CASH READER FOR THE VISUALLY IMPAIRED INDIVIDUALS

BY

TAN NIAN HERNG

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

MAY 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: CASH READER FOR THE VISUALLY IMPAIRED INDIVIDUALS

Academic Session: _MAY____________

I TAN NIAN HERNG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

Address:
_NO 452 JALAN RJ 1/9
TAMAN RASAH JAYA
70300 SEREMBAN
NEGERI SEMBILAN

Leung Kar Hang
Supervisor's name

Date:___24/8/2024___________________

Date: 12 Sep 2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF _INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __24/8/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that __*TAN NIAN HERNG*__________________________ (ID No: 19ACB04197) has completed this final year project/ dissertation/ thesis* entitled "_CASH READER FOR THE VISUALLY IMPAIRED" under the supervision of PROF DR LEUNG KAR HANG (Supervisor) from the Department of FICT, Faculty/Institute* of__Computer Science Faculty/Institute* of _Information and Communication Technology__________________________.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

____

(*TAN NIAN HERNG*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**CASH READER FOR THE VISUALLY IMPAIRED**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____

Name : ______TAN NIAN HERNG___________________

Date : ____24/8/2024_____________________

ACKNOWLEDGEMENTS

I extend my gratitude toward Prof Dr. Leung Kar Hang, who agreed on being my supervisor for graciously affording me this invaluable opportunity to contribute at the Cash Reader project. This will help me to establish a promising career in the dynamic field of Computer Vision. I am blessed and indebted to receive this experience and guidance.

In addition, I want to express my deepest gratitude to my parents and family. Their unwavering love, support and constant encouragement have been a pillar of strength throughout this entire course. This became my belief and my strength for my academic pursuits.

Together, these mentors and loved ones have played instrumental roles in shaping my academic and for that I extend my sincerest appreciation.

ABSTRACT

Visually impaired individuals continuously face countless challenges in their daily lives. Cash Reader stands at the forefront of accurate detection and recognition for banknotes and coins that could be used by visually impaired individuals. Traditional methods that are used to address the challenges of visually impaired individuals in recognizing the value of currency, such as the use of tactile features or Braille labels on banknotes, are helpful but come with significant limitations. Tactile features and Braille labels are not universally available, and even when present, they may not be effective for all users, particularly those with diminished tactile sensitivity or those who do not read Braille. Banknote size and shape differences, while useful, can lead to confusion, especially when new designs are introduced or when notes are worn. Folding techniques, although practical, can be cumbersome and prone to errors, while banknote templates are often inconvenient to carry and use. Additionally, reliance on assistance from others compromises the independence of visually impaired individuals and increases the risk of errors or fraud.

To overcome these challenges, the Cash Reader harnesses advanced deep learning techniques and AI to offer a robust and intuitive solution for currency recognition. Leveraging powerful models such as OpenCV and YOLOv8, it provides an accessible and highly accurate method for visually impaired individuals to independently identify Malaysian Ringgit, Malaysian coins, USD banknotes, EURO banknotes, SGD banknotes, Thai Baht banknotes and coins. The system is engineered to accurately distinguish between these denominations, enabling users to manage financial transactions with ease. By integrating these technologies, the Cash Reader promotes greater autonomy and confidence in daily financial interactions, reducing dependence on others and minimizing errors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

YOLOv8	You Only Look Once version 8
MYR	Malaysia Ringgit
SGD	Singapore Dollar
USD	United States Dollar
Baht	Thai Baht
EUR	Euro

Chapter 1

Introduction

For visually impaired individuals, managing financial transactions poses considerable obstacles, particularly when it comes to accurately identifying currency. While there are existing solutions like tactile features and Braille labels on some banknotes, these methods have significant limitations. Tactile features may not be universally available, and Braille is not always practical for users who lack the necessary sensitivity or do not use Braille. Furthermore, relying on physical characteristics such as banknote size and shape can lead to confusion, especially with newly issued or worn-out currency. Techniques like folding notes and using templates can be effective but are often cumbersome and susceptible to mistakes. Dependence on assistance from others further undermines independence and introduces potential for error and fraud.

To overcome these limitations, the Cash Reader for the Visually Impaired offers an advanced and effective solution for currency recognition. Utilizing state-of-the-art technology, this tool employs sophisticated deep learning algorithms and artificial intelligence to deliver accurate and reliable identification of various currency denominations. The Cash Reader integrates high-performance models such as OpenCV and YOLOv8 to facilitate real-time recognition of 5 different currencies which is USD, MYR, Baht, SGD and EURO.

This system addresses the shortcomings of traditional methods by providing a streamlined, user-friendly interface that enhances financial autonomy. With its advanced image processing and machine learning capabilities, the Cash Reader can accurately identify and differentiate between denominations, irrespective of changes in currency design or physical wear. This innovation not only helps visually impaired individuals manage their finances independently but also improves their confidence and reduces errors.

The Cash Reader represents a significant leap forward in supporting visually impaired individuals by integrating modern technology to meet their needs. It promotes greater independence and security in everyday financial dealings, making it a valuable tool for enhancing their overall quality of life.

1.1 Problem Statement

One of the significant challenges faced by the visually impaired community is the accurate identification and differentiation of various denominations of paper currency notes and coins. This difficulty often necessitates reliance on others for conducting financial transactions, which highlights a critical gap in social and economic inclusion. To address these needs, specialized cash readers have been developed to assist visually impaired individuals in recognizing currency.

However, achieving high accuracy and rapid processing speed while minimizing computational power consumption remains a major challenge. Researchers such as Da Costa et al. [1] and Mitsukura et al. [2] have encountered substantial computational overhead in their efforts to deliver accurate recognition, leading to longer computation times. Although their methods achieved impressive accuracy rates (e.g., 98% for coin identification), they are not well-suited for mobile applications due to their high computational demands.

Additionally, recognizing currency in uncontrolled environments presents significant difficulties that adversely impact system performance. Factors such as wrinkled or folded banknotes, varying distances from the camera, and inconsistent lighting conditions can drastically affect accuracy. Ali et al. [3] investigated these issues and found that recognition accuracy dropped from 100% to 54% when dealing with noisy or degraded currencies. The presence of additional elements, like sticky tape, can further obscure distinctive features of the currency.

A specific challenge faced in Malaysia is the difficulty in distinguishing between 20 cents and 50 cents coins due to recent design changes. The new coins have a minimal size difference of only 2.05 mm (22.65 mm for 50 cents and 20.60 mm for 20 cents), making it difficult for both visually impaired and sighted individuals to differentiate them by touch alone. This issue underscores the need for improved solutions that can effectively handle such closely similar denominations.

1.2 Motivation

The development of the Cash Reader system is driven by a profound commitment to enhancing the financial autonomy of visually impaired individuals. Financial independence is crucial for quality of life, yet for those with visual impairments, distinguishing between various currency denominations and managing money independently remains a significant challenge. Traditional methods, such as tactile features and Braille labels, have limitations due to their uneven availability and susceptibility to wear and tear, particularly when applied to frequently handled currency. The Cash Reader system aims to bridge this gap, offering a solution that not only ensures practicality but also restores dignity and self-reliance, allowing users to conduct financial transactions without relying on external assistance.

A major motivation for developing this system is the need to address the specific challenges faced by visually impaired individuals in Malaysia. While numerous assistive tools are available worldwide, very few are tailored to the unique characteristics of Malaysian currency. The Cash Reader system is designed with this local context in mind, enabling the identification of MYR, USD, EURO, SGD and Baht. This localized focus ensures that the system effectively meets the needs of the growing population of visually impaired individuals in Malaysia and globally.

Globally, visual impairment poses a significant challenge. The Global Burden of Disease Study has documented a substantial increase in blindness and vision impairments, with blindness rising by 50.6% and vision impairments by 91.7% from 1990 to 2020. According to the World Health Organization (WHO), approximately 161 million people worldwide are visually impaired, including 37 million who are completely blind. When uncorrected refractive errors are included, this number increases to around 259 million, as reported by Dandona et al. (2006) [4]. In Malaysia, the prevalence of blindness is a8pproximately 1.2% of the population, with cataracts being the leading cause, followed by diabetic retinopathy and glaucoma, as noted by Tan Sri Dr Noor Hisham Abdullah and the National Eye Survey Malaysia [5][6].

By tackling both global and local challenges in currency recognition, the Cash Reader system aims to promote greater financial independence and inclusion, significantly improving the quality of life for visually impaired individuals.

