Detection of Indonesian Food to Estimate Nutritional Information Using YOLOv5

Gina Cahya Utami[1], Chyntia Raras Ajeng Widiawati[2], Pungkas Subarkah[3*]

[1,3] Department of Informatics, Universitas Amikom Purwokerto, Purwokerto, Central Java
[2] Department of Information Technology, Universitas Amikom Purwokerto, Purwokerto, Central Java
Email: [1] ginacu12@gmail.com, [2] chyntiaraw@amikompurwokerto.ac.id, [3*] subarkah@amikompurwokerto.ac.id

(Naskah masuk: 25 Mei 2023, direvisi: 10 Jun 2023, diterima: 15 Jun 2023)

Abstract

Currently, the development of online food delivery service applications is very popular. The application offers convenience in finding and fulfilling food needs. That circumstance has an impact such as not controlling the type and amount of food consumed. Therefore, to maintain a healthy lifestyle, people need to eat healthy and nutritious food. The goal of this research is to build a model using the YOLOv5 model that can detect images of Indonesian food so that nutritional estimation can then be carried out by taking information per serving data sourced from the FatSecret Indonesia website. The methods of this research include data collection, data pre-processing, training, testing, evaluation, image detection, and model export. The outcome of this research is an object detection model that is ready to be implemented in android applications or websites to detect images of Indonesian food which can be estimated for each nutrient. Based on the detection results, 98.6% for an average of a curacy, 95% for precision, 95.3% for recall, and 95% for F1-Score were obtained. The results of the detection are then used to estimate nutrition by taking information per portion from the FatSecret Indonesia website. From the experiments that were carried out on seven pictures of Indonesian food, the estimation was carried out well by displaying various nutritional information including energy, protein, fat, and carbohydrates.

Keywords: YOLOv5, Object Detection, Nutritions.

I. INTRODUCTION

The Industrial Revolution 4.0 has brought about technological advancements that have made it easier for people to find and meet their needs. One of them concerns the need for food [1]. Currently, the convenience of online food delivery service applications is the answer to finding and meeting people's food needs. People sometimes encounter conditions such as feeling lazy, doing piles of work, or avoiding traffic jams, so they prefer to order food using an online food delivery service application [2]. Research by Nielsen Media Research Singapore in 2019 shows that Indonesia is a large market for food delivery services, especially during the COVID-19 pandemic, this is evidenced by the many food delivery services such as GoFood and ShopeeFood [3].

This convenience will have an impact on the type and amount of food consumed to become uncontrolled. Research [4] explains that the increase in food consumption is quite large due to the use of social media. Lack of information regarding nutritional needs as well as nutrition can also lead to excess nutritional status such as obesity. In essence, the average energy adequacy rate for Indonesian people is 2,100 kcal and for protein, adequacy is 57 grams per person per day at the consumption level [5]. This education is sometimes still not understood by the community so people still often eat food carelessly. The solution to this issue is to develop a model that can recognize the type of food and then estimate the nutritional information.

Several studies have started to develop systems regarding food detection and then estimate its nutrition. One of them is research by [6] building an application to detect food that was captured and then display the item's nutritional value using YOLOv3. According to the study's findings, an accuracy of 100% for images with only one object, 88% for images with three objects, and 68% for images with five objects. From this study, it can be concluded that the more objects in an image, the less accuracy will decrease.

A system to assist users in controlling their eating behaviors through food detection and calorie analysis was proposed in prior research titled "Food Calorie and Nutrition Analysis System based on Mask R-CNN" by [7]. This research involved the detection of food using the Mask R-CNN, analysis of the food proportion using a food mask, and

DOI: 10.34148/teknika.v12i2.636

the weight was determined using linear regression. The food's nutritional content can be examined using the estimated weight. The Food-256 dataset and the Ville café dataset were used in this research, and the accuracy was 99.86%.

Another study [1] regarding building applications for recognizing fruit objects, including apples, grapes, pears, oranges, and bananas using YOLOv3 resulted in an accuracy of 97.4%. In this study, there was a case of overfitting because the data used as testing data was training data.

Research [8] regarding food detection using the Single Shot Detector (SSD) in the android application. The model can detect food on a plate and then estimate the number of calories by taking calorie information per serving from the FatSecret website.

Research on the detection of mold in food was carried out by [9] comparing the detection models using YOLOv3, YOLOv4, and YOLOv5. The results of this study are YOLOv5 managed to outperform in a precision, recall, and average precision (AP) score with respective scores of 98.10%, 100%, and 99.60%. Researchers suggest increasing the number of images to improve better results.

The study [10] compares different object identification and localization algorithms based on accuracy, execution time, and parameter values with different input image sizes. This study has discovered a novel single-stage model methodology that increases speed without significantly reducing accuracy. The comparing findings demonstrate that YOLOv3-Tiny accelerated object detection while retaining result accuracy.

In the study, the YOLO algorithm is also utilized to detect prostate cancer and determine important regions on input biopsy images automatically in the research [11]. The first set contains 50 actual photos of prostate tissue that are comparable to the train set and have a 97% accuracy rate. The accuracy rate for the test set, which comprises 137 fully different actual prostate tissue biopsy photos, was 89%. The misdetected and misclassified cases can be found in four of the test images. Data augmentation is the primary cause of poor detection and classification. Although the augmented image data increases the system's accuracy, it can also lead to errors in detection and classification. Nevertheless, according to the test results, tools for diagnosing prostate cancer with high accuracy can be created utilizing artificial intelligence techniques such as object detection algorithms.

