ENHANCING RETAIL PRODUCT RECOGNITION USING MODIFIED YOLOV8 AND SELF-SUPERVISED LEARNING

FELIX CORPUTTY[1,2], SURYO ADHI, WIBOWO[1,2,*], UNANG SUNARYA[3], RISSA RAHMANIA[4], SIDDIQ WAHYU HIDAYAT[5]

[1]School of Electrical Engineering, Telkom University, Bandung, Rep. of. Indonesia
[2]Center of Excellence Artificial Intelligence for Learning and Optimization, Telkom University, Bandung, Rep. of. Indonesia
[3]School of Applied Science, Telkom University, Bandung, Rep. of. Indonesia
[4]Computer Science Department, School of Computer Science, Bina Nusantara University, Bandung Campus, Jakarta, Rep. of. Indonesia
[5]National Research and Innovation Agency, BRIN, Jakarta, Rep. of. Indonesia
[*]Corresponding Author

E-mail: [1]felixcorputty@student.telkomuniversity.ac.id, [2]suryoadhiwibowo@telkomuniversity.ac.id, [3]unangsunarya@telkomuniversity.ac.id, [4]rissa.rahmania@binus.ac.id, [5]siwahid@gmail.com

ABSTRACT

Artificial intelligence has several parts, one of them is computer vision. Computer vision is a technology that allows computers to recognize objects as humans do. Computer vision has been widely applied in various applications, one of an example is in retail product recognition. However, the current computer vision technology is still difficult to distinguish between one product and another in the same category known as intra-class variation. Therefore, this research developed an algorithm that uses the concept of computer vision to be able to distinguish one product and another in the same category. The research was conducted using two stages. In the first stage the dataset was trained using YOLOv8. There are four experiments conducted using YOLOv8, namely YOLOv8 original, YOLOv8 with 4 detection heads (YOLOv8-4DH), YOLOv8 with additional convolutional layer and C2f layer on the backbone (YOLOv8-Conv) and the last is YOLOv8 with 4 detection heads, additional convolutional layer and C2f layer on the backbone (YOLOv8-Conv-4DH). The best model is selected based on the highest mAP value. The model with the highest mAP value is YOLOv8-4DH at 91%. The best model is used to crop the image to be used as input in the second stage. In the second stage, the cropped image is trained using SimCLR. The training weights from SimCLR are stored and loaded back into the SimCLR model for training and evaluation. The results of the second stage showed that the best model YOLOv8-4DH combined with SimCLR algorithm got an accuracy of 97.76%.

Keywords: *Artificial Intelligence, Computer Vision, Retail Product, SimCLR, YOLOv8*

1. INTRODUCTION

Artificial intelligence (AI) currently has a very significant development. There are many main components of AI, for example computer vision. Computer vision is a technology that allows computers to see and recognize objects around them like how human does. Technology that applies the concept of computer vision has been widely used in several things such as fruit detection [1], face tracking [2] and autonomous driving [3]. The application of computer vision is used to recognize products in one class in the retail industry. Amazon dash cart [4][10] is a computer vision-based technology in the retail product industry created by the amazon company using a cashless system "Just Walk Out". Amazon Dash Cart was first used in the amazon company's self-service store, Amazon Go. With a variety of computer vision technologies that already exist and are used in the retail product industry, the application of computer vision in the retail product industry still has several problems. One of the problems often seen in retail product recognition is the similarity between one product and another in the same category, making it difficult for models to distinguish these products. The similarity between one product and another in the same category is called intra-class variation [5][33].

Based on the problems mentioned previously, our contribution for this research is focused on designing retail product recognition using the YOLOv8 algorithm and a self-supervised learning algorithm called SimCLR. There are two stages carried out in this research. The first stage is YOLOv8 process, namely the dataset was trained on

