Performance Improvement of YOLOv8-RTDETR Method Based Retail Product Detection

Andi Wahyu Maulana [1,2], Suryo Adhi Wibowo [1,2,*], Unang Sunarya [3], Rissa Rahmania [4], and Asep Insani [5]

[1] School of Electrical Engineering, Telkom University, Bandung, Indonesia
[2] Center of Excellence Artificial Intelligence for Learning and Optimization, Telkom University, Bandung, Indonesia
[3] School of Applied Science, Telkom University, Bandung, Indonesia
[4] Computer Science Department, School of Computer Science, Bina Nusantara University, Bandung Campus, Jakarta, Indonesia
[5] National Research and Innovation Agency (BRIN), Jakarta, Indonesia
Email: suryoadhiwibowo@telkomuniversity.ac.id (S.A.W.); awahyumaulana@student.telkomuniversity.ac.id (A.W.M.); unangsunarya@telkomuniversity.ac.id (U.S.); rissa.rahmania@binus.ac.id (R.R.); asep035@brin.go.id (A.I.)
*Corresponding author

***Abstract*—Recently people often purchase their daily needs at retail stores. Therefore, crowds might happen due to a manual queueing system. To overcome the problem, the smart system based on object detection has been conducted using several object detection methods. This study proposed YOLOv8 combined with transformer Real-Time Detection Transformer (RT-DETR) model to enhance the method performance in detecting the detail products. The intra-Class Variation method has been used to recognize the characteristics of the products such as size, color, and variant of the product. To validate the proposed model, three different datasets have been applied that is grocery dataset that displays products one by one in the training and validation process, the RPC-dataset that has many products in one image, and the D2S dataset with products that have varying lighting and stacked products. Results showed that the proposed model outperformed compared to other models, with a mean Average Precision (mAP) of 99.5% for the grocery dataset, 99.3% RPC-dataset, and 85.5% D2S dataset, respectively.**

***Keywords*—intra-class variation, mean Average Precision (mAP), retail product, Real-Time Detection Transformer (RT-DETR), YOLOv8**

I. Introduction

Vision computer is an Artificial Intelligence (AI) that performs deep learning methods to analyze image or visual data to obtain information from the image data [1]. Its applications are diverse, including segmentation for separating parts of images such as organs in medical imaging, recognition of objects and text for identification and security, and object detection used in intelligent traffic systems and autonomous vehicles.

The application of computer vision focusing on object detection has many variations. One example is an application in retail product detection. One existing technology is Amazon Dash, which provides a convenient shopping experience without the need to queue. Amazon Dash utilizes a cashless system called "Just Walk Out" to implement a self-service store. Although Amazon Go's "Just Walk Out" technology has been discontinued, this does not mean that retail product detection technology lacks significant potential in Indonesia. On the contrary, this technology can bring substantial benefits. With a population of 279,585,034 people [2] and a total of 34,715 retail stores, object detection technology can help improve operational efficiency and reduce human errors in inventory management. Retail product detection has received a lot of attention from researchers because it has a variety of information such as product size, type, and color. Therefore, image processing on products is very challenging and much needed for retail stores to advance the technology in their stores.

Several deep learning methods are usually used in object detection such as Convolutional Neural Network (CNN) [3], You Only Look Once (YOLO) [4], Faster Region-Based Convolutional Neural Network (R-CNN) [5], Single Shot Multibox Detector (SSD) [6], Vision Transformer [7] and others. The mentioned models are among the deep learning models that are often used for object detection with maximum accuracy and lower time consumption. In this study, the YOLO model was enhanced with a vision transformer model to detect retail products. This approach improves the detection of product variations and focuses on intra-class variations [8], resulting in a high mean Average Precision (mAP)

- This study used the object detection method on retail products using YOLOv8 combined with the Real-Time Detection Transformer (RT-DETR) vision transformer model to produce good product detection accuracy and fast computation vision transformer model to produce good product detection accuracy and fast computation.
- Intra-class variation is tested with a variety of retail product datasets with the aim of training and emphasizing intra-class variation in the model to

Manuscript received April 23, 2024; revised June 11, 2024; accepted June 18, 2024; published September 6, 2024.

enhance performance. We are robust with decoder in Real-Time Detection Transformer (RT-DETR) that uses residual block.

The rest of this study is organized as follows. Section II shows the literature review, Section III describes the proposed method in detail. Experiments and results are shown in Section IV and finally, we conclude this study in Section V.

II. Literature Review

Several previous studies have attempted to address the challenge of detecting products with visual similarities but different types. For example, research conducted by Santra *et al*. [9] took an approach using a Reconstruction-Classification Network (RC-Net), a network combining reconstruction and classification to enhance detail classification accuracy. This method integrates two main stages: reconstruction to reduce noise and improve image quality, and classification to accurately identify objects. RC-Net has demonstrated its effectiveness in managing variations in image quality and enhancing overall classification accuracy. The method managed to achieve an accuracy rate of around 90% on the various datasets tested, although it still requires improvement to reach above 80% overall.

On the other hand, Hsia *et al*. [10] designed an experiment by applying the Faster Region-based Convolutional Neural Network (R-CNN) method with data augmentation. They utilized various data augmentation techniques such as rotation, flipping, and scaling to enrich the dataset and enhance model performance. The study results showed that data augmentation significantly improved the model's accuracy and robustness to input data variations. The results of this study achieved a mAP accuracy rate of 99.27%, but the resulting model still faced obstacles in recognizing very subtle differences in products, due to the limitation of the size of the dataset used.

Another study conducted by Lee *et al*. [11] used the YOLOv5 model with Mobile Neural Network version 3 (MobileNetv3) architecture for retail product detection. This combination is designed to optimize detection speed and efficiency without compromising accuracy. Experimental results demonstrated that this model could detect retail products quickly and accurately, making it suitable for real-time applications in retail environments. Despite achieving 98.5% mAP accuracy, this study also faced limitations due to the use of datasets on a limited scale.