1.3 Project Scope and Direction

The primary deliverable of this project is a currency detection and recognition system, named Cash Reader, specifically designed to assist visually impaired individuals in accurately identifying and classifying currency. This system focuses on five main currencies: Malaysian Ringgit, United States Dollar (USD), Euro (EURO), Singapore Dollar (SGD), and Thai Baht (Baht)

The core of the project involves testing and implementing various deep learning models to develop a robust object detection system capable of recognizing different currency denominations. By addressing and overcoming the challenges identified in prior research such as difficulties with folded banknotes, partial currency capture, and dirty or worn banknotes this project aims to enhance the accuracy and efficiency of currency recognition.

Key objectives include:

1. **Developing a Deep Learning Model**: Implement and train several deep learning models to create an object detection system that can accurately identify and classify currency denominations.
2. **Addressing Technical Challenges**: Tackle issues related to partial or obscured currency capture and ensure that the system remains effective despite variations in currency condition.
3. **Optimizing Computational Efficiency**: Design the system to minimize computational power requirements while maintaining high accuracy, ensuring practical deployment on user-friendly devices.

Upon deployment, the Cash Reader system will provide real-time currency recognition, enabling users to confidently and independently manage financial transactions. This system represents a significant advancement in assistive technology for visually impaired individuals, addressing both practical and technical challenges to improve financial accessibility and independence.

1.4 Research Objectives

1. Data Collection and Creating a Ground Truth Dataset

The objective of this phase is to establish a robust system for the accurate recognition and classification of 5 different currency denominations, thereby enabling users to handle currency transactions with confidence and independence. The process begins with the meticulous collection of high-resolution images of banknotes and coins of MYR, USD, EURO, SGD, and Baht using an iPhone 13. The choice of a high-quality camera ensures that the images capture fine details essential for effective currency recognition.

Following the image collection, the next step involves annotating these images to create a comprehensive ground truth dataset. This annotation process is carried out using Roboflow, where bounding boxes are precisely drawn around each banknote. This step is crucial for training the object detection model, as it provides the necessary labels and spatial information required for the model to learn and identify different currency denominations accurately.

The dataset will include a diverse range of images captured under varying conditions, such as different lighting, angles, and backgrounds, to simulate real-world scenarios. Additionally, images of banknotes in various states of wear, from pristine to heavily used, will be included to enhance the system's robustness and accuracy. By systematically organizing and labelling these images, this phase ensures that the ground truth dataset is comprehensive and representative, laying a solid foundation for developing a reliable currency recognition system.

2. To Develop a Model to Detect and Recognize Banknote denominations

This research aims to develop a sophisticated detection and recognition model for accurately identifying and classifying banknote denominations across multiple currencies using state-of-the-art deep learning techniques. The study involves a comparative analysis of several models, including VGG19 and ResNet50, to identify the most effective approach, ultimately finalizing YOLOv8 for its superior accuracy, robustness, and computational efficiency. The model will be trained on an extensive and diverse dataset comprising images captured under varying conditions such as different lighting, angles, and backgrounds to ensure reliable performance in real-world environments. Moreover, the model will be fine-tuned to recognize and distinguish between multiple currencies, specifically MYR, USD, Euro, SGD, and Baht, across a range of denominations. The ultimate goal is to fully automate the currency recognition process, providing a highly accurate and accessible solution for visually impaired individuals. This system will enable rapid and precise banknote identification, even in complex and uncontrolled environments, thus significantly enhancing the autonomy and financial inclusion of its users.

3. Deploy model to detect and recognize currency denomination at real-time

This phase involves deploying the YOLOv8 model to enable real-time detection and recognition of banknote denominations using a live webcam feed. The deployment process includes initializing the YOLOv8 model with a pre-trained weight file and configuring the system to capture frames from an external webcam.

The system processes each frame by running object detection through the YOLO model, which identifies and classifies banknote denominations. The model's predictions are displayed in real time, with bounding boxes and confidence labels overlaid on the detected banknotes. Performance will be evaluated based on the model's accuracy in recognizing different currencies, such as MYR, USD, Euro, SGD, and Baht, and its efficiency in processing frames with minimal delay. This setup demonstrates the model’s capability to provide immediate and accurate currency recognition, enhancing usability for applications requiring real-time financial transactions.

1.5 Contributions

The proposed Cash Reader system primarily contributes to improving financial independence for visually impaired individuals by providing an accessible, fully automated solution for currency detection and recognition. The system accurately identifies and classifies different currency denominations, ensuring ease of use and reliability. Utilizing YOLOv8's real-time detection capabilities, the system enables users to distinguish between various currencies Malaysian Ringgit, USD, Euro, SGD, and Thai Baht without requiring assistance. Additionally, the integration of advanced deep learning models, along with precise data annotation tools like Roboflow, enhances the accuracy and robustness of the detection system across different conditions, such as varied lighting, angles, and background scenarios. This development represents a significant leap forward in assistive technology for the visually impaired, empowering users to conduct financial transactions independently and securely.

1.6 Background Information

The ability to distinguish currency denominations is crucial for everyday financial transactions. For visually impaired individuals, the challenge of recognizing different banknotes has been addressed by several traditional methods. One of the earliest approaches was the incorporation of tactile features, such as Braille dots, embossed symbols, and raised patterns on the banknotes [7][8]. These physical features allowed visually impaired individuals to distinguish between different denominations by touch [9].

Countries like Canada, India, and Malaysia have incorporated tactile marks on banknotes for this purpose. For instance, Malaysian Ringgit notes have varying tactile lines that indicate the denomination [10]. Similarly, the Euro banknotes feature raised printing to assist in identifying the currency value. However, despite these measures, tactile identification can be error-prone and dependent on the physical wear and tear of the currency, limiting its long-term effectiveness [11].

Visual aids such as magnifying devices and currency templates have also been used to help identify notes. Currency templates are plastic or metal devices that visually impaired users can slide banknotes through to determine their size, which helps in distinguishing denominations. However, these methods are often inconvenient and not fully reliable in day-to-day use.

With advancements in computer vision and deep learning, more sophisticated methods have been developed to address this problem. One of the most promising solutions in this area is the use of convolutional neural networks (CNNs) for automatic currency recognition [12]. These models are trained on large datasets of currency images under various conditions and can predict the denomination of a banknote with high accuracy.

YOLO (You Only Look Once), a real-time object detection system, has emerged as a highly efficient solution for such tasks. YOLO models have undergone several iterations, with YOLOv8 being the latest and most robust version for detecting multiple objects in images with high precision and speed [13]. YOLOv8 is particularly suited for applications like currency recognition due to its real-time processing capabilities, making it ideal for integration into mobile applications that visually impaired users can utilize to independently identify banknotes.

The strength of YOLOv8 lies in its ability to detect banknotes in various conditions, such as different lighting environments, obstructions, or overlapping currency notes. Moreover, it provides flexibility for further training on diverse currencies, including Malaysian Ringgit, USD, Euro, and Thai Baht, as part of the Cash Reader system for the visually impaired.

By leveraging such state-of-the-art deep learning methods, currency recognition systems have evolved into automated, scalable, and highly accurate tools that are paving the way for greater autonomy among visually impaired individuals in handling financial transactions [14]. These systems reduce the dependence on others, enhance confidence, and minimize errors in identifying banknotes.

Automated systems built on models like YOLOv8 are not only fast but also capable of adapting to new challenges, such as recognizing partially obscured notes or notes in non-ideal conditions (e.g., crumpled or worn-out). With such predictive capabilities, these systems outperform traditional tactile methods and are more reliable over extended periods of usage, independent of physical currency degradation.

This shift towards AI-driven solutions represents a significant milestone in accessibility technology, where individuals with visual impairments can utilize mobile devices equipped with advanced AI models to perform tasks that were once heavily reliant on manual or tactile methods.

1.7 Report Organization

The specifics of this research are outlined in the subsequent chapters. Chapter 2 comprises a literature review and discussion of the existing research. Following this, Chapter 3 provides a detailed explanation and visualization of the proposed methods for the micrometeorology forecasting system. Moving forward, Chapter 4 details how the system is set up, including both software and hardware aspects, and provides a thorough overview of how the system operates. Chapter 5 delves into a detailed evaluation and discussion on the findings and outcomes outlined in Chapter 4. Lastly, Chapter 6 concludes the research, summarizing the work done and suggesting future research directions.

Chapter 2

Literature Review

2.1 Indian Currency Recognition System Using CNN And Comparison With YOLOv5

2.1.1 Overview of the Proposed Method

In Achar et al. [15], a system for recognizing Indian currencies using Convolutional Neural Networks (CNNs) is proposed and compared with YOLOv5. The primary goal is to assist visually impaired individuals in identifying and reading out the values of Indian paper currencies with high accuracy. The system is designed to process and classify currency notes, providing audio feedback to the users.