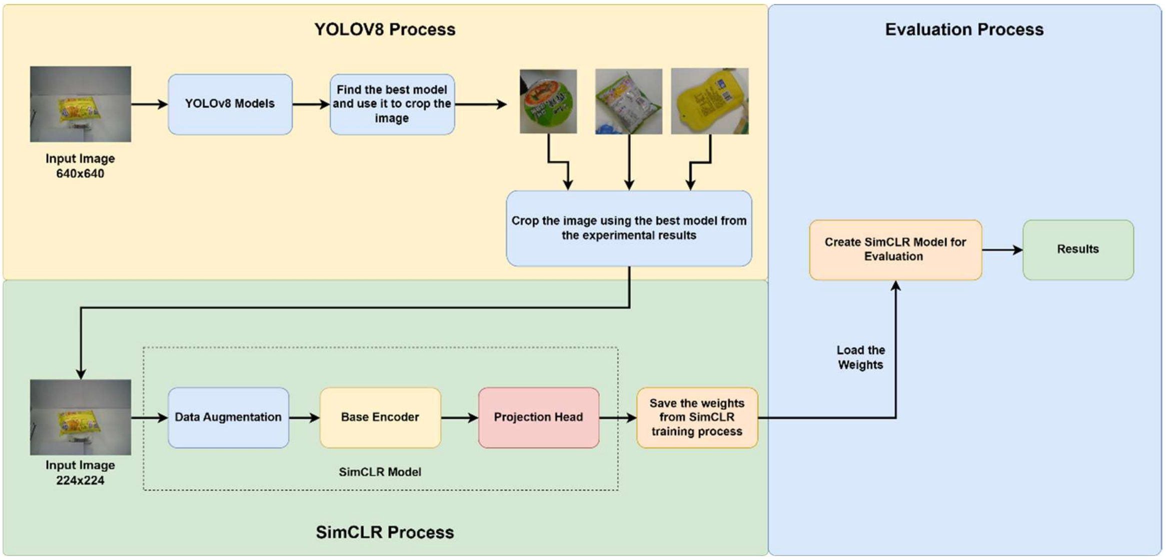
YOLOV8 Process
Input Image 640x640
YOLOv8 Models
Find the best model and use it to crop the image
Crop the image using the best model from the experimental results
Evaluation Process
Create SimCLR Model for Evaluation
Results
Load the Weights
Input Image 224x224
Data Augmentation
Base Encoder
Projection Head
Save the weights from SimCLR training process
SimCLR Model
SimCLR Process

Figure 1: Block Diagram of the Proposed Method

several algorithms built from YOLOv8 then the model is selected based on the highest mAP value. The second stage is SimCLR process, namely the image generated using the best model from the training results using YOLOv8 used in SimCLR for the training and evaluation process. There are four experiments using the YOLOv8 algorithm in the first stage, namely the original YOLOv8, YOLOv8 with 4 detection heads (YOLOv8-4DH), YOLOv8 with additional convolutional layer and c2f layer on the backbone (YOLOv8-Conv), lastly is combination method of the second and the third method (YOLOv8-Conv-4DH). The selected model, the best one from the experimental results of several models using YOLOv8 in the first stage is than applied to crop the images based on the training result. Subsequently, the cropped images are used as the training input for the SimCLR algorithm in the second stage. Finally, weights of the training process are stored and reused into the SimCLR algorithm for training and evaluating the images.

The structure of this paper is divided to five sections; section 2 discusses the literature review related to retail product recognition. Section 3 contains the theoretical basis of the method used and the proposed method in the research. Section 4 contains the results of the experiments conducted in this study. Section 5 discusses the conclusions of the research results.

2. RELATED WORK

The application of computer vision in the retail product industry has been applied in many previous studies to solve the problem that is often seen in retail products, namely intra-class variation. Wang *et al.* proposed an improved siamase neural network to perform the task of oneshot retail product identification, meaning that for each product identified only one image used for training. Besides that, in order to enhance feature extraction method, the dual attention technique is added into proposed algorithm while binary cross-entropy function with an euclidean penalty function [6]. Furthermore, the method could be an effective solution for increasing stock keeping unit (SKU) categories because SKUs could change at any time. The research conducted by Santra *et al.* uses two stages, the first using the reconstruction classification network (RCNet) which is basically a deep supervised convolutional auto encoder (SCAE) similar to the supervised auto encoder (SAE) [7]. RCNet will serve as a classifier and assist in product recognition under various kinds of lighting in the store [7]. The second stage in [7] is to improve the classification performance of the first stage on the discriminative part of the searched products (in an unsupervised way) and organized as an ordered sequence to uniquely describe the products. The sequence is modeled using convolutional LSTM (ConvLSTM) and the classification results from both stages are merged and determine the product label of the test image [7]. In the research proposed by Jungjie Wang *et al.* using retail product identification and localization based on an improved convolutional neural network [8]. First, the model improves YOLOv3 by using group convolution and depth-separable convolution to optimize the backbone network structure and reduce the number of calculations [8]. Second, the multiscale structure is converted to two scales and k-means clustering is used to reorder 6 anchors of different sizes [8]. Lastly, spatial pyramid pooling (SPP) is introduced to replace pooling with convolution to effectively improve the robustness to image distortion caused by cropping and scaling and finally, a novel mosaic data enchancement method is used to enrich the data set and improve the network's robustness to small data [8]. Research by Wenyong Wang *et al.* proposed an improved fine-grained image classification method using the self-attention method [9]. The method used in [9] uses the self-attention destruction and construction learning (SADCL) method with the VGG-16 and Resnet-50 base models and the proposed method is used to calculate precise fine-grained classification predictions and large information areas in the reasoning process.

