International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Jul-2024)

http://dx.doi.org/10.12785/ijcds/160125

Evaluation of Deep Learning Models for Detection of Indonesian Rupiah

Charleen Charleen[1] and Gede Putra Kusuma[2]

[1,2] *Computer Science Department, Bina Nusantara University, Jakarta, Indonesia, 11480*

Received 13 Nov. 2023, Revised 18 Apr. 2024, Accepted 23 Apr. 2024, Published 1 Jul. 2024

Abstract: This study compares the performance of current object detection models, namely YOLOv7-tiny, YOLOv8n, and EfficientDet-d0, using YOLOv5n as the baseline model in addressing the challenge of Rupiah banknote detection. The challenge involves recognizing unique features on the banknotes, which may have higher complexity compared to common objects in object detection tasks. The dataset used covers 2022 Emission Year Rupiah banknotes, is manually created, and covers various real-world scenarios for comprehensive evaluation. This research also explores the impact of data augmentation to optimize model performance. Results show that YOLOv8 is the top-performing model, with mAP@0.5 scoring 0.995 and mAP@0.5:0.95 scoring 0.994 on the test data, also consistently maintaining high performance even without augmentation. YOLOv5 also showed impressive mAP scores of 0.995 and 0.973 with augmentation. YOLOv7, although it did not surpass YOLOv8 and YOLOv5 in accuracy, achieved good results, especially with data augmentation. In terms of inference time, YOLOv5 excels with 6.7 ms without augmentation and 6.5 ms with augmentation, emphasizing its efficiency. YOLOv8, although slightly less efficient, with inference time of 7.8 ms without augmentation and 8.2 ms with augmentation, provides higher accuracy. The choice between the two depends on the balance between accuracy and efficiency. This research also highlights the positive impact of data augmentation, especially in YOLOv5's responsiveness to additional data. While EfficientDet is efficient in inference time and resource usage, it suffers in performance, especially without augmentation. This study attempts to develop a dependable method for identifying banknotes. By achieving this, the aim was to improve accessibility in financial tasks and everyday life, particularly benefiting those with visual impairments or other disabilities.

Keywords: Deep Learning, Object Detection, Indonesian Rupiah, YOLOv5 model, YOLOv7 model, YOLOv8 model, EfficientDet model

1. INTRODUCTION

Object detection is a complex and vital challenge within the field of computer vision. It revolves around the exact localization and recognition of objects in input data. These algorithms work by analyzing the distinctive characteristics of objects and applying machine learning techniques to both characterize and identify these objects [1]. Object detection has a wide range of uses, from enabling autonomous vehicles to assisting in the field of medical imaging [2]. The recognition of Indonesian Rupiah banknotes, however, is a specialized aspect of object detection that is the focus of this study.

The general availability of money in daily financial transactions serves to highlight the importance of this particularized object detection task [3]. This is especially important in the context of Indonesia, where Bank Indonesia issues and governs the country's official currency, the Indonesian Rupiah (IDR). Coins and banknotes are the two primary forms of currency. In this study, we utilized the 2022 issuance of the Indonesian Rupiah, which represents the most recent design iteration. It is important to note that the Indonesian Rupiah's design has undergone multiple revisions over the years, with the primary objective being the enhancement of currency quality and the reinforcement of security features [4].

Disabled individuals often encounter numerous challenges in their daily activities. These difficulties span a range of aspects, including mobility, dining, and shopping, with many of these activities involving financial transactions. For physically disabled individuals, maintaining these operations can prove to be significantly more demanding than for those who are able-bodied [5]. However, among the disabled community, visually impaired individuals face even greater obstacles compared to others. Despite the growing popularity of electronic financial transactions, cash remains a primary form of exchange [6]. This presents a unique challenge, as effective strategies are required to enable precise handling and recognition of banknotes, something that can be especially challenging for those with visual impairments. Consequently, this group of people

E-mail address: charleen@binus.ac.id, inegara@binus.edu

often encounters substantial difficulties when attempting to carry out financial transactions securely and independently [7].

Visual impairments have wide-ranging effects on people's lives, affecting everything from their level of independence overall to their ability to access education and employment opportunities [8]. For this group, being able to identify and distinguish banknotes emerges as a crucial component of financial wellness. While Braille text, tactile features, and currency readers have all offered some support, there is still a critical need for cutting-edge solutions. Here, computer vision technology holds out the possibility of helping those who are blind due to its potential for advancement [9].

Object detection is an important subject that is constantly researched in computer vision research [10]. Deep learning-based object detection technology has made significant advances in recent decades [11]. These cutting-edge techniques, demonstrated by models such as You Only Look Once (YOLO) [12], have attracted considerable attention due to their real-time detection capabilities and astounding levels of accuracy. On the other hand, EfficientDet was proposed as an object detector model that offers better efficiency. This research consistently demonstrated superiority in terms of accuracy and efficiency under various resource conditions [13]. This highlights the significant potential of applying these models in the context of Rupiah banknote detection.

This study compares four cutting-edge object detection models: YOLOv5 [14], YOLOv7 [15], YOLOv8 [16], EfficientDet [13] in the context of detecting Indonesian Rupiah banknotes. The aim of this research is to determine the most effective model for developing an application that helps visually impaired people in accurately recognizing and differentiating Indonesian Rupiah banknotes. The study's goal is to identify these models' distinct strengths and their applicability in the context of banknote recognition.

2. RELATED WORKS

Several methods for detecting banknotes have been investigated. Notably, feature-based approaches like Oriented FAST and Rotated BRIEF (ORB) algorithm have shown promise. Sarker and his colleagues proposed ORB as a system for assisting visually impaired individuals in real-time Bangladeshi currency detection [17]. This method demonstrated rapid matching times and achieved a 100% accuracy rate. Their system, which was created as a mobile app, proved to be a valuable tool for visually impaired people, assisting them in precise and real-time banknote identification.

Another noteworthy feature-based algorithm for banknote detection is Speeded Up Robust Features (SURF). To identify different key points and extract relevant features from the banknote image, SURF employs an arbitrary feature transformation technique. Gillich et al. applied SURF to detect the position and potential occlusion of randomly distributed textured Egyptian banknotes using a smartphone camera [18]. This method employs Random Sample and Consensus (RANSAC) algorithm to filter out false results. In all categories, the accuracy was reported to be 93%.

Another study conducted by Sufri and his team also investigates the development of an automated banknote recognition system to assist visually impaired individuals in identifying Malaysian Ringgit banknotes [19]. It assesses the influence of region and orientation on the performance of feature extraction-based Machine Learning algorithms (K-Nearest Neighbor (KNN), Direct Torque Control (DTC), Support Vector Machine (SVM), Bayesian Classifier (BC)) as well as Deep Learning via AlexNet. SVM and BC achieved 100% accuracy, whereas Deep Learning (AlexNet) performed well with similar orientation but struggled with new orientation. With the goal of enhancing the independence and standard of living for the blind and visually impaired individuals during monetary transactions, this dual approach offers valuable insights for creating strong and flexible banknote identification systems.