YOLO is also utilized for detecting fruit as done by this research [12]. This research has developed the YOLO-Tomato, a more accurate and compact version of the YOLOv3 tomato detection model. Compared to other state-of-the-art detection techniques, the YOLO-Tomato performed the best.

Another study [13] suggests a better method for dealing with mask-wearing detection based on YOLO-v4. The results demonstrate that the accuracy of mask detection in this study is higher than in other algorithms. Regardless, the situation of false and missed detection has evolved. Furthermore, the algorithm presented in this paper reduces the model's training costs and complexity requirements, allowing the algorithm to be extended to other object detection tasks like the detection of mask wear by staff, passengers, students, and other patients in addition to being deployed on medium devices.

The research [14] has introduced LabelStoma, an application for measuring stomatal density. LabelStoma uses the YOLO algorithm to detect stomata and has 0.91 in F1-Score when tested on photos from the species used for training. This application will contribute to a better understanding of processes related to plant gas exchange as well as the carbon and water cycles.

This study [15] created a deep learning target identification method called YOLO-SASE based on the YOLO algorithm to enhance the ability to detect infrared tiny targets in complex backgrounds. The algorithm proposed in this study's experiment had accuracy and recall rates that were 2% and 3% higher than those of the original model, and the stability of the results was substantially better than during training.

This research will build a model for detecting Indonesian food using YOLOv5, which is known to be a very good object detection model recently. This is proven in research [16] about the development of all five versions of the YOLO and can be concluded that the last version of YOLO, YOLOV5, performs better than YOLOv4 in terms of speed and accuracy. After food has been successfully detected, the nutritional estimation will then be carried out by taking information per serving sourced from the FatSecret Indonesia website.

II. RESEARCH METHOD

The research process consisted of seven stages, including data collection, data pre-processing, training, testing, evaluation, image detection to display nutritional information, and model export to a file with the .pt extension. The process is presented in Figure 1.

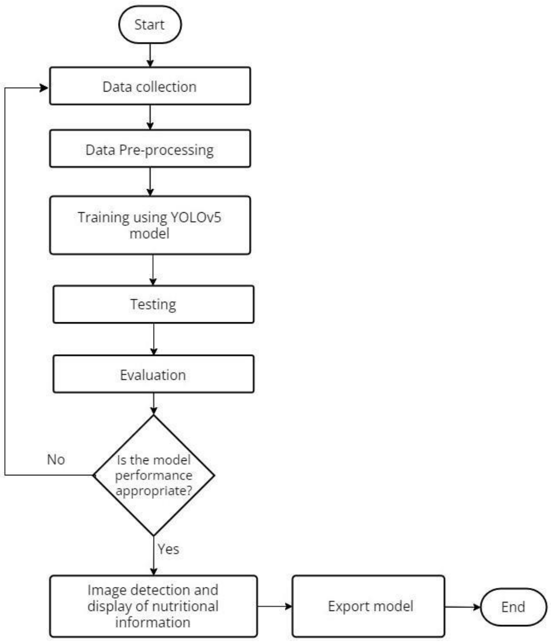

Start
Data collection
Data Pre-processing
Training using YOLOv5 model
Testing
Evaluation
No
Is the model performance appropriate?
Yes
Image detection and display of nutritional information
Export model
End

Figure 1. Research Flow

The first process in this research was data collection, both data in the form of images and nutritional information. Data collection is in the form of image datasets sourced from Kaggle, Roboflow, and Google Images sites. As for the nutrition dataset, it is sourced from the FatSecret Indonesia website. Indonesian food objects or classes used in this study include bakso kuah, batagor kering, gado-gado, nasi goreng, pempek, sate ayam, and soto Lamongan.

The data pre-processing consists of image annotation, splitting image datasets, resizing images, and image augmentation. Here is the explanation:

1. Image annotation is the stage of class labeling and bounding boxes on the image. The result of this process is a file with the extension .txt [17]. The file contains object coordinates which consist of object classes that are represented by numbers, x coordinates, y coordinates, width, and object height [18].
2. Split image datasets. The dataset must then be divided into training, validation, and testing data after the image has been annotated.
3. Resize the image. This process aims to uniform image sizes [19]. This process is carried out to reset the image size so that it can be adjusted to the size read by the model [20].
4. Image augmentation. According to Perez & Wang, the image augmentation process will change and modify the image so that the computer detects that the image being changed is a different image, but humans can still know that the image being changed is the same image [21]. This process is carried out on the training data by cutting the image into certain parts, rotating, brightening, providing exposure, and making the image into a collage or mosaic.

Training is the process of using several prepared training data [20]. The model used in this research is YOLOv5. Validation data is also used in this process, this data aims to test accuracy when carrying out the training process [22].

YOLOv5 can outperform other YOLO models in terms of accuracy and speed [16]. In research conducted by Yan, where robots were given the task of picking apples, the YOLOv5 model was compared to the YOLOv3 and YOLOv4 models. From this study, mAP increased by 14.95% and 4.75%, respectively [9].

