Jurnal Infotel
Vol. 16, No. 3, August 2024, pp. 502–517. doi:10.20895/infotel.v16i3.1189

Research Article

Rupiah Banknotes Detection: Comparison of The Faster R-CNN Algorithm and YOLOv5

Muhammad Zuhdi Hanif[1,*], Wahyu Andi Saputra[2], Yit Hong Choo, [3] and Andi Prademon Yunus[4]

[1,2,4]Informatics Department, Institut Teknologi Telkom Purwokerto, 53147, Indonesia
[3]Institute for Intelligent Systems Research and Innovation, Deakin University, Melbourne, 3125, Australia

*Corresponding email: 20102074@ittelkom-pwt.ac.id

Received: June 10, 2024; Revised: August 19, 2024; Accepted: August 21, 2024.

Abstract:
This research compares the performance of the Faster R-CNN algorithm, utilizing the ResNet-50 architecture, with YOLOv5 in recognizing rupiah banknotes. A dataset comprising 1120 images from 8 different classes was used for training and evaluation. The results indicate that the YOLOv5 model trained on RGB data achieved the highest accuracy of 1. In comparison, other models—YOLOv5 with HSV, Faster R-CNN with RGB, and Faster R-CNN with HSV—attained accuracies of 0.986, 0.875, and 0.821, respectively. The best-performing model demonstrated excellent capability in predicting the correct labels and accurately placing bounding boxes on the test images. This study provides valuable insights that can be utilized in the future development of systems designed for the recognition of rupiah banknotes, contributing to advancements in automated financial transactions.

Keywords: Faster R-CNN, Object detection, ResNet-50, Rupiah Banknotes, YOLOv5

1 Introduction

The usage of banknotes for physical transactions has become essential since everyone can fulfil their daily needs with money in the form of banknotes such as food, drinks, clothing, and shelter [1]. Money is used as a conventional transaction tool for buying and selling between people and to carry out transactions between humans and machines. One of the conveniences that the public can feel is the existence of technology or systems that can recognize the nominal value of banknotes [2]. Therefore, we need a technology to differentiate banknotes based on their nominal value. One that can be used is Object Detection.

The use of object detection has increased in recent years. Object detection gives computers the ability to recognize many things in input, such as images, videos, and similar forms of images [3]. This technology can assist navigation because its function is similar to the human vision system [4]. Some techniques commonly used for object detection are Faster R-CNN and YOLO. In this paper, we compare YOLOv5 and Faster R-CNN with ResNet50 to detect the Indonesia Rupiah banknotes.

Faster R-CNN is a machine object detection method where object recognition is done by looking for the characteristics of the object in the image. Faster R-CNN has advantages over its predecessor algorithms, primarily because it no longer uses selective search. Instead, it employs Region Proposal Network (RPN), which creates a region proposals more quickly, reducing computing time. This search is performed through layers, as in neural networks through a convolution process or Convolutional Neural Network (CNN) [5]. However, CNN architectures often suffer from vanishing gradients, where the gradient of the loss function decreases to near zero, hindering convergence [6]. ResNet (Residual Network) architecture, introduced by Kaiming He, overcomes this problem with skip connections, batch normalization, and the elimination of the fully connected layer at the end [7]. ResNet's method effectively counters vanishing gradients, maintaining performance even as the architecture deepens [8]. Nevertheless, more than Faster R-CNN is required for object detection tasks; a more responsive and efficient method, such as YOLO, is needed.

YOLO (You Only Look Once) is a unified, real-time object detection system that predicts bounding boxes and class probabilities directly from full images in a single evaluation [9]. YOLO divides the input image into several boxes and predicts each bounding box and its probability [10], making it more efficient and effective in object detection.

Several related studies concerning Faster R-CNN, YOLO, and object detection on rupiah banknotes exist. The first research, "Implementation of the Faster Region Convolutional Neural Network (Faster R-CNN) Method for Recognizing Lovebird Types," conducted by Fino Charli et al. in 2020 [11], used a dataset of 808 lovebird images across 8 classes. The results showed a model accuracy of 78% to 99%, demonstrating the efficacy of the Faster R-CNN method for image prediction.

The second study, "Implementation of Deep Learning Using the Convolutional Neural Network Method and the YOLO Algorithm in the Rupiah Banknote Detection System for People with Low Vision," by Kevin Maulana Azhar et al. in 2021 [12], used the Darknet framework with the YOLO algorithm to detect rupiah banknote images. The dataset consisted of 1400 images of 2016 emission banknotes across 14 classes, divided into train, validation, and test sets with a 70:20:10 ratio. The model achieved an mAP value of 88% with a detection response time of 1.28 seconds.