3. MATERIALS AND METHODS

This research aims to design a retail product recognition system using the YOLOv8 and the self-supervised learning algorithm called SimCLR. There are two types of datasets used in this research, namely RPC Dataset [11] and RPC Dataset which has been managed and simplified by Samrat Sahoo [12]. The RPC dataset simplified by Samrat Sahoo [12] is used for the first stage in this research while the RPC dataset [11] is used for the second stage. The first stage in this research is YOLOv8 process, namely the dataset is trained on several YOLOv8 models that have been prepared and the images is cropped using the best model. The second stage is SimCLR process, namely the SimCLR model trained using the images generated by the best model resulting from experiments using several YOLOv8 models then the weights from the training results are stored. The stored weights are reused into the SimCLR architecture for training and evaluating images from the RPC dataset [12]. Figure 1 shows the block diagram of the proposed method in this study.

3.1 Dataset

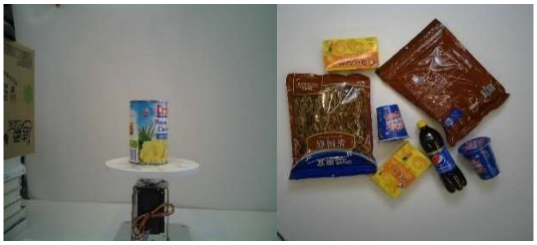

Figure 2: Sample images of RPC dataset - image for each product (left) and checkout images (right)

The datasets used in this research are RPC Dataset [11] and RPC dataset that has been managed and simplified by Samrat Sahoo [12]. RPC dataset [11] consists of 200 classes with a division of 53739 images for training data, 6000 images for validation data and 24000 images for test data while RPC dataset managed and simplified by Samrat Sahoo [12] consists of 17 classes with a division of 58589 images for training data, 16740 images for validation data, 8370 images for test data. RPC Dataset has several characteristics such as RPC dataset is the largest dataset for retail Automatic Checkout (ACO) in terms of product categories [11] which is almost double the size compared to the previous dataset [13]. The next characteristic is that it collects two types of images: images for each product and checkout images at the cashier [12]. Then the RPC dataset has a total of 200 classes and can be categorized into 17 meta-categories that cover a variety of appearances such as bottles, boxes, tubes, bags [12]. The next characteristic is that the dataset is made as similar as possible to mimic a retail payment scenario and the checkout images are divided into three different levels of clutter and lighting [12]. The last characteristic is that the checkout images of the RPC dataset come with three different types of annotations representing weak supervision to strong supervision [12]. For an example image of the RPC dataset, it can be seen in figure 2.

3.2 YOLOv8

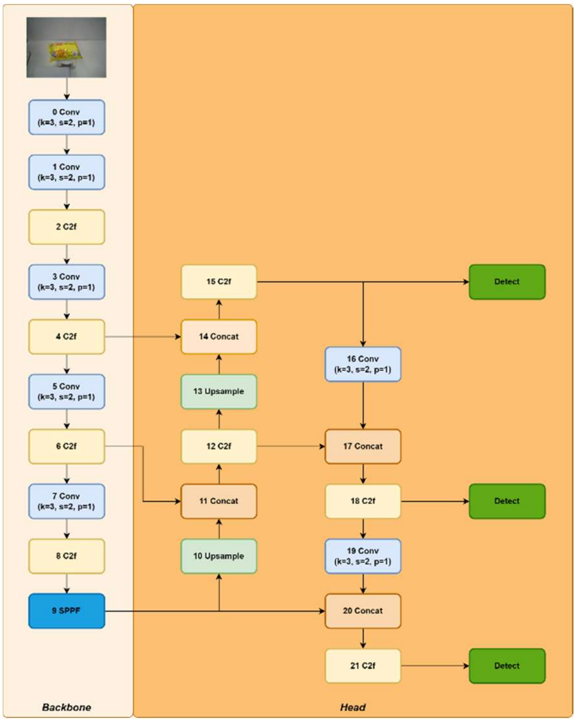

Figure 3: YOLOv8 Original Architecture

YOLOv8 [14] is a new real-time object detection algorithm for the YOLO series released in

2023 by ultralytics. YOLOv8 supports various vision tasks such as object detection and tracking, segmentation, pose estimation and image classification. YOLOv8 has several layers that build its network structure such as Convolutional, Concat, C2f, Upsample, SPPF layers. YOLOv8 uses the same backbone as YOLOv5 [15], namely CSPDarknet53 with some changes to the CSP layer which is now called the C2f module (cross-stage partial bottleneck with two convolutions). The C2f module learns from the idea of ELAN in YOLOv7 [17] and combines C3 and ELAN to form the C2f module. The C2f module combines high-level features with contextual information to improve detection accuracy [16]. The features that have been generated in the backbone section will be processed in the SPPF (Spatial Pyramid Pooling Fast) layer which includes a Convolutional layer and a Maxpooling layer and this SPPF serves to increase the computational speed by more than two times [24]. YOLOv8 uses a new architecture that combines Feature Pyramid Network (FPN) [18] and Path Aggregation Network (PAN) [19] modules. FPN is used to generate feature maps of various scales and resolutions while PAN is used to combine features from different levels of the network to produce accuracy [20].