Deep learning-based models have brought a transformative impact to banknote detection. One successful example is the work conducted by Park and his team [20]. This research suggests a three-stage banknote and coin detection technology using a smartphone camera. To overcome the limitations of earlier approaches, this technique combines Pretrained Faster Region-based Convolutional Neural Network (R-CNN) with ResNet architecture and geometric constraints. Experiments conducted using Dijkstra's Algorithm with Buckets v1 (DKB v1) and Jordanian dinar (JOD) databases demonstrate greater accuracy when compared to current techniques. The proposed model's accuracy rates for coins, banknotes, and coins and banknotes were 95.48%, 98.8%, and 97.21%, respectively, for DKB v1 and JOD, and they were 92.11%, 97.47%, and 96.04%, respectively, for DKB v1 and JOD, respectively.

YOLOv3 is a prominent deep learning technique for detecting banknotes [11]. Park and his teams [21] present the Multinational Banknote Detecting Model (MBDM). By adding specialized structures including convolution layers, residual layers, and downsampling techniques, MBDM employs Improved YOLOv3 in this study. MBDM, with its 69 convolution layers, enhances the operations of feature extraction, prediction, and upsampling, leading to better performance in banknote detection. The alterations are intended to enhance the model's precision in identifying and localize banknotes, particularly by utilizing mosaic data enhancement during the training process. MBDM performed better than current techniques, with 83.96% accuracy. MBDM performs better in the detection of different currencies thanks to its efficient feature extraction capabilities.

The YOLOv5 algorithm is a promising option for real-time banknote detection applications because of its efficiency and versatility, which have propelled it to the forefront of banknote detection in recent developments. Notably, with an image size of 640 pixels, YOLOv5x significantly obtained an Average Precision (AP) of 50.7% with an image size of 640 pixels when tested on the Microsoft Common Objects in Context (MS COCO) dataset test-dev 2017. Additionally, it can attain a remarkable 200 frames per second (FPS) using a 32-batch batch size on an NVIDIA V100. With test-time augmentation and a larger input size of 1536 pixels, YOLOv5 attains an even higher AP of 55.8% [22]. According to a study by Dande and colleagues, the YOLOv5 model can successfully identify Indian banknotes, as evidenced by the consistently high mean Average Precision (mAP), precision, and recall in banknote detection [23]. These results collectively underscore YOLOv5's competence in delivering accurate and efficient banknote recognition.

Another noteworthy advancement is the YOLOv7 [15], as introduced by Wang and his team, presents a novel architecture for real-time object detection and model scaling, offering high accuracy of 56.8% AP, with YOLOv7-E6 delivering outstanding performance at 56 FPS and 55.9% AP. Several architectural adjustments and a set of "bag-of-freebies" were suggested by YOLOv7 to increase accuracy while maintaining the same inference speed and training time [22].

Furthermore, the state-of-the-art YOLOv8 pushes the boundaries of speed and accuracy, outperforming its predecessor YOLOv5 with an AP of 53.9% on the MS COCO dataset for 640-pixel images. The exceptional processing speed of YOLOv8 distinguishes it. It runs at an impressive 280 frames per second (FPS) on an NVIDIA A100 with TensorRT. This high FPS indicates that YOLOv8 can process and analyze a large number of frames or images per second, making it ideal for real-time applications and scenarios requiring speed. All of these improvements put YOLOv8 at the forefront of object detection technology at this time [22].

EfficientDet proposes an object detector that achieves better efficiency. By introducing a bi-directional feature pyramid network (BiFPN) and a compound scaling method, this research improves accuracy and efficiency. EfficientDetD7, the proposed model, achieves an AP value of 52.2% on the COCO test-dev dataset, while being significantly smaller and requiring fewer floating-point operations compared to previous detectors. The approaches in the study consistently excel in terms of accuracy and efficiency across various resource constraints [13].

Based on the development of these deep learning models, a comparison will be made for banknote detection between several models, namely YOLOv5, YOLOv7, YOLOv8 and EfficientDet. The goal of this study is to evaluate each model's ability to detect banknotes in order to determine which model will results in the most accurate and efficient results.

3. METHODOLOGY

The planning stage of the research involved developing a research idea and conducting an extensive literature review. The problem identification served as the foundation for directing the research toward a specific goal. Following the planning stage, the initiation stage involves dataset creation and pre-processing to ensure data integrity. The training stage of the research is where a currency detection model is developed. Validation data is used to monitor the model's performance. Following training, a fine-tuning phase is implemented to improve currency detection performance. This study also compares four object detection algorithms in the context of Rupiah banknote detection: YOLOv5, YOLOv7, YOLOv8, and EfficientDet.

A. Data Collection

The data for this study was manually collected using a mobile phone camera. The dataset consists of images of Rupiah banknotes from the emission year 2022, as shown in Figure 1. The dataset's diversity was carefully selected to include a diverse range of real-world scenarios and challenges. There are 15 background variations, each with 5 images per class, contributing to the diversity of the dataset. It contains images taken in a variety of lighting conditions, including outdoor sunlight, outdoor nighttime, well-lit indoor, and low-light indoor settings. The dataset includes banknotes held by individuals and placed on flat surfaces, as well as various background conditions and colors, flash usage, and instances of folded or crumpled banknotes. The backgrounds utilized for indoor settings included a variety of materials such as wood, tiles, and cardboard, alongside colors including red, pink, black, white, and light blue. For outdoor settings, backgrounds included park scenery. Furthermore, images were obtained from a variety of angles, including flat and oblique perspectives, to assess the method's ability to detect banknotes accurately from various perspectives. This meticulous dataset curation aims to provide a robust evaluation of object detection methods, ensuring reliable Rupiah banknote detection results under a variety of real-world conditions and scenarios.

Figure 1. 2022 Edition Rupiah Banknotes

TABLE I. Data Information [24]

Nominal	Front Image	Back Image	Size	Color
Rp100.000	Dr. (H.C.) Ir. Soekarno and Dr. (H.C.) Drs. Mohammad Hatta	Topeng Betawi dance, Raja Ampat landscape and Moon Orchid flower	151 mm x 65 mm	Red
Rp50.000	Ir. H. Djuanda Kartawidjaja	Legong dance, Komodo National Park landscape and Bali Jepun flower	146 mm x 65 mm	Blue
Rp20.000	Dr. G. S. S. J. Ratulangi	Gong dance, Derawan landscape and Black Orchid flower	141 mm x 65 mm	Green
Rp10.000	Frans Kaisiepo	Pakarena dance, Wakatobi National Park landscape, and Cempaka Kasar forest	136 mm x 65 mm	Purple
Rp5.000	Dr. K. H. Idham Chalid	Gambyong dance, Mount Bromo, and Sedap malam flower	131 mm x 65 mm	Brown
Rp2.000	Mohammad Hoesni Thamrin	Piring dance, Ngarai Sianok landscape, and Jeumpa flower	126 mm x 65 mm	Grey
Rp1.000	Tjun Meutia	Tifa dance, Banda Neira landscape, and Larat Orchid flower	121 mm x 65 mm	Green

A representative dataset was created using Bank Indonesia (BI) Rupiah banknotes from the 2022 emission year. This dataset contains seven Rupiah banknote denominations issued in the year 2022: Rp.100,000, Rp.50,000, Rp.20,000, Rp.10,000, Rp.5,000, Rp.2,000, and Rp.1,000. We captured separate images for each banknote denomination, encompassing both the front and back classes, resulting in a total of 14 classes. The resolution was set to 640 x 640 pixels for the input size. Table I contains information about the dataset that was used.