The architecture of YOLOv5 was created using PyTorch and divided into three parts, namely the backbone, neck, and head as shown in Figure 2. On the backbone, this version uses the Focus and CSPDarknet5 structures which are to reduce the required CUDA memory, reduce layers, and increase forward propagation, and backpropagation. The neck component uses PANet which adopts the Feature Pyramid Network (FPN) which includes several bottom-top and top-bottom layers. This increases the propagation of low-level features in the model. The last component in this architecture, namely the head, uses the same components as in the fourth and third versions which produce three different feature map outputs to achieve multiscale prediction. This helps improve the prediction of small to large objects efficiently in the model.

An explanation of the YOLOv5 architecture presented in Figure 2 includes the focus layer on the backbone evolving from the YOLOv3 structure, which previously had three layers to become one layer in this fifth version. Conv here shows layer convolution. C3 consists of three convolution layers and modules that flow by various bottlenecks or barriers. SPP is a pooling layer that is used to remove fixed network size limitations. Upsample is used in upsampling the previous layer's fusion at the closest node. Concat is a slicing layer and is used to slice the previous layer. The third Conv2d is a detection module that is used on the head.

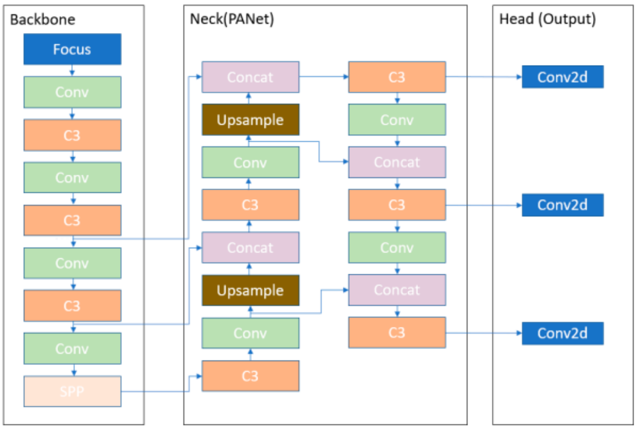


Figure 2. YOLOv5 Architecture

The testing process aims to test/evaluate the model that has been trained using test data. The testing phase will continue with the process of predicting the bounding box for each detected object frame. The output of this process is in the form of pictures of food with the results of objects being detected and a confidence score in detecting the object [23].

The evaluation process is used to evaluate the performance of a model using evaluation metrics such as the confusion matrix, accuracy, precision, recall, and F1-score.

1. Confusion Matrix

The Confusion Matrix is a table that describes model performance on a test data set with known values [24]. The table is shown in Table 1.

Table 1. Confusion Matrix Table

		True Values	
		+	-
Prediction	+	*TP*	*FP*
	-	*FN*	*TN*

Based on Table 1:

a. The amount of positive data images that have been identified as real or positive values is known as True Positives (TP).
b. The amount of negative data images that are mistakenly identified as positive or false values is called a False Positive (FP).

c. The amount of positive data images that are mistakenly or wrongly identified as negative values is called a False Negative (FN)
d. The amount of negative data images that are recognized as being true negatives or negative values is known as True Negatives (TN).

Table 1 shows the accuracy, precision, and recall values contained in the values of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) [25].

2. Accuracy

Based on calculations from Sokolova & Lapalme, the accuracy score describes how accurate and efficient the model as a whole is in detecting data correctly, the accuracy calculation is stated in equation (1) [25].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{1}$$

3. Precision

Precision is the comparison of data images correctly detected as positive by the system and all data detected as positive. Precision is stated in equation (2) [26].

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

4. Recall

A recall is a measurement of data images that are correctly detected as positive by the system divided by the total positive samples [18] [26]. A recall is stated in equation (3).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

5. F1-Score

F1-Score is the result of combining precision and recall computations. The worst F1-Score is 0, while the best F1-Score is 1. A good F1-Score indicates that the resulting model has good precision and recall scores [27]. F1-Score is stated in equation (4) [26].

$$F1\,Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

After the evaluation results on the model achieve the desired results, it can proceed to the next process. However, if the evaluation results have not achieved the expected results, then the process can be repeated from the beginning to improve the process again until the expected evaluation results are achieved.

After the evaluation results are following the expected results, it can proceed to the image detection stage and then display nutritional information. In this process, the model will detect images taken from the training dataset and then the detection results in the form of strings are used to retrieve nutritional information from the FatSecret Indonesia website. The Python programming language is utilized in this research, which is executed in Google Colaboratory.

The next process is exporting or saving the model to local storage using the PyTorch library with the output in the form of a .pt extension file which can then be used for implementation in the form of a website or Android.

PyTorch is an open-source machine learning library utilized for applications such as computer vision. This system was created by Facebook and it was made to supply models that are simpler to compose than other systems such as TensorFlow. The YOLOv5 model is accessible as it were for PyTorch at this point, which is why we used PyTorch for this research [28].

III. RESULT AND DISCUSSION

The first process was data collection. The collection of nutritional information data uses the assumption that the portions of Indonesia's food are from the FatSecret Indonesia website. This information is used to estimate the nutritional value of each object class. The nutritional information includes energy, fat, carbohydrates, and protein.