The third study, "Detecting the Amount of Money in Images Using Convolutional Neural Network: Integration of Image Pre-Processing Methods and CNN-Based Classification," conducted by Muhamad Malik Ibrahim et al. in 2023 [13], used a dataset of 1076 images across 8 classes with the MobileNetV2 architecture. The results showed sequential probabilities in the IDR 1.000, IDR 2.000, IDR 5.000, IDR 10.000, IDR 20.000, IDR 50.000, IDR 75.000, IDR 100.000 classes of 20%, 10%, 70%, 50%, 40%, 100%, 80%, 90%.

Based on the background explained above, Faster R-CNN, especially with the ResNet-50 architecture, offers significant advantages in overcoming vanishing gradients. However, a more efficient method, such as YOLOv5, is needed for real-time object detection. In previous studies, the YOLO model was only trained on the 2016 emission Rupiah banknote dataset, while currently, there is already money for the 2022 emission. In previous studies using the MobileNetV2 architecture, only classification tasks were performed without object detection, which is less applicable. Therefore, this study aims to:

1. Implement the Faster R-CNN with ResNet-50 and YOLOv5 algorithms to identify rupiah banknotes with datasets, including the 2016 and 2022 emissions.
2. Measure the model's performance between the Faster R-CNN with ResNet-50 and YOLOv5 algorithms in identifying rupiah banknotes.

2 Research Method

2.1 Research Flowchart

In compiling this research, several steps must be followed. Figure 1 shows the research flowchart used in compiling the report.

2.2 Dataset

The author carried out data collection using secondary data sourced from the kaggle.com website called rupiah banknotes [14]. This dataset consists of 30 classes consisting of various nominal rupiah banknotes ranging from IDR 1,000 to IDR 100,000 and an additional nominal value of IDR 75,000 for the 2020 issue. The total data used was 1120 images, which were divided into 8 different classes.

2.3 Data Preprocessing

At the preprocessing stage, the obtained data is processed more systematically so that it can be used optimally during data training. After the data is obtained, the next step is to annotate it. Data annotation is done by placing a bounding box on the banknote object in the image. The author uses the platform Computer Vision Annotation Tools (CVAT) as shown on Figure 2. Annotated data is saved with the .xml extension in PASCAL VOC format for the Faster R-CNN model and YOLO txt for the YOLOv5 model.

After annotation, the data is divided into train data and test data with a ratio of 80:20. The next stage is carrying out preprocessing, such as various combination schemes to change the image's colour. Two colour change schemes are used in this research, namely RGB data and HSV data combined with HOG.

2.3.1 Histogram of Oriented Gradient (HOG)

A histogram of Oriented Gradients (HOG) is a feature extraction technique in image processing that groups pixel gradient values according to the orientation of each local part of the image. The appearance and shape of local objects can often be characterized quite well by the distribution of local intensity gradients or edge directions, even though the exact position of the corresponding gradients or edges is not known [15].

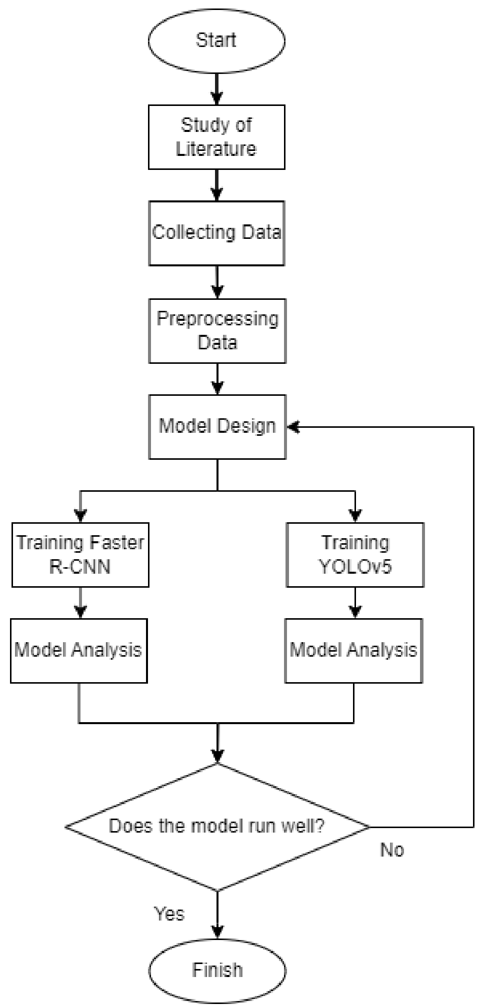

Figure 1: Research flowchart.

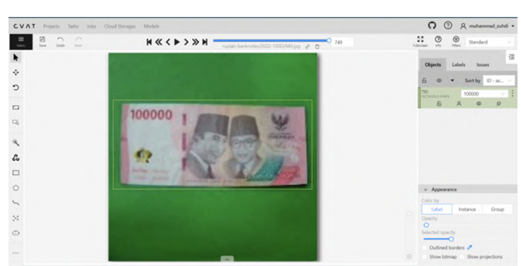

Figure 2: Data annotation using CVAT.