2.1.2 Methodology

The methodology presented involves the following steps:

1. **Data Collection and Preprocessing**:
   - **Data Acquisition**: Images of Indian banknotes are collected and pre-processed. The preprocessing steps include normalization and quantization of features from separate Red, Green, and Blue (RGB) layers.
   - **Feature Extraction**: Features from the RGB layers are extracted, normalized, and converted into machine-readable data.

2. **Model Training**:
   - **Convolutional Neural Networks (CNNs)**: The CNN model is trained using the Adam optimizer, which helps in probabilistic prediction of each currency type. The model is designed to classify the banknotes and provide audio output indicating the denomination.
   - **YOLOv5**: YOLOv5, known for its efficiency in object detection, is used as a comparison model. YOLOv5's speed and performance are evaluated against the CNN model.

3. **Evaluation and Comparison**:
   - **Accuracy Measurement**: The accuracy of both CNN and YOLOv5 models is compared. The CNN model achieved an accuracy of 79.83%, which is just 1% less than the YOLOv5 model.

2.1.3 Strengths

The proposed system offers several advantages:

- **High Accuracy**: The CNN model achieves a high accuracy rate, making it effective for currency recognition.
- **Real-Time Feedback**: The system provides immediate audio feedback, which is crucial for visually impaired users.
- **Comparison with YOLOv5**: The comparison with YOLOv5 provides insights into the performance of different models, highlighting the strengths and weaknesses of CNN in currency recognition tasks.

2.1.4 Weaknesses

Some limitations of the system include:

- **Slight Accuracy Difference**: While the CNN model performs well, it is slightly less accurate than YOLOv5, which might impact its effectiveness in practical applications.
- **Computational Requirements**: Both models may require significant computational resources, which could affect real-time performance on limited hardware.
- **Dataset Limitations**: The dataset used for training and evaluation may not cover all possible variations in currency appearance, which could limit the generalizability of the models.

2.1.5 Recommendations

To address the limitations and improve the system:

- **Enhance Dataset**: Expand the dataset to include a wider variety of banknote images under different conditions to improve model robustness and accuracy.
- **Optimize Models**: Explore optimization techniques to reduce computational requirements and improve real-time performance.
- **User Feedback**: Conduct user studies to gather feedback from visually impaired users and refine the system based on their experiences and needs.

2.1.6 Future Work

Future research could focus on:

- **Integration of Advanced Models**: Combining CNN and YOLOv5 or integrating other advanced models to achieve better performance in currency recognition.

- **Scalability**: Adapting the system to handle currencies from other countries, considering variations in design and features.
- **User-Centric Design**: Further developing the system based on user feedback to enhance usability and accessibility for visually impaired individuals.

2.2 Currency recognition using a smartphone: Comparison between colour SIFT and grayscale SIFT algorithms.

2.2.1 Overview of the Proposed Method

In Doush et al. [16], a system for recognizing Jordanian coins and paper currency was proposed using the Scale-Invariant Feature Transform (SIFT) algorithm. The system aimed to accurately identify Jordanian banknotes and coins through a mobile recognition application. The focus was on evaluating the effectiveness of color SIFT versus gray SIFT in terms of processing speed and accuracy. The dataset included images of various Jordanian denominations captured under different conditions.

2.2.2 Methodology

The methodology presented involves the following steps:

1. **Data Collection and Preprocessing:**
   - **Data Acquisition:** Images of Jordanian banknotes and coins were captured using a smartphone. The dataset comprised 500 images, including diverse conditions such as different angles, distances, scales, lighting, folding, and wrinkling.
   - **Image Preprocessing:** Images were compressed to sizes under 20 KB to accelerate processing and optimize RAM usage. Automatic background removal was achieved by cropping the images to isolate the main object and define its boundaries accurately.

2. **Feature Extraction and Matching:**
   - **Using SIFT Descriptors:** The SIFT algorithm was used to extract features that were invariant to transformations such as rotation and scaling. This involved key-point detection and creating descriptors through the construction of a scale space and applying the Laplacian of Gaussian technique. Harris corner detection was used to refine key-points, which were then used to create descriptors for banknotes.
   - **Key-Point Matching:** Techniques were employed to build precise reference descriptors by counting key-points in training images and comparing them to test images using Euclidean distance. The approach identified 50 common features to improve accuracy.

3. **Evaluation and Comparison:**
    - **Performance Comparison:** The performance of color SIFT and gray SIFT was evaluated. Color SIFT demonstrated superior processing speed and accuracy compared to gray SIFT, as it utilized additional color information for object description.

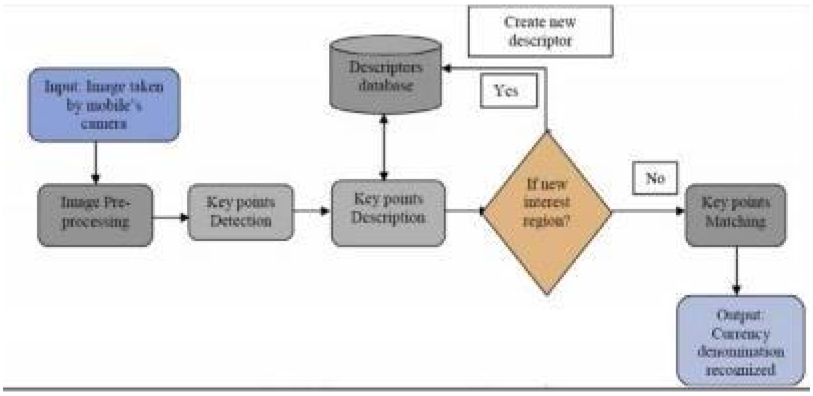


Figure 2.1 Pipeline of proposed system.

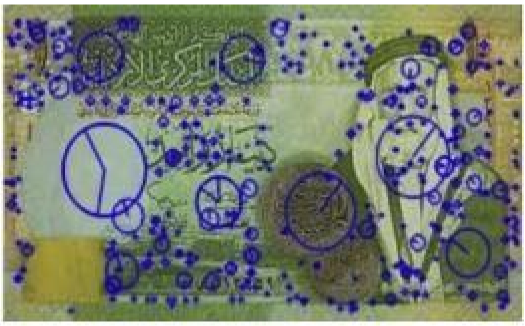


Figure 2.2 Output result of the key-points.

2.2.3 Strengths

The proposed system offers several advantages:

- **Enhanced Accuracy:** Color SIFT outperformed gray SIFT in terms of processing speed and accuracy, leveraging color information to improve feature detection.
- **Robustness:** The system effectively handled diverse conditions such as varying lighting and image distortions, demonstrating robustness in real-world scenarios.

2.2.4 Weaknesses

Some limitations of the system include:

- **Challenges with Wrinkled and Folded Banknotes:** Matching descriptors became difficult when banknotes were excessively wrinkled or folded, as key-points were obscured or altered.
- **Issues with Close-Up and Distant Shots:** Images taken from very close or distant distances either cropped significant portions or included excessive background, complicating descriptor matching.
- **Illumination Variability:** Highly illuminated images caused problems with key-point visibility, affecting the accuracy of descriptor matching.

2.2.5 Recommendations

To address the limitations and improve the system:

- **Enhance Preprocessing Techniques:** Implement restoration methods to correct distortions caused by wrinkles or folds and adaptive thresholding to manage varying lighting conditions.
- **Expand Dataset:** Include a broader variety of images with different conditions to improve model robustness and accuracy.

2.2.6 Future Work

Future research could focus on:

- **Integration of Advanced Algorithms:** Combining SIFT with other advanced algorithms to enhance performance in currency recognition.
- **Handling Variations:** Adapting the system to handle currencies from other regions and considering design variations.
- **Improving Usability:** Further developing the system based on user feedback to better meet the needs of visually impaired individuals.

2.3 Expert Systems with Applications

2.3.1 Overview of the Proposed Method

Hassanpour et al. [17] propose a method for paper currency recognition based on the Hidden Markov Model (HMM). This method achieved an impressive accuracy rate of 98%. The system was tested on Euro, Dollar, and Middle-Eastern currencies.

2.3.2 Methodology

The methodology presented involves the following steps:

1. **Pre-processing:**
   - **Wiener Filter:** The initial step involves using a two-dimensional Wiener filter for preprocessing. This filter is applied to greyscale images to remove noise, ensuring cleaner input for subsequent processing stages.
2. **Feature Extraction:**
   - **Size and Colour Considerations:** Since banknotes can become worn or torn over time, size and colour alone may not be sufficient for accurate recognition. The system employs a colour histogram approach to differentiate between various banknotes, leveraging the unique colour characteristics of each denomination.
   - **Hidden Markov Model (HMM):** To enhance the performance, the HMM approach is integrated to handle the variability in banknote appearance and improve recognition accuracy.
3. **Greyscale Quantization:**
   - **Optimizing Performance:** To minimize the execution time of the HMM algorithm, the greyscale levels of banknote images are reduced. The optimal quantization level is determined using the Jensen function, which compares the similarity between vectors to optimize performance.