Food image data collection was sourced from Kaggle, Roboflow, and Google Images sites. The objects or classes of the images include bakso kuah, batagor kering, gado-gado, nasi goreng, pempek, sate ayam, dan soto Lamongan. The total images collected were 2,625 images with 375 images in each class. From the picture, there is only one food object in it.

The data pre-processing consists of these steps:

1. Image Annotations. This stage is carried out using tools from roboflow. The annotation process is shown in Figure 3.

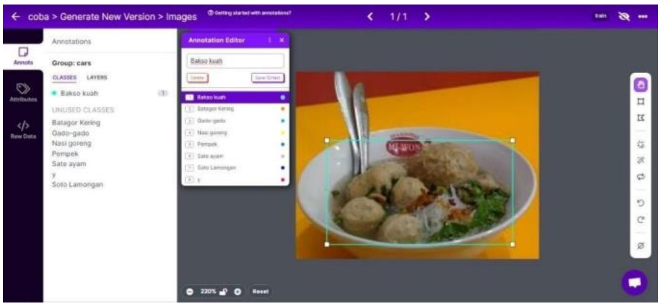
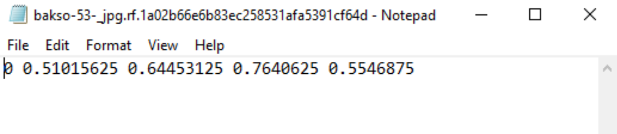


Figure 3. Annotation Process

From this process, a file is generated containing the label coordinates of each image with the .txt extension. The following are the contents of the file.


bakso-53-_jpg.rf.1a02b66e6b83ec258531afa5391cf64d - Notepad

```
0 0.51015625 0.64453125 0.7640625 0.5546875
```

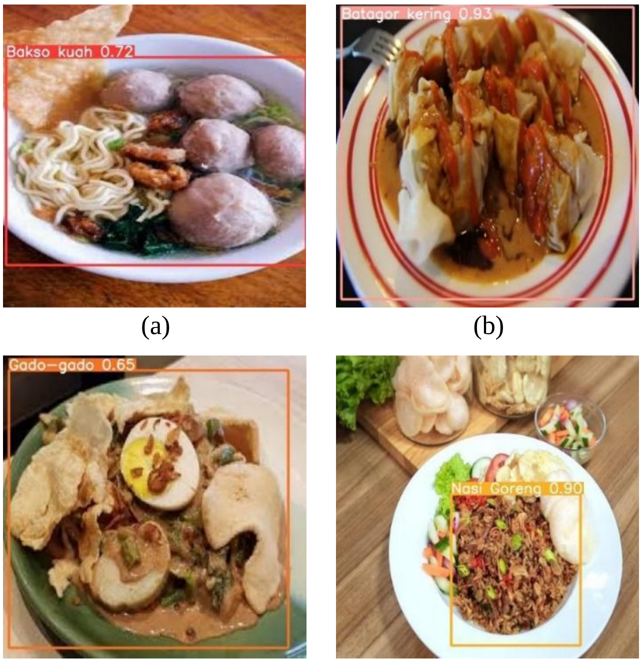
Figure 4. Inside File .txt

Figure 4 contains information about object coordinates consisting of <object-class> <x-center> <y-center> <width> <height>. The value "0" in Figure 4 is the value of <object-class> which represents the class of the bakso kuah object. <x-center> and <y-center> are the center coordinates of the bounding box. Meanwhile, <width> and <height> are the width and height of the object in the image [18].

2. Split Image Dataset. At this point, an 80:10:10 ratio is used to separate the images. This splitting is done using tools from Roboflow. Of the total images, namely 375, resulting in training data of 300 images, validation data of 38, and testing data of 37 for each class.
3. Resizing the image. The size of the image used before the training process is 640 x 640 pixels.
4. Image Augmentation. The augmentations used in this process include crop, rotation, brightness, exposure, and mosaic. The augmentation process is only carried out on training data, so that the training data, which was originally 300 images per class, becomes 900 because each class will be added two types of augmentation techniques to add variety to the images. From this process, the initial 2.635 images increased to 6.825.

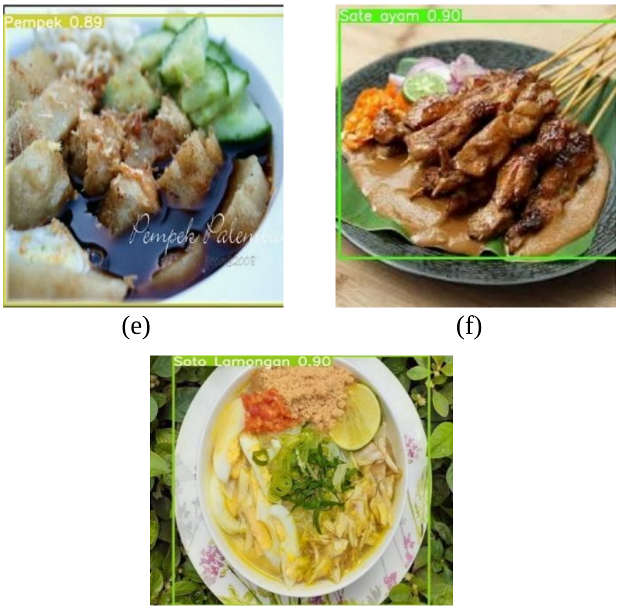
The training process is using YOLOv5. This process is carried out using the training data that we have carried out the pre-processing process first so that the training process runs smoothly. This process is executed on the Google Colaboratory platform using the PyTorch library. This process uses 150 epochs and takes about 3 hours and 39 minutes.