3.2.1 yolov8-original

The first experiment in this research uses the original YOLOv8. Figure 3 shows the original architecture of the YOLOv8 algorithm. The original YOLOv8 architecture consists of 22 stages and 3 detection heads. The backbone section consists of 10 stages and the head section consists of 12 stages and 3 detection heads. The layers that build up the original YOLOv8 architecture consist of 7 Convolutional layers, 8 C2f layers, 4 Concat layers, 2 Upsample layers and 1 SPPF layer. The output of the original YOLOv8 architecture to go to the detect section comes from layer C2f on stages 14, 17 and 20.

3.2.2 yolov8-4dh

The second experiment in this research is YOLOv8 with 4 detection heads (YOLOv8-4DH). The architecture of YOLOv8-4DH can be seen in figure 4. The architecture of YOLOv8-4DH consists of 28 stages and 4 detection heads. The YOLOv8-4DH architecture is similar to the original YOLOv8 architecture but there are additions to the head part, namely on stages 16-21 of the YOLOv8-4DH architecture. In this section, several layers are added such as Upsample, Concat, C2f and Convolutional. In addition, 1 detection head is also added to the output part of YOLOv8-4DH. Output for 4 detection heads on YOLOv8-4DH The filter sizes used in stages 18, 21, 24 and 27 are 256, 512, 1024, 2048 respectively. The head detection in YOLO is based on FPN and PAN output features where FPN-based networks such as YOLOv3 [22] directly utilize one-way fusion features for output and PAN-based YOLOv4 [23] and YOLOv5 [15] algorithms add a low-level channel to the high level above which directly transmits low-level information upwards [21].

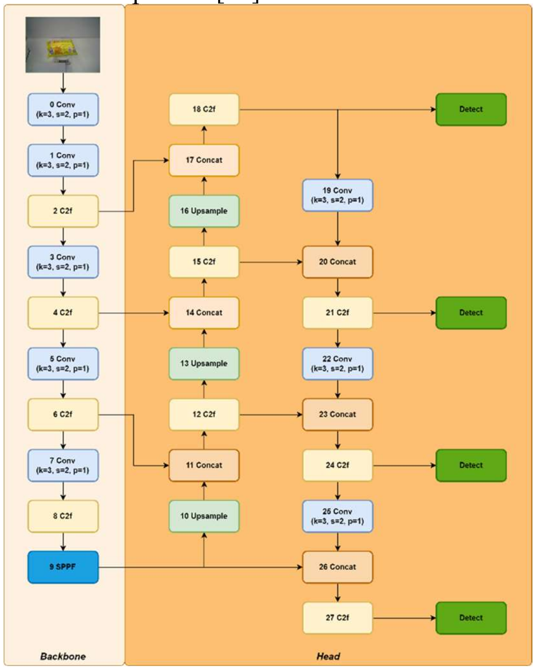

Figure 4: YOLOv8-4DH Architecture

3.2.3 yolov8-conv

The third experiment in this research is YOLOv8 with a Convolutional layer and an additional C2f layer on the backbone (YOLOv8-Conv). The architecture of YOLOv8-Conv can be seen in figure 5. The YOLOv8-Conv architecture consists of 24 stages and 3 detection heads. YOLOv8-Conv has the same architecture as the original YOLOv8 but there are C2f layers and additional Convolutional layers on stages 2 and 3 so that the backbone of YOLOv8-Conv consists of 12 stages. In the YOLOv8-Conv backbone section, the filter size used in the Convolutional stage 0 layer is the same as the original YOLOv8 which is 64. However, in the YOLOv8-Conv SPPF layer the number of filters used becomes 2048 while in the original YOLOv8 it is 1024. In the YOLOv8-Conv detection head, the filter sizes used are 512, 1024 and 2048 unlike the original YOLOv8 which uses filter sizes 256, 512, 1024. The function of the C2f layer is to combine features of various resolutions generated by previous layers into one data structure that will be further used by the network to detect objects at various scales. The function of the Convolutional layer is to perform the feature extraction process by performing a convolution process. The output for the YOLOv8-Conv detection head comes from layer C2f at stages 17, 20 and 23.