Wang *et al*. [12] proposed an improved Siamese neural network to identify products through one-shot learning. First, a spatial channel dual attention mechanism is introduced to improve the network architecture. Second, a binary cross-entropy loss function with a distance penalty is adopted to replace the conventional contrastive loss function. The proposed network can better model the details of the products. Experimental results from two publicly available databases demonstrate that the proposed method outperforms conventional approaches and effectively addresses data insufficiency during the training stage. The drawback of this study is that the implementation of the dual attention mechanism and the new loss function can increase computational complexity and resource requirements, resulting in longer computation times.

III. Materials and Methods

This research aims to design a retail product detection system using the YOLOv8 model and the RT-DETR transformer model, utilizing three public datasets: the grocery dataset [13], the Retail Product Checkout (RPC) dataset [14], and the Densely Segmented Supermarket (D2S) dataset [15]. The integration of these two models involves removing the detect part from the YOLOv8 architecture and replacing it with the 'decoder head' architecture from RT-DETR.

A. Dataset

The dataset used in this research was not created by the authors, but we used a public dataset. There are three public datasets: the grocery dataset, the Retail Product Checkout (RPC) dataset, and the Densely Segmented Supermarket (D2S) dataset (see Fig. 1).

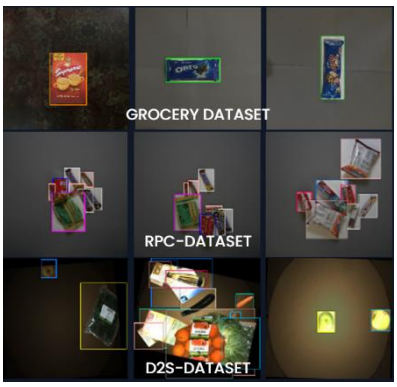

Fig. 1. Sample images of the grocery dataset (top), RPC dataset (middle), and D2S dataset (bottom).

1) Grocery dataset

Grocery dataset [13] is a dataset that focuses on a single product to conduct validation tests. The grocery dataset has 33,919 images with a partition 85% for training, 10% for validation, and 5% for testing. This dataset has almost the same features and colors, so it is suitable for use as a test material to overcome intra-class variation.

2) RPC-dataset

The RPC-Dataset was developed by Wei *et al.* [14] RPC-Dataset is challenging to detect large-scale retail products in a very large number of classes, there are 200 classes. This dataset provides 83,699 images with a partition of 70% for training, 20% for validation, and 10% for testing.

3) D2S-dataset

D2S-Dataset [15] provides data that shows in terms of lighting and product stacking. This dataset only has a total of 3,729 images with a partition of 70% for training, 20% for validation, and 10% for testing. This dataset is very challenging because there are almost all similar products,

the lighting of the images in various kinds of lighting, and the products are stacked.

B. You Only Look Once version 8 (YOLOv8)

YOLOv8 is the latest version of the YOLO model variant released in early 2023. YOLOv8 boosts the accuracy of real-time object detection [16]. It is adapted from the YOLOv5 [17] model by taking features of Cross Stage Partial (CSP), feature fusion method Path Aggregation Network-Feature Pyramid Network (PAN-FPN), and Spatial Pyramid Pooling Fusion (SPPF) modules [18]. It introduces P5 640 and P6 1280 resolutions and features instance segmentation based on you Only Look at Coefficients (YOLACT) [19]. With n/s/m/l/x scaling like YOLOv5 [17], the model can adapt to various situations. Enhancements to the backbone network and neck module are inspired by YOLOv7's [20] Efficient Layer Aggregation Network (ELAN) concept [21], with YOLOv5's C3 module replaced by C2f. However, this C2f module has operations such as split and concat that are less practical. The head module is also updated with separate structures for classification and detection, and a transition from anchor-based to anchor-free. Loss calculation uses taskalignedassigner and includes Distribution Focal Loss (DFL) in the regression loss [18]. Overall, YOLOv8 incorporates concepts from recent algorithms such as [22, 23], and brings various improvements to the model structure, loss calculation, training strategy, model inference process, and data augmentation (see Fig. 2).