The size of each class in the dataset is 75 images per class, with a total of 1050 images of Rupiah banknotes were collected for the 2022 emission year. The dataset is carefully balanced, with each class having an equal proportion of 75 images. This balanced distribution ensures that each banknote denomination is adequately represented in the dataset, facilitating a fair evaluation of the object detection methods under consideration.

The goal of gathering this data is to create a large and representative dataset from which to conduct an accurate evaluation of the object detection method to be tested. With a targeted dataset, this study can facilitate a comprehensive assessment of the effectiveness of the object detection techniques under consideration. By collecting a dataset that includes a variety of conditions, backgrounds, viewing angles, and irregularities as described above, this study hopes to provide accurate and reliable results in evaluating the performance of the compared object detection methods in Rupiah banknote detection.

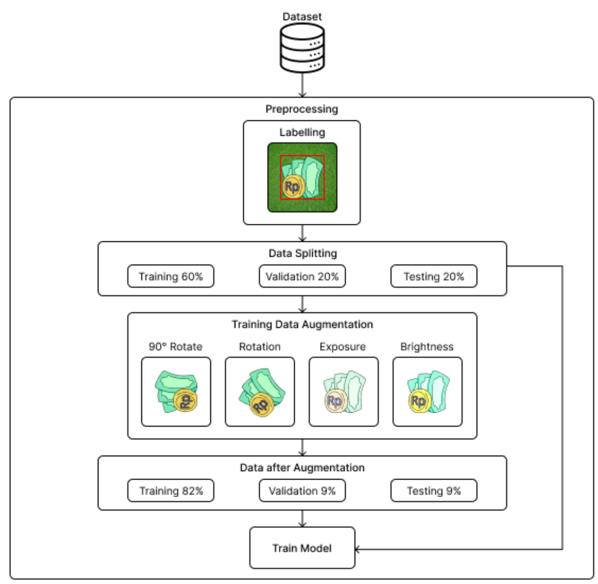


Figure 2. Preprocessing Steps

B. Preprocessing

The preprocessing stage is shown in Figure 2. The preprocessing of the banknote dataset involves several crucial steps and is primarily managed using the Roboflow platform. First, the data collection phase includes manual labelling, where each banknote instance in an image is meticulously annotated with a bounding box, and a corresponding label specifying its denomination is added. Following this labelling process, the next step is to adjust the resolution of the images. The original resolution of 3024 x 3024 is modified to 640 x 640. This resolution adjustment is necessary because the chosen YOLO detection model has a maximum resolution limitation of 640 x 640. This change ensures that the dataset complies with the requirements of the YOLO model. After the resolution adjustment is complete, the resolved dataset is ready to proceed to the data augmentation stage.

Data augmentation is an essential step in dataset processing aimed at enhancing the variety and volume of the training data. 3-time augmentation approach is applied, where one image is created with the preprocessing settings applied, and the other two undergo augmentation. This results in tripling the number of images for each source image. The augmentations used involve rotations of 90° rotations in clockwise, counter-clockwise, and upside-down orientations, as well as rotations ranging from -15° to +15°. Additionally, brightness adjustments are applied in the range of -25% to +25%, and exposure variations between -20% and +20% were also applied. The main purpose of this augmentation is to introduce diversity and increase the volume of the training dataset. These variations allow the model to learn effectively and handle various situations that may arise during banknote recognition.

The datasets are divided into two distinct sets: one without augmentation and another with augmentation. The dataset without augmentation comprises 1050 images, and it is further split into three subsets: 60% for training (630 images), 20% for validation (210 images), and 20% for testing (210 images). In contrast, the dataset with augmentation contains 2310 images, with 82% designated for training (1890 images) to benefit from the augmented data, while 9% is allocated to both the validation (210 images) and testing (210 images) subsets. These divisions allow the model to be trained on a diverse range of data examples, while the validation and testing sets remain consistent and unaltered to assess the model's performance under real-world conditions.

C. Model Building

In this study, the performance of object detection on banknote datasets will be compared using the YOLOv5, YOLOv7, YOLOv8 and EfficientDet methods. The model used are YOLOv5n, YOLOv7-tiny, YOLOv8n and EfficientDet-d0, selecting these specific versions for their streamlined and lightweight architecture, making them optimal choices for efficient object detection tasks. Each method will be customized and trained using appropriately labelled training data. Here are some details about the model that was used:

1) YOLOv5n

Jocher and his team introduced the YOLOv5n detection algorithm, which distinguishes itself through its simplicity, speed, and portability [25]. The core structure of the YOLOv5n model comprises three essential components: Backbone, Head, and Output. To extract features, the CSP-Darknet serves as the backbone network. Within the CSP-Darknet, the architecture leverages both the Focus and CSP structures. The Focus structure executes an image-slicing index operation that transforms spatial dimensions information into channel dimensions. This operation results in a double downsampled feature map, enhancing the model's inference speed. The CSP structure adopts the design principles from the CSPNet network, enabling the model to acquire a richer set of features while addressing the issue of over-computation during inference, an outcome of smart structural design [25]. The Head component is a single head architecture that takes the output of the backbone network and generates predictions. The Output component converts the predictions from the Head component into a format that can be used for the object detection task.

2) YOLOv7-tiny

YOLOv7, is built upon the ELAN (Efficient Layer Aggregation Network) architecture, known for its efficiency and accuracy. YOLOv7-tiny architecture consists of three key components: the backbone, responsible for feature extraction using CSPNet (Cross Stage Partial Network); the neck, which fuses features from different backbone levels using E-ELAN (Extended Efficient Layer Aggregation Network); and the head, which makes object predictions. YOLOv7 employs novel architecture called E-ELAN, which enhances learning capabilities through group convolution without disrupting gradient flow paths. Furthermore, it employs a strategy of using coarse features from the neck for the auxiliary head and fine features for the lead head, which improves accuracy for objects of varying sizes. YOLOv7 improves model stability and accuracy by using both auxiliary and lead heads, making it a significant advancement in real-time object detection [15].

3) YOLOv8n

YOLOv8 is an advanced real-time object detection model that enhances YOLOv5 architecture with additional features. It improves detection accuracy by fusing contextual information with high-level features using the C2f module (cross-stage partial bottleneck with two convolutions). YOLOv8 achieves higher accuracy by processing objectness, classification, and regression tasks independently through the use of a decoupled head and an anchor-free model approach. The model uses the softmax function for class probabilities and the sigmoid function for objectness score activation in the output layer. Moreover, YOLOv8 incorporates sophisticated loss functions for classification loss and bounding box loss, such as DFL and CIoU, which enhance object detection, especially for small objects [22].

YOLOv8n is a lightweight version of YOLOv8 that is optimized for speed and efficiency. YOLOv8n acquires residual features using an innovative C2f structure that preserves gradient-flow information while ensuring a lightweight design [26]. However, the YOLOv8n architecture achieves high accuracy on object detection tasks, making it an excellent choice for applications requiring speed and efficiency, such as real-time object detection on mobile devices.

