Implementation of YOLOv8-seg on store products to speed up the scanning process at point of sales

Hanna Arini Parhusip[1], Suryasatriya Trihandaru[1], Denny Indrajaya[1], Jane Labadin[2]
[1]Department of Data Science, Faculty of Science and Mathematics, Satya Wacana Christian University, Salatiga, Indonesia
[2]Data Science Centre, Faculty of Computer Science and Information Technology, Universitas Sarawak, Samarahan, Malaysia

Article Info

Article history:

Received Nov 27, 2023
Revised Feb 18, 2024
Accepted Mar 15, 2024

Keywords:

Deep learning
Instance segementation
Point of sales
Store products
You only look once v8-seg

ABSTRACT

You only look once v8 (YOLOv8)-seg and its variants are implemented to accelerate the collection of goods for a store for selling activity in Indonesia. The method used here is object detection and segmentation of these objects, a combination of detection and segmentation called instance segmentation. The novelty lies in the customization and optimization of YOLOv8-seg for detecting and segmenting 30 specific Indonesian products. The use of augmented data (125 images augmented into 1,250 images) enhances the model's ability to generalize and perform well in various scenarios. The small number of data points and the small number of epochs have proven reliable algorithms to implement on store products instead of using QR codes in a digital manner. Five models are examined, i.e., YOLOv8-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg, and YOLOv8x-seg, with a data distribution of 64% for the training dataset, 16% for the validation dataset, and 20% for the testing dataset. The best model, YOLOv8l-seg, was obtained with the highest mean average precision (mAP)^{box} value of 99.372% and a mAP^{mask} value of 99.372% from testing the testing dataset. However, the YOLOv8m-seg model can be the best alternative model with a mAP^{box} value of 99.330% since the number of parameters and the computational speed are the best compared to other models.

Corresponding Author:

Hanna Arini Parhusip
Department of Data Science, Faculty of Science and Mathematics, Satya Wacana Christian University
52-60 Diponegoro Street, Salatiga 50711, Indonesia
Email: hanna.parhusip@uksw.edu

1. INTRODUCTION

The buying and selling process in modern stores, restaurants, and supermarkets has changed compared to the traditional buying and selling process, where buying and selling transactions use cash. Nowadays, the use of QR codes is an example of modern transactions. However, the use of QR codes is still considered too long in digital transactions because the use of QR codes with many customers at one cashier can cause queues [1]. If the line is long, sometimes people become reluctant to shop at that place. In addition, sometimes it will also have an impact on the use of parking because many people bring vehicles but are still waiting in line to pay. Again, this will make people not shop at the store because the parking is full. Similarly, the addition of cashiers still cannot be considered optimal because the company needs to add workers who pose a burden to employers. Therefore, technology is needed that can overcome these problems.

One solution that can overcome these problems is the development of technology that can speed up the scanning process of purchased goods. In general, the process of scanning or product data collection is carried out one by one, which will take a lot of time if the number of products is too large. Therefore, the solution provided is the application of one method in artificial intelligence (AI) specifically instance segmentation, to detect several products purchased in one scan to speed up the scanning process. Instance segmentation is a part of computer vision that can detect objects with detection results that follow the shape of the object [2]. There have been many studies using instance segmentation for various things, such as building edge detection [3], identification and extraction of oil well sites using the mask-region-based convolutional neural network (RCNN) architecture [4], and fruits detection [5]. One of the architectures used to model instance segmentation in the previous research mentioned is you only look once (YOLO).

The YOLO architecture can provide good results and fast detection capabilities. This model architecture is good for making an instance segmentation model for the detection of store products as a solution to speed up queues [6], the development of autonomous driving systems [7], or managing distance estimation [8], depth estimation [9] and real-time object detection [10]. The way YOLO works is different from the faster R-CNN architecture, mask R-CNN, or other architectures, that apply the two-stage model. YOLO applies a one-stage model so the model can detect objects faster [11]. In this research, the YOLO architecture can provide good results and fast detection capabilities. Compared to several types of YOLO architecture, which was developed in 2023, the YOLOv8-seg architecture is created and it can generate instance segmentation models at a fairly fast speed, for instance, to detect tomatoes [12]. Therefore, the research here refers to YOLOv8-seg to solve the problem above.

Finally, the new material we offer here is the customization and optimization of YOLOv8-seg for detecting and segmenting 30 specific Indonesian products as the novelty of this article. The use of augmented data (125 images augmented into 1,250 images) enhances the model's ability to generalize and perform well in various scenarios. Therefore, the major issue we address here is the implementation of the 5 versions of YOLOv8-seg i.e., YOLOv8-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg, and YOLOv8x-seg. These models are examined to get the best model for a further business process. Thus, this study aims to obtain the best model for making digital transaction devices based on YOLOv8-seg that is better than QR codes, where several objects at once can be directly calculated for buying and selling transactions at the point of sale (POS) to shorten queues or avoid queues, where this has not been done much in supermarkets in Indonesia. The article here proposes to find the best algorithm from the five variants of YOLOv8-seg implemented for the 30 specific Indonesian products.

2. METHOD

The purpose of this research is to produce instance segmentation models with five variants of YOLOv8-seg architecture that can be used for the detection of Indonesian products. In addition, considering the process of creating a model that needs to do labeling on the object to be detected, in this study the amount of data taken for the training process is not so many, with the hope that in updating the model when implementing it in a store there is no need to do labeling data too many. But on the other hand, the use of a lot of data is important in modeling instance segmentation. Therefore, in this research, the data augmentation process is also carried out.

The five variants of YOLOv8-seg architecture used also have different numbers of parameters. Based on the model developed by the developer, when the number of parameters increases, the mean average precision (mAP) values obtained are better, but the computational speed decreases [13]. Therefore, the model created first is the model with the least number of parameters. Then, the resulting model is evaluated. If the evaluation results show bad results, then the training process will be carried out again by adjusting the hyperparameters, until the produced model is good enough. When the model with the least number of parameters is good, other models with more parameters than the first model is trained and evaluated. In this case, the other model creation uses the same hyperparameter values as the first model. After all models have been created, an overall evaluation of all models is conducted and conclusions are drawn.

Based on the outline of the research process that has been described, the research steps are data collection, data preprocessing (data distribution and data augmentation), model creation (training process), evaluation and conclusion. The flowchart of the research process is shown in Figure 1. In the flowchart, each step represents a key stage in the research process, from data collection to conclusion. In addition, the flowchart also shows an overview of the data distribution that was done and the models that were created. A more detailed explanation of the research process will be given in the next subsections.

2.1. Data collection

In this study, making an instance segmentation model for POS where in its application, the detection location can be conditioned. To improve the performance of the model in detection, a white background was used, which when compared to existing products, makes it rare to find products that have a plain white packaging color. Figure 2 shows the data sample where the data are 125 images with products as many as 30. From all the existing images, there are 25 objects for each product.