2.4 Model Design

This research compares two models, Faster R-CNN ResNet-50 and YOLOv5 on the Indonesia Rupiah banknotes.

2.4.1 Faster R-CNN

Faster Region Convolutional Neural Network co, commonly called Faster R-CNN, is an object detection algorithm that is a refinement of the previous method, namely R-CNN and also Fast R-CNN, which in this method uses Selective Search as a form of input that will be processed in CNN. Selective search found obstacles that could hinder the computing process, which took longer [16]. In the Faster R-CNN method, the role of selective search is replaced by utilizing the Region Proposal Network (RPN). The following is the architecture of Faster R-CNN. Figure 3 shows the architecture of Faster R-CNN [17].

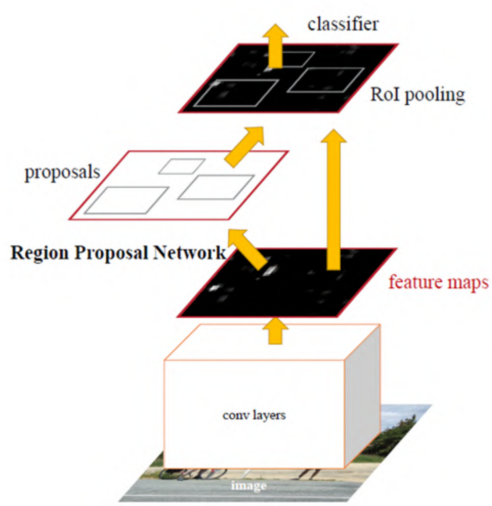

Figure 3: Faster R-CNN architecture.

Region Proposal Network is a critical element of the Faster R-CNN network. The primary function of RPN is to predict proposals for regions that potentially contain objects in the input image. The result of the RPN is a list of regional proposals, which are then used by the object detection component in Faster R-CNN to classify and identify objects in the image. The proposals produced by RPN are then compared with those of the Ground Truth Box.

2.4.2 ResNet-50

ResNet is an architecture built by Kaiming He et al., who won first place in the 2015 ILSVRC competition [18]. ResNet is a CNN architecture known to have good accuracy because it uses a different method from previous CNN architectures, namely residual blocks and skip connections [19]. The ResNet architecture has many versions, one of which is ResNet-50.

ResNet applies the skip connection concept, where the features that are the input of the previous layer are also used as input for the output of that layer [20]. Figure 4 is the general architecture of ResNet-50 [18].

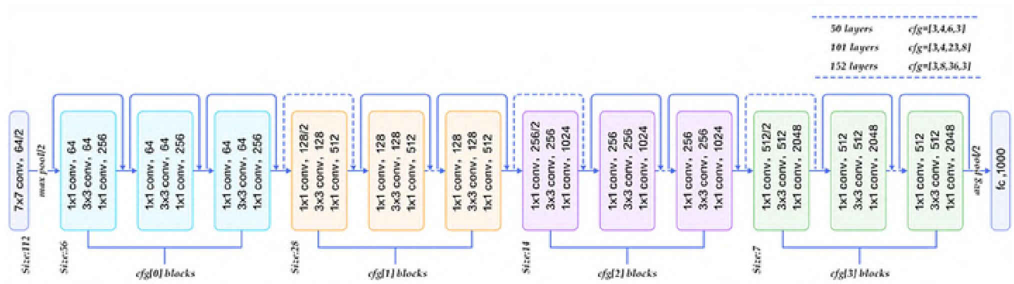

Figure 4: ResNet-50 architecture.

Figure 4 shows the ResNet 50 architecture, which begins with an input layer with 7x7 Convolution and 64 Filters. It uses a large 7x7 filter to extract basic features. ResNet-50 consists of a series of residual blocks, each with a set of convolutional layers. After the residual blocks, the architecture includes a fully connected (FC) layer.

2.4.3 YOLOv5

YOLO (You Only Look Once) is the first single-stage object detection algorithm proposed by Redmon J. This algorithm eliminates the candidate box extraction step in a two-stage algorithm and combines bounding boxes and classification into a regression problem [21]. YOLOv5 is a version of YOLO that was published in 2020. YOLOv5 primarily depends on deep CNN structure, which is skilled at locating objects in images. The set of rules uses a single neural network to expect item bordering and class probabilities. This differs from traditional object detection algorithms that require more than one level to detect objects, making YOLOv5 quicker and more accurate than conventional algorithms [22]. Figure 5 shows architecture of YOLOv5.

2.5 Model Training

Training data in this research was carried out to train the model designed in the previous stage. The model is trained to recognize and differentiate images of rupiah banknotes. The model in the train dataset process is targeted to produce a good model with the lowest possible loss value and the highest possible accuracy value to recognize objects well.