2.3.3 Strengths

The proposed method offers several advantages:

- **Diverse Dataset:** The approach was evaluated using a comprehensive dataset of 150 banknotes from 101 distinct denominations representing 23 countries. This extensive dataset demonstrates the method's applicability across a wide range of currencies.
- **Versatile Recognition:** The system can recognize banknotes in any orientation due to its use of both texture and colour characteristics.

2.3.4 Weaknesses

Some limitations of the system include:

- **Recognition Failure:** The system may fail in scenarios where a portion of the currency is missing or torn, or when unusual objects such as sticky tape or additional text are present on the banknote. Examples of these issues are illustrated in Figure 2.3.1.

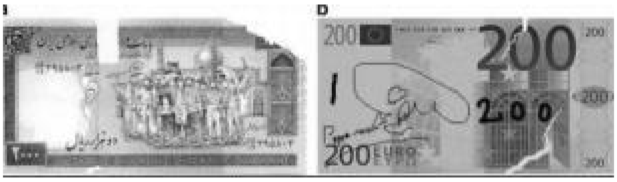


Figure 2.3. Examples of banknotes causing recognition failure: (a) Torn right corner and sticky tape; (b) Severely torn currency with a signature on the bottom left corner.

2.3.5 Recommendations

To address these limitations:

- **Handling Unusual Objects:** Apply adaptive thresholding techniques or edge detection algorithms like Canny edge detection to distinguish the banknotes and identify regions of interest despite the presence of unusual objects.
- **Restoring Torn Currency:** Implement image restoration techniques such as inpainting, interpolation, and texture synthesis to reconstruct missing or damaged portions of banknotes. These techniques aim to fill in the gaps and create plausible textures for the missing regions.

2.3.6 Future Work

Future research could focus on:

- **Improving Robustness:** Enhancing the system's robustness to handle more severe damage and unusual objects on banknotes.
- **Expanding Dataset:** Increasing the diversity of the dataset to include more variations of currency and conditions.
- **Advanced Techniques:** Exploring advanced image processing and machine learning techniques to further improve recognition accuracy and reliability.

2.4 Currency Recognition App for Blind People

2.4.1 Brief

In the paper by M. Sinthuja et al. [18], a currency recognition application is proposed to assist visually impaired individuals by identifying different currency denominations and providing audio output for easy understanding. The application focuses on detecting various currencies using mobile devices, ensuring accessibility and independence for visually impaired users.

2.4.2 Approach Method

The system involves capturing images of banknotes using a smartphone camera. The primary method used for currency recognition relies on image processing techniques and a neural network model for classification.

Pre-processing:

The first step includes noise reduction and image enhancement using filters. This process ensures that the input images are of high quality for accurate detection.

Feature Extraction and Classification:

After pre-processing, relevant features like patterns, textures, and edges are extracted. A convolutional neural network (CNN) is employed to classify the images into different currency denominations. The CNN is trained using a dataset of banknote images, enabling it to identify currency with high precision.

2.4.3 Strength

One of the major strengths of this system is its ability to operate on low-end mobile devices, making it widely accessible to users. Additionally, the system provides real-time feedback, ensuring prompt identification of currency. The integration of a speech synthesis engine for audio output further enhances user experience by offering immediate voice-based feedback.

2.4.4 Weakness

A limitation of the system is its reliance on good lighting conditions for accurate currency recognition. Poor lighting may result in incorrect detection or failure to recognize the banknote. Another weakness is the restricted dataset used for training the model, which may limit the system's performance when exposed to different variations of currency notes.

2.4.5 Recommendation

To address these issues, it is recommended to enhance the system's performance by incorporating data augmentation techniques to increase the robustness of the model. Improving the ability to recognize currency in various lighting conditions can also be achieved by using more advanced image processing techniques like adaptive histogram equalization.

2.4.6 Future Work

Future research could focus on expanding the currency types supported by the app, allowing it to handle international currencies. Additionally, introducing a feedback loop that lets users correct any misidentified currency could improve the system’s overall accuracy and user satisfaction.

2.5 YOLO-v3 Based Currency Detection and Recognition System for Visually Impaired Persons

2.5.1 Overview of the Proposed Method

Joshi et al. [19] present a currency detection and recognition system based on YOLO-v3 designed to assist visually impaired individuals. The system aims to provide accurate identification and recognition of various currency denominations, enhancing financial independence for users by leveraging advanced deep learning techniques.

2.5.2 Approach Method

Data Collection and Preparation

- **Image Acquisition:** High-resolution images of different currency notes are gathered to build a comprehensive dataset.
- **Annotation:** Currency notes are annotated with bounding boxes to train the YOLO-v3 model effectively.

Model Training

- **YOLO-v3 Training:** The YOLO-v3 model is utilized for its robust real-time object detection capabilities. The model is trained with specific parameters such as learning rate, batch size, and number of epochs to optimize performance.
- **Training Configuration:** The system involves configuring YOLO-v3's architecture to recognize currency denominations accurately.

Evaluation and Testing

- **Performance Metrics:** Evaluation is conducted using metrics like accuracy, precision, recall, and F1-score to gauge the effectiveness of the model.
- **User Testing:** The system undergoes practical testing with visually impaired users to ensure it meets their needs and provides reliable assistance in real-world scenarios.

2.5.3 Strengths

- **High Detection Accuracy:** YOLO-v3's real-time object detection capabilities contribute to high accuracy in recognizing different currency denominations.
- **Designed for Users:** The system is tailored to meet the needs of visually impaired individuals, offering practical support in daily financial transactions.

- **Real-Time Performance:** The efficiency of YOLO-v3 ensures that the system operates in real-time, enhancing usability.

2.5.4 Weaknesses

- **Detection Challenges:** YOLO-v3 may encounter difficulties in detecting currencies under suboptimal conditions such as poor lighting or image blurriness.
- **Dataset Constraints:** The effectiveness of the system is contingent on the diversity and quality of the training dataset, which may not cover all currency variations.

2.5.5 Recommendations

- **Dataset Expansion:** Increase the variety of currency notes in the dataset to improve the model's robustness and accuracy.
- **Model Enhancement:** Explore additional techniques or modifications to YOLO-v3 to improve its performance in challenging conditions.
- **User Feedback Integration:** Collect and incorporate feedback from visually impaired users to refine the system's functionality and effectiveness.

2.5.6 Future Work

- **Advanced Model Integration:** Investigate combining YOLO-v3 with other deep learning models to enhance detection accuracy and robustness.
- **Adaptation Techniques:** Develop methods to improve performance under varying lighting conditions and image qualities.
- **Enhanced User Experience:** Continuously improve the system based on user feedback to ensure it provides effective support for visually impaired users.

2.6 Innovative Currency Identifier for the Blind through Audio Output using Deep Learning

2.6.1 Overview of the Proposed Method

Maganti et al. [20] propose an innovative system designed to assist the blind in identifying currency through audio output. Utilizing deep learning techniques, the system accurately recognizes various currency denominations and communicates the results to users audibly. This approach leverages advanced neural network models to enhance the accessibility and independence of visually impaired individuals in handling currency.

2.6.2 Approach Method

Data Collection and Preprocessing

- **Data Acquisition:** A comprehensive dataset of currency images is collected, including various denominations and conditions to ensure robustness.
- **Preprocessing:** Images are standardized through resizing, normalization, and enhancement techniques to improve the performance of the deep learning models. Preprocessing steps include color normalization, noise reduction, and image augmentation to simulate diverse real-world conditions.

Deep Learning Model Implementation

- **Model Selection:** Convolutional Neural Networks (CNNs) are employed due to their effectiveness in feature extraction and image classification. A well-established CNN architecture is chosen for its ability to learn and generalize from the currency images.
- **Training and Validation:** The model is trained using a large and diverse dataset. During training, various techniques such as data augmentation, dropout, and regularization are applied to prevent overfitting and enhance the model's generalization capabilities.
- **Audio Output:** Once the currency is recognized, the system generates audio feedback to communicate the denomination to the user. Text-to-speech (TTS) technology is utilized to provide clear and accurate auditory information.

2.6.3 Strengths

- **High Accuracy:** The deep learning approach demonstrates high accuracy in recognizing different currency denominations, making it effective for practical use.
- **Accessibility:** The system provides immediate audio feedback, significantly improving accessibility for visually impaired users.
- **Robustness:** By incorporating data augmentation and preprocessing techniques, the system handles a variety of conditions and currency types, ensuring reliable performance in diverse scenarios.