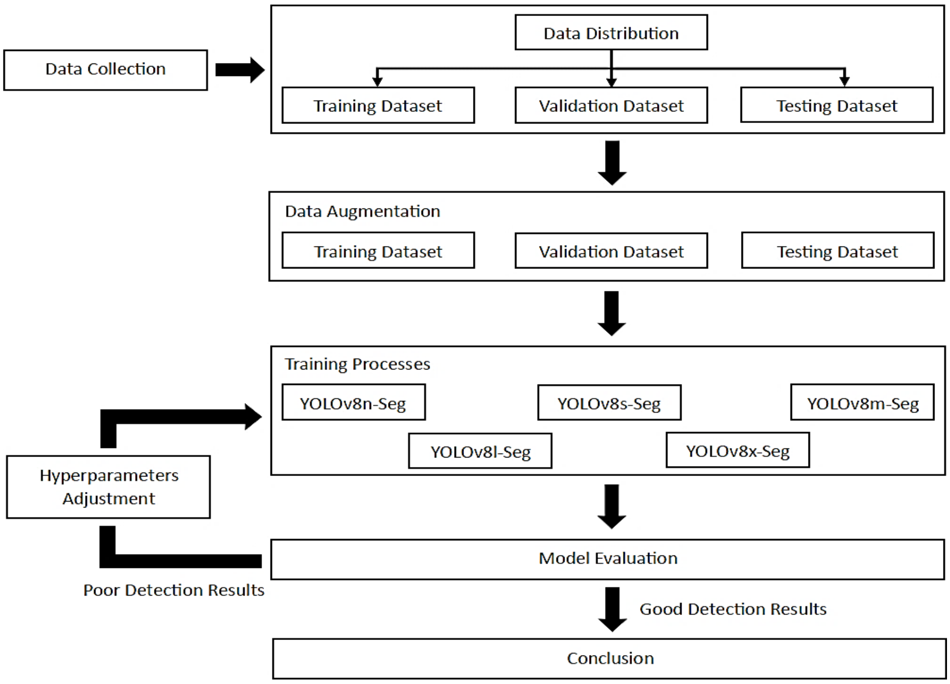


Figure 1. The flowchart of the research process

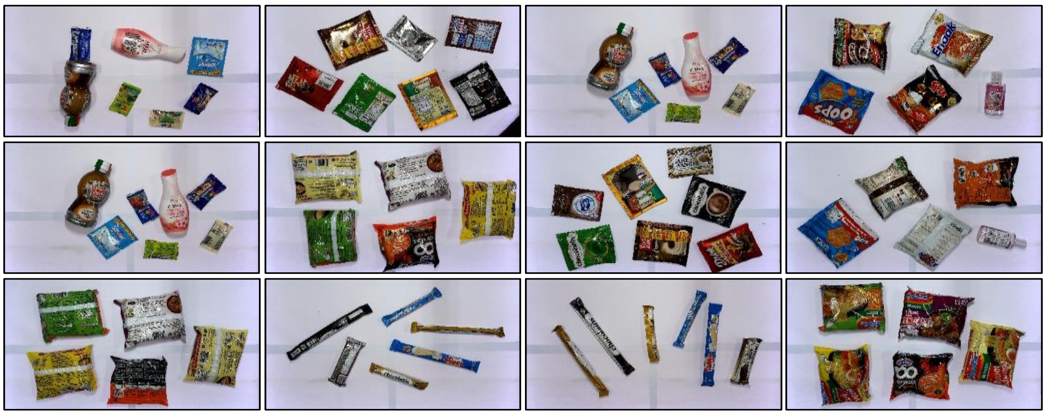


Figure 2. Sample data taken from a store

2.2. Data preprocessing

Before data preprocessing, data was divided into 3 parts, namely 64% data for the training dataset, 16% for the validation dataset, and 20% for the testing dataset. After the data distribution process, each dataset is augmented by providing color filters, such as summer and winter, and various brightness and rotation settings, so that the amount of data in the training dataset becomes 800 images, the validation dataset as much as 200 images, and the testing dataset as much as 250 images. The sample of augmented data is depicted in Figure 3.

The data augmentation is done to overcome the small amount of data and increase the ability of the model to perform detection [14]. In addition, in product detection, white padding was also added to the image so the shape of the image became square, and normalization was carried out with the formula shown in (1) before the image was entered into the model for the detection process, i.e.

$$x' = \frac{x}{255} \qquad (1)$$

where x is the pixel value of the image and x' is the pixel value of the image after normalization.

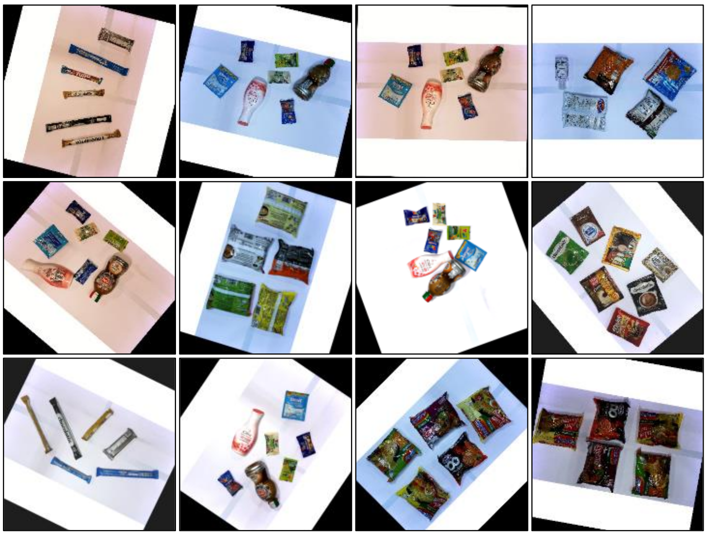


Figure 3. Sample augmented data

2.3. The used models

YOLO is an architecture that continues to be developed from 2016 to 2023. YOLOv8 has better speed and mAP than the previous version [9]. YOLOv8 uses a backbone similar to YOLOv5, with some changes, such as the CSPLayer on YOLOv5 to the C2f module [10]. Compared to YOLOv8 architecture, YOLOv8-seg has additional layers used to segment. The YOLOv8-seg architecture has an architecture that almost resembles YOLOv8 where the YOLOv8-seg uses the CSPDarknet53 extractor feature as shown in Figure 4, followed by C2f and 2 segmentation heads [15], [16].

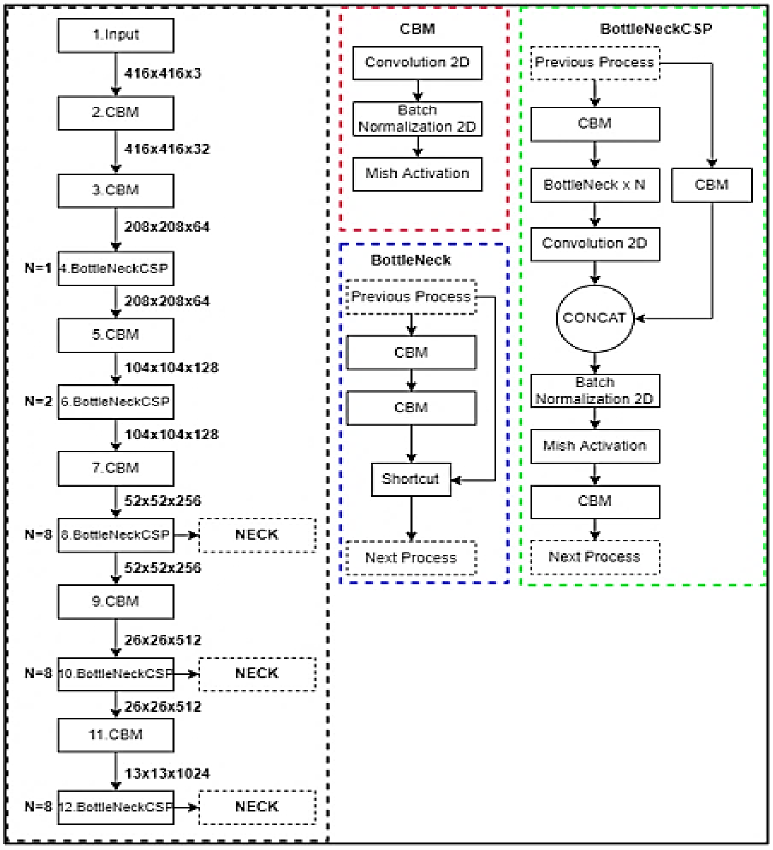


Figure 4. CSPDarknet53 architecture (YOLOv8-seg architecture heads [15], [16])

These architectures have already been evaluated by their developers, as shown in Table 1, but the used data by the developer had a colorful background for each image (COCO dataset) [13]. Table 1 lists that each version of YOLOv8-seg has the same input size with a different number of parameters. If the number of parameters increases, the mAP value also increases and the speed becomes slower. Therefore, the best YOLOv8-seg version will be searched in this research in terms of mAP and the number of parameters.

