir, P. A. Shofo, and A. Darmawan, "Efektivitas Xgboost Lightgbm dan Catboost Pada Dataset Imbalanced Predictive Maintenance," vol. 3, 2026.
- [13] H. Wu, Q. Liu, and X. Liu, "A Review on Deep Learning Approaches to Image Classification and Object Segmentation," *Comput. Mater. Contin.*, vol. 60, no. 2, pp. 575–597, 2019, doi: 10.32604/cmc.2019.03595.
- [14] M. Farooqui *et al.*, "Food Classification Using Deep Learning: Presenting a New Food Segmentation Dataset," *Math. Model. Eng. Probl.*, vol. 10, no. 3, pp. 1017–1024, Jun. 2023, doi: 10.18280/mmep.100336.
- [15] Usha Ruby Dr.A, "Binary cross entropy with deep learning technique for Image classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020, doi: 10.30534/ijatcse/2020/175942020.