4) EfficientDet-d0

EfficientDet uses the EfficientNet technique as the basis of its architecture, which is a CNN model designed to improve computational efficiency with respect to limited computing resources. In addition, the EfficientDet architecture also uses the BiFPN technique to combine features from different levels of image resolution and produce richer and more representative features [27]. EfficientDet uses two heads namely, Classification and Regression head. Classification Head is responsible for classifying the detected objects into predefined classes. The Classification Head generates confidence scores for each class of objects to be detected, allowing EfficientDet to recognize objects more accurately. The Regression Head is responsible for generating bounding box coordinates for each detected object, thus allowing EfficientDet to recognize objects more precisely [13].

5) Model Implementation

This research refrains from using pre-trained weights or transfer learning due to the following reasons. Pre-trained models are tailored to broader datasets like COCO, resulting in inefficient feature extraction for banknotes. Moreover, the absence of pre-trained models for banknote detection limits the advantages of transfer learning [28]. To address these challenges, we opt for a training-from-scratch approach, enabling custom model design specifically tuned to banknotes' unique characteristics, reducing uncertainty, and enhancing efficiency.

The experimentation process takes place on the Google Colab platform, which offers a cloud-based and collaborative environment for research and development. Google Colab is favored for its accessibility, as it provides a free and powerful computing resource, particularly beneficial for resource-intensive tasks like deep learning [29]. The use of Google Colab eliminates the need for extensive local hardware and eases the setup process. The chosen framework, PyTorch, continues to serve as the foundation for model development and evaluation, as it is fully compatible with Google Colab [29].

Two main experiments were conducted regarding model training. First, model training was conducted without augmented data, which means that the training data was not increased by adding variance or diversity. Second, model training was conducted with augmented data, where various augmentation operations were applied to the training data to introduce variety and diversity into the dataset. Furthermore, both experiments involve a hyperparameter tuning process. Hyperparameter tuning is the process of optimizing key parameters of the model, such as the number of training epochs (the number of iterations through the entire training dataset) and the learning rate (the rate at which the model learns from the data), to achieve optimal model performance. In the context of this research, the focus of hyperparameter tuning is to fine-tune these parameters so that the model can achieve maximum performance in detecting Indonesian Rupiah banknotes.

D. Evaluation

During the assessment step, several metrics are employed to comprehensively examine the performance of the developed object detection model. The mAP is the fundamental statistic for measuring performance. mAP is a widely used object detection assessment metric. The average precision (AP) throughout all classes is calculated by mAP at a given IoU threshold [30].

The evaluation results are graphed to provide an in-depth overview of the model's efficacy. The graph displays mean Average Precision (mAP) scores at different IoU thresholds, with a focus on mAP@0.5 and mAP@0.5:0.95. "mAP@0.5" denotes the average mAP at an IoU threshold of 0.5, whereas "mAP@0.5:0.95" denotes the average mAP calculated with a step size of 0.05 across multiple IoU thresholds ranging from 0.5 to 0.95 [31]. This graph serves as a valuable tool for conducting a nuanced analysis of the model's precision-recall trade-off at different IoU levels, allowing for a deeper understanding of its accuracy under varying conditions.

The loss function, with its components "box," "obj" (objectness), and "cls" (classification), is integral to the training and evaluation of object detection models and is visualized as a loss graph to represent its performance over time. The "box" component quantifies errors in bounding box localization, ensuring alignment with ground truth [32]. "Obj" assesses the model's ability to distinguish objects from non-objects, contributing to accurate identification [32]. Meanwhile, "cls" gauges the model's proficiency in categorizing objects into predefined classes [32]. Examining these loss functions in both training and validation phases through the loss graph offers valuable insights into the model's capacity to localize objects effectively, discern their presence, and achieve precise classification, aiding in model refinement and optimization.

During testing, images from the dataset are processed through the model, generating prediction results that are meticulously compared with the ground truth labels within the dataset. This assessment calculates key performance indicators such as precision and recall at varying thresholds. Moreover, the mAP@0.5 and mAP@0.5:0.95 score for the testing is calculated based on the precision and recall metrics obtained, signifying the model's overall performance.

Average inference time per image is a crucial metric that indicates the efficiency of a device in completing image processing tasks. It reflects the speed at which the device can process images, with faster inference times indicating better performance. In the context of analyzing the results of inference time measurements on test data, an evaluation was conducted. These measurements, recorded in milliseconds (ms), encompassed both model implementations without and with the application of data augmentation. The test data comprised 210 images depicting various scenarios. Each processed image was timed from the moment it entered the processing phase until the result was available, providing insights into the efficiency of the models in completing object detection tasks.

4. RESULT AND ANALYSIS

In this section, we present the results and analysis of the experiments conducted using YOLOv5, YOLOv7, YOLOv8 and EfficientDet, with and without data augmentation. The experiments were carried out with specific hyperparameters, including 320 epochs, an image size of 640x640, and a batch size of 16. The evaluation metrics used for analysis include Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95.

A. Training and Validation Results

1) YOLOv5

a) Without Augmentation

The results for YOLOv5 without data augmentation in the validation set are presented in the Table **??**. The line graph shown in Figure 3 presents YOLOv5 performance without data augmentation.

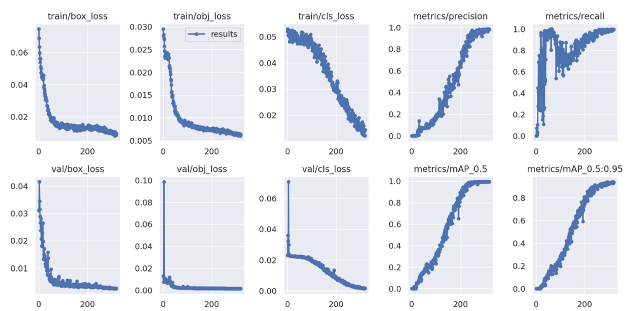


Figure 3. YOLOv5 Results without Data Augmentation

TABLE II. YOLOv5 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.983	0.998	0.995	0.941
1000-B	0.998	1.000	0.995	0.946
1000-D	0.989	0.974	0.995	0.934
10000-B	0.998	1.000	0.995	0.965
10000-D	1.000	1.000	0.995	0.968
100000-B	0.974	1.000	0.995	0.941
100000-D	0.986	1.000	0.995	0.904
2000-B	0.963	1.000	0.995	0.956
2000-D	0.987	1.000	0.995	0.958
20000-B	0.985	1.000	0.995	0.956
20000-D	0.991	1.000	0.995	0.942
5000-B	1.000	1.000	0.995	0.938
5000-D	0.978	1.000	0.995	0.945
50000-B	0.941	1.000	0.995	0.919
50000-D	0.969	1.000	0.995	0.908

b) With Augmentation

The results for YOLOv5 with data augmentation in the validation set are presented in Table **??**. The line graph shown in Figure 4 presents YOLOv5 performance with data augmentation.