For each model, the used image size was 640×640 with the same hyperparameters. We choose here first that the number of epochs is 10. This is done by some experiments that are not presented here where the number of epochs is varied to find the minimum number of epochs with the good lost functions. The experiments give conclusion that the best number of epochs is 10 so that we can continue to consider other parameters. Other hyperparameters are: batch of 1, stochastic gradient descent (SGD) optimizer with a learning rate of 0.01, momentum of 0.937, weight decay of 0.0005, and warmup momentum of 0.8.

Table 1. Model evaluation by developer

Model	Size (pixels)	mAP$^{\text{box}}$ 50-95	mAP$^{\text{mas}}$ $^{\text{k}}$ 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	Params (M)	FLOPs (B)
YOLOv8-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

In making bounding boxes on YOLOv8, distribution focal loss (DFL) and complete intersection over union (CIoU) are used to calculate regression loss from bounding boxes [17]. The measure of the bounding box is determined by CIoU in the object detection architecture of the CIoU [18], i.e.

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b^p, b^{gt})}{(c_w)^2 + (c_h)^2} + \frac{4}{\pi^2}\left(tan^{-1}\frac{w^{gt}}{h^{gt}} - tan^{-1}\frac{w^p}{h^p}\right) \qquad (2)$$

Where $\rho^2(b^p, b^{gt})$ showing Euclidean distance between centroids of the ground-truth bounding box (b^{gt}) with predicted bounding box (b^p); h^p and w^p are the height and width of b^p respectively; h^{gt} and w^{gt} are the height and width of b^{gt} respectively, and IoU is the ratio between the intersection and the union of b^{gt} with b^p that formulated by (3) [19], i.e.

$$IoU = \frac{area\ (b^p \cap b^{gt})}{area\ (b^p \cup b^{gt})} \qquad (3)$$

In optimizing the model to more quickly find the distribution of adjacent target areas [20], the DFL formula is implemented [21], i.e.

$$DFL(s_i, s_{i+1}) = -\left((y_i + 1 - y)\log(s_i) + (y_i - y)log(s_{i+1})\right) \qquad (4)$$

where s_i is output from the activation function;y_i and y_{i+1} are the interval order in *i*-th and (*i*+1)-th respectively; and y is the related label.

In the YOLOv8 architecture, several loss formulas are used. One of them is binary cross-entropy loss. The purpose of this formula is to measure the error for the classification task [22], i.e.

$$L_{cls} = \sum_{i=0}^{s^2} 1_i^{obj} \sum_{c \in classes}\left(p_i(c) - \widehat{p_\iota}(c)\right)^2 \qquad (5)$$

Where $1_i^{obj} = 1$ if the object appears inside the *block*, otherwise it is 0. The notion $\widehat{p_\iota}(c)$ refers the conditional probability for class c on the i-th block. We have $p_i(c) = 1$ if ground truth = c, otherwise it is 0. The total loss classification value is the sum of all loss values on each S grid.

Furthermore, in semantic segmentation, there is also a loss for fully convolutional network (FCN), which is also defined as binary cross-entropy loss, which only includes the k-th mask if the region is related to the ground truth class of $k\text{-}th$ class [22], i.e.

$$L_{FCN} = \sum_{1 \le i,j \le m}\left[y_{ij} \ln\hat{y}_{ij}^{\,k} + \left(1 - y_{ij}\right)\ln\left(1 - \hat{y}_{ij}^{\,k}\right)\right] \qquad (6)$$

where y_{ij} is the label of the (i, j) cell on the true mask for regions with size m^2, then $\hat{y}_{ij}^{\,k}$ is the predicted value of the same cell inside the learned mask for the ground truth of $k\text{-}th$ class.

2.4. Method of evaluation

The mAP value is the significant parameter to evaluate the performance of object detection and instance segmentation, which requires an additional parameter, i.e., the intersection over union (IoU) threshold [23]. The purpose of the IoU value is to determine whether the detected object is true or false. Suppose an IoU threshold of 0.5 is selected. We have the standard 2×2 confusion matrix where each component is denoted by true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The model correctly detects the object and $\text{IoU} \geq 0.5$ leading to TP and IoU < 0.5 leading to FP. If the model fails to detect the object then it gives FN and FP will be indicated by wrong detection by the model on the background or other objects. With the result of the confusion matrix, one can define recall and precision [19], i.e.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{8}$$

The accuracy model is indicated by measuring the mAP value [23], i.e.

$$\text{mAP} = \frac{\sum_{i=1}^{k} \text{AP}_i}{k} \tag{9}$$

where k = the number of the used classes and AP_i: the average precision (AP) for each class first, i.e.

$$\text{AP} = \int_0^1 p(r)dr \tag{10}$$

The formula is discretized into

$$\text{AP} = \sum_{i=0}^{n-1} (r_{i+1} - r_i) P_{interp}(r_{i+1}) \tag{11}$$

where:

$$P_{interp}(r_{i+1}) = \max_{r' \geq r_{i+1}} P(r') \tag{12}$$

3. RESULTS AND DISCUSSION

3.1. Results

As stated in the chapter 2, we implemented 5 models of YOLOv8-seg, i.e., YOLOv8-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg, and YOLOv8x-seg models. With the result data of the augmentation process, a summary of the training used dataset is shown in Figure 5. The obtained models are based on the value of the mAP validation dataset from the training process with 10 epochs and IoU of 0.5.

We discuss the results of the models below. Note that the computation is limited due to the ability of the used laptop with specifications: Processor: AMD Ryzen 7 6800H with Radeon Graphics (16 CPUs), ~3.2 GHz, GPU: NVIDIA GeForce RTX 3050, RAM: 16 GB. The result of each model YOLOv8-seg is shown below.

3.1.1. The results of the YOLOv8-seg

The YOLOv8-seg model has the smallest number of parameters out of five instance segmentation models of Indonesian products. The loss value functions are illustrated in Figure 6. With 10 epochs, the loss functions are rapidly tending to zeros as we expected. However, the values for each measure (box_loss, cls_loss, and dfl_loss) need to be improved, leading to adding more epochs to the exercise. On the other hand, the specified architecture and hyperparameters already give segmentation results that are close to the maximum, leading to the addition of epochs, that cannot significantly improve the quality of segmentation. Additionally, the loss values obtained from testing the validation dataset from each epoch also look not much different from the loss values from the training dataset. It means that the results of the training process have run well and do not make an overfitting model, even the loss values are lower than the training dataset.

The results of image detection and segmentation obtained from the testing dataset are shown in Figure 7 with the confusion matrix in Figure 8. Many detection errors occur in the background and quite a lot of background is detected as objects following the loss values graph shown in Figure 6. The objects detected correctly provide good segmentation according to the seg_loss graph obtained. Now, the confusion matrix is observed and it is depicted in Figure 8. The detection errors in the background are dominated by the appearance of two detection results on one object, one of which gives the wrong classification. However, the mAP^{box} and mAP^{mask} values obtained are good as shown by Table 2, which is above 94% in the three datasets.