2.6.4 Weaknesses

- **Dataset Limitations:** The effectiveness of the system is dependent on the diversity and quality of the dataset. Any limitations in dataset coverage could affect the model's performance in recognizing currencies not well-represented in the training data.
- **Processing Speed:** Depending on the model complexity and computational resources, there may be delays in processing and providing audio feedback, especially in real-time applications.

2.6.5 Recommendations

- **Expand Dataset:** Increase the diversity and volume of the dataset to include a wider range of currency types and conditions, improving the model's robustness and accuracy.
- **Optimize Processing:** Implement optimization techniques to enhance processing speed and efficiency, ensuring timely audio feedback for users.
- **User Testing:** Conduct extensive user testing with visually impaired individuals to gather feedback and refine the system based on real-world usability and performance.

2.6.6 Future Work

Future research could focus on:

- **Integration with Mobile Devices:** Developing a mobile application version of the system for on-the-go currency recognition.
- **Enhanced Audio Feedback:** Exploring advanced audio feedback methods to provide additional information or context about the currency.

- **Adaptability:** Improving the system's adaptability to different currencies and denominations, including potential integration with global currency recognition capabilities.

2.7 A Feature-Based Classifier for Bangla Currency Using Deep Learning

2.7.1 Overview of the Proposed Method

Nahid et al. [21] introduce a feature-based classifier for Bangla currency recognition utilizing deep learning techniques. The system is designed to enhance the accuracy of Bangla currency identification by leveraging advanced deep learning models to extract and analyze key features from currency images. This method aims to improve the reliability and efficiency of currency recognition, specifically for Bangla banknotes.

2.7.2 Approach Method

Data Collection and Preprocessing

- **Data Acquisition:** A dataset of Bangla currency images is collected, including various denominations and conditions. The dataset is curated to ensure comprehensive coverage of different features present in Bangla banknotes.
- **Preprocessing:** Images undergo preprocessing steps such as resizing, normalization, and feature enhancement. Techniques like color normalization and contrast adjustment are applied to improve the quality of the images and the performance of the deep learning models.

Feature Extraction and Model Training

- **Feature Extraction:** Key features from the currency images are extracted using deep learning models. The focus is on identifying distinctive patterns and characteristics unique to Bangla banknotes.
- **Model Implementation:** A deep learning model, specifically designed for feature extraction and classification, is employed. The model is trained using a combination of the extracted features and raw image data to enhance its ability to recognize and classify Bangla currency.
- **Training Process:** The training involves using a large dataset with diverse examples to ensure the model can generalize well across different banknote conditions. Techniques such as dropout and data augmentation are used to improve model performance and prevent overfitting.

2.7.3 Strengths

- **Feature-Based Approach:** The use of a feature-based classifier enhances the model's ability to focus on important characteristics of Bangla banknotes, leading to improved recognition accuracy.
- **Deep Learning Efficiency:** Deep learning models effectively learn and generalize from the data, providing robust performance in identifying Bangla currency.
- **Comprehensive Dataset:** The dataset covers a wide range of Bangla banknotes, ensuring that the system can handle various denominations and conditions.

2.7.4 Weaknesses

- **Dataset Constraints:** The system's performance is contingent on the quality and diversity of the dataset. Any gaps in dataset coverage could impact the model's ability to recognize less-represented banknote features.
- **Model Complexity:** The deep learning model's complexity may lead to increased computational requirements, potentially affecting real-time performance and processing speed.

2.7.5 Recommendations

- **Expand Dataset:** Enhance the dataset by including more diverse examples of Bangla currency to improve the model's robustness and accuracy.
- **Optimize Model:** Explore optimization techniques to reduce computational demands and improve the efficiency of the feature-based classifier.
- **User Feedback:** Engage with end-users to gather feedback and make necessary adjustments to the system based on practical use cases and real-world conditions.

2.7.6 Future Work

Future research could focus on:

- **Integration with Mobile Platforms:** Developing a mobile application for on-the-go Bangla currency recognition using the feature-based classifier.
- **Cross-Currency Recognition:** Extending the system to recognize and classify currencies from other regions using a similar feature-based approach.
- **Enhanced Features:** Investigating additional features and characteristics that could further improve the accuracy and reliability of the currency recognition system.

2.2.8 Comparison of the Currency Recognition system.

Table 2.1 Comparison of different currency recognition systems.

Literature Review	Method Used	Strength	Weakness
2.1	1. **Data Collection & Preprocessing:** Images of Indian banknotes are processed, with features extracted from RGB layers. 2. **Model Training:** CNN and YOLOv5 are used for currency classification and comparison. 3. **Evaluation:** CNN accuracy is 79.83%, slightly lower than YOLOv5.	• High accuracy in currency recognition. • Immediate audio feedback for visually impaired users. • Insightful comparison with YOLOv5.	• CNN accuracy is slightly lower than YOLOv5. • High computational demands may affect performance. • Dataset may not cover all currency variations.
2.2	1. **Data Collection & Preprocessing:** 500 images of Jordanian banknotes and coins were captured under various conditions. Images were compressed and background removed. 2. **Feature Extraction & Matching:** SIFT algorithm was used for feature extraction with key-point matching via Euclidean distance. Color SIFT showed superior accuracy and speed compared to gray SIFT. 3. **Evaluation:** Color SIFT demonstrated better	• Enhanced Accuracy: Color SIFT improved speed and accuracy by using color information. • Robustness: Effective in handling various conditions like lighting and distortions.	•Wrinkled/Folded Banknotes: Difficulties with matching descriptors due to obscured key-points. •Close-Up/Distant Shots: Issues with significant cropping or excessive background. •Illumination Variability: Problems with key-point visibility in highly illuminated images.

	performance in processing speed and accuracy.		
2.3	1. **Pre-processing:** Uses a Wiener filter on greyscale images to reduce noise. 2. **Feature Extraction:** Employs color histogram and Hidden Markov Model (HMM) to improve recognition by accounting for color and variability in banknote appearance. 3. **Greyscale Quantization:** Reduces greyscale levels to optimize HMM performance, using Jensen function for quantization.	• Diverse Dataset: Evaluated on 150 banknotes from 23 countries, showing broad applicability. • Versatile Recognition: Recognizes banknotes in any orientation using texture and color.	•Recognition Failure: Issues with missing, torn banknotes, or additional objects like tape or text
2.4	1. **Pre-processing:** Images are captured via smartphone and enhanced with noise reduction filters. 2. **Feature Extraction & Classification:** Features like patterns and textures are extracted, and a CNN classifies the currency into denominations. The CNN is trained on a banknote dataset.	• **Accessibility:** Operates on low-end mobile devices, making it widely accessible. • **Real-Time Feedback:** Provides immediate audio feedback through speech synthesis.	•**Lighting Dependency:** Performance can be affected by poor lighting conditions. •**Dataset Limitations:** Limited dataset may impact performance with currency variations.
2.5	1. **Data Collection & Preparation:** High-resolution images of currency notes are gathered and annotated with bounding boxes.	• **High Detection Accuracy:** YOLO-v3 provides accurate currency recognition.	• **Detection Challenges:** Issues in poor lighting or with blurry images. • **Dataset**

	2. **Model Training:** YOLO-v3 is trained for real-time object detection with specific parameters like learning rate and epochs. 3. **Evaluation & Testing:** Performance is measured using accuracy, precision, recall, and F1-score, and the system is tested with visually impaired users.	• **User-Centric Design:** Tailored for visually impaired users, aiding in financial transactions. • **Real-Time Performance:** Efficient real-time operation enhances usability.	**Constraints:** Effectiveness depends on the diversity and quality of the training dataset.
2.6	1. **Data Collection & Preprocessing:** Images of various currencies are collected, standardized, and enhanced through resizing, normalization, and augmentation. 2. **Deep Learning Model Implementation:** CNNs are used for feature extraction and classification. Techniques like data augmentation and dropout are employed during training. Audio feedback is provided via TTS technology.	• **High Accuracy:** Effective in recognizing different denominations. • **Accessibility:** Provides immediate audio feedback for visually impaired users. • **Robustness:** Handles various conditions and currency types well.	• **Dataset Limitations:** Performance depends on dataset diversity and quality. • **Processing Speed:** Model complexity and computational resources may affect real-time processing and feedback speed.
2.7	1. **Data Collection & Preprocessing:** A dataset of Bangla currency is collected and preprocessed through resizing, normalization, and feature enhancement. 2. **Feature Extraction &**	• **Feature-Based Approach:** Enhances recognition accuracy by focusing on distinctive features. • **Deep Learning**	• **Dataset Constraints:** Performance is dependent on dataset quality and diversity. • **Model Complexity:** High

	Model Training: Key features are extracted using deep learning models. A deep learning classifier is trained with diverse data to recognize Bangla currency accurately. Techniques like dropout and data augmentation are used to improve performance and prevent overfitting.	**Efficiency:** Provides robust performance and generalization. • **Comprehensive Dataset:** Covers a wide range of Bangla banknotes, improving the model's handling of various denominations.	computational requirements may affect real-time performance and processing speed.