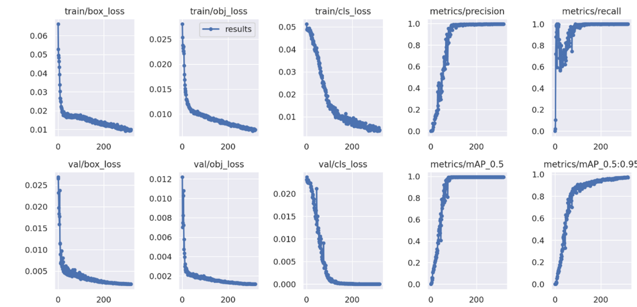


Figure 4. YOLOv5 Results with Data Augmentation

TABLE III. YOLOv5 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.975
1000-B	0.997	1.000	0.995	0.960
1000-D	0.997	1.000	0.995	0.969
10000-B	0.996	1.000	0.995	0.962
10000-D	0.998	1.000	0.995	0.981
100000-B	0.994	1.000	0.995	0.970
100000-D	0.995	1.000	0.995	0.970
2000-B	0.995	1.000	0.995	0.980
2000-D	0.996	1.000	0.995	0.957
20000-B	0.993	1.000	0.995	0.989
20000-D	0.993	1.000	0.995	0.984
5000-B	0.996	1.000	0.995	0.984
5000-D	0.993	1.000	0.995	0.977
50000-B	0.996	1.000	0.995	0.981
50000-D	1.000	1.000	0.995	0.977

2) YOLOv7

a) Without Augmentation

The results for YOLOv7 without data augmentation in the validation set are presented in the Table IV. The line graph shown in Figure 5 presents YOLOv7 performance without data augmentation.

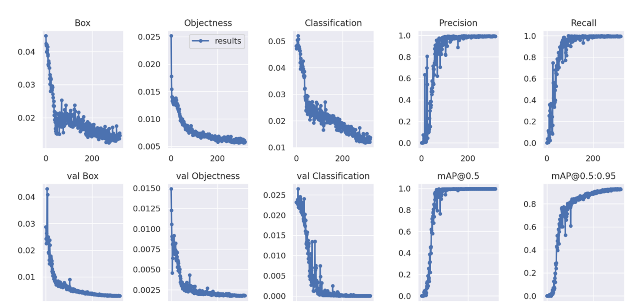


Figure 5. YOLOv7 Results without Data Augmentation

b) With Augmentation

The results for YOLOv7 with data augmentation in the validation set are presented in the Table **??**. The line graph shown in Figure 6 presents YOLOv7 performance with data augmentation.

TABLE IV. YOLOv7 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.992	0.995	0.997	0.927
1000-B	1.000	0.976	0.997	0.936
1000-D	1.000	0.956	0.997	0.922
10000-B	1.000	0.995	0.996	0.927
10000-D	0.988	1.000	0.996	0.947
100000-B	0.988	1.000	0.996	0.946
100000-D	0.986	1.000	0.997	0.919
2000-B	0.989	1.000	0.997	0.898
2000-D	1.000	1.000	0.997	0.924
20000-B	0.997	1.000	0.998	0.947
20000-D	0.983	1.000	0.997	0.927
5000-B	0.989	1.000	0.997	0.913
5000-D	0.990	1.000	0.997	0.929
50000-B	0.988	1.000	0.996	0.911
50000-D	0.984	1.000	0.997	0.933

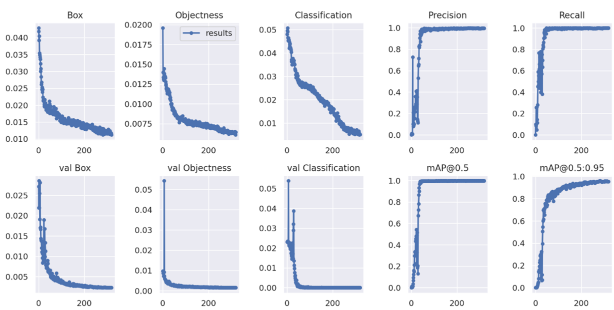


Figure 6. YOLOv7 Results with Data Augmentation

TABLE V. YOLOv7 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.995	1.000	0.998	0.954
1000-B	0.998	1.000	0.997	0.938
1000-D	0.994	1.000	0.999	0.956
10000-B	1.000	1.000	0.998	0.951
10000-D	0.996	1.000	0.998	0.964
100000-B	0.994	1.000	0.999	0.956
100000-D	0.993	1.000	0.999	0.967
2000-B	0.996	1.000	0.997	0.960
2000-D	0.994	1.000	0.999	0.959
20000-B	0.994	1.000	0.998	0.978
20000-D	0.994	1.000	0.997	0.921
5000-B	0.998	1.000	0.998	0.967
5000-D	0.993	1.000	0.997	0.952
50000-B	0.995	1.000	0.997	0.946
50000-D	0.996	1.000	0.997	0.943

3) YOLOv8

a) Without Augmentation

The results for YOLOv8 without data augmentation in the validation set are presented in the Table VI. The line graph shown in Figure 7 presents YOLOv8 performance without data augmentation.

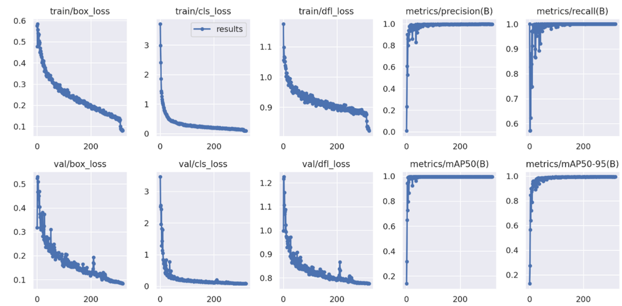


Figure 7. YOLOv8 Results without Data Augmentation

TABLE VI. YOLOv8 without Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.995
1000-B	1.000	1.000	0.995	0.995
1000-D	0.997	1.000	0.995	0.995
10000-B	0.996	1.000	0.995	0.995
10000-D	0.996	1.000	0.995	0.995
100000-B	0.996	1.000	0.995	0.995
100000-D	0.995	1.000	0.995	0.995
2000-B	0.996	1.000	0.995	0.995
2000-D	0.997	1.000	0.995	0.995
20000-B	0.996	1.000	0.995	0.995
20000-D	0.995	1.000	0.995	0.995
5000-B	0.995	1.000	0.995	0.995
5000-D	0.995	1.000	0.995	0.995
50000-B	0.996	1.000	0.995	0.995
50000-D	0.996	1.000	0.995	0.995

b) With Augmentation

The results for YOLOv8 with data augmentation in the validation set are presented in the Table **??**. The line graph shown in Figure 8 presents YOLOv8 performance with data augmentation.