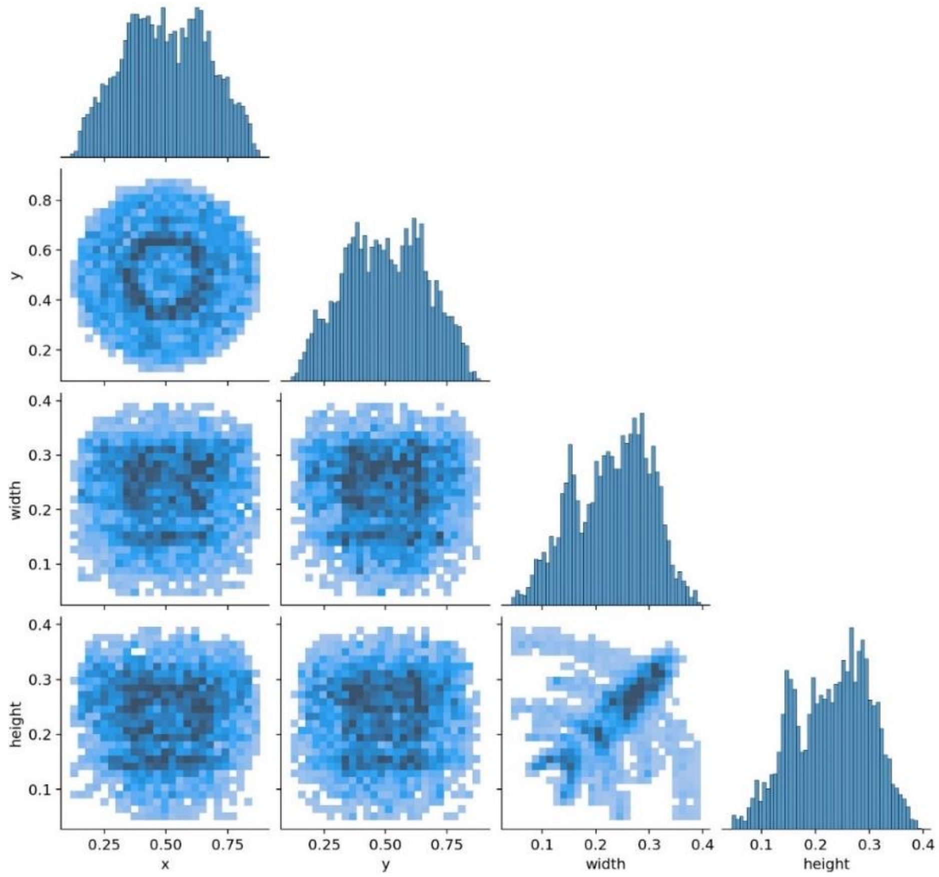


Figure 5. Summary of the training dataset

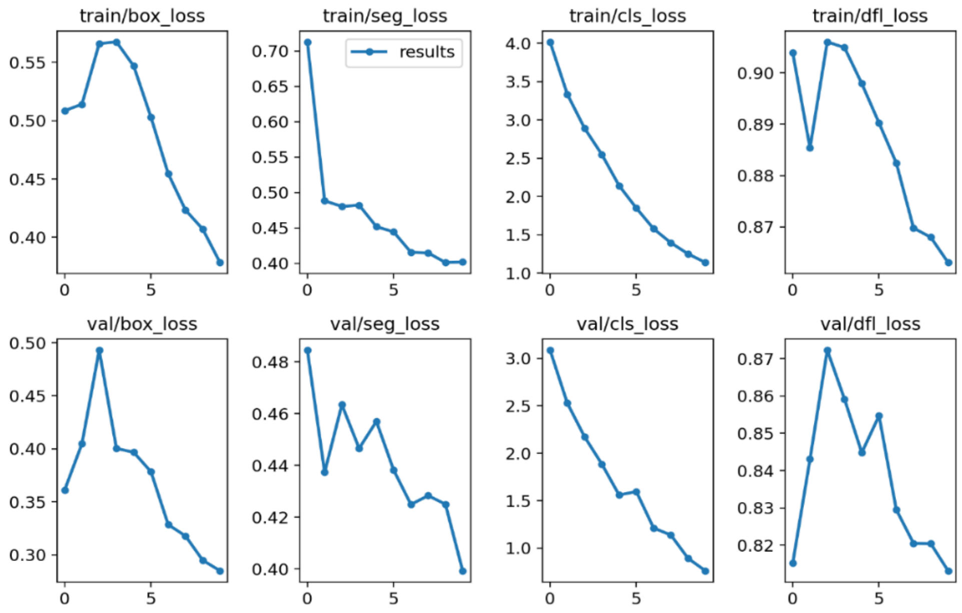


Figure 6. Loss values graph of the YOLOv8-seg model

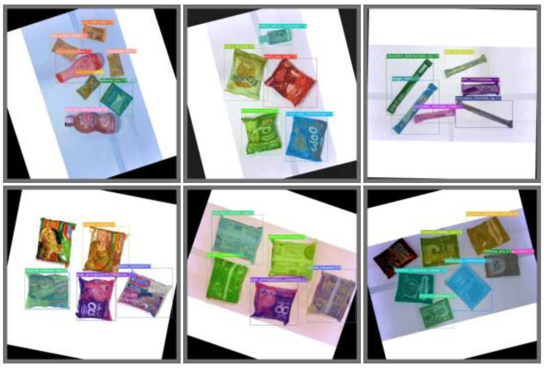


Figure 7. Results of detection and segmentation of the YOLOv8-seg model on the testing dataset

.

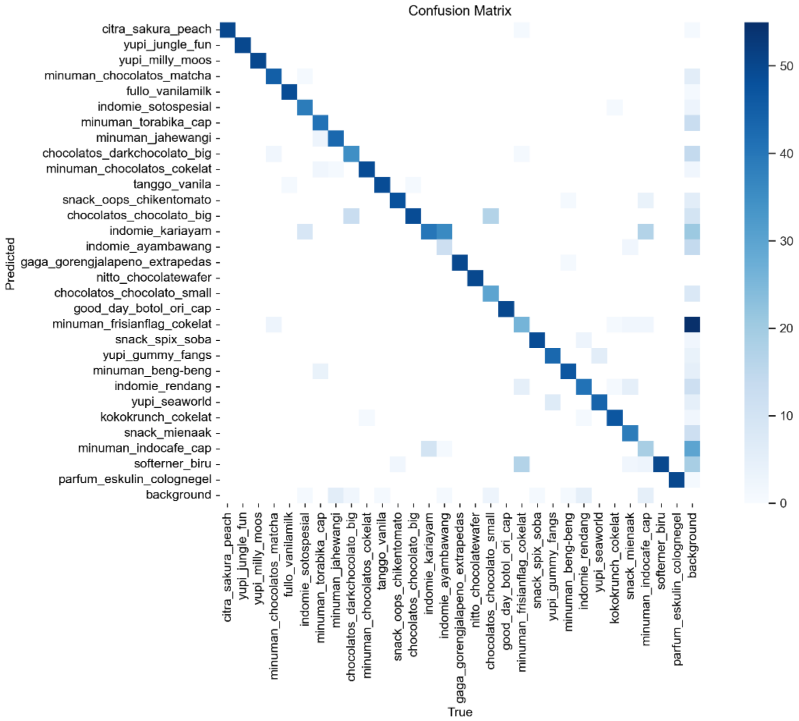


Figure 8. Confusion matrix of the YOLOv8-seg model testing on the testing dataset (vertical and horizontal: the list of Indonesian products)

Table 2. The value of mAP obtained from the YOLOv8-seg model

Datasets	Box (%)	Mask (%)
Training dataset	96.578	96.575
Validation dataset	95.930	95.930
Testing dataset	94.647	94.647

3.1.2. The results of the YOLOv8s-seg

Similarly, we study the data in the YOLOv8s-seg model, where the training process uses the same hyperparameters as the YOLOv8-seg model. We show similar results with similar procedures. The graph of loss values obtained during the training process of this model is shown in Figure 9. In this model, the loss values obtained look better than the previous model, with the loss value at the 10 epochs of the box_loss, seg_loss, clc_loss, and dfl_loss lower than the YOLOv8-seg model. Even though the shape of the loss value graph obtained from the 10 epochs is still not seen optimally and it is still possible to improve model performance. Based on the loss values from the validation dataset in this model, there is also no overfitting; just like in the previous model, the loss values of the validation dataset are lower than the training dataset.