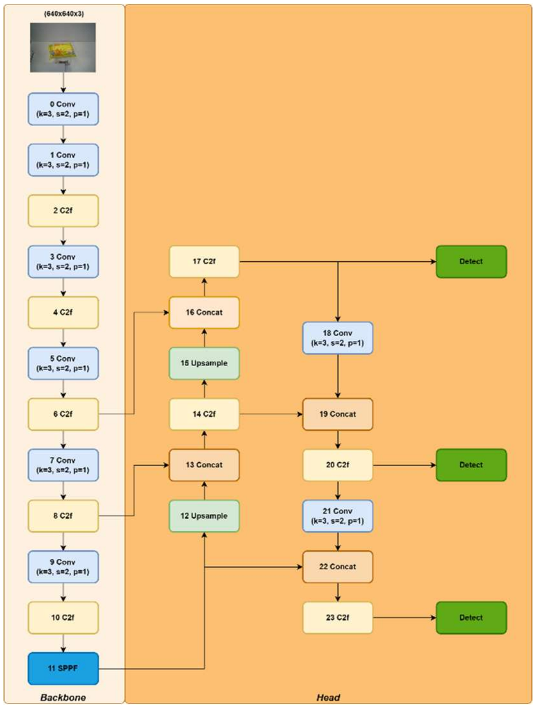

Figure 5: YOLOv8-Conv Architecture

3.2.4 yolov8-conv-4dh

The fourth experiment in this research is YOLOv8 with four detection heads and the addition of Convolutional layer and C2f layer on the backbone (YOLO-Conv-4DH). The architecture of YOLOv8-Conv-4DH can be seen in figure 6. This experiment is a combination of experiment two and experiment three. The YOLO-Conv-4DH architecture consists of 30 stages and 4 detection heads. Just like YOLOv8-Conv, the number of filters used in the backbone of YOLOv8-Conv-4DH is more precisely the Convolutional layer at stage 0 starting from 64. Then, in the SPPF layer of YOLOv8-Conv-4DH the filter size used is 2048. In the YOLOv8-Conv-4DH head section, there are additional layers at stages 18-23. Layers added at stages 18-23 such as Upsample, Concat, C2f, Convolutional. The output for the 4 detection heads in YOLOv8-Conv-4DH comes from layer C2f at stages 20, 23, 26 and 29. For the 4 detection heads of YOLOv8-Conv-4DH, the filter sizes used are the same as YOLOv8-Conv which are 256, 512, 1024, 2048.

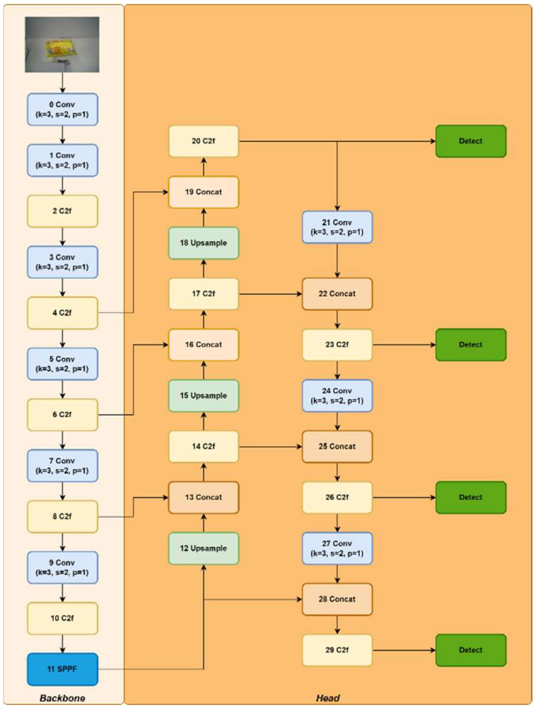

Figure 6: YOLOv8-Conv-4DH Architecture

3.3 SimCLR

SimCLR [25] stands for Simple Framework for Contrastive Learning of Visual Representation. The idea of SimCLR is very simple that an image is taken and a random transformation is applied to get two images (x_i and x_j) then each image is passed through an encoder to get a representation (h_i and h_j) and a connected non-linear layer is applied to get a representation (z_i and z_j). The framework of SimCLR can be seen in figure 7. Based on the SimCLR framework shown in figure 7, there is a three-step process in SimCLR:

a. Data augmentation: In the data augmentation process, from the incoming input image, two correlated images will be displayed as a result of 3 types of simple augmentation in sequence, namely random crop and resize, random color distortion including jitter and grayscale conversion and random gaussian blur [26]. This process produces different perspectives of the same image.

b. Base encoder: The image obtained from the data augmentation process is fed to the encoder and the resulting vector encoding of the image and the encoders used such as ResNet-18 and ResNet-50 [26]. In the base encoder, the feature representation is obtained from the extraction of enhanced samples