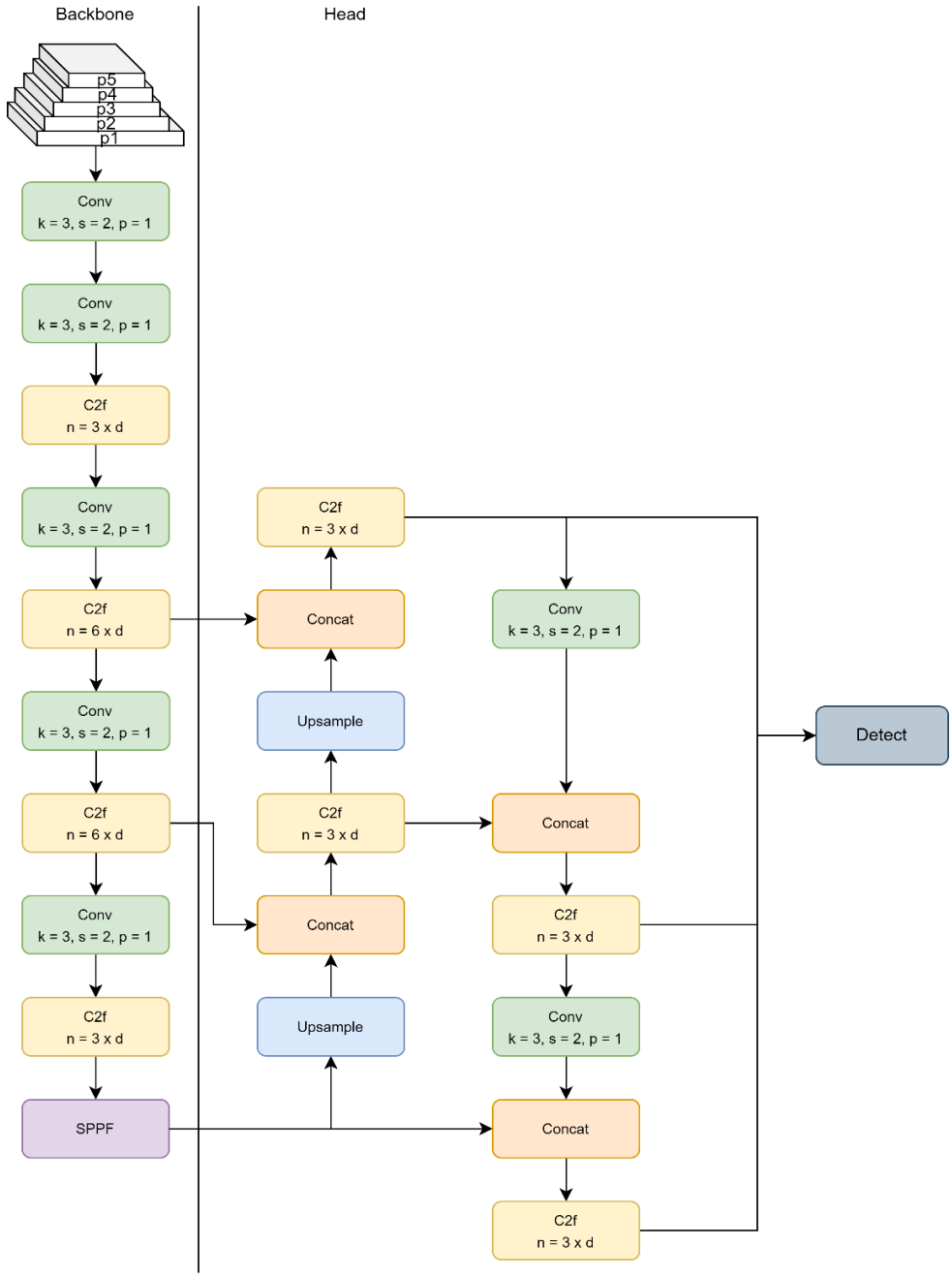

Fig. 2. YOLOv8 original architecture.

C. Real-Time Detection Transformer (RT-DETR)

RT-DETR [24] (Fig. 3) is the latest transformer model optimized for real-time object detection, offering speed and efficiency without sacrificing accuracy. Compared to DETR (Detection Transformer) [25], which uses backbones like Residual Networks (ResNet) [26] to extract rich features but suffers from high latency, RT-DETR employs lighter and faster backbones such as Efficient-Net [27] or MobileNet [28], This makes RT-DETR ideal for real-time applications like video surveillance and autonomous vehicles. The RT-DETR architecture includes a backbone for feature extraction, a transformer encoder [7] for encoding spatial and contextual information, a transformer decoder [7] for generating bounding box predictions and class labels, and a prediction head for the final predictions. With latency optimization and efficient design, RT-DETR provides superior performance in fast and efficient object detection compared to Detection Transformer (DETR).

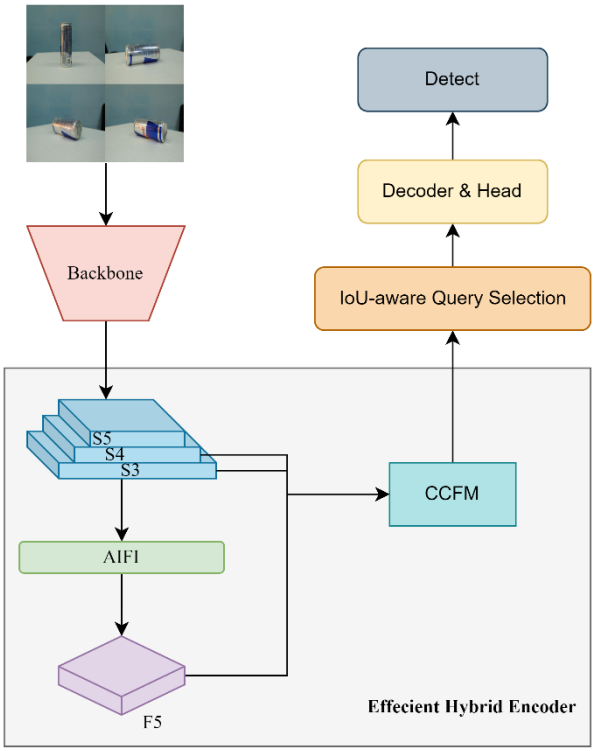

Fig. 3. RT-DETR original architecture.

D. YOLO-RTDETR

YOLOv8 is still lacking for small object detection so it is not optimal when product recognition. With the RT-DETR [24], detection performance improves because the RT-DETR Decoder convolves the deformed image to recognize features previously identified by YOLOv8. This allows the model to detect objects with maximum accuracy.

1) Decoder head of RT-DETR

The decoder [29] in RT-DETR consists of several main components that work together to generate accurate predictions in object detection, as shown in Fig. 4. These main components include Self-Attention, Cross-Attention, and the Feed-Forward Network (FFN) [30]. Self-Attention employs Multi-Head Attention, Dropout, and LayerNorm to capture relationships between elements within the input sequence. Multi-head attention [30] enables the model to attend to various aspects of the input simultaneously, while Dropout prevents overfitting and LayerNorm maintains training stability by normalizing the output from the previous layer.