2.6 Evaluation Methods

2.6.1 Negative Log Likelihood (NLL)

Negative Log-likelihood is a loss metric that measures how well a model predicts data distribution. This metric is calculated by summing the log-likelihood of the observed data probabilities, where the probability is calculated based on the model predictions. The

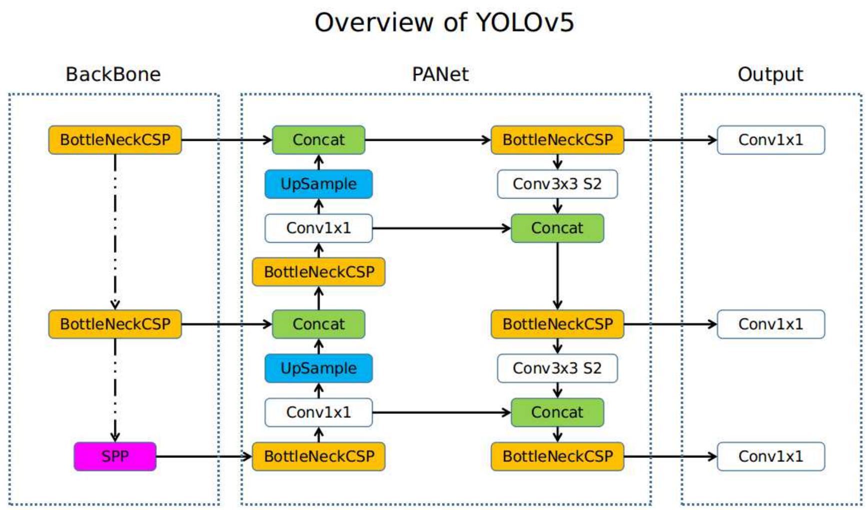

Figure 5: YOLOv5 architecture [23].

smaller the NLL value, the better the model predicts the data distribution [24]. NLL is calculated using the Eq. 1

$$l(\theta) = -\sum_{i=1}^{n} (y_i \log \hat{y}_{\theta,i} + (1 - y_i) \log(1 - \hat{y}_{\theta,i})) \tag{1}$$

2.6.2 Confusion Matrix

A confusion matrix is a table that shows the performance of an algorithm, mainly how well it performs in classifying [25]. In this research, the confusion matrix as in Table 1 used includes accuracy, precision, recall, and F1 score.

Table 1: Confusion matrix

	Predicted Class 1	**Predicted Class 2**
Ground Truth Class 1	TP	FN
Ground Truth Class 2	FP	TN

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TN} + \text{FP}} \tag{4}$$

$$\text{F1} - \text{Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

3 Results

3.1 Preprocessing

1. Train and Test Data Split Datasets previously annotated in PASCAL VOC format were converted into data frames to make the process easier to use. Table 2 is a sample of the dataset used. In the table above, the total number of datasets used is 1120. The

Table 2: Dataset sample

	filename	**name**	**xmin**	**ymin**	**xmax**	**ymax**
0	rupiah-banknnotes/1000/0.jpg	1000	10.87	31.92	219.21	148.48
1	rupiah-banknotes/1000/1.jpg	1000	14.78	34.53	216.16	147.18
2	rupiah-banknotes/1000/2.jpg	1000	9.24	59.44	214.03	180.86

"filename" column is the address of the image, which will later be used to retrieve images during the preprocessing process. The "width" and "height" columns are the width and length measurements where all images are 224x224, corresponding to the input image desired by the Faster R-CNN model. The "name" column is the class label of each image. The columns "xmin", "ymin", "xmax," and "ymax" are the coordinates of the bounding box, which has previously been annotated on the image. We can use the YOLO *.txt format directly in the data loader.

Table 3: Annotation format on YOLO txt

label	**xmin**	**ymin**	**xmax**	**ymax**
0	0.483795	0.498371	0.946607	0.521027

2. Image Normalization The next preprocessing step after dividing the dataset is normalizing the data. The goal of data normalization is to change the pixel values in the image uniformly. This process can improve model performance in the training process. The normalization carried out in this research involves changing the pixel values, which originally had a range of 0-255, to a range of 0-1. There are various ways to carry out normalization.
3. Color Change Scheme The following preprocessing stage applied in this research involves several colour change schemes. The purpose of using multiple schemes is to see which preprocessing is better for the model in this research. There are two color change schemes used, including:
   (a) Uses RGB image format The first color change scheme uses the RGB image format. The data used in this scheme is the same image as the normalization results. Data that has previously been normalized will be trained in the model.
   (b) Change the image format to HSV with HOG
   The second color change scheme is to change the image color format from RGB to HSV with HOG. Hue Saturation Value (HSV) is carried out to convert images

so that they can separate color information (Hue) from light intensity (Value), which allows object detection to be more robust to changes in lighting, which is then combined with HOG. Heuristic of Oriented Gradients (HOG) is a preprocessing technique that aims to calculate pixel intensity in horizontal and vertical directions. In this research, the value component of the HSV image is used to extract HOG features, which are then combined with the hue and saturation channels. Figure 6 is a visualization of the image after normalization after using HSV with HOG.