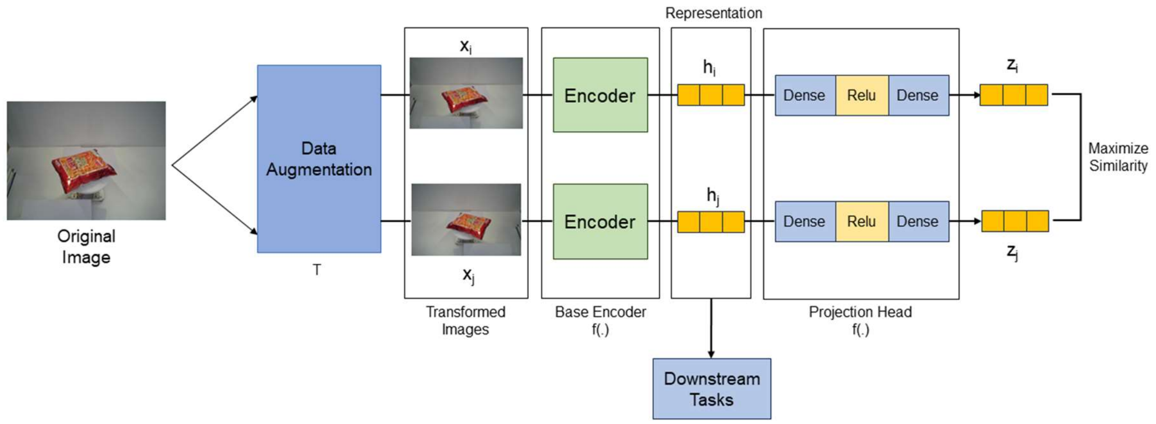

Figure 7: SimCLR Framework

c. Projection head: An encoder such as ResNet-50 will produce an encoding with a dimension of 2048. To reduce the dimensionality of the vector encoding, a Multi Layer Perceptron (MLP) is used and also uses a ReLU activation function with a hidden layer. The encoding in the high-dimensional space is mapped into a 128-dimensional latent space where contrastive loss is applied [26].

The loss function used in the SimCLR algorithm for the training process is Normalized Temperature-scaled Cross Entropy loss (NT-XEnt loss). The NT-XEnt loss function equation can be seen in equation 1:

$$l_{i,j} = -log \frac{\exp\left(sim(z_i, z_j)/\tau\right)}{\sum_{k=1}^{2N} l_{[k\neq 1]} \exp\left(sim(z_i, z_j)/\tau\right)} \qquad (1)$$

where $l_{[k\neq 1]}$ is an indicator function that takes value 1 if [k≠1] and τ denotes the temperature hyperparameter. The final loss is calculated for all positive pairs of both (i,j) and (j,i) in the mini batch [25]. Experiments conducted on SimCLR are that the image will be cropped using the best model selected from the results of several experiments conducted using the YOLOv8 algorithm. Then the image forwarded to the SimCLR algorithm and the image will be trained first on the SimCLR algorithm. After that, the weights from the training results will be saved and will be loaded to evaluate the image produced by the best model from the results of several experiments using the YOLOv8 algorithm.

3.4 Evaluation Metrics

In this research, the performance of the model will be evaluated using two assessment criteria, namely mean average precision (mAP) and accuracy. In the first stage, the model is evaluated using the mAP evaluation metric and in the second stage it uses the accuracy evaluation metric. mAP is a widely applied evaluation metric for object detection of certain categories that indicates the performance level of the object detection process and its location and mAP can be calculated from the average average precision for all classes [28]. The mAP value can be calculated using equation 2.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \qquad (2)$$

where N is the number of classes and AP is Average Precision. Average Precision is used to calculate the average value of precision at various recall levels and the higher the mAP value, the better the performance of a model and vice versa [28]. The formula for Average Precision can be seen in equation 3.

$$AP = \sum (R_n - R_{n-1})P_n \qquad (3)$$

where Rn and Pn are recall and precision at the n-th threshold. The formula for the accuracy used in the second stage can be seen in equation 4 as follows

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad (4)$$

in this equation TP, TN, FN and FP represent true positive, true negative, false negative and false positive.

4. RESULTS AND DISCUSSION

The datasets used in this research are RPC Dataset [11] and RPC dataset that has been managed and simplified by Samrat Sahoo [12]. RPC dataset [11] consists of 200 classes with a division of 53739 images for training data, 6000 images for validation data and 24000 images for test data while RPC dataset managed and simplified by Samrat Sahoo [12] consists of 17 classes with a division of 58589 images for training data, 16740 images for validation data, 8370 images for test data. The RPC Dataset managed and simplified by Samrat Sahoo [12] was used in the first stage of this research while the RPC Dataset [12] was used in the second stage of this research. There are two stages in this study, the first is that the dataset trained using several YOLOv8 models such as YOLOv8 original, YOLOv8-Conv, YOLOv8-4DH, YOLOv8-Conv-4DH which have been prepared previously. After the YOLOv8 models have been trained, the test data images are cropped using the best model selected based on the results of several experiments using YOLOv8. Then the second stage of this research is that the test image cropped using the best YOLOv8 model training results and then the image is trained again using the SimCLR algorithm. After the training process using the SimCLR algorithm is complete, the weights of the SimCLR training results are stored and reused to training and evaluation process of the RPC dataset that has been prepared.

The first stage in this research is the detection process, namely the dataset is trained using several YOLOv8 models that have been prepared. At this stage, the best model is selected based on the highest mean average precision (mAP) value from several experiments using the YOLOv8 algorithm. Table 1 shows the experimental results using several YOLOv8 models. Based on the table, it can be seen that the model with the highest mAP value is YOLOv8-4DH with a mAP value 91%. In contrast to other YOLOv8 models such as the original YOLOv8 which got a mAP value of 90,2% then YOLOv8-Conv which got a mAP value 90,4% and YOLOv8-Conv-4DH which got a mAP value 90,5%. From these results, it can be seen that YOLOv8-4DH has a mAP value that is 0,04% greater than YOLOv8 Original then 0,06% greater than YOLOv8-Conv and 0,05% greater than YOLOv8-Conv-4DH.