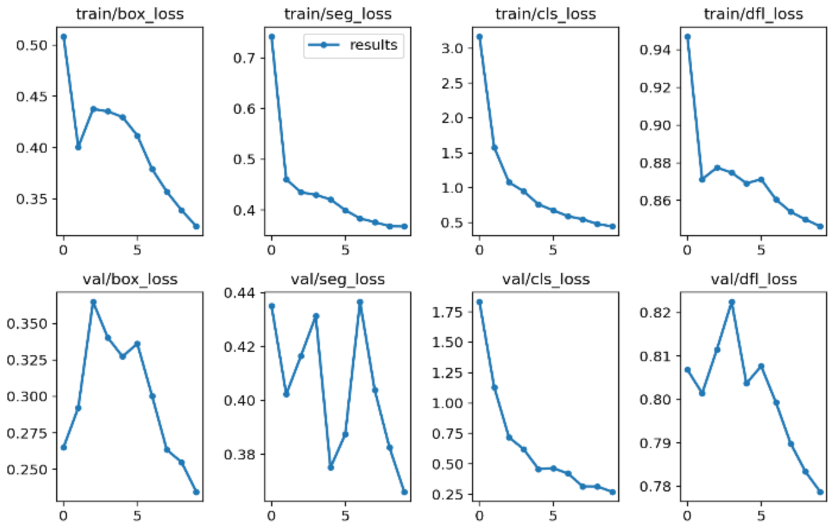


Figure 9. Loss values graph of the YOLOv8s-seg model

Figure 10 shows the confusion matrix of the YOLOv8s-seg model on the testing dataset. In addition, the classification of detected objects is also shown. Thus, the classification results produced in this model are in line with the obtained loss value. However, there are still objects that are not detected or considered background. Now, the results of the YOLOv8s-seg model in the testing dataset are shown in Figure 11. The model can detect and segment well. The obtained segmentation results are also more precise with the detected objects when compared to the previous model. The mAP values obtained from this model are shown in Table 3, where the mAP values obtained are better than the previous model. The increase in mAP obtained in this model is quite significant, where in all datasets the mAP value for box and mask is above 98%.

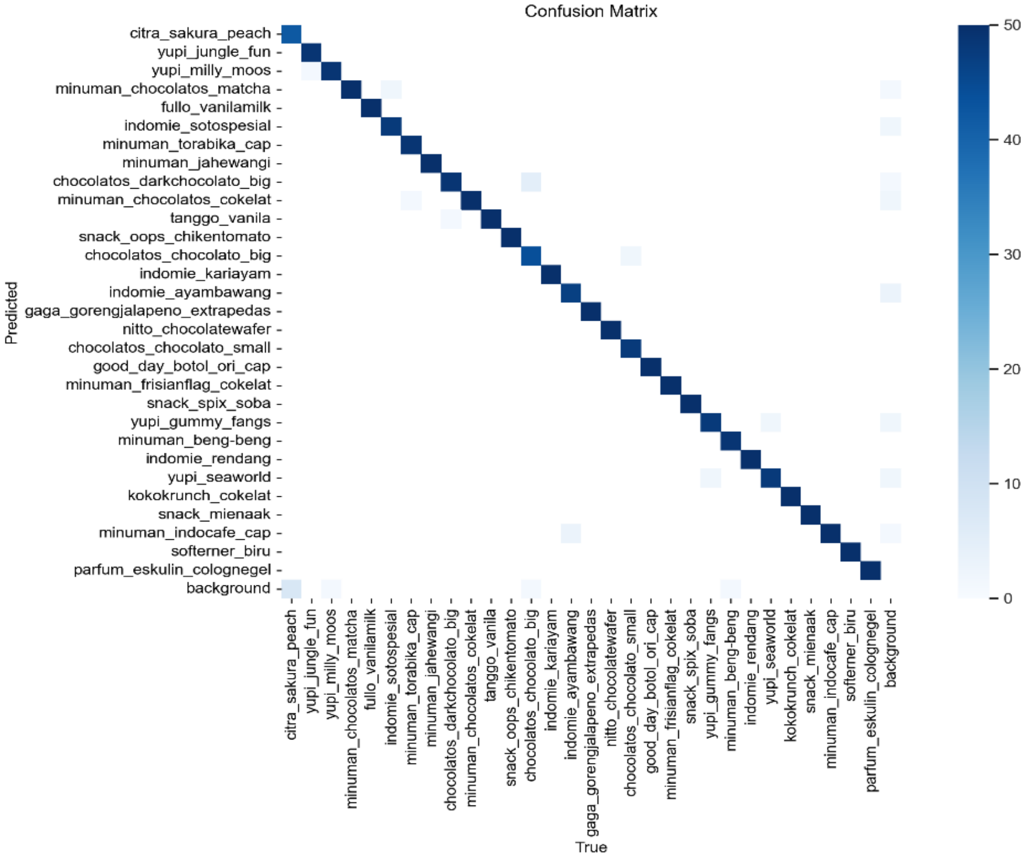


Figure 10. Confusion matrix of the YOLOv8s-seg model on the testing dataset (names are in Indonesian)

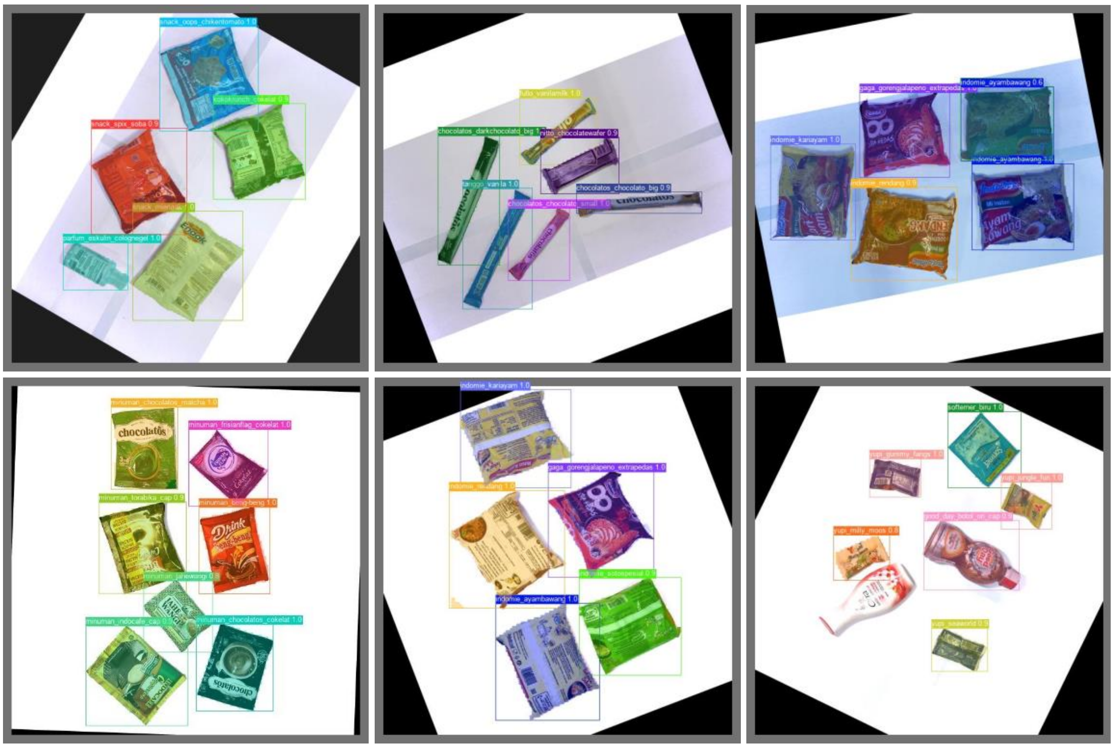


Figure 11. Detection and segmentation results from the YOLOv8s-seg model on the testing dataset (names are in Indonesian)