The testing process is carried out using test data. The following image shown in Figure 4 is a sample image of detection results using test data.



(a) (b)

(c) (d)



(e) (f)

(g)

Figure 5. Results of The Testing Process. (a) Bakso Kuah, (b) Batagor Kering, (c) Gado-Gado, (d) Nasi Goreng, (e) Pempek, (f) Sate Ayam, and (g) Soto Lamongan.

From the several samples shown in Figure 5, a bounding box is obtained that frames the detected objects as well as the confidence score of the model in detecting each of these samples. When detecting bakso kuah, a confidence score obtained 72% for recognizing objects as bakso kuah. For batagor kering detection, the model recognizes the object with 93% confidence. In the gado-gado object, the confidence score obtained at 65%. Meanwhile, the nasi goreng gets a score of 90%. For the pempek object, the confidence score obtained at 89%. The sate ayam object obtained a score of 90%. And for the last object, namely soto Lamongan, the confidence score is obtained at 90%.

When the testing procedure is completed successfully, the following step is to evaluate the results using the confusion matrix to determine the evaluation's accuracy, precision, recall, and F1-Score. The following is the evaluation result of each class.

Table 2. Results of the Evaluation Process

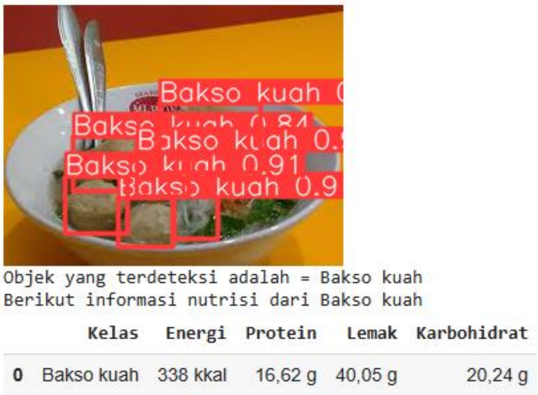
No	**Class**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
1.	Bakso kuah	99.19%	0.98	0.98	0.98
2.	Batagor kering	98.78%	1.00	0.89	0.94
3.	Gado-gado	97.56%	0.92	0.92	0.92
4.	Nasi goreng	98.37%	0.94	0.94	0.94
5.	Pempek	98.37%	0.93	0.97	0.95
6.	Sate ayam	98.37%	0.95	1.00	0.97
7.	Soto Lamongan	99.59%	0.97	1.00	0.98
Average		98.61%	0.95	0.96	0.95

DOI: 10.34148/teknika.v12i2.636

Based on Table 2, the highest accuracy value is obtained in the Soto Lamongan class with a value of 99.59%. The highest precision value is found in the batagor kering class with a value of 1.00. The highest recall values were found in the sate ayam with and Soto Lamongan classes, each with a value of 1.00. And the highest F1-score is in the Bakso kuah and Soto Lamongan classes with each value of 0.98.

The conclusion from evaluation results in Table 2, average accuracy value for the entire class of this study was 98.61%, the average precision value was 0.95, the average recall value was 0.96, and the average F1-Score value of 0.95.

After the authors tested the model using a nutritional dataset, the next process was to display nutritional information on each food that was successfully detected. The results of this process are shown in Figure 6.

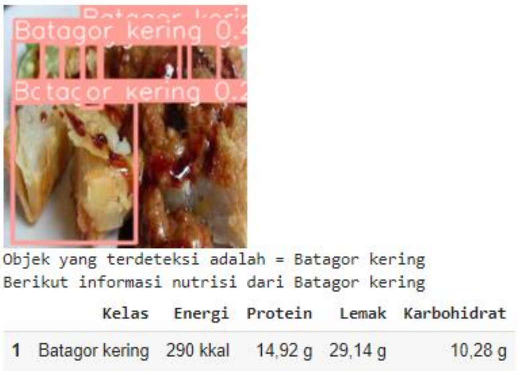


Objek yang terdeteksi adalah = Bakso kuah
Berikut informasi nutrisi dari Bakso kuah

	Kelas	Energi	Protein	Lemak	Karbohidrat
0	Bakso kuah	338 kkal	16,62 g	40,05 g	20,24 g

Figure 6. Food Detection and Nutritional Information of Bakso kuah

Based on Figure 6, the model was successful in detecting the bakso kuah image and estimating its nutrition based on nutritional information from FatSecret Indonesia's website. The nutrients contained in bakso kuah include energy of 338 kcal, protein of 16.62 g, fat of 40.05 g, and carbohydrates of 20.24 g.