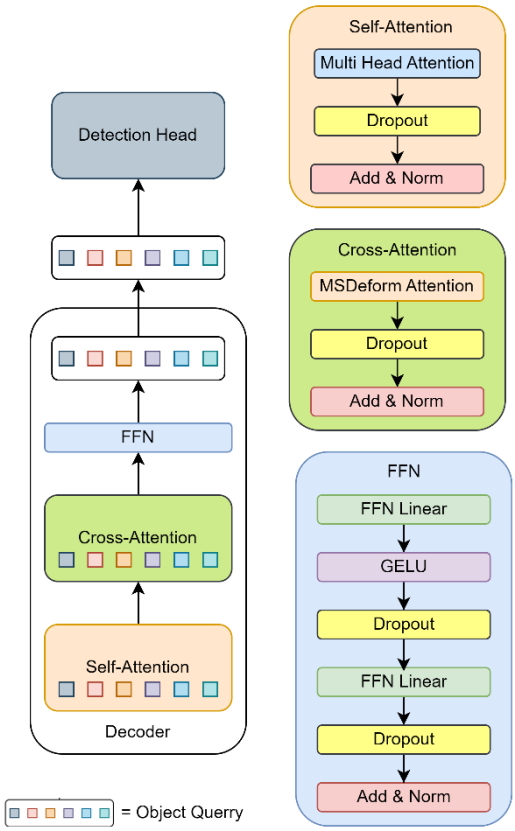

Fig. 4. The main components of the decoder include self-attention in the first component, cross-attention in the second component, and a Feed-Forward Network (FFN) layer in the final component.

Cross-attention in the RT-DETR decoder uses Multi-Scale Deformable Attention, Dropout, and LayerNorm to focus on significant features from the input sequence based on information from the encoder. Multi-scale deformable Attention [31] allows the model to concentrate on more relevant areas across different scales, enhancing detection accuracy. Dropout is applied to reduce overfitting, and LayerNorm ensures stable and consistent output. Following this, the FFN [30] consisting of two linear layers with a non-linear activation in between, processes the output from cross-attention to generate the final prediction. This FFN helps integrate the information obtained from self-attention and cross-attention to produce a more representative and accurate output in object detection tasks.

2) Residual block

A residual block [32] is a crucial component in neural network architectures that helps address gradient vanishing and gradient explosion issues, particularly in very deep networks. It allows a direct path (skip

connection) from the input to the output, enabling information from previous layers to be directly added to subsequent layers without passing through all intermediate layers [26]. This design preserves gradients by allowing them to flow directly from the output to the input, reducing the risk of gradient vanishing in deep networks. Additionally, residual blocks accelerate convergence by simplifying the network's learning of identity mapping and help reduce overfitting by incorporating dropout, making the network more robust against noise in the training data [26]. The basic formula for the residual block can be seen in Eq. (1)

$$y = F(x, \{W_i\}) + x \tag{1}$$

where y is the output of the residual block, x is the original input. The function $F(x, \{W_i\})$ is the transformation function performed by several layers (e.g., convolutional layers, activation, and normalization) with parameters $(\{W_i\})$. The (x) is directly added to the result of the transformation F through skip connection. In this case, residual block was added to the decoder block after FFN. In the context of a FFN on a Transformer, based on Eq. (1) the residual block can be described in Eq. (2).

$$y = LayerNorm(x + Dropout(Linear_2 \left(Activation\left(Linear_1(x)\right)\right))) \tag{2}$$

where x is the input to FFN, $Linear_1$, and $Linear_2$ is the first and second linear in FFN. $Activation$ is a non-linear activation function Gaussian Error Linear Unit (GELU) [33]. $Dropout$ is a regularization mechanism to prevent overfitting and $LayerNorm$ is layer normalization to improve training stability. From the explanations above, we added a residual block after the FFN which can be seen in Fig. 5.

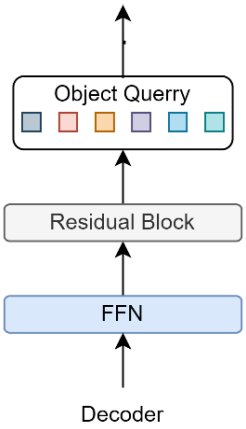

Fig. 5. The residual block is added after the FFN component.

With the addition of residual blocks, retail product detection with intra-class variation [8] will become more accurate because residual blocks help maintain more stable gradients during the training process. Residual blocks allow information from previous layers to be directly added to subsequent layers, making it easier for the network to learn relevant features for detecting variations of products within the same class. Additionally, residual blocks [32] reduce the risk of overfitting by preventing the network from becoming too deep without significantly increasing complexity. Consequently, the model can capture small variations in retail products, such as differences in shape, color, or texture, which are often challenging in object detection tasks.

3) Combining YOLOv8 and RT-DETR

YOLOv8 is renowned for its real-time inference speed and efficiency, while RT-DETR excels in detecting objects with multi-scale deformable attention for higher accuracy. In this approach, the YOLOv8 backbone is used for rapid feature extraction, and these features are then passed to the RT-DETR decoder [29], replacing the standard detect layer. The combination of the two models can be seen in Fig. 6.

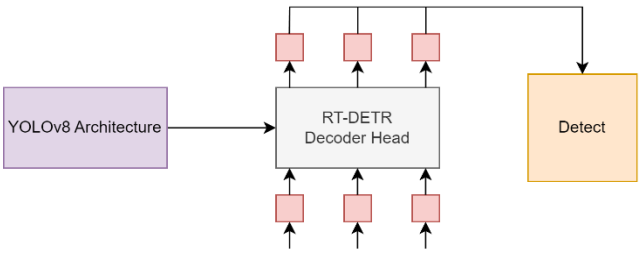

Fig. 6. YOLOv8 combined to decoder Head of RT-DETR.

E. Intra-Class Variation

Intra-class variation [8] (Fig. 7) refers to the degree of variation or difference that exists between members of the same class or group within a dataset. In the context of pattern recognition or classification, this concept relates to the extent to which data within a class can vary in terms of certain traits or attributes. When applied to object detection problems, such as in smart cart systems, intra-class variation refers to visual or attribute fluctuations found among products belonging to a uniform product category. Intra-class differences can arise through several factors, namely: brand, size, and variant.