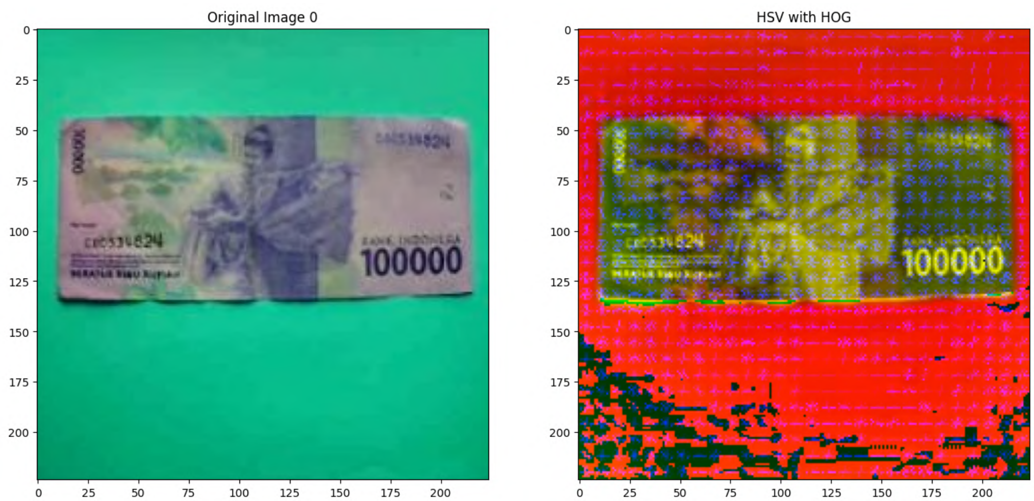

Figure 6: Visualization of data samples using a combination of HSV with HOG.

3.2 Model Training

After the image has been preprocessed, the next stage is modeling. This research compared Faster R-CNN using ResNet50 and YOLOv5.

3.2.1 Faster R-CNN with ResNet50

Modelling begins by creating a data loader consisting of a train loader and a test loader. After that, the hyperparameters that will be used in the model, such as epoch, batch size, learning rate, momentum, and weight decay, will be determined. The batch size used in this research is 8. The learning rate controls how many steps are taken when updating the neural network model weights during the training process. The learning rate used is 0.005. Momentum is a technique used to speed up training and help break out of local minima. The momentum used is 0.9. Weight decay is done by adding a small amount of the L2 norm of the weights to the loss function during training. This causes the weights to be more likely to be minor, which in turn can help prevent the model from fitting too well to the training data and improve the model's generalization ability. The weight decay used is 0.0005.

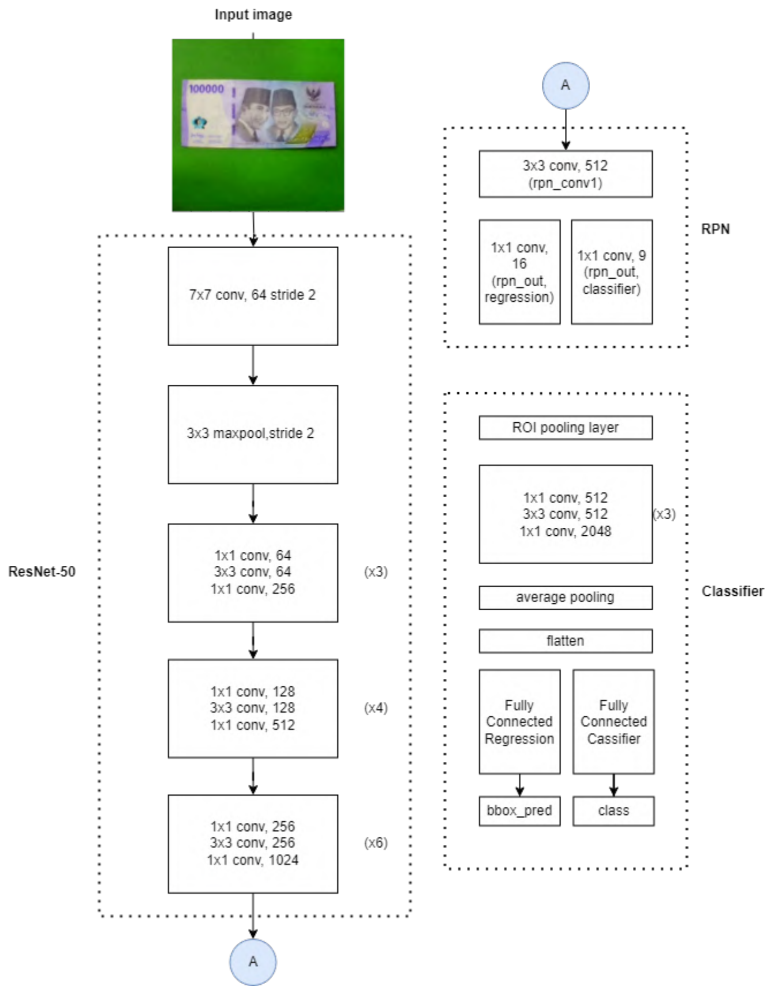

Figure 7: Faster R-CNN with ResNet50 architecture.