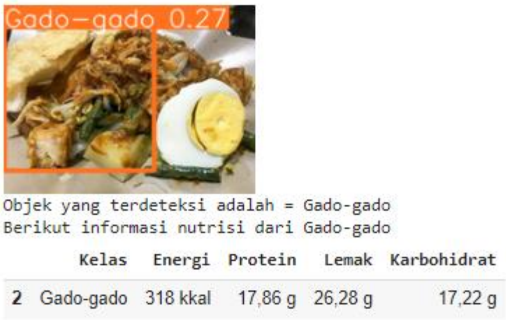


Objek yang terdeteksi adalah = Batagor kering
Berikut informasi nutrisi dari Batagor kering

	Kelas	Energi	Protein	Lemak	Karbohidrat
1	Batagor kering	290 kkal	14,92 g	29,14 g	10,28 g

Figure 7. Food Detection and Nutritional Information of Batagor Kering

Based on Figure 7, the model was successful in detecting the batagor kering image and estimating its nutrition. The nutrients contained in batagor kering include energy of 290 kcal, protein of 14.92 g, fat of 29.14 g, and carbohydrates of 10.28 g.

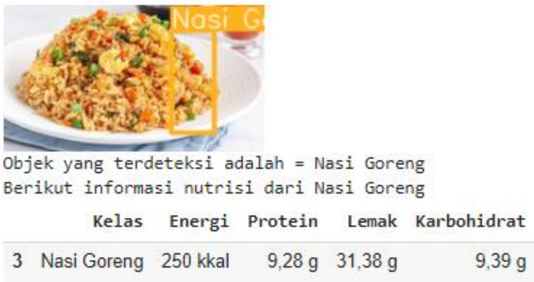


Objek yang terdeteksi adalah = Gado-gado
Berikut informasi nutrisi dari Gado-gado

	Kelas	Energi	Protein	Lemak	Karbohidrat
2	Gado-gado	318 kkal	17,86 g	26,28 g	17,22 g

Figure 8. Food Detection and Nutritional Information of Gado-gado

Based on Figure 8, the model was successful in detecting the gado-gado image and estimating its nutrition. The nutrients contained in gado-gado include energy of 318 kcal, protein of 17.86 g, fat of 26.28 g, and carbohydrates of 17.22 g.

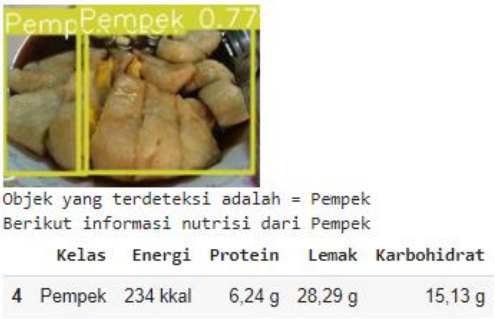


Objek yang terdeteksi adalah = Nasi Goreng
Berikut informasi nutrisi dari Nasi Goreng

	Kelas	Energi	Protein	Lemak	Karbohidrat
3	Nasi Goreng	250 kkal	9,28 g	31,38 g	9,39 g

Figure 9. Food Detection and Nutritional Information of Nasi Goreng

Based on Figure 9, the model was successful in detecting the nasi goreng image and estimating its nutrition. The nutrients contained in nasi goreng include energy of 250 kcal, protein of 9.28 g, fat of 31.38 g, and carbohydrates of 9.39 g.

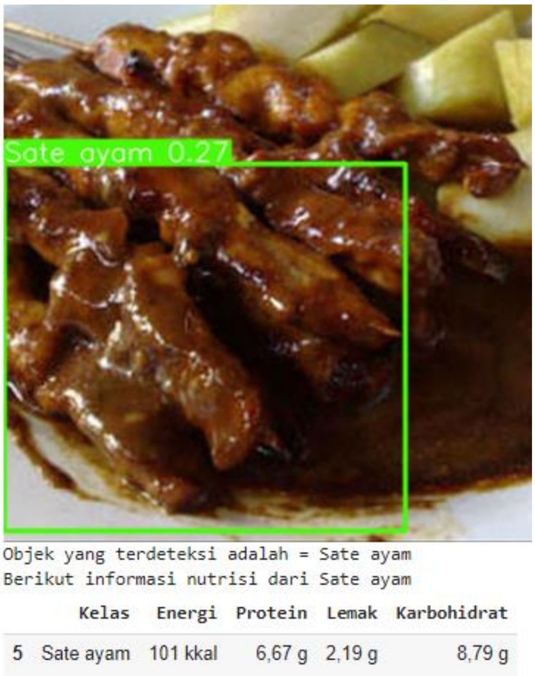


Objek yang terdeteksi adalah = Pempek
Berikut informasi nutrisi dari Pempek

	Kelas	Energi	Protein	Lemak	Karbohidrat
4	Pempek	234 kkal	6,24 g	28,29 g	15,13 g

Figure 10. Food Detection and Nutritional Information of Pempek

Based on Figure 10, the model was successful in detecting the pempek image and estimating its nutrition. The nutrients contained in pempek include energy of 234 kcal, protein of 6.24 g, fat of 28.29 g, and carbohydrates of 15.13 g.