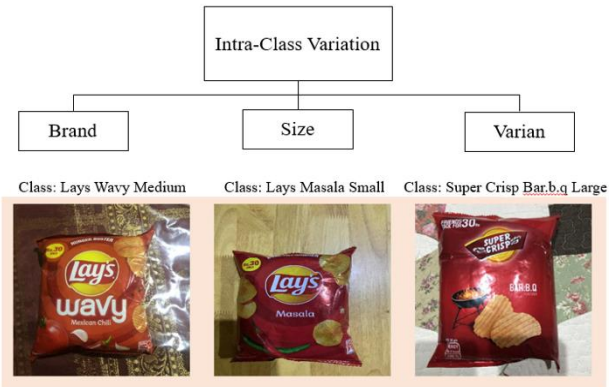

Fig. 7. Intra-class variations.

F. Performance Evaluation Metrics

In this research, the performance of the model will be evaluated using precision, recall and mAP. Precision is an

evaluation metric that measures how often a model predicts data correctly. The formula for precision can be seen in Eq. (3).

$$Precession = \frac{TP}{TP+FP} \quad (3)$$

Based on Eq. (3), the following information is provided. True Positive (TP) represents a condition where the data classified as true has a true actual answer as well and False Positive (FP) represents a condition where the data classified as true but has a false actual answer.

Recall is defined as the ratio of TP to the total number of actual positive data. The recall calculation can be seen in Eq. (4).

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Based on Eq. (4), False Negative (FN) represents a condition where the data is classified as false but the actual answer is true.

Mean average precision is an important evaluation metric for measuring the performance of a detected object. The mapped value is the average of the Average Precision (AP) values for each class. The mAP calculation is based on the comparison of ground truth data with the bounding box data generated by the detection system. The mAP calculation can be seen in Eq. (5).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (5)$$

where, N is the total number of tested classes, i is the iteration starting from 1, and AP is the average precision.

IV. Result and Discussion

For this experiment, alternate trials of each type of dataset were conducted. This experiment was conducted using an DGX DGX A100 device to perform heavy computation. This section will discuss the results of the experiments carried out.

A. Experiment Preparation

In this experiment, we tested 3 types of datasets. These datasets were tested with the model we created by combining YOLOv8 with the RT-DETR decoder head. For the grocery [13] and RPC datasets [14], 50 epochs were set for training because the datasets are large enough to build a proper learning model. For D2S-Dataset [15], 150 epochs are given because this dataset has a total of only 3,729 images from the results of image preprocessing augmentation and also the dataset tested is a dataset that has different types of lighting and stacked products.

B. Grocery Dataset Result and Discussion

Fig. 8 displays the prediction data that has been labeled before and the results of the prediction have an average confidence score of 90%. Fig. 9 is the overall result by looking at the stable training results with high accuracy. These results of each training conducted and can be seen if these results can be said to be good fitting. Table I shows the results of several approaches that can be compared with our approach. It can be seen if the approach we made can be superior to several approaches that have been done. The results of YOLOv8-RTDETR for precision reached 99.8%, the recall performance level reached 99.9% and the result can be seen from the mAP reaching 99.5%. Other YOLO models such as YOLOv8, YOLOv8-Ghost, and YOLOv8-P2 also show excellent performance with precision and recall of 99.8%. However, their mAP50 is slightly lower than YOLOv8-RTDETR, with values of 99.4% and 99.3%, respectively. Although the difference is small, it confirms that YOLOv8-RTDETR has a slight edge in terms of detection. RetinaNet with ResNet50 and ResNet101 backbones shows decent performance but does not match up to the YOLOv8 models. RetinaNet (ResNet101) has higher precision and recall (93.6% and 93.5%) compared to ResNet50, which has 80.2% precision and 81% recall. The mAP50 of ResNet101 (92.5%) is also higher than that of ResNet50 (72.3%). Despite the better performance of RetinaNet (ResNet101), the YOLOv8 models significantly outperform it in all metrics. YOLOv5 demonstrates good performance with 89.4% precision, 88.2% recall, and 88.1% mAP50. However, compared to YOLOv8 and its variants, YOLOv5's performance lags. This indicates that the development from YOLOv5 to YOLOv8 has brought significant improvements in terms of accuracy and detection. These three performances beat some of the performances of other approaches, so our approach achieves very good results and can be used to detect multiple retail products, but this detection only focuses on a single product and does not include multiple products in one image with different lighting levels.

Fig. 8. Predict the result for grocery-dataset.

TABLE I. RESULT FOR GROCERY-DATASET

Approach	Precision (%)	Recall (%)	mAP50 (%)
YOLOv5 [11]	89.4	88.2	88.1
RetinaNet (ResNet50) [34]	80.2	81	72.3
RetinaNet (ResNet101) [34]	93.6	93.5	92.5
YOLOv8 [35]	99.8	99.8	99.4
YOLOv8-Ghost [35]	99.8	99.8	99.4
YOLOv8-P2 [35]	99.8	99.8	99.3
YOLOv8-RTDETR	99.8	99.9	99.5

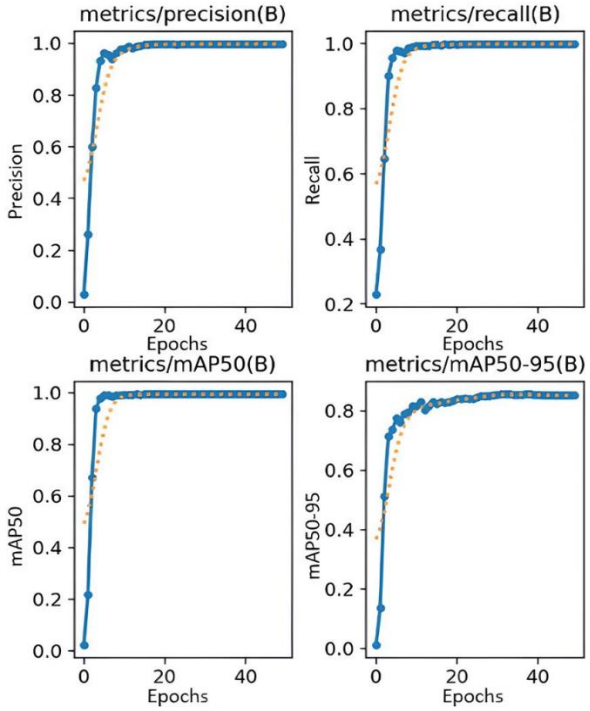

Fig. 9. Evaluate metrics of grocery-dataset.