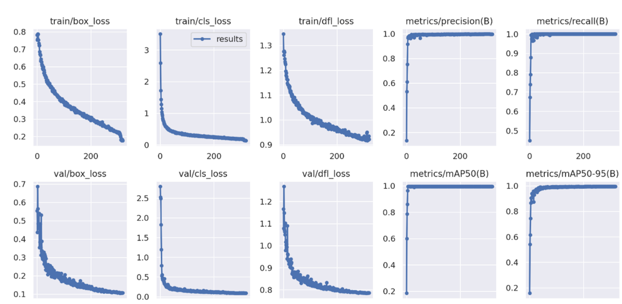


Figure 8. YOLOv8 Results with Data Augmentation

TABLE VII. YOLOv8 with Data Augmentation

Class	P	R	mAP@0.5	mAP@0.5:0.95
All	0.996	1.000	0.995	0.995
1000-B	0.997	1.000	0.995	0.995
1000-D	0.997	1.000	0.995	0.995
10000-B	0.996	1.000	0.995	0.995
10000-D	0.996	1.000	0.995	0.995
100000-B	0.996	1.000	0.995	0.995
100000-D	1.000	1.000	0.995	0.995
2000-B	0.998	1.000	0.995	0.995
2000-D	0.998	1.000	0.995	0.995
20000-B	0.994	1.000	0.995	0.995
20000-D	0.994	1.000	0.995	0.995
5000-B	0.996	1.000	0.995	0.995
5000-D	0.996	1.000	0.995	0.995
50000-B	0.995	1.000	0.995	0.995
50000-D	0.995	1.000	0.995	0.995

4) *EfficientDet*

The results for EfficientDet without and with data augmentation in the validation set are presented in Table VIII. The line graph shown in Figure 9 illustrates EfficientDet performance without data augmentation, while Figure 10 presents EfficientDet performance with data augmentation.

TABLE VIII. EfficientDet Validation Results

	mAP@0.5	mAP@0.5:0.95
Without Augmentation	0.245	0.217
With Augmentation	0.784	0.679

B. *Testing Results*

The testing results for each model, both with and without data augmentation, are summarized in Table IX.

C. *Inference Time Results*

The average inference time for each model, both with and without data augmentation, are summarized in Table X.

D. *Analysis*

In this section, we provide a comprehensive analysis of the results, covering the performance of each model, the impact of data augmentation.

1) *Model Performance*

YOLOv5 also performs well in both validation and testing scenarios, and its performance benefits significantly from data augmentation. When augmentation is applied, YOLOv5 achieves notable improvements in mAP@0.5:0.95, particularly for specific classes. This suggests that YOLOv5 is adaptive and can benefit from additional data during training. It offers a good balance between performance and adaptability, making it a strong choice

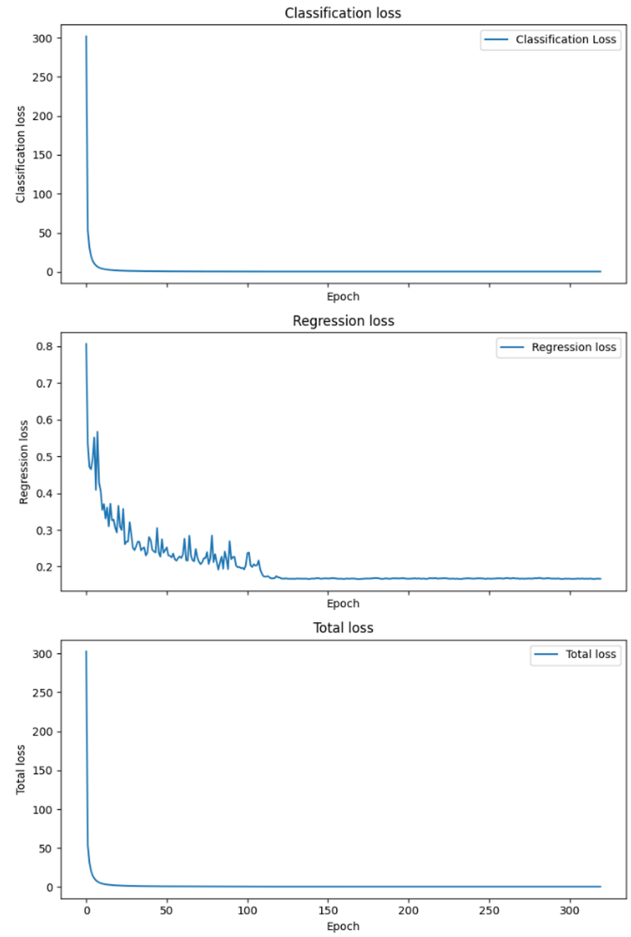


Figure 9. EfficientDet without Data Augmentation

TABLE IX. Testing Results

Model	mAP@0.5	mAP@0.5:0.95
YOLOv5 without augmentation	0.995	0.934
YOLOv5 with augmentation	0.995	0.973
YOLOv7 without augmentation	0.997	0.917
YOLOv7 with augmentation	0.998	0.947
YOLOv8 without augmentation	0.995	0.995
YOLOv8 with augmentation	0.995	0.994
EfficientDet without augmentation	0.239	0.209
EfficientDet with augmentation	0.740	0.632

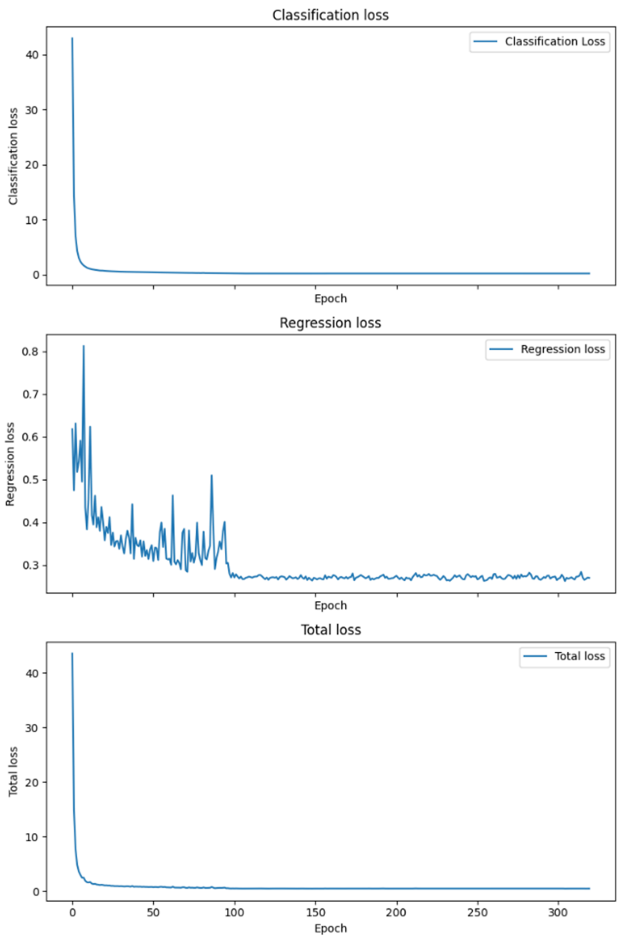


Figure 10. EfficientDet Results with Data Augmentation

TABLE X. Inference Time Results

Model	Inference Time
YOLOv5 without augmentation	6.7 ms
YOLOv5 with augmentation	6.5 ms
YOLOv7 without augmentation	5.8 ms
YOLOv7 with augmentation	6.1 ms
YOLOv8 without augmentation	7.8 ms
YOLOv8 with augmentation	8.2 ms
EfficientDet without augmentation	4.9 ms
EfficientDet with augmentation	4.5 ms

for various applications. While YOLOv5's performance is commendable, it falls slightly behind YOLOv8, which maintains consistent performance without augmentation.