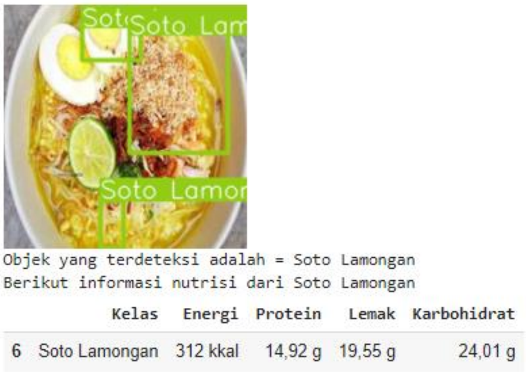


```
Objek yang terdeteksi adalah = Sate ayam
Berikut informasi nutrisi dari Sate ayam
```

	Kelas	Energi	Protein	Lemak	Karbohidrat
5	Sate ayam	101 kkal	6,67 g	2,19 g	8,79 g

Figure 11. Food Detection and Nutritional Information of Sate Ayam

Based on Figure 11, the model was successful in detecting the sate ayam image and estimating its nutrition. The nutrients contained in sate ayam include energy of 101 kcal, protein of 6.67 g, fat of 2.19 g, and carbohydrates of 8.79 g.



```
Objek yang terdeteksi adalah = Soto Lamongan
Berikut informasi nutrisi dari Soto Lamongan
```

	Kelas	Energi	Protein	Lemak	Karbohidrat
6	Soto Lamongan	312 kkal	14,92 g	19,55 g	24,01 g

Figure 12. Food Detection and Nutritional Information of Soto Lamongan

Based on Figure 12, the model was successful in detecting the Soto Lamongan image and estimating its nutrition. The nutrients contained in Soto Lamongan image include energy of 312 kcal, protein of 14.92 g, fat of 19.55 g, and carbohydrates of 24.01 g.

The results of the weights of the model that have been trained are automatically stored after the training process is complete, this process is the export model process. The model is stored in the .pt file extension which is a common convention of the PyTorch library.

IV. CONCLUSION

An image detection system for Indonesian food to estimate nutritional information has been built using the YOLOv5 model. The YOLOv5 model performs very well for the detection of Indonesian specialties, as evidenced by the results obtained with an average accuracy of 98.6%, an average precision value of 0.95, an average recall value of 0.96, and an average F1-Score value of 0.95. The results of the detection are then used to estimate nutrition by taking information per portion from the FatSecret Indonesia website. From the experiments that were carried out on seven pictures of Indonesian food, the estimation was carried out well by displaying various nutritional information including energy, protein, fat, and carbohydrates. The outcome of this research is an object detection model that is ready to be implemented in android applications or websites. Suggestions for further research are to be able to implement the model as an android application or website with various supporting features such as recording calories per day to make it easier for users.

REFERENSI

[1] M. A. Assuja, S. Saniati, and Y. Nurhasanah, “Artificial Intelligence (AI) Fruit Detection for Nutrition Information - using Convolutional Neural Network,” thesis, Universitas Teknokrat Indonesia, Bandar Lampung, 2019 .

[2] R. Kurniasari, L. R. Sefrina, and S. Sabrina, “Edukasi Penggunaan aplikasi Pesan Antar Makanan Online Dengan Bijak Untuk Menciptakan status gizi optimal,” *SELAPARANG: Jurnal Pengabdian Masyarakat Berkemajuan*, vol. 6, no. 1, pp. 230–234, 2022. doi:10.31764/jpmb.v6i1.7208.

[3] A. Poerna Wardhanie and Ahmad Nizar Yogatama, “Peran E-Service Quality Dalam Meningkatkan Penjualan Makanan Melalui Penggunaan Aplikasi ShopeeFood,” *JoTI*, vol. 3, no. 2, pp. 67–77, 2022, doi: 10.37802/joti.v3i2.225.

[4] Y. M. Fristanti and A. Ruhana, “Hubungan Intensitas Penggunaan Smartphone Terhadap Aktivitas Fisik dan Tingkat Konsumsi Pangan Pada Mahasiswa Gizi Unesa Di Era Pandemi Covid–19,” *Jurnal Gizi Unesa*, vol. 1, no. 2, pp. 89–94, 2021.

[5] Menteri Kesehatan, Jakarta: Direktur Jenderal Peraturan Perundang-Undangan Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia, 2019.

[6] M. Dandi, H. Fauzi Tsp, and S. Rizal, “Perancangan Aplikasi Perhitungan Nutrisi Pada Makanan Berbasis Android Dengan Metode Convolutional Neural Network (Cnn),” *e-Proceeding Eng.*, vol. 8, no. No.5, pp. 5000–5008, 2021.

[7] M. L. Chiang, C. A. Wu, J. K. Feng, C. Y. Fang, and S. W. Chen, “Food Calorie and Nutrition Analysis System

based on Mask R-CNN," *2019 IEEE 5th Int. Conf. Comput. Commun. ICCC 2019*, pp. 1721–1728, 2019, doi: 10.1109/ICCC47050.2019.9064257.

[8] R. E. Jayaputra, "Estimasi Kalori pada makanan melalui Citra Makanan menggunakan Model SSD (Single Shot Detector) pada Mobile Device," thesis, Universitas Islam Indonesia, Yogyakarta, 2020.