C. RPC-Dataset Result and Discussion

Fig. 10 displays the prediction data that has been labeled before and the results of the prediction have an average confidence score of 90%. Fig. 11 is the overall result by looking at the stable training results with high accuracy. These results are the results of each training conducted and it can be seen if these results can be said to be good fitting. Table II shows the results of several approaches that can be compared with our approach. It can be seen if the approach we made can be superior to several approaches that have been done. YOLOv8-RTDETR shows exceptional performance with a mAP50 of 99.3% and a mAP50-95 of 86.4%. This makes it one of the best models in terms of overall accuracy across both metrics. The high mAP50-95 indicates that YOLOv8-RTDETR maintains excellent performance across a wide range of Intersection of Union (IoU) thresholds, making it highly reliable for various object detection tasks. Other YOLOv8 variants such as YOLOv8, YOLOv8-Ghost, and YOLOv8-P2 also demonstrate strong performance. YOLOv8-Ghost stands out slightly with a mAP50-95 of 86.5%, marginally higher than YOLOv8-RTDETR. YOLOv8 and YOLOv8-P2 have slightly lower mAP50-95 scores of 86.1% and 85.9%, respectively. Despite these minor differences, all YOLOv8 variants perform very well, confirming the robustness of the YOLOv8 architecture. Syn+Render and Coarse to Fine Grid Feature Extraction Network (CGFENet): Syn+Render models also perform admirably with mAP50 scores of 99.3% and 99.1%, respectively. However, Content Generative Framework for Efficient Network: Synthesis and Rendering (CGFENet: Syn+Render) excels in the mAP50-95 metric with a score of 86.2%, indicating strong performance across various Intersections Over Union (IoU) thresholds. The standard Syn+Render model, while having a high mAP50, falls behind in the mAP50-95 metric with a score of 84.1%. Baseline: Syn shows significantly lower performance with a mAP50 of 81.5% and a mAP50-95 of 56.3%. This indicates that this model is less effective compared to the other approaches, particularly across a range of IoU thresholds. The large gap in mAP50-95 suggests that Baselane: Syn struggles to maintain accuracy when the IoU threshold increases. YOLOv8-RTDETR emerges as one of the top-performing models, particularly excelling in the mAP50-95 metric, which highlights its robustness across different IoU thresholds. While all YOLOv8 variants show strong performance, YOLOv8-RTDETR and YOLOv8-Ghost are particularly noteworthy. Overall, YOLOv8-RTDETR stands out as a reliable and highly accurate model for object detection, capable of maintaining high performance across various IoU thresholds.

TABLE II. RESULT FOR RPC-DATASET

Approach	mAP50 (%)	mAP50-95 (%)
Syn+Render [14]	**99.3**	84.1
CGFENet: Syn+Render [14]	99.1	86.2
Baseline: Syn [14]	81.5	56.3
YOLOv8 [35]	99.2	86.1
YOLOv8-Ghost [35]	99.2	**86.5**
YOLOv8-P2 [35]	99.1	85.9
YOLOv8-RTDETR	**99.3**	86.4

Fig. 10. Predict the result for RPC-dataset.

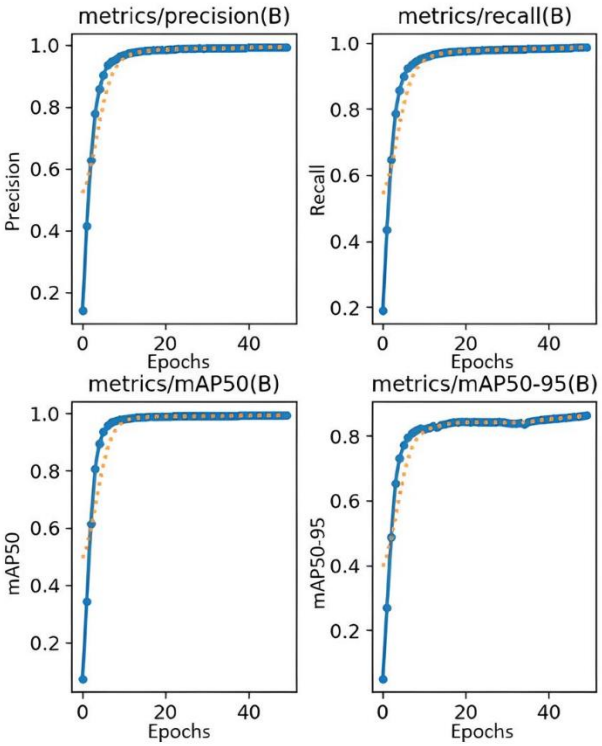

Fig. 11. Evaluate metrics of RPC-dataset.

metrics. The results from this dataset are smaller than the results from the previous dataset because the tested dataset has data with poor lighting, piled up products, and products that have similar characteristics.

Fig. 12. Predict the result for D2S-dataset.