Table 1: Experimental Results using Several YOLOv8 Models

Model	mAP 0.5:0.95
YOLOv8 Original	0,902
YOLOv8-4DH	**0,91**
YOLOv8-Conv	0,904
YOLOv8-Conv-4DH	0,905

Based on the experimental results using several models from the YOLOv8 algorithm, it can be seen that the best model is YOLOv8-4DH with a mAP value 91%. YOLOv8-4DH can get better results than other models because of the addition of 1 detection head so that YOLOv8-4DH has 4 detection heads. Detection head is very influential on the detection process because the feature representation that has been learned by the model during training will be forwarded to the detection head to perform the object prediction process. In addition, the detection heads in YOLOv8 have different feature map sizes so that the model can easily detect objects with various scale sizes or in other words, different feature map sizes on the detection head can help the model detect objects that are small or large without sacrificing relevant information. After obtaining the best model from the experimental results using several models built from YOLOv8, the model are used to crop test images from the RPC dataset and the images are forwarded to SimCLR for classification process. The dataset that has been cropped and generated is entered into the training process using SimCLR then the training weights are stored then the stored weights are reused for the training and evaluation process.

From the results obtained by the second stage, it can be seen that the accuracy reaches 97,76%. SimCLR will learn what features are present in the unlabeled image. The information in the image will be extracted in this training process so that when entering the evaluation process, the model can easily recognize the image based on the results of the extracted information. In addition to comparing the experimental results using the model built using the YOLOv8 algorithm, this research also compares the proposed model with other models that have been trained using the same benchmark dataset, namely the RPC dataset. Table 2 shows the results of the performance comparison between the proposed model and other models that have been trained using the RPC dataset. The table shows that the proposed model has an accuracy of 97,76% when compared to other models such as B-CNN + VGGnet which gets an accuracy of 53,2% then TASN + ResNet-50 at 63,9% and DCL + ResNet-50 has an accuracy of 71,8%. In addition, the Siamase + dual-attention + Euclidean Cross Entropy, NTS-NET (k=4) + ResNet-50 and SADCL models got accuracy values of 95,32%, 54,7% and 81,4%, respectively. The combination result between YOLOv8-4DH and SimCLR makes the accuracy value of the proposed method reach 97,76%. It makes the proposed method has achieved quite good accuracy on the RPC dataset.

Table 2: Experimental Results of Several Models Trained using the RPC Dataset

Model	Accuracy on RPC Dataset
B-CNN + VGGnet [29]	53,2%
TASN + ResNet-50 [30]	63,9%
DCL + ResNet-50 [31]	71,8%
Siamase + dual-attention + Euclidean Cross Entropy [6]	95,32%
NTS-NET (k=4) + ResNet-50 [32][9]	54,7%
SADCL [9]	81,4%
Proposed Method	**97,76%**

5. CONCLUSION

In this research, the automatic retail product recognition system is carried out using two stages. In

the first stage, the RPC dataset which has been managed and simplified is trained using several models built using the YOLOv8 algorithm. There are 4 models built using the YOLOv8 algorithm, namely YOLOv8 Original, YOLOv8 with 4 detection heads (YOLOv8-4DH), YOLOv8 with additional Convolutional layer and C2f layer on the backbone (YOLOv8-Conv) and the last is YOLOv8 with 4 detection heads, additional Convolutional layer and C2f layer on the backbone (YOLOv8-Conv-4DH). Based on the results obtained in the first stage, the model that gets the highest mAP value is YOLOv8-4DH with a mAP value of 91%. The use of detection heads is very influential on the detection process because the feature representation obtained is forwarded to the detection head for object prediction. In addition, the detection heads in YOLOv8 have different feature map sizes so that the model can easily detect objects with various scale sizes or in other words, different feature map sizes in the detection head can help the model detect objects that are small or large without sacrificing relevant information. Although YOLOv8-Conv-4DH also has the addition of 1 detection head, the results obtained by YOLOv8-Conv-4DH are smaller than YOLOv8-4DH because the information extracted by the model is not very efficient due to the number of layers added to the model, making the model learn too much and more information is obtained. The cropped images from the first stage are forwarded to the second stage to be trained first. The training weights are saved and loaded back into the model for training and evaluation using the RPC dataset. In the second stage, the SimCLR model got an accuracy result of 97,76% which is still better when compared to other models trained using the same dataset. The use of the SimCLR algorithm in this second stage is to learn the features contained in the unlabeled image. The information in the unlabeled image will be extracted in the training process so that when entering the evaluation process, the model can easily recognize the image based on the results.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] F. Corputty, S. A. Wibowo and S. Rizal, "Implementation of Object Detection and Recognition Based On Exploration Deep Neural Network Features for Quadcopter," 2022 5th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2022, pp. 485-490, doi: 10.1109/ICOIACT55506.2022.9971943.