This research carried out modelling using a train and eval per epoch scheme, which immediately displays the train and eval results in one epoch. The epoch used is 40. The model used in this research is Faster R-CNN with Resnet-50. The model used is available in the Torchvision library. Figure 7 shows the architecture of Faster R-CNN with ResNet50.

3.2.2 YOLOv5

The YOLOv5 model used in this research also uses the same two schemes as those in Faster R-CNN. The dataset previously annotated in YOLO txt format is divided into train tests with a ratio of 8:2. The number of epochs used in this research is 100. This number is given considering that YOLO's computing time is much lighter than Faster R-CNN. Figure 8 shows the architecture of YOLOv5.

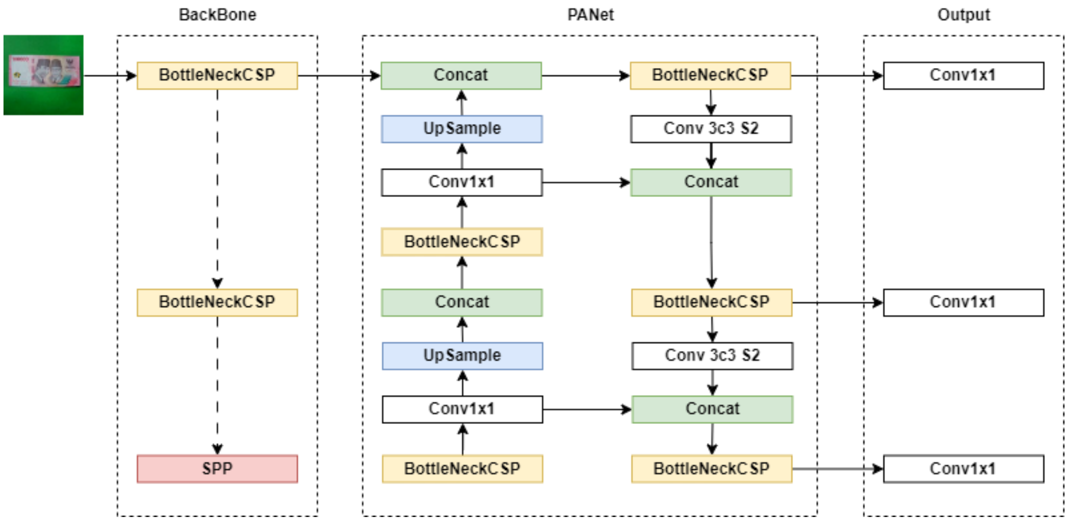

Figure 8: YOLOv5 architecture.

4 Experiment Results

4.0.1 RGB Scheme

In the preprocessing scheme using RGB format, the Faster R-CNN model with ResNet50 obtains the following evaluation results in Table 4. The number of epochs used in this scheme is 14 times. The number of epochs at the beginning was 40 epochs. However, in the 14th epoch, the mAP value was 0.5, mAP 0.75, and the eval accuracy was 0.875. In the calculations, mAP 0.50 and mAP 0.75, and recall each show the same value, namely 0.875. The precision calculation results show the lowest value is 0.786. F1-Score shows a figure of 0.823

Meanwhile, the RGB preprocessing scheme in the YOLOv5 model is in Table 5. The number of epochs used in this scheme is 100 times. The number of epochs in yolov5 is greater because it has lighter computing. The YOLOv5 model's results on the RGB scheme are convincing. It has a very small loss with an accuracy of up to 1. In other evaluation metrics, recall shows a result of 1 as well, and other metrics also show figures above 90%.

Table 4: Faster R-CNN result on RGB scheme

Epoch	Train Losses	Accuracy	mAP 0.5	mAP 0.75	Precision	Recall
14	0.02	0.875	0.875	0.875	0.786	0.875

Table 5: YOLOv5 result on RGB scheme

Epoch	Train Losses	Accuracy	mAP 0.5	mAP 0.5:0.95	Precision	Recall
100	0.004	1	0.995	0.917	0.996	1

Figure 9: Prediction result on RGB scheme using Faster R-CNN model.

4.0.2 HSV with HOG Scheme

In the preprocessing scheme using HSV with HOG format, the Faster R-CNN model with ResNet50 obtains the evaluation results in Table 6. The number of epochs used in this scheme is 14 times. Train losses in this scheme show a value of 0.025. The mAP 0.50 and mAP 0.75 calculation shows the same result at 0.869. The recall shows the same value, namely 0.821. The precision calculation results show the lowest value, namely 0.763. The F1-Score shows a figure of 0.77.