[9] F. Jubayer *et al.*, "Detection of mold on the food surface using YOLOv5," *Curr. Res. Food Sci.*, vol. 4, pp. 724–728, 2021, doi: 10.1016/j.crfs.2021.10.003.

[10] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, no. August, pp. 687–694, 2020, doi: 10.1109/ICACCS48705.2020.9074315.

[11] M. E. Salman, G. Çakirsoy Çakar, J. Azimjonov, M. Kösem, and İ. H. Cedimoğlu, "Automated prostate cancer grading and diagnosis system using deep learning-based Yolo object detection algorithm," *Expert Syst. Appl.*, vol. 201, no. April, 2022, doi: 10.1016/j.eswa.2022.117148.

[12] G. Liu, J. C. Nouaze, P. L. T. Mbouembe, and J. H. Kim, "YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3," *Sensors (Switzerland)*, vol. 20, no. 7, pp. 1–20, 2020, doi: 10.3390/s20072145.

[13] J. Yu and W. Zhang, "Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4," *Eng. Lett.*, vol. 30, no. 4, pp. 1493–1503, 2022, doi: 10.3390/s21093263.

[14] A. Casado-García *et al.*, "LabelStoma: A tool for stomata detection based on the YOLO algorithm," *Comput. Electron. Agric.*, vol. 178, no. August, p. 105751, 2020, doi: 10.1016/j.compag.2020.105751.

[15] X. Zhou, L. Jiang, C. Hu, S. Lei, T. Zhang, and X. Mou, "YOLO-SASE: An Improved YOLO Algorithm for the Small Targets in Complex Backgrounds," *Sensors*, vol. 22, no. 12, pp. 1–14, 2022, doi: 10.3390/s22124600.

[16] D. Thuan, "Evolution of YOLO Algorithm and YOLOv5: The State-of-the-art Object Detection Algorithm," thesis, Oulu University of Applied Sciences, Finland, 2021.

[17] F. Ramasari, F. Firdaus, S. Nita, and K. Kartika, "Penggunaan Metode You Only Look Once dalam Penentu Pindah Tanaman Cabai Besar Ternotifikasi Telegram ," *Elektron : Jurnal Ilmiah*, vol. 13, no. 2, pp. 45–52, 2021. doi:10.30630/eji.13.2.229.

[18] A. Amwin, "Deteksi dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO)," thesis, Universitas Islam Indonesia, Yogyakarta, 2021.

[19] M. Y. A. Thoriq, I. A. Siradjuddin, and K. E. Permana, "Deteksi Wajah Manusia berbasis One Stage Detector Menggunakan Metode You Only Look Once (YOLO)," *J. Teknoinfo*, vol. 17, no. 1, p. 66, 2023, doi: 10.33365/jti.v17i1.1884.

[20] M. T. Stefanus Christian Adi Pradhana, Untari Novia Wisesty S.T.,M.T., Febryanthi Sthevanie S.T., "Pengenalan Aksara Jawa dengan Menggunakan Metode Convolutional Neural Network," *e-Proceeding Eng.*, vol. 7, no. 1, pp. 2558–2567, 2020.

[21] O. Rochmawanti, F. Utaminingrum, and F. A. Bachtiar, "Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, p. 805, 2021, doi: 10.25126/jtiik.2021844441.

[22] I. W. Prastika and E. Zuliarso, "Deteksi penyakit kulit wajah menggunakan tensorflow dengan metode convolutional neural network," *J. Manaj. Inform. dan Sist. Inf.*, vol. 4, no. 2, pp. 84–91, 2021, doi: 10.36595/misi.v4i2.418.

[23] B. Putra, G. Pamungkas, B. Nugroho, and F. Anggraeny, "Deteksi dan Menghitung Manusia Menggunakan YOLO-CNN," *J. Inform. dan Sist. Inf.*, vol. 02, no. 1, pp. 67–76, 2021.

[24] A. Rosadi *et al.*, "Analisis Sentimen Berdasarkan Opini Pengguna pada Media Twitter Terhadap BPJS Menggunakan Metode Lexicon Based dan Naïve Bayes Classifier," *J. Ilm. Komputasi*, vol. 20, no. 1, pp. 39–52, 2021, doi: 10.32409/jikstik.20.1.401.

[25] N. Hadianto, H. B. Novitasari, and A. Rahmawati, "Klasifikasi Peminjaman Nasabah Bank Menggunakan Metode Neural Network," *J. Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 163–170, 2019, doi: 10.33480/pilar.v15i2.658.

[26] K. A. Shianto, K. Gunadi, and E. Setyati, "Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN," *J. Infra*, vol. 7, no. 1, pp. 157–163, 2019, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/8065

[27] M. Meiriyama, S. Devella, and S. M. Adelfi, "Klasifikasi Daun Herbal Berdasarkan Fitur Bentuk dan Tekstur Menggunakan KNN," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 3, pp. 2573–2584, 2022, doi: 10.35957/jatisi.v9i3.2974.

[28] A. Malta, M. Mendes, and T. Farinha, "Augmented reality maintenance assistant using yolov5," *Appl. Sci.*, vol. 11, no. 11, pp. 1–14, 2021, doi: 10.3390/app11114758.