Table 3. mAP obtained from the YOLOv8s-seg model

Datasets	Box (%)	Mask (%)
Training dataset	98.886	98.886
Validation dataset	98.852	98.852
Testing dataset	98.657	98.657

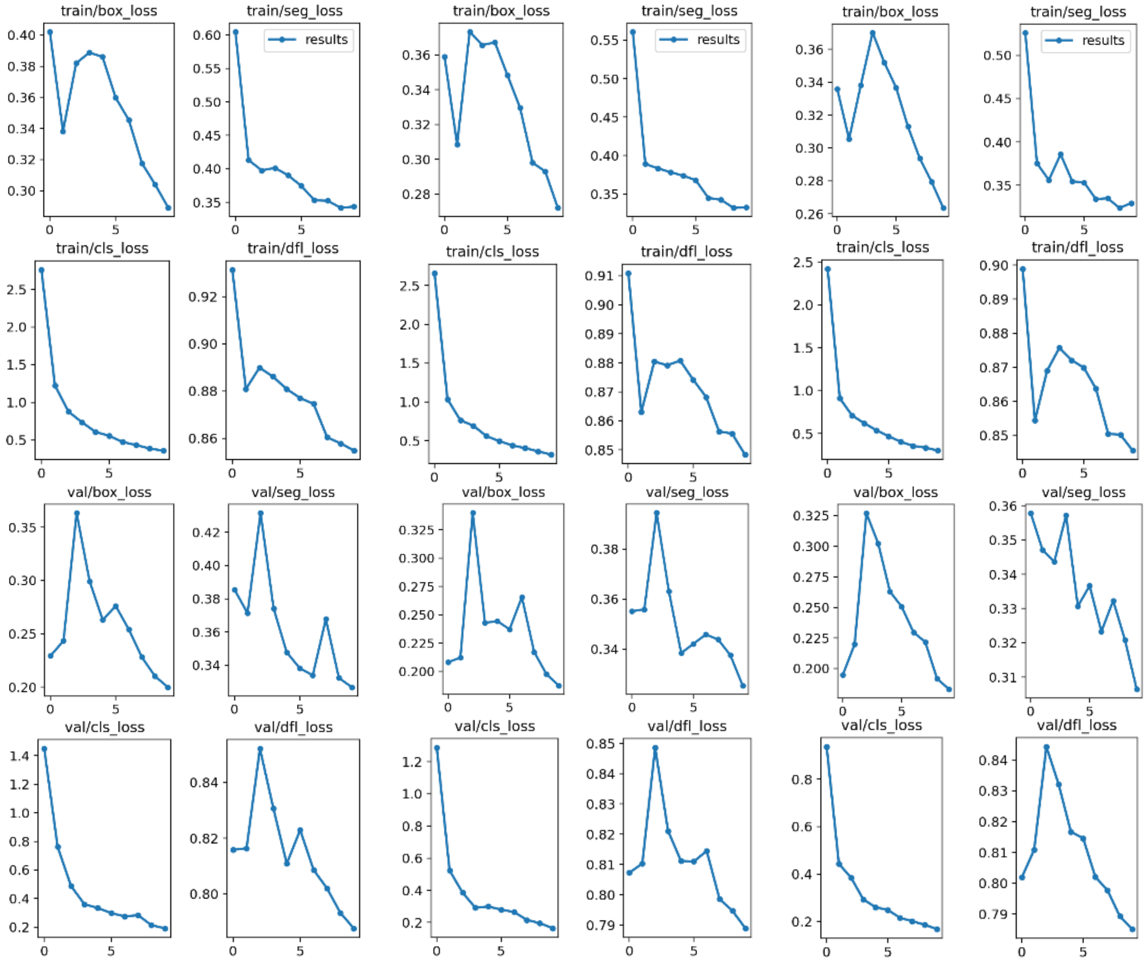
3.1.3 The results of the models YOLOv8m-seg, YOLOv8l-seg, YOLOv8x-seg

We have observed the first two models that are used to develop the application for a store to speed up the consumers to make payment more quickly by introducing object detections in the process. We provide here 3 more models to have the best performance of AI before being implemented in the casier machine. We will compare the results for the same cases here to draw quicker visible conclusion for each model. The 3 models that will be observed are YOLOv8m-seg, YOLOv8l-seg, and YOLOv8x-seg.

The YOLOv8m-seg model provides loss values in the training process that are not much different from the YOLOv8s-seg model. In this model, the loss values for box_loss, cls_loss, and seg_loss are better but not much different, while the dfl_loss values at the 10-th epoch are higher (see Figure 12(a)). In this model, as also seen in the box_loss graph, the loss value in each epoch has not moved in a convergent direction. In the loss values validation dataset graph, the same thing happens where the loss values in box_loss, cls_loss, and seg_loss are better, but for dfl_loss the loss value increases. In this model, it also appears that there is no overfitting and the mAP value in the validation dataset is lower than the training dataset.

The YOLOv8l-seg model is a model that has a greater number of parameters than the previous three models. In this model, the loss values obtained from 10 epochs are better than the previous model, but the difference is not significant (see Figure 12(b)). Of course, this difference is not significant because, in terms of mAP, the mAP value obtained in the previous model is close to perfect, which means that the loss values of the training dataset and validation dataset are also close to perfect. However, it still looks like it can be improved again because the change in loss value in each epoch in box_loss and df_loss has not been seen converging to a certain value, so it can be even better. This model also does not look overfitting because the loss values of the validation dataset are not much different from the loss values of the training dataset.

The last model is YOLOv8x-seg with the lost function is depicted in Figure 12(c). The model has the highest number of parameters out of the five models. Based on previous models, the loss value of the training process is obtained by more parameter models. However, the loss value obtained from the training process is not significantly different from the YOLOv8l-seg model. In this model, the loss values in box_loss, seg_loss, and dfl_loss have not been seen converging towards certain values, such as cls_loss, that have been seen towards certain values, so the model is still being improved. Overfitting is not observed here yielding to have good results.



(a) (b) (c)

Figure 12. Illustration of loss values graph of the (a) YOLOv8m-seg, (b) YOLOv8l-seg, (c) YOLOv8x-seg

Now we mimic the confusion matrices for the three models without presenting the matrices here for simplicity. The YOLOv8m-seg shows that the misdetection occurs only in products whose packaging is very similar, such as chocolatos_chocolato_big and chocolatos_chocolato_small, which only differ in size due to the obtained box_loss value providing excellent detection results. The testing dataset using this model also provides mAP^{box} and mAP^{mask} values, directly proportional to the confusion matrix results. The segmentation results are good as the previous model. However, there is an image as the result of augmentation that shows objects whose segmentation is slightly inappropriate because the object is white and becomes merged with the white background. The mAP value obtained in this model has an increase of about 1% from the two previous YOLOv8s-seg models, where the mAP^{box} and mAP^{mask} values are above 99% for all three datasets as shown in Table 4. The mAP value is almost close to 100% yielding better results.