YOLOv7 exhibits strong performance, especially when data augmentation is applied, resulting in higher mAP@0.5 and mAP@0.5:0.95 scores. However, when compared to YOLOv8 and YOLOv5, YOLOv7 falls slightly behind in terms of overall performance. While YOLOv7 offers good results and benefits from data augmentation, it does not outperform YOLOv8 and YOLOv5. Slightly lower mAP scores suggest that, in comparison, YOLOv7 may have limitations in specific scenarios where the highest precision and recall are critical.

YOLOv8 demonstrates consistent and robust performance in both validation and testing scenarios. The model excels without the need for data augmentation, showcasing its inherent strength. This is a significant advantage, as it implies that YOLOv8 can perform reliably in a wide range of real-world applications where augmenting data might not always be feasible or practical. YOLOv8 maintains high precision and recall across various classes, making it a top performer.

Meanwhile, EfficientDet shows a lower level of accuracy both in validation and testing without augmentation. However, with data augmentation, there was a significant improvement. Loss function analysis shows that augmentation has a positive impact on EfficientDet, improving object classification and reducing total loss.

2) Model Efficiency

YOLOv5 exhibited an average inference time of 6.7 ms without data augmentation and 6.5 ms with data augmentation. In comparison, the YOLOv7 model achieved an average inference time of 5.8 ms without data augmentation and 6.1 ms with data augmentation. On the other hand, YOLOv8 demonstrated an average inference time of 7.8 ms without data augmentation and 8.2 ms with data augmentation.

With superior inference time efficiency, EfficientDet showed the best performance. It achieved the average inference time of 4.9 ms without data augmentation, and 4.5 ms with data augmentation. EfficientDet emerged as a highly efficient choice for inference time, with stable performance across the test scenarios. However, the decision in choosing the best model should consider the balance between accuracy and efficiency.

3) Impact of Data Augmentation

Comparing the results with and without data augmentation across all models, it's evident that data augmentation significantly impacts the model's performance, particularly in terms of mAP@0.50:0.95. Augmentation enhances the models' ability to generalize and detect objects under various conditions, resulting in higher precision and recall.

The analysis of loss functions reveals the positive impact of augmentation, particularly on YOLOv5 and EfficientDet models, enhancing object classification and reducing total loss. These findings offer valuable insights into the effectiveness and adaptability of each model in the banknote object detection task. Overall, the addition of data augmentation proves beneficial for all models, particularly in scenarios with tighter Intersection over Union (IOU) thresholds.

Specifically, YOLOv5 exhibits a notable increase in mAP@0.5:0.95 when augmentation is applied, showcasing its adaptability to additional training data. Similarly, YOLOv7 demonstrates significant improvement in mAP@0.5:0.95 with augmentation, indicating the model's responsiveness to data enrichment. Although YOLOv8 maintains high performance even without augmentation, it still benefits from data augmentation, although with less pronounced differences in performance, suggesting its robustness and generalization capabilities. Notably, EfficientDet shows a dramatic improvement with augmentation, emphasizing its responsiveness to enhanced training data.

These results underscore the importance of augmentation in enhancing object detection performance across different models. They provide comprehensive insights into the effectiveness and adaptability of each model in object detection tasks, emphasizing the value of data augmentation in improving model performance and adaptability to diverse scenarios.

4) Hyperparameter Tuning

Hyperparameter tuning plays a crucial role in optimizing the performance of deep learning models. Throughout this research, extensive efforts were dedicated to fine-tuning these parameters. It's noteworthy that despite these efforts, the default hyperparameters provided for the models consistently produced the best results. The number of epochs chosen for training was set to 320, based on empirical evidence indicating that YOLOv5, the baseline model, achieved its highest accuracy at this epoch count. This decision aimed to allow the model to iterate sufficiently through the training data to converge to its optimal performance. Regarding the learning rate, a value of 0.01 was used for all models. Experimentation with different rates, including 0.1 and 0.05 for YOLOv5, did not yield satisfactory results. Higher learning rates may lead to faster convergence but can also cause overshooting or suboptimal solutions. Conversely, lower learning rates promote smoother convergence and better generalization to unseen data. By selecting a learning rate of 0.01, the aim was to strike a balance between convergence speed and generalization ability, ensuring robust performance across diverse datasets and conditions. Despite the exploration of alternative rates, the default value consistently demonstrated superior accuracy and generalization capabilities.

5. CONCLUSIONS AND FUTURE WORK

In this study, our primary objective was to evaluate the performance of cutting-edge object detection models—YOLOv5, YOLOv7, YOLOv8 and EfficientDet—in the context of Indonesian Rupiah banknote detection. Our findings indicate that YOLOv8n stands out as the top performer in terms of accuracy, consistently delivering robust results with remarkable precision and recall. YOLOv8n achieved an outstanding mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.995, both with and without data augmentation, highlighting its exceptional accuracy. YOLOv5n also performed admirably, showcasing adaptability to additional training data and substantial improvement with data augmentation. Without augmentation, it achieved mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.941, while with augmentation, it reached mAP@0.5 of 0.995 and mAP@0.5:0.95 of 0.975, emphasizing its high precision and recall. YOLOv7-tiny demonstrated strong performance, particularly with data augmentation, achieving a notable mAP@0.5 of 0.997 and mAP@0.5:0.95 of 0.927 without augmentation, and mAP@0.5 of 0.998 and mAP@0.5:0.95 of 0.954 with augmentation, though it slightly trailed YOLOv8n and YOLOv5n in overall performance. However, the EfficientDet model does not match the performance of the other three models, with mAP@0.5 and mAP@0.5:0.95 values of 0.740 and 0.632, even with the application of data augmentation. Data augmentation significantly enhanced model performance, improving their generalization and object detection capabilities, with YOLOv5 and EfficientDet benefiting the most from increased data diversity.

In terms of inference time, EfficientDet stands out as a highly efficient model with the lowest time of 4.9 ms without augmentation and 4.5 ms with augmentation. However, it should be noted that this advantage comes with low accuracy. YOLOv7 also proved to be efficient with an inference time of 5.8 ms without augmentation and 6.1 ms with augmentation, making it the second most efficient model. YOLOv5 shows quite good efficiency with an inference time of about 6.7 ms without augmentation and 6.5 ms with augmentation. On the other hand, YOLOv8, although slightly less efficient, provides adequate performance with an inference time of about 7.8 ms without augmentation and 8.2 ms with augmentation. In choosing the best model, consideration is needed regarding the balance between accuracy and efficiency of inference time. The YOLOv8 and YOLOv5 models can be considered as superior choices, depending on the prioritization between accuracy and efficiency.

Future work includes developing a mobile app with the most effective model for real-time Indonesian Rupiah banknote recognition, prioritizing mobile optimization, user feedback, and usability enhancements. Integration of features such as voice commands and multi-currency support will be explored. Extensive real-world testing with visually impaired users is planned to validate the app's effectiveness in improving their financial independence and daily transaction experiences.