Chapter 3
System Methodology/Approach

3.1 Design Specifications

This section provides a concise introduction to the methodology employed in the implementation of the project.

3.1.1 Methodology

The project management approach employed for this initiative is the agile methodology. This method structures the project into discrete phases, each serving as a foundational element for the next. This phased approach facilitates focused attention on specific tasks while supporting iterative development and ongoing refinement. As a result, productivity is enhanced, and project delivery is accelerated, which is particularly advantageous given the project's constrained timeline and the need to manage multiple concurrent tasks. The agile approach is crucial for addressing these constraints and ensuring successful completion within the limited timeframe. Additionally, each phase includes a testing process to verify that deliverables meet expectations. This continuous feedback loop allows the project to adapt to evolving requirements, address errors, and incorporate changes effectively.

3.1.2 General Work Procedure

The diagram illustrates the workflow of a Cash Reader system, beginning with the initialization of the process. The first step involves setting up the camera module, followed by the collection of image data for banknotes and coins. Once the data is collected, it is annotated to prepare it for training the model. The trained model is then loaded into the system for currency detection. After detection, live inference is performed on the captured images, allowing the system to identify and classify the currency denominations in real-time. The results are then output, either through a display or an audio announcement, marking the conclusion of the process.

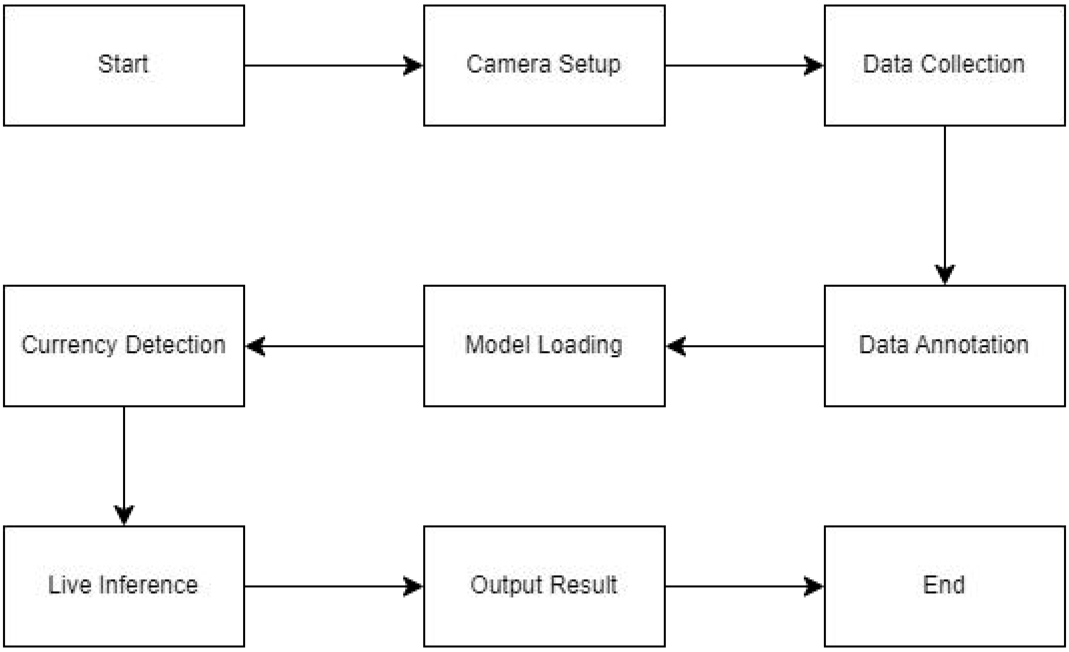


Figure 3.1 Block diagram for general work procedure.

3.2 System Design / Overview

This section contains the system block diagram and the detail explanation of each block in the block diagram. The complete system block diagram is shown in Figure 3.2.

Set up camera module:

The system is designed to utilize a camera module for capturing currency images, it is crucial to have a camera with adequate resolution and clarity to capture fine details on currency notes, such as security features, text, and colour variations.

Data Collection:

The banknote and coin images are collected using an iPhone 13, ensuring the dataset includes every possible denomination of each currency type the Cash Reader system needs to recognize, including Malaysian Ringgit (MYR), United States Dollar (USD), Euro, Singapore Dollar (SGD), and Thai Baht, with diverse perspectives, lighting conditions, backgrounds, image quality, mixed currencies, note conditions, and handling styles to enhance model accuracy.

Data Annotation:

Data annotation is a crucial step for training the YOLOv8 model to accurately recognize and classify different currency notes. The process begins with labeling each image according to the currency type and its denomination. For instance, an image might be labeled as "USD_5" if it shows a five-dollar bill, or "MYR_10" if it features a ten-ringgit note. This detailed labeling helps the YOLOv8 model understand what each image represents, enabling it to learn to identify and differentiate between various currencies during the training phase.

Roboflow offers an intuitive interface for annotating images, allowing you to create and manage labels and bounding boxes with precision. These boxes mark the exact location of the notes, providing the model with precise coordinates to focus on. This step is essential for object detection, as it trains the model to not only recognize the currency but also to locate it accurately within an image. If the bounding boxes are not drawn correctly, it could hinder the model's ability to detect and classify the currency notes effectively.

Data Preprocessing:

Data preprocessing is a critical step to ensure the dataset is well-prepared for training the You Only Look Once (YOLO) version 8 model. The preprocessing phase involves several key tasks:

1. Image Resizing: Images are resized to a consistent resolution to match the input requirements and maintaining uniformity across the dataset. This allow the model to process all images at a similar scale and aspect ratio of 640x640.
2. Data Augmentation: To enhance the robustness of the model and to prevent overfitting, data augmentation technique is implemented such as rotation, flipping, brightness, exposure and noise is added. Augmentation increases the diversity of the training dataset by artificially expanding it, which helps the model generalize better to unseen data.
3. Splitting the Dataset: The dataset is divided into training, validation, and test sets. This split is essential for evaluating the model's performance and ensuring that it generalizes well. The training set is used to train the model, the validation set helps in tuning hyperparameters and monitoring performance during training, and the test set is used for the final evaluation of the model.

Model Loading:

The YOLOv8 model is loaded into the system which is known for its speed and accuracy for object detection and classification task. The model is integrated into the system to leverage these enhancements, providing a robust solution for currency detection and classification. The system employs pre-trained weights that have been fine-tuned on a specialized currency dataset. These weights encapsulate the model's learned features, which are critical for accurately detecting and classifying various currency notes. By loading these pre-trained weights, the system ensures that the model is well-prepared to handle the specific task of currency recognition with high precision.

Currency Detection:

The model systematically scans the entire image, identifying regions where currency notes are likely to be present. This detection mechanism is foundational to the system's ability to accurately locate and process currency notes in various contexts. Once a currency note is detected, the YOLO model will predict bounding box around it, along with confidence scores that indicate the certainty of the detection.

Classification and Recognition:

After detecting a currency note, the system classifies it into one of the predefined categories. This classification is based on distinctive features like color, text and design elements that are unique to each currency. To ensure the highest level of accuracy, the system may perform additional verification checks, particularly if the model's confidence score is low. These checks help reduce the likelihood of errors, especially in cases where the visual characteristics of the currency notes are ambiguous, or the image quality is suboptimal.

Live Inference:

In live inference mode, the system processes each frame from the camera in real-time, detecting and recognizing currency notes as they appear. This real-time processing capability is essential for applications that require immediate feedback, such as for visually impaired individuals to perform transactions. The system aims to minimize latency since it is crucial in time-sensitive environments, where quick decisions and actions are necessary.

Output Result

The recognized currency type and denomination are visually displayed on a screen, providing clear and immediate results to the user or operator. This visual output facilitates easy verification and confirmation of the recognized currency. For accessibility purposes, the system can announce the recognized currency and denomination out loud. This auditory feedback is especially valuable for users with visual impairments, allowing them to interact with the system independently.