Table 6: Faster R-CNN result on RGB scheme

Epoch	Train Losses	Accuracy	mAP 0.5	mAP 0.75	Precision	Recall
14	0.025	0.8210	0.869	0.869	0.763	0.821

Meanwhile, the HSv with HOG preprocessing scheme in the YOLOv5 model is as follows Table 7. The number of epochs used in this scheme is 100 times. The number of epochs in yolov5 is greater because it has lighter computing. The YOLOv5 model's results on the HSV with the HOG scheme are convincing. It has a small loss with 0.007 values and an accuracy of up to 0.986. In other evaluation metrics, recall shows a result of 0.981 as well, and other metrics also show figures above 90%.

Table 7: YOLOv5 result on RGB scheme

Epoch	Train Losses	Accuracy	mAP 0.5	mAP 0.5:0.95	Precision	Recall
100	0.007	0.986	0.994	0.917	0.992	0.981

4.0.3 RGB Scheme

Figure 9 shows the results of experiments on the RGB scheme with the Faster R-CNN ResNet50 model. Faster R-CNN models trained on RGB images do not show such good results. In the first picture, the model can predict the class correctly but with a confidence score that is not too high, 0.42. Placing the box in the image is good enough because it covers the entire object. In the second image, the model predicts a class that does not match the image. The correct class is "75000", but the model predicts the object is class "2000" even though the confidence score is low. The box also does not draw well because it is distorted and does not cover the entire object. In the third image, the model can predict the image well. The predicted class matches that of the object with a confidence score that is also

Figure 10: Prediction result on RGB scheme using YOLOv5 model.

Figure 11: Prediction result on HSV with HOG scheme using Faster R-CNN model.

relatively high at 0.61. Boxes also draw well because they cover the entire object. Figure 10 shows the results of experiments on the RGB scheme with the YOLOv5 model.

The YOLOv5 model trained with the RGB scheme shows excellent results. In the first picture, the model can predict the class correctly and get a good confidence score of 0.71. The box placement is also good because it covers the entire object. In the second picture, the model can also correctly predict the class and get an excellent confidence score of 0.85. The box placement is also good because it covers the entire object. In the third picture, the model can also predict the class correctly, but with a confidence level that is not too high at 0.54. The box's alignment is also quite good, even though a small part of the right edge of the object is not included in the box.

4.0.4 HSV with HOG Scheme

Figure 11 shows the results of experiments on the HSV with HOG scheme with the Faster R-CNN ResNet50 model. The Faster R-CNN model trained using HSV data with HOG shows poor results. The model mispredicts the class in the first image, and the confidence score is high at 0.74. Even so, the box drawn is quite good because almost all parts of the object are included. The second image shows worse results; the model predicts more than one class, and both show wrong predictions with confidence scores of 0.29 and 0.24. The two boxes drawn are also not good because they do not contain all the objects. In the third picture, the new model shows a correct prediction even though the confidence score is not too high at 0.55. The box drawn is also entirely accurate, leaving only a few parts of the object that are not included in the box. Figure 12 shows the results of experiments on the RGB scheme with the YOLOv5 model. The YOLOv5 model trained using HSV data with HOG shows poor results in the first picture. The model predicts the wrong class with a confidence score of 0.27. Even so, the predicted box is quite good because it contains

Figure 12: Prediction result on HSV with HOG scheme using YOLOv5 model.

almost all the image objects. In the other two images, the model does not make predictions about the object even though previously, the threshold determined for the confidence score was already shallow at 0.1.

5 Data and Code Availibility

We publicly open our data and source code committed on our GitHub repo https://github.com/AndiDemon-Lab/RupiahBankNotes.

6 Conclusion

This research compares two object detection methods, Faster R-CNN using ResNet50 with YOLOv5 on the rupiah banknote dataset. This study uses two training schemes: RGB preprocessing and HSV with HOG values. Based on the research conducted, the YOLOv5 model trained using RGB data showed the best results. The model obtained an accuracy value of 1.0 after 100 epochs. In comparison, other models—YOLOv5 with HSV, Faster R-CNN with RGB, and Faster R-CNN with HSV—achieved accuracies of 0.986, 0.875, and 0.821, respectively. The prediction results with the YOLOv5 model on RGB data also showed very accurate results where, after testing 3 images, the model could predict all classes appropriately and with accurate box placement. Preprocessing with RGB had better results than the combination of HSV with HOG, which showed less promising results from both models.

References

[1] Y. Banusu and A. Firmanto, "Kebahagiaan dalam ruang keseharian manusia. forum, 49 (2), 51–61," 2020.

[2] W. Hamidah, T. S. B. Irawan, N. A. P. Hasbullah, and A. B. Kaswar, "Deteksi nominal uang kertas menggunakan ocr (optical character recognition)," *Techno Xplore: Jurnal Ilmu Komputer dan Teknologi Informasi*, vol. 7, no. 2, pp. 72–76, 2022.