6. LIMITATIONS AND CHALLENGES

The data collection process had limitations that could have influenced the dataset's representativeness and generalizability. Sampling bias was a concern due to limitations in capturing diverse scenarios and differences in image quality from mobile phone cameras. While efforts were made to include a wide range of environmental conditions and lighting scenarios in the dataset, it's possible that some conditions were not adequately represented. Furthermore, focusing solely on 2022 Rupiah banknotes and taking separate images for each denomination may result in imbalances. Additionally, ensuring consistency and accuracy in the manual annotation and labeling process posed challenges, potentially affecting dataset reliability. Recognizing these challenges is essential to improving the quality of datasets and guaranteeing the validity of research findings.

References

[1] K. Potdar, C. D. Pai, and S. Akolkar, "A convolutional neural network based live object recognition system as blind aid," *arXiv preprint arXiv:1811.10399*, Nov. 2018.

[2] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, p. 103514, Jun. 2022.

[3] S. W. Sari, "Perkembangan dan pemikiran uang dari masa ke masa," *An-Nisbah: Jurnal Ekonomi Syariah*, vol. 3, no. 1, Oct. 2016.

[4] T. Hartono and F. R. Cahaya, "The effectiveness of blind codes and security features for the blind and the prevention of fraud on the indonesian rupiah," *Asia Pacific Fraud Journal*, vol. 2, no. 2, p. 165, Feb. 2018.

[5] S.-C. Ng, C.-P. Kwok, S.-H. Chung, Y.-Y. Leung, and H.-S. Pang, "An intelligent banknote recognition system by using machine learning with assistive technology for visually impaired people," *2020 10th International Conference on Information Science and Technology (ICIST)*, Sep. 2020.

[6] J. Lee, H. Hong, K. Kim, and K. Park, "A survey on banknote recognition methods by various sensors," *Sensors*, vol. 17, no. 2, p. 313, Feb. 2017.

[7] L. P. Sousa, R. M. S. Veras, L. H. S. Vogado, L. S. Britto Neto, R. R. V. Silva, F. H. D. Araujo, and F. N. S. Medeiros, "Banknote identification methodology for visually impaired people," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020.

[8] N. Jones, H. E. Bartlett, and R. Cooke, "An analysis of the impact of visual impairment on activities of daily living and vision-related quality of life in a visually impaired adult population," *British Journal of Visual Impairment*, vol. 37, no. 1, p. 50–63, Nov. 2018.

[9] A. Orabi, A. Nasr, and A. Al-Marakeby, "Banknote recognition system for visually impaired people," *Journal of Al-Azhar University Engineering Sector*, vol. 15, no. 57, p. 1069–1077, Oct. 2020.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, p. 3212–3232, Nov. 2019.

[11] J. Park, J. Baek, K. You, S. W. Nam, and J. Kim, "Microalgae detection using a deep learning object detection algorithm, yolov3," *Journal of Korean Society on Water Environment*, vol. 37, no. 4, pp. 275–285, 2021.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[13] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10 781–10 790, 2020.

[14] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, Z. Yifu, C. Wong, D. Montes *et al.*, "ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation," *Zenodo*, 2022.

[15] 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," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[16] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics (version 8.0.0)," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[17] M. F. Rahman Sarker, M. Israfil Mahmud Raju, A. A. Marouf, R. Hafiz, S. A. Hossain, and M. Hossain Khandker Protik, "Real-time bangladeshi currency detection system for visually impaired person," *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2019.

[18] E. Gillich, A. Fritze, K.-F. Henning, A. Pfeifer, H. Dorksen, and V. Lohweg, "Camera-based occlusion detection for banknote sorting," *2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI)*, Sep. 2019.

[19] N. A. J. Sufri, N. A. Rahmad, N. F. Ghazali, N. Shahar, and M. A. As'ari, "Vision based system for banknote recognition using different machine learning and deep learning approach," *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*, Aug. 2019.

[20] C. Park, S. W. Cho, N. R. Baek, J. Choi, and K. R. Park, "Deep feature-based three-stage detection of banknotes and coins for assisting visually impaired people," *IEEE Access*, vol. 8, p. 184598–184613, 2020.

[21] C. Park and K. R. Park, "Mbdm: Multinational banknote detecting model for assisting visually impaired people," *Mathematics*, vol. 11, no. 6, p. 1392, 2023.

[22] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *arXiv preprint arXiv:2304.00501*, Apr. 2023.

[23] S. A. Dande, G. R. Uppunuri, and A. S. Raghuvanshi, "Yolov5 based web application for indian currency note detection," *Int. Res. J. Eng. Technol*, pp. 2801–2807, 2022.

[24] D. Sihwinarti and I. D. M. Sutedja, "Wacana redenominasi kaitanya uang rupiah baru di indonesia," *SENADA (Seminar Nasional Manajemen, Desain dan Aplikasi Bisnis Teknologi)*, vol. 6, pp. 304–309, 2023. [Online]. Available: https://eprosiding.idbbali.ac.id/index.php/senada/article/view/769

[25] Y. Jin, Y. Lu, G. Zhou, Q. Liu, and Y. Wang, "Glass wool defect detection using an improved yolov5," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2023.

[26] Y. Du, X. Liu, Y. Yi, and K. Wei, "Optimizing road safety: Advancements in lightweight yolov8 models and ghostc2f design for real-time distracted driving detection," *Sensors*, vol. 23, no. 21, p. 8844, Oct. 2023.

[27] M. L. Mekhalfi, C. Nicolò, Y. Bazi, M. M. Al Rahhal, N. A. Alsharif, and E. Al Maghayreh, "Contrasting yolov5, transformer,

and efficientdet detectors for crop circle detection in desert," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

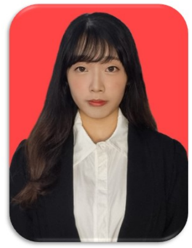
[28] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with yolo network," *Computers and Electronics in Agriculture*, vol. 169, p. 105214, Feb. 2020.

[29] S. Ray, K. Alshouiliy, and D. P. Agrawal, "Dimensionality reduction for human activity recognition using google colab," *Information*, vol. 12, no. 1, p. 6, Dec. 2020.

[30] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with yolov8," *arXiv preprint arXiv:2305.09972*, 2023.

[31] C.-y. SONG, F. ZHANG, J.-s. LI, J.-y. XIE, C. YANG, H. ZHOU, and J.-x. ZHANG, "Detection of maize tassels for uav remote sensing image with an improved yolox model," *Journal of Integrative Agriculture*, vol. 22, no. 6, p. 1671–1683, Jun. 2023.

[32] X. Zhang, Y. Feng, S. Zhang, N. Wang, and S. Mei, "Finding nonrigid tiny person with densely cropped and local attention object detector networks in low-altitude aerial images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, p. 4371–4385, 2022.



Charleen is a dedicated college student in the Bina Nusantara University Graduate Program (BGP). Pursuing a degree in Computer Science, Charleen has actively contributed to this paper through writing this paper. showcasing a strong foundation in computer vision. With a passion for research and a well-rounded approach to education, Charleen aspires to make significant future contributions to the field of computer science. She is currently in her 9th semester of her college year at Bina Nusantara University, Jakarta 11480, Indonesia. She can be contacted at her email: charleen@binus.ac.id

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.