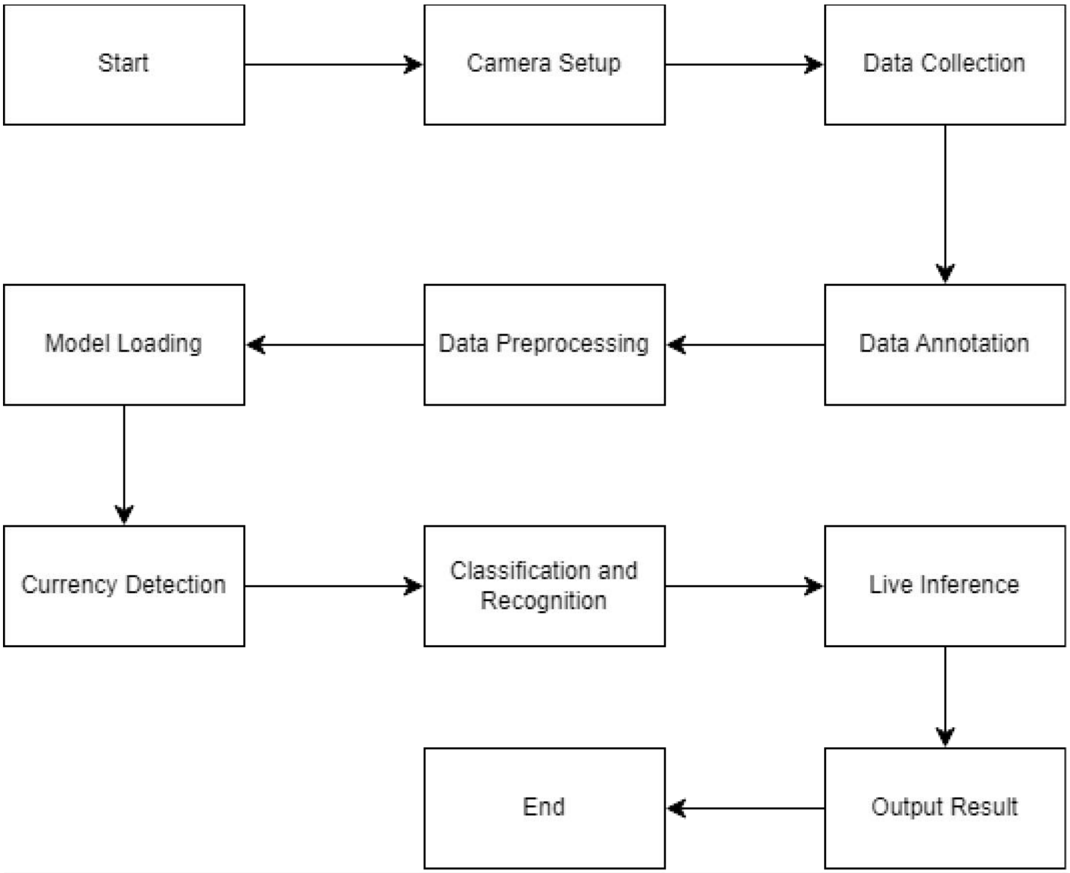


Figure 3.2 Complete system block diagram.

3.3 Use Case Design

The use case design section describes how users interact with the system and outlines the specific tasks they can perform [22]. Additionally, it encompasses use case from various perspectives, shedding light on how different stakeholders engage with the system and the tasks pertinent to their roles. Furthermore, it illustrates the system's handling of input from users and the external environment to produce outputs or responses tailored to each stakeholder's needs.

3.3.1 Use Case Diagram from System Developer's Perspective

This subsection provides an insight into the system's functionality and behaviour as perceived by the developer or creator. Besides that, this subsection also elucidates the underlying architecture and design choices made by the developers to meet the system's requirements and objectives. It offers an overview of the system's behaviour and capabilities from a development standpoint.

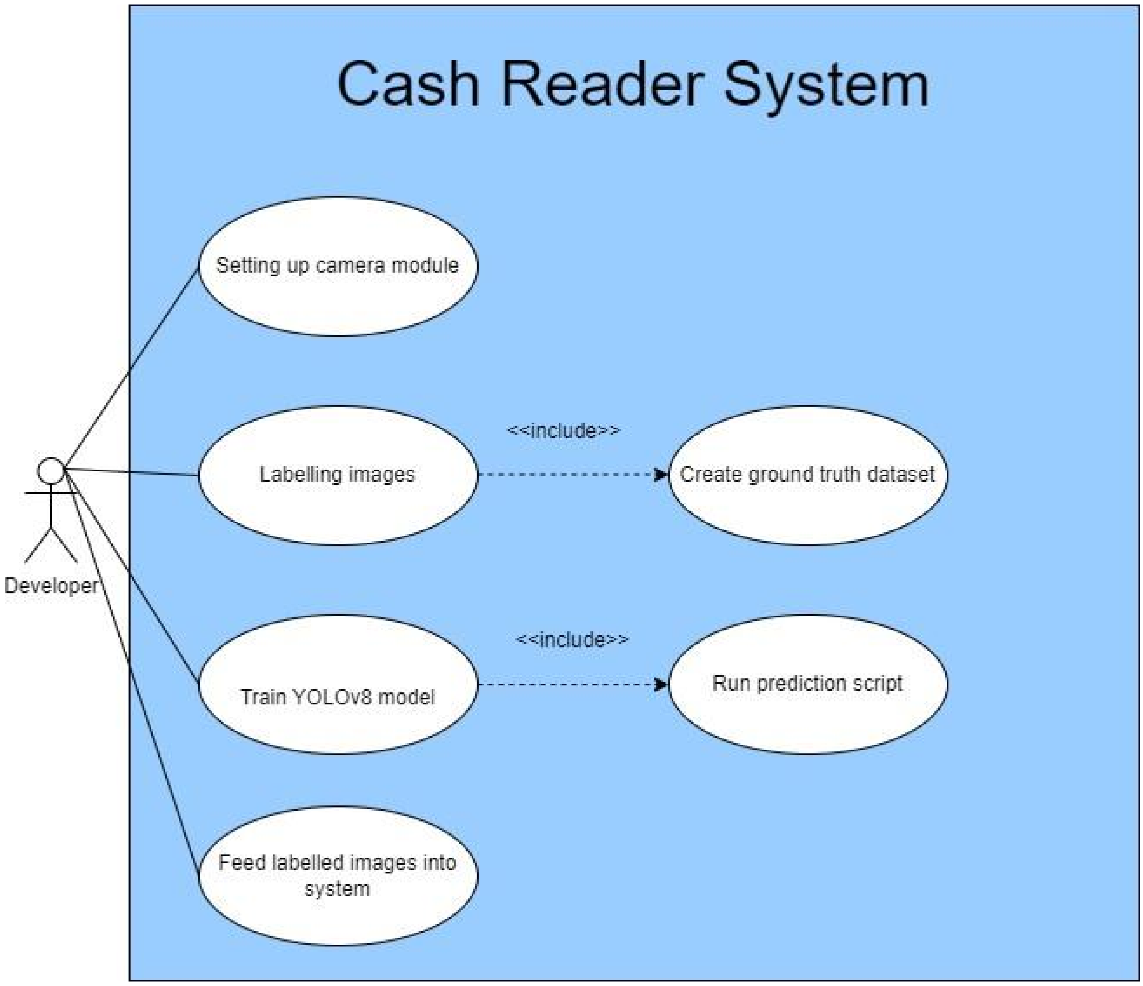


Figure 3.3 Use case diagram from system developer's perspective

3.3.2 Use Case Description

Table 3.1 Use case description – Setting up camera module

Use Case ID:	1
Use Case Name:	Setting up camera module
Brief Description:	This use case describes the process of ensuring the camera module is connected to the system
Actor:	System developer
Preconditions:	1. The camera module is powered on
Postconditions:	-
Basic Flow:	1. The system developer downloads the necessary application to set up the camera module 2. Once the camera module has been done setting up, check if it is working
Alternate Flows:	Instead of using external camera module, system developer can use camera that is built in on their phone or laptop
Exceptions:	-
Relationship:	-

Table 3.2 Use case description – Labelling images

Use Case ID:	2
Use Case Name:	Labelling images
Brief Description:	This use case describes the process of labeling images for the creation of a ground truth dataset.
Actor:	System developer
Preconditions:	1. Images have been captured and stored
Postconditions:	1. Labelled images are ready for use in the system
Basic Flow:	1.The system developer opens the image labeling tool. 2.The system developer manually labels the images by drawing bounding boxes around relevant features. 3.The labeled images are saved to the system.
Alternate Flows:	Instead of manual labeling, the system developer can use an auto-labeling tool to accelerate the process.
Exceptions:	1. Labeling errors occur due to poor image quality, requiring the system developer to re-label the images.
Relationship:	<<includes>> Create ground truth dataset