[3] M. A. Rasyad, F. Dewanta, and S. Astuti, "All-in-one computation vs computational-offloading approaches: a performance evaluation of object detection strategies on android mobile devices," *JURNAL INFOTEL*, vol. 13, no. 4, pp. 216–222, 2021.

[4] C. Rahmad, K. ST M, M. P. Rawansyah, and T. K. Rochastu, "Object detection system sebagai alat bantu mendeteksi objek sekitar untuk penyandang tunanetra.," 2019.

[5] Y. Rizki, R. M. Taufiq, H. Mukhtar, and D. Putri, "Klasifikasi pola kain tenun melayu menggunakan faster r-cnn," *IT Journal Research and Development*, vol. 5, no. 2, pp. 215–225, 2021.

[6] G. Thiodorus, A. Prasetia, L. A. Ardhani, and N. Yudistira, "Klasifikasi citra makanan/non makanan menggunakan metode transfer learning dengan model residual network," *Teknologi: Jurnal Ilmiah Sistem Informasi*, vol. 11, no. 2, pp. 74–83, 2021.

[7] D. Suyanto, K. N. Ramadhani, and S. Mandala, "Deep learning modernisasi machine learning untuk big data," *Informatika*, 2019.

[8] B. M. Sujatmiko, E. Yudaningtyas, and P. M. Raharjo, "Convolution neural network dengan desain jaringan resnet sebagai metode klasifikasi tumor kulit," *Jurnal Simantec*, vol. 11, no. 1, pp. 53–64, 2022.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[10] A. N. Sugandi and B. Hartono, "Implementasi pengolahan citra pada quadcopter untuk deteksi manusia menggunakan algoritma yolo," in *Prosiding Industrial Research Workshop and National Seminar*, vol. 13, pp. 183–188, 2022.

[11] F. Charli, H. Syaputra, M. Akbar, S. Sauda, and F. Panjaitan, "Implementasi metode faster region convolutional neural network (faster r-cnn) untuk pengenalan jenis burung lovebird," *Journal of Information Technology Ampera*, vol. 1, no. 3, pp. 185–197, 2020.

[12] K. M. Azhar, I. Santoso, and Y. A. A. Soetrisno, "Implementasi deep learning menggunakan metode convolutional neural network dan algoritma yolo dalam sistem pendeteksi uang kertas rupiah bagi penyandang low vision," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 10, no. 3, pp. 502–509, 2021.

[13] M. M. Ibrahim, R. Rahmadewi, and L. Nurpulaela, "Pendeteksian nominal uang pada gambar menggunakan convolutional neural network: Integrasi metode prapemrosesan citra dan klasifikasi berbasis cnn," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, pp. 1394–1400, 2023.

[14] N. Alfiyah, "Rupiah Banknotes — kaggle.com." https://www.kaggle.com/datasets/nurulalfiyyah/rupiah-banknotes, 2022.

[15] M. Muhathir, M. H. Santoso, and R. Muliono, "Analysis naã¯ ve bayes in classifying fruit by utilizing hog feature extraction," *Journal of Informatics and Telecommunication Engineering*, vol. 4, no. 1, pp. 151–160, 2020.

[16] M. F. Rahman and B. Bambang, "Deteksi sampah pada real-time video menggunakan metode faster r-cnn," *Applied Technology and Computing Science Journal*, vol. 3, no. 2, pp. 117–125, 2020.

[17] R. Girshick, "Fast r-cnn," 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[19] S. Suprihanto, I. Awaludin, M. Fadhil, and M. A. Z. Zulfikor, "Analisis kinerja resnet-50 dalam klasifikasi penyakit pada daun kopi robusta," *J. Inform*, vol. 9, no. 2, pp. 116–122, 2022.

[20] F. Nashrullah, S. A. Wibowo, and G. Budiman, "Investigasi parameter epoch pada arsitektur resnet-50 untuk klasifikasi pornografi," *Journal of Computer, Electronic, and Telecommunication*, vol. 1, no. 1, pp. 1–8, 2020.

[21] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-time vehicle detection based on improved yolo v5," *Sustainability*, vol. 14, no. 19, p. 12274, 2022.

[22] L. Manikandan, R. Malavika, B. Anjali, and S. Chandana, "Object detection using yolo v5," *European Chemical Bulletin*, 2023.

[23] "Overview of model structure about yolov5." https://github.com/ultralytics/yolov5/issues/280, 2020. [Accessed 07-06-2024].

[24] Liu, Shenghua, Jamieson, Nathan, Lannon, Kevin, Mohrman, Kelci, Negash, Sirak, Wan, Yuyi, and Yates, Brent, "Using a neural network to approximate the negative log likelihood function," *EPJ Web of Conf.*, vol. 295, p. 09041, 2024.

[25] M. Dandi Darojat, *Convolutional Neural Network Untuk Klasifikasi Citra Makanan Khas Indonesia*. PhD thesis, Universitas Brawijaya, 2021.