Table 4. The mAP obtained from the YOLOv8m-seg model

Datasets	Box (%)	Mask (%)
Training dataset	99.392	99.392
Validation dataset	99.424	99.424
Testing dataset	99.330	99.330

According the confusion matrix of the YOLOv8l-seg, the classification error has been very small, even almost non-existent. Additionally, there are still some backgrounds detected as an object, namely fullo_vanilamilk and there are still objects that have not been detected or considered background. The error occurs because the packaging of these goods looks very similar, namely chocolatos_chocolato_big and chocolatos_chocolato_small. Similarly, we obtain the confusion matrix of the YOLOv8x-seg model which provides good results but are not much different from the YOLOv8l-seg model. There are still some backgrounds detected as objects or vice versa. The background is detected to be objects chocolatos_chocolato_small, yupi_gummy_fangs, and yupi_seaworld leading to the errors.

Detection and segmentation results from the last 3 models are also examined. The results are not shown here visually. The mAP values for this model are shown in Table 5. Similarly, the computation proceeded for the YOLOv8x-seg model with the value of mAP nearly the same as in Table 5. This is shown in Table 6.

Table 5. mAP obtained from the YOLOv8l-seg model

Datasets	Box (%)	Mask (%)
Training dataset	99.493	99.493
Validation dataset	99.367	99.367
Testing dataset	99.372	99.372

Table 6. mAP obtained from the YOLOv8x-seg model

Datasets	Box (%)	Mask (%)
Training dataset	99.448	99.448
Validation dataset	99.130	99.130
Testing dataset	99.354	99.354

3.2. Discussion

We have shown details of our observations based on the YOLOv8-seg models. We collect the mAP values from each observation by category. The number of epochs (10) can already provide a high mAP value in the entire dataset as shown in Table 7. In addition, the mAP^{box} and mAP^{mask} values obtained on each model are also relatively the same, which means that the YOLOv8-seg architecture has good performance when used to detect store products where the detection location can be conditioned. The segmentation results obtained are also very good; this happens because the detection location is carried out on a background that is a monotone color (not colorful). In terms of loss values, graphs obtained in all models also show no overfitting, even though they use a relatively small amount of data, that is, 800 images containing 30 types of products for the training process. Even these images are the result of the augmentation process. These conclude the benefits of using YOLOv8-seg with its 5 variants.

Table 7. The mAP values obtained from the entire YOLOv8-seg model (bold is indicating the highest percentage on each column)

Models	Training dataset		Validation dataset		Testing dataset	
	Box (%)	Mask (%)	Box (%)	Mask (%)	Box (%)	Mask (%)
YOLOv8-seg	96.578	96.575	95.930	95.930	94.647	94.647
YOLOv8s-seg	98.886	98.886	98.852	98.852	98.657	98.657
YOLOv8m-seg	99.392	99.392	99.424	99.424	99.330	99.330
YOLOv8l-seg	99.493	99.493	99.367	99.367	99.372	99.372
YOLOv8x-seg	99.448	99.448	99.130	99.130	99.354	99.354

Table 7 shows a significant difference that occurred in the YOLOv8-seg model compared to the other models, which is 94.647% as the result of the testing dataset. However, the YOLOv8-seg model has the least number of parameters of all the models so it has the fastest computational process as shown in Table 8. The small number of parameters in this model means it generates a small number of map features or less important information in the image is obtained. According to the mAP value, this model can already provide excellent results, with mAP value of up to 94.647%. Additionally, the detection and segmentation capabilities of this model can still be improved by increasing the number of epochs.

The detection and segmentation results obtained from the entire model have been very good. On the other hand, there are still objects that are not detected. The confidence score value of the object does not reach the limit of the confidence score used, which is 0.5. As a result, the detection results obtained are ignored. The confidence score is very important in object detection; hence, the acquisition of detection results with a high confidence score can minimize the occurrence of misclassification. In the obtained models, the confidence score obtained is very good; many confidence scores are above 0.8, but there are still some that are below 0.8, which are finally ignored because they are smaller than 0.5.

Table 8. The number of parameters of the models in this study to detect 30 products

Models	Number of parameters
YOLOv8-seg	3.269.466
YOLOv8s-seg	11.801.706
YOLOv8m-seg	27.257.018
YOLOv8l-seg	45.959.178
YOLOv8x-seg	71.779.738

The obtained models are considerably sophisticated. Moreover, the mAP value is already very high. By obtaining a mAP value that is determined perfectly with an epoch of 10, it is also possible to obtain a mAP value of up to 100% in all datasets by conducting a training process using a higher epoch value. Focusing on the objectives of this study, namely comparing models obtained from five different architectures to speed up the scanning process at the POS. The best model, YOLOv8l-seg, was obtained with mAP values from the training dataset and testing dataset with an epoch of 10 for the detection and segmentation of 30 store products. In the application of the model, there are several other considerations, such as the used computer device. It is necessary to consider the computational speed of the model. If the computational speed is important, including the goodness of the model in detecting and segmentation, the YOLOv8m-seg model can be used, which has a smaller number of parameters than YOLOv8l-seg and the mAP value that is similar to YOLOv8l-seg. The products can have more complex structures. One may refer to the development of the variant YOLOv8-seg by considering the lightweight [24]. Other versions of detecting small objects can be considered, such as YOLOv4 [25] or DC-YOLOv8, where detection is based on the camera sensor [26].

4. CONCLUSION

The article here presents the implementation of AI for object detection of sells products in a store to build an AI system in the process of selling products to customers. This study aims to create a model that can be used to accelerate the speed up of the scanning process at POS where the number of products in the store is usually very large. The used model architectures are YOLOv8-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg, and YOLOv8x-seg. The data are 125 images, and each image data is augmented to have 1,250 new image data with 30 objects. Based on the obtained models using 125 images with a data distribution of 64% for the training dataset, 16% for the validation dataset, and 20% for the testing dataset. The YOLOv8-seg architecture can provide good results with only an epoch of 10. From the 5 types of YOLOv8-seg models, the best YOLOv8l-seg model was obtained with the best mAP value for box and mask from testing the testing dataset. The obtained mAP^{box} value was 99.372%, and the mAP^{mask} value was 99.372%. Furthermore, the algorithm should work with the required speed and optimal parameters. In these cases, one should use the YOLOv8m-seg model, which can be an alternative with a mAP^{box} of 99.330% and a mAP^{mask} value of 99.330%. Further research can be developed, i.e. using more products based on the results here. By applying the algorithms to the enough number data for object detection and segmentation, the models are expected to give good accuracy with a record of the detection location that can be conditioned and carried out the augmentation process on images and labels on images.

ACKNOWLEDGEMENTS

The research here is supported by internal university funding from Satya Wacana Christian University in Indonesia, No. 146/SPK-PT/RIK/9/2023 for producing the data.

REFERENCES

[1] D. R. Wibowo and S. Hergiyan, “Simulation analysis of cashier service queue system at a supermarket,” in *Proceedings of the Second Asia Pacific International Conference on Industrial Engineering and Operations Management*, 2021, pp. 2033–2043.

[2] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 87–93, 2018, doi: 10.1007/s13735-017-0141-z.