Table 3.3 Use case description – Train YOLOv8 Model

Use Case ID:	3
Use Case Name:	Tran YOLOv8 Model
Brief Description:	This use case describes the process of training the YOLOv8 model using labeled images.
Actor:	System developer
Preconditions:	1. Labeled images and annotations are available. 2. The YOLOv8 model is installed on the system.
Postconditions:	1. A trained YOLOv8 model is ready for deployment.
Basic Flow:	1. The system developer configures the training parameters for the YOLOv8 model. 2. The system developer initiates the training process. 3. The system developer monitors the training process and adjusts parameters as needed.
Alternate Flows:	The system developer may use pre-trained weights to accelerate training.
Exceptions:	1. Training is interrupted due to hardware limitations, requiring the system developer to restart the process.
Relationship:	<<includes>> Feed labeled images into system

Table 3.4 Use case description – Run Prediction Script

Use Case ID:	4
Use Case Name:	Run Prediction Script
Brief Description:	This use case describes the process of running a prediction script to test the trained YOLOv8 model.
Actor:	System developer
Preconditions:	1.The YOLOv8 model is trained and saved. 2.Test images are available.
Postconditions:	1. Prediction results are generated and saved.
Basic Flow:	1. The system developer loads the trained YOLOv8 model. 2. The system developer runs the prediction script on the test images. 3. The system developer reviews the prediction results.
Alternate Flows:	The system developer may run the prediction script on live camera feed instead of pre-stored images.
Exceptions:	1. Prediction script errors out due to incompatibility or missing dependencies.
Relationship:	<<includes>> Train YOLOv8 model

3.3.3 Use Case Diagram from User's Perspective

This section highlights the potential interactions and functionalities of the system from the viewpoint of the users perspective who intend to utilize the system.

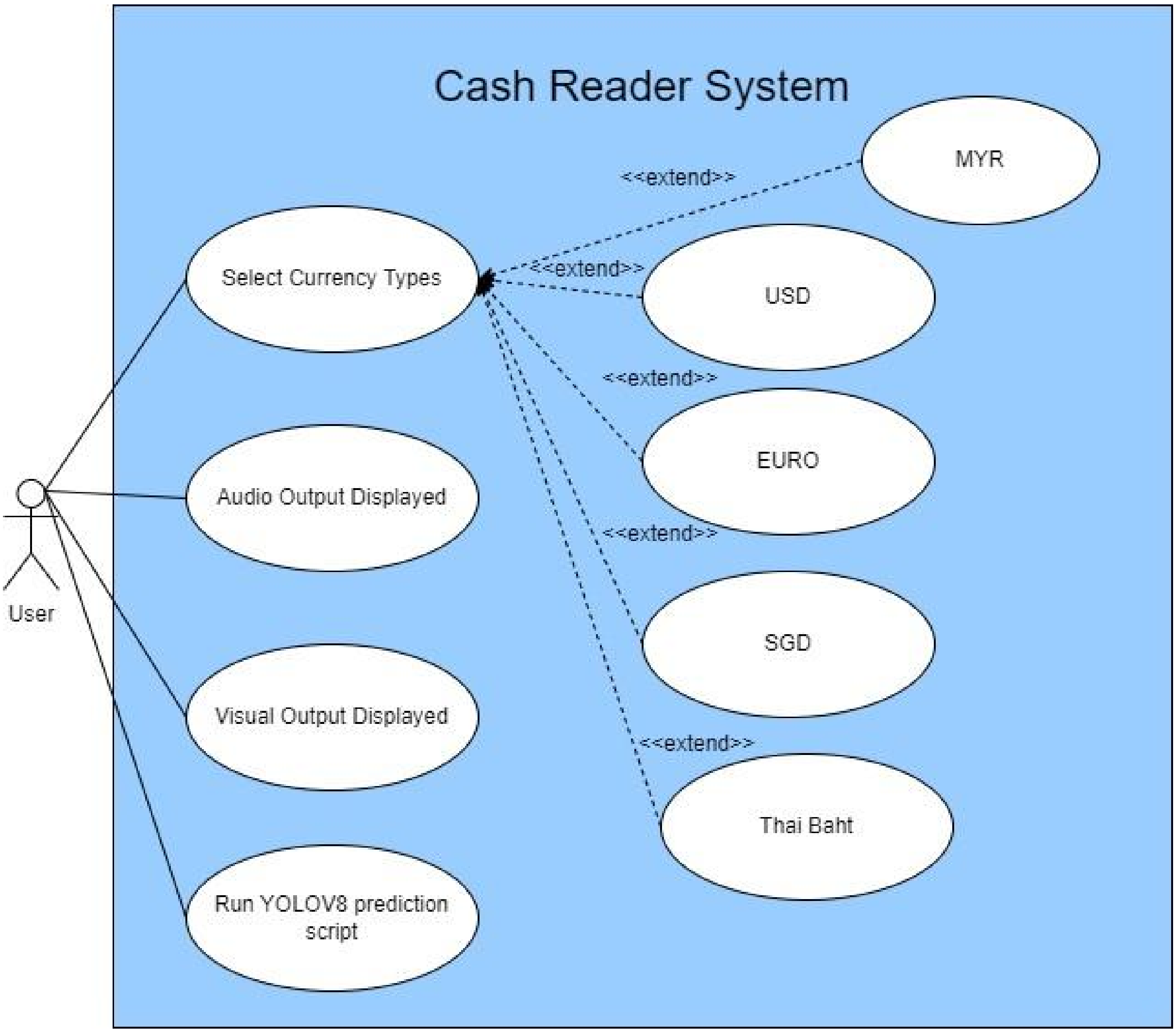


Figure 3.4 Use case diagram from user's perspective

3.3.4 Use Case Description

Table 3.5 Use case description – Select Currency Types

Use Case ID:	1
Use Case Name:	Select Currency Types
Brief Description:	This use case describes the process by which the user selects the type of currency for recognition.
Actor:	User
Preconditions:	1.The Cash Reader system is operational
Postconditions:	1. The selected currency type is used for further processing in the system.
Basic Flow:	1. The user accesses the "Select Currency Types" option. 2. The user chooses the desired currency from the list provided (MYR, USD, EURO, SGD, Thai Baht).
Alternate Flows:	1. The system automatically detects the currency type without user selection.
Exceptions:	1. The desired currency type is not available in the system, leading the user to choose the closest alternative.
Relationship:	<<extend>> MYR <<extend>> USD <<extend>> EURO <<extend>> SGD <<extend>> Thai Baht

Table 3.6 Use case description – Audio Output Displayed

Use Case ID:	2
Use Case Name:	Audio Output Displayed
Brief Description:	This use case describes the process of providing audio feedback to the user after the currency is recognized.
Actor:	User
Preconditions:	1. The system has successfully identified the currency type.
Postconditions:	1. The identified currency type is communicated to the user via audio.
Basic Flow:	1. The system generates an audio output based on the identified currency type. 2. The user listens to the audio feedback.
Alternate Flows:	1. The user can choose a different audio language or format.
Exceptions:	1. Audio output fails, requiring the system to fall back on visual output.
Relationship:	-

Table 3.7 Use case description – Visual Output Displayed

Use Case ID:	3
Use Case Name:	Visual Output Displayed
Brief Description:	This use case describes the process of providing audio feedback to the user after the currency is recognized.
Actor:	User
Preconditions:	1. The system has successfully identified the currency type.
Postconditions:	1. The identified currency type is communicated to the user via visual display.
Basic Flow:	1. The system generates a visual output (e.g., on screen) based on the identified currency type. 2. The user views the visual feedback.
Alternate Flows:	1. The user can adjust the display settings (e.g., text size, color).
Exceptions:	1. Visual output fails, requiring the system to fall back on audio output.
Relationship:	-

Table 3.8 Use case description – Run YOLOv8 Prediction Script

Use Case ID:	4
Use Case Name:	Run YOLOv8 Prediction Script
Brief Description:	This use case describes the process of running the YOLOv8 prediction script to identify the currency.
Actor:	User
Preconditions:	1. The YOLOv8 model is trained and loaded into the system.
Postconditions:	1. The prediction results are generated and used for output.
Basic Flow:	1. The user initiates the YOLOv8 prediction script. 2. The system processes the currency image and generates classification results.
Alternate Flows:	1. The user may re-run the prediction script if the results are unsatisfactory.
Exceptions:	1. Prediction script fails, requiring the user to troubleshoot or restart the process.
Relationship:	-

3.4 Activity Diagram

This section presents activity diagrams from two distinct viewpoints, which is system developer and end users. These diagrams are essential tools in both developing and comprehending software systems, providing visual depictions of activity flows within the system.

3.4.1 Activity Diagram from System Developer's Perspective

This subsection provides the perspective of system developers engaged in the development and construction of the system. The activity diagram created from the system developer viewpoint offers valuable insights into the internal mechanisms of the software, detailing how the system is built. Figure 3.6 shows the activity diagram from system developer's perspective.