D. D2S-Dataset Result and Discussion

In Fig. 12 displays the prediction data that has been labeled before and the results of the prediction have an average confidence score of 75%. Fig. 13 is the overall result by looking at the stable training results with high accuracy. Although there is an insignificant increase due to the dataset, the learning results in each epoch increase. Table III shows the results of several approaches that can be compared with our approach. YOLOv8-RTDETR demonstrates the best performance among all the models tested, with a mAP50 of 85.5% and a mAP50-95 of 61.9%. This indicates superior accuracy and robustness across different Intersections Over Union (IoU) thresholds. The high mAP50-95 score highlights YOLOv8-RTDETR's ability to maintain performance even with stricter IoU requirements. Other YOLOv8 variants also show strong performance. YOLOv8-P2 achieves a mAP50 of 82.1% and a mAP50-95 of 58.7%, which is slightly better than YOLOv8 and YOLOv8-Ghost. YOLOv8 has a mAP50 of 81.9% and a mAP50-95 of 58.2%, while YOLOv8-Ghost records 80.3% for mAP50 and 57.4% for mAP50-95. Despite these variations, all YOLOv8 models significantly outperform the other models, emphasizing the robustness and effectiveness of the YOLOv8 architecture. Traditional models such as Mask R-CNN, FCIS, and Faster R-CNN show lower performance compared to the YOLOv8 models. Mask R-CNN and Fully Convolutional Instance Segmentation (FCIS) both have a mAP50 of approximately 57-58% and a mAP50-95 of 51.3%. Faster R-CNN has the lowest performance with a mAP50 of 55.2% and a mAP50-95 of 49.7%. These models, while popular and widely used, lag significantly behind the YOLOv8 models in terms of both mAP50 and mAP50-95

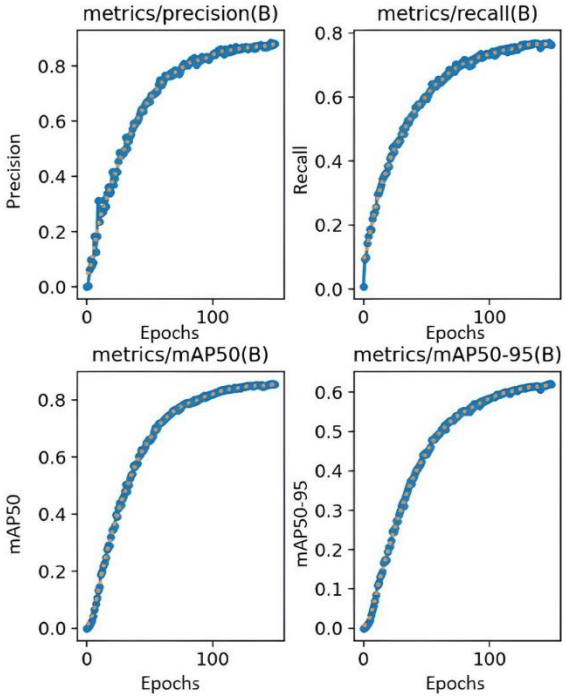

Fig. 13. Evaluate metrics of D2S-dataset.

TABLE III. RESULT FOR D2S-DATASET

Approach	mAP50 (%)	mAP50-95 (%)
Mask R-CNN [36]	57.6	51.3
FCIS [37]	58.3	51.3
Faster R-CNN [10, 38]	55.2	49.7
YOLOv8 [35]	81.9	58.2
YOLOv8-Ghost [35]	80.3	57.4
YOLOv8-P2 [35]	82.1	58.7
YOLOv8-RTDETR	**85.5**	**61.9**

V. Conclusion

Retail products are detected using the intra-class variation technique in the YOLOv8-RTDETR model. This model can detect products that are similar in terms of brand, size, and variety from each type of dataset that has been tested. Across the three datasets analyzed, namely the Grocery Dataset, RPC-Dataset, and D2S-Dataset, the performance of various object detection models was thoroughly evaluated. In the Grocery Dataset analysis, YOLOv8-RTDETR emerged as the best model, achieving almost perfect precision, recall, and mAP50 scores, indicating its exceptional accuracy and reliability for grocery item detection. Other YOLOv8 variants also performed extremely well, showcasing the strength of the YOLOv8 architecture, while traditional models like RetinaNet showed good performance but were significantly outperformed by the YOLOv8 variants.

In the RPC-Dataset analysis, YOLOv8-RTDETR and YOLOv8-Ghost demonstrated the highest performance with excellent mAP50 and mAP50-95 scores, indicating their robustness and reliability in detecting retail products. CGFENet: Syn+Render also performed well, particularly in the mAP50-95 metric, highlighting its capability across a range of IoU thresholds. Conversely, Baseline: Syn underperformed significantly, showing that it is less effective compared to the other models tested.

The D2S-Dataset analysis revealed that YOLOv8-RTDETR again showed the best performance, with the highest mAP50 and mAP50-95 scores, indicating its superior accuracy and consistency for detecting small objects in various contexts. Other YOLOv8 variants also outperformed traditional models, showcasing the advancements in the YOLOv8 architecture. Traditional models like Mask R-CNN, FCIS, and Faster R-CNN had lower performance, particularly in the mAP50-95 metric, indicating less robustness across different IoU thresholds.

Overall, across all three datasets, YOLOv8-RTDETR consistently demonstrated the highest performance, making it the most reliable and accurate model for object detection tasks. The YOLOv8 variants, in general, showed significant improvements over traditional models, highlighting the advancements and robustness of the YOLOv8 architecture. Therefore, YOLOv8-RTDETR becomes the preferred choice for applications that demand high precision and recall in diverse object detection scenarios.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

S.A.W. provides ideas, then validates and organizes the manuscript. A.W.M. provides datasets, performs coding, and analyzes results. U.S. provides datasets and organizes the manuscript. R.R. does code and testing. A.I. performs testing. All authors had approved the final version.

Acknowledgment

This work was supported by the RIIM Batch 3 program grant funded by LPDP, Ministry of Finance, Rep. of. Indonesia, and BRIN (no. 81/IV/KS/05/2023) and was supported by Telkom University.