[3] D. Prabhakar and P. K. Garg, “Building edge detection from very high-resolution remote sensing imagery using deep learning,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 48, pp.

189–196, 2023, doi: 10.5194/isprs-Archives-XLVIII-M-3-2023-189-2023.

[4] H. He *et al.*, "Mask R-CNN based automated identification and extraction of oil well sites," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, pp. 1–29, 2022, doi: 10.1016/j.jag.2022.102875.

[5] O. M. Lawal, "YOLOv5-LiNet: a lightweight network for fruits instance segmentation," *PLOS ONE*, vol. 18, no. 3, pp. 1–19, 2023, doi: 10.1371/journal.pone.0282297.

[6] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.

[7] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115–122, 2023, doi: 10.1016/j.patrec.2023.03.009.

[8] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-YOLO: fast object detection with distance estimation," *Applied Sciences*, vol. 12, no. 3, pp. 1–13, 2022, doi: 10.3390/app12031354.

[9] J. Yu and H. Choi, "YOLO mde: object detection with monocular depth estimation," *Electronics*, vol. 11, no. 1, 2022, doi: 10.3390/electronics11010076.

[10] F. Wang, X. Wang, J. Zhou, and M. Liu, "An underwater object detection method for sonar image based on YOLOv3 model," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 44, no. 10, pp. 3419–3426, 2022, doi: 10.11999/JEIT220260.

[11] M. C. -García, J. T. -Mateo, P. L. -Benítez, and J. G. -Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, 2021, doi: 10.3390/rs13010089.

[12] X. Yue, K. Qi, X. Na, Y. Zhang, Y. Liu, and C. Liu, "Improved YOLOv8-seg network for instance segmentation of healthy and diseased tomato plants in the growth stage," *Agriculture*, vol. 13, no. 8, 2023, doi: 10.3390/agriculture13081643.

[13] G. Jocher, Q. Burhan, and Q. Laughing, "Instance segmentation," *Ultralytics*, 2023. Accessed: Nov. 18, 2023. [Online]. Available: https://docs.ultralytics.com/tasks/segment/#export/

[14] R. S. Passa, S. Nurmaini, and D. P. Rini, "YOLOv8 based on data augmentation for MRI brain tumor detection," *Scientific Journal of Informatics*, vol. 10, no. 3, pp. 363–370, 2023, doi: 10.15294/sji.v10i3.45361.

[15] RangeKing, "Brief summary of YOLOv8 model structure," *GitHub*, 2023. Accessed: Jun. 15, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics/issues/189

[16] J. Terven and D. C. -Esparza, "A comprehensive review of YOLO: From YOLOv1 and beyond," *arXiv-Computer Science*, pp 1-36, 2023.

[17] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y. I. Cho, "An improved wildfire smoke detection based on YOLOv8 and UAV images," *Sensors*, vol. 23, 2023, doi: 10.3390/s23208374.

[18] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.

[19] R. Padilla, S. L. Netto, and E. A. B. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242. doi: 10.1109/IWSSIP48289.2020.9145130.

[20] W. Y. Feng, Y. F. Zhu, J. T. Zheng, and H. Wang, "Embedded YOLO: a real-time object detector for small intelligent trajectory cars," *Journal of Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/6555513.

[21] M. Shroff, A. Desai and D. Garg, "YOLOv8-based waste detection system for recycling plants: a deep learning approach, " *2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, Erode, India, 2023, pp. 1-9, doi: 10.1109/ICSSAS57918.2023.10331643.

[22] Z. Wang, "SEG-YOLO: Real-time instance segmentation using YOLOv3 and fully convolutional network," M.Sc. Thesis, Department of Information and Network Engineering, School of Electrical Engineering and Computer Science, 2019.

[23] G.-S. Peng, "Performance and accuracy analysis in object detection," M.Sc. Thesis, Department of Computer Science, California State University at San Marcos, 2019.

[24] Y. Wu, Q. Han, Q. Jin, J. Li, and Y. Zhang, "LCA-YOLOv8-Seg: An improved lightweight YOLOv8-seg for real-time pixel-level crack detection of dams and bridges," *Applied Sciences*, vol. 13, no. 19, 2023, doi: 10.3390/app131910583.

[25] S. J. Ji, Q. H. Ling, and F. Han, "An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information," *Computers and Electrical Engineering*, vol. 105, pp. 1–9, 2023, doi: 10.1016/j.compeleceng.2022.108490.

[26] H. Lou *et al.*, "DC-YOLOv8: small-size object detection algorithm based on camera sensor," *Electronics*, vol. 12, 2023, doi: 10.3390/electronics12102323.

BIOGRAPHIES OF AUTHORS

Hanna Arini Parhusip holds a Doctor of Mathematics degree from ITB, 2005. She received her B. Sc.in Mathematics from Gadjah Mada Univeristy in Indonesia and her M.Sc. (in industrial mathematics) from Kaiserslautern University, in 1992 and 1997, respectively. In her master's program she was a DAAD scholar, and the sandwich program supported by DAAD, ICTP, and the Indonesian government for her Doctoral program. She is currently an associate professor in the Master of Data Science Department at Satya Wacana Christian University in Indonesia. She has been one of the founders of the Master program of Data Science at this university since 2020 and developing the application of artificial intelligence for mining and the Internet of Things for several purposes in some surrounding companies as a collaboration between the university and the industries. Her research includes machine learning, artificial intelligence, the internet of things, and the application of ordinary differential equations. She has published over 10 papers in international journals and conferences in the last 5 years. She has been the head in the Master program of Data Science. She can be contacted at email: hanna.parhusip@uksw.edu.

Suryasatriya Trihandaru holds a Doctor of Mathematics degree from ITB, 2005. He received his B.Sc. in Physics from Gadjah Mada University in Indonesia and his M.Sc. (in industrial mathematics) from Kaiserslautern University, in 1992 and 1998, respectively. In his master's program, he was a DAAD scholar, and the sandwich program was supported by DAAD, ICTP, and Indonesian government for his doctoral program. He is currently an associate professor in the Master of Data Science Department at Satya Wacana Christian University in Indonesia. He has been one of the founders of the Master's program in Data Science at this university since 2020 and has been developing the application of artificial intelligence for mining, medical physics and the Internet of Things for several purposes in some surrounding companies as a collaboration between the university and the industries. His research includes machine learning, artificial intelligence, the internet of things, and medical physics. He has published over 10 papers in international journals and conferences in the last 5 years. He can be contacted at email: suryasatriya@uksw.edu.

Denny Indrajaya has held a Bachelor of Mathematics (B.Sc.) in Mathematics of Science and Mathematics Faculty, Satya Wacana Christian University, since 2022. He is directly studying for a Master of Data Science and achieved a national award in the Data Science Competition in December 2022. In 2023, he won the Best Idea award for the other national competition in data science. He got sponsorship from a university and from a company to do his master's program in object detection. He can be contacted at email: 632022002@student.uksw.edu.

Prof. Dr. Jane Labadin is a Professor at the Faculty of Computer Science and Information Technology as well as a Research Fellow at the Institute of Social Informatics and Technological Innovation, Universiti Malaysia Sarawak (UNIMAS). She received her Ph.D. in computational mathematics specializing in fluid dynamics from the Imperial College of Science, Technology, and Medicine, London, UK, in 2002. Her bachelor's degree in applied mathematics was from the same university in 1995. She obtained her Master in Computation in 1997 from the University of Manchester, Institute of Science and Technology, UK. She is an active researcher, as evident from the many internal and external research projects that she leads as well as from her publications. Her research interest is generally in computational modeling of dynamical systems, which includes infectious disease modeling. She can be contacted at email: ljane@unimas.my.