[2] X. Liu et al., "Collaborative Edge Computing With FPGA-Based CNN Accelerators for Energy-Efficient and Time-Aware Face Tracking System," in IEEE Transactions on Computational Social Systems, vol. 9, no. 1, Feb. 2022, pp. 252-266, doi: 10.1109/TCSS.2021.3059318.

[3] H. Caesar et al., "nuscenes: A multimodal dataset for autonomous driving," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621–11631.

[4] "Amazon Dash Cart: Grocery \& Gourmet Food." Amazon. https://www.amazon.com/b?ie=UTF8\&node=21289116011 (accessed August 03, 2023).

[5] Y. Wei, S. Tran, S. Xu, B. Kang, M. Springer, and others, "Deep learning for retail product recognition: Challenges and techniques," Computational intelligence and neuroscience, vol. 2020, 2020.

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

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

[8] J. Wang, H. Chengwei, L. Zhao, and Z. Li, "Lightweight identification of retail products based on improved convolutional neural network," Multimed. Tools Appl., 2022, pp. 31313–31328, doi: 10.1007/s11042-022-12872-6.

[9] W. Wang, Y. Cui, G. Li, C. Jiang, and S. Deng, "A self-attention-based destruction and construction learning fine-grained image classification method for retail product recognition," Neural Computing and Applications, vol. 32, 2020, pp. 14613–14622.

[10] D. Kumar. "The next evolution of the Dash Cart: New features and expansion to first Whole Foods Market Store". Amazon. https://www.aboutamazon.com/news/retail/amazon-dash-cart-new-features-whole-foods (accessed September 05, 2023)

[11] X.-S. Wei, Q. Cui, L. Yang, P. Wang, and L. Liu, "RPC: A large-scale retail product checkout dataset," arXiv preprint arXiv:1901.07249, 2019.

[12] S. Sahoo. "Groceries Computer Vision Project". roboflow. https://universe. roboflow.com/ samrat-sahoo/groceries-6pfog (accessed September 05, 2023)

[13] P. Follmann, T. Bottger, P. Hartinger, R. Konig, and M. Ulrich, "MVTec D2S: densely segmented supermarket dataset," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 569–585.

[14] G. Jocher, S. Waxmann, A. Chaurasia, Laughing "Ultralytics YOLOv8". ultralytics. https://docs.ultralytics.com (accessed September 05, 2023)

[15] G. Jocher et al., ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo, 2022. doi: 10.5281/zenodo.7347926.

[16] J. Terven and D. Cordova-Esparza, A Comprehensive Review of YOLO: From YOLOv1 and Beyond. 2023.

[17] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475.

[18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[19] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.

[20] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," arXiv preprint arXiv:2305.09972, 2023.

[21] D. Wan, R. Lu, S. Wang, S. Shen, T. Xu, and X. Lang, "YOLO-HR: Improved YOLOv5 for Object Detection in High-Resolution Optical Remote Sensing Images," Remote Sensing, vol. 15, no. 3, 2023, p. 614.

[22] J. Redmon and A. Farhadi, YOLOv3: An Incremental Improvement. 2018.

[23] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020.

[24] H.-K. Jung and G.-S. Choi, "Improved yolov5: Efficient object detection using drone images under various conditions," Applied Sciences, vol. 12, no. 14, 2022, p. 7255.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning, 2020, pp. 1597–1607.

[26] V. Margapuri and M. Neilsen, "Classification of Seeds using Domain Randomization on Self-Supervised Learning Frameworks," in 2021 IEEE Symposium Series on Computational Intelligence (SSCI), 2021, pp. 01–08.

[27] J. Ma, H. Duan, C. Yang, X. Wang, Y. Niu, and Y. Han, "SimCLR-Unet: An ECG Feature wave segmentation algorithm based on a self-supervised learning strategy," in Proceedings of the 2022 4th International Conference on Robotics, Intelligent Control and Artificial Intelligence, 2022, pp. 1354–1359.

[28] M. Ahmed, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Survey and performance analysis of deep learning based object detection in challenging environments," Sensors, vol. 21, no. 15, pp. 1–

28, 2021, doi: 10.3390/s21155116.

[29] T.-Y. Lin, A. RoyChowdhury, and S. Maji, “Bilinear CNN models for fine-grained visual recognition,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1449–1457.

[30] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo, “Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5012–5021.

[31] Y. Chen, Y. Bai, W. Zhang, and T. Mei, “Destruction and construction learning for fine-grained image recognition,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5157–5166.

[32] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang, “Learning to navigate for fine-grained classification,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp.420–435

[33] Rahmania, R., et al, "Object Size Recognition as Intra-class Variations using Transfer Learning," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023, pp. 568–573.