References

[1] S. Xu, J. Wang, W. Shou, T. Ngo, A. M. Sadick, and X. Wang, "Computer vision techniques in construction: A critical review," *Arch. Comput. Methods Eng.*, vol. 28, no. 5, pp. 3383–3397, 2021. doi: 10.1007/s11831-020-09504-3

[2] A. C. Septadarman and R. A. Rambe, "Analysis of the influence of population growth, education, and health on poverty in Indonesia from 2018 to 2022," *East Asian J. Multidiscip. Res.*, vol. 3, no. 1, pp. 129–142, 2024.

[3] L. Alzubaidi *et al.*, *Review of Deep Learning: Concepts, CNN* Architectures, *Challenges*, *Applications*, *Future Directions*, Springer International Publishing, vol. 8, no. 1. 2021. doi: 10.1186/s40537-021-00444-8

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

[5] M. Maity, S. Banerjee, and S. S. Chaudhuri, "Faster R-CNN and YOLO based vehicle detection: A survey," in *Proc. 2021 5th* International *Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1442–1447. doi: 10.1109/ICCMC51019.2021.9418274

[6] S. A. Magalhães *et al.*, "Evaluating the single-shot multi-box detector and yolo deep learning models for the detection of tomatoes in a greenhouse," *Sensors*, vol. 21, no. 10, pp. 1–24, 2021, doi: 10.3390/s21103569

[7] K. Islam, "Recent advances in vision transformer: A survey and outlook of recent work," arXiv preprint, arXiv:2203.01536, 2022.

[8] A. Parashar, R. S. Shekhawat, W. Ding, and I. Rida, "Intra-class variations with deep learning-based gait analysis: A comprehensive survey of covariates and methods," *Neurocomputing*, vol. 505, pp. 315–338, 2022.

[9] B. Santra, A. K. Shaw, and D. P. Mukherjee, "Part-based annotation-free fine-grained classification of images of retail products," *Pattern Recognit.*, vol. 121, 108257, 2022.

[10] C. H. Hsia, T. H. Chang, C. Y. Chiang, and C. H. Tse, "Mask R-CNN with new data augmentation features for smart detection of retail products," *Appl. Sci.*, vol. 12, 2902, 2022.

[11] R. Y. Lee, S. Y. Chua, Y. L. Lai, T. Y. Chai, S. Y. Wai, and S. C. Haw, "Cashierless checkout vision system for smart retail using deep learning," *J. Syst. Manag. Sci.*, vol. 12, no. 4, pp. 232–250, 2022. doi: 10.33168/JSMS.2022.0415

[12] C. Wang, C. Huang, X. Zhu, and L. Zhao, "One-shot retail product identification based on improved Siamese neural networks," *Circuits, Syst. Signal Process.*, vol. 41, no. 11, pp. 6098–6112, 2022.

[13] New-workspace-wfzw3, grocery DATASET dataset. [Online]. Available: https://universe.roboflow.com/new-workspace-wfzw3/grocery-dataset-q9fj2

[14] X.-S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu. (2019). RPC: A large-scale retail product checkout dataset. [Online]. Available: http://arxiv.org/abs/1901.07249

[15] P. Follmann, T. Böttger, P. Härtinger, R. König, and M. Ulrich, "MVTec D2S: Densely segmented supermarket dataset," *Lect. Notes* Comput. *Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, LNCS, vol. 11214, pp. 581–597, 2018.

[16] J. Terven and D. C. Esparza, "A comprehensive review of YOLO: From YOLOv1 and beyond," arXiv preprint, arXiv:2304.00501, pp. 1–34, 2023.

[17] M. Jani, J. Fayyad, Y. A. Younes, and H. Najjaran, "Model compression methods for YOLOv5: A review," arXiv preprint, arXiv:2307.11904, 2023.

[18] X. Wang, H. Gao, Z. Jia, and Z. Li, "BL-YOLOv8: An improved road defect detection model based on YOLOv8," *Sensors*, vol. 20, 2023.

[19] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9157–9166.

[20] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 7464–7475.
[21] Y. Zeng, T. Zhang, W. He, and Z. Zhang, "YOLOv7-UAV: An unmanned aerial vehicle image object detection algorithm based on improved YOLOv7," *Electron.*, vol. 12, no. 14, pp. 1–14, 2023. doi: 10.3390/electronics12143141
[22] C. Li *et al.*, "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint, arXiv:2209.02976, 2022.
[23] S. Xu *et al.*, "PP-YOLOE: An evolved version of YOLO," arXiv preprint, arXiv:2203.16250v3, pp. 1–7, 2022.
[24] W. Lv *et al.*, "Detrs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16965–16974.
[25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. European Conference on Computer Vision*, 2020, pp. 213–229.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. International Conference on Machine Learning*, 2019, pp. 6105–6114.
[28] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint, arXiv1704.04861, 2017.
[29] Y. Li, Y. Lin, T. Xiao, and J. Zhu, "An efficient transformer decoder with compressed sub-layers," in *Proc. AAAI Conference on Artificial Intelligence*, 2021, pp. 13315–13323.
[30] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
[31] Y. Liu *et al.*, "Exploring multi-scale deformable context and channel-wise attention for salient object detection," *Neurocomputing*, vol. 428, pp. 92–103, 2021.
[32] S. Zagoruyko and N. Komodakis, "Wide residual networks," arXiv preprint, arXiv1605.07146, 2016.
[33] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint, arXiv1606.08415, 2016.
[34] H. Zhang, H. Chang, B. Ma, S. Shan, and X. Chen, "Cascade RetinaNet: Maintaining consistency for single-stage object detection," arXiv preprint, arXiv1907.06881, 2019.
[35] G. Jocher, A. Chaurasia, and J. Qiu. (2023). Ultralytics YOLOv8. [Online]. Available: https://github.com/ultralytics/ultralytics
[36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE international conference on* computer *vision*, 2017, pp. 2961–2969.
[37] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.
[38] R. Girshick, "Fast R-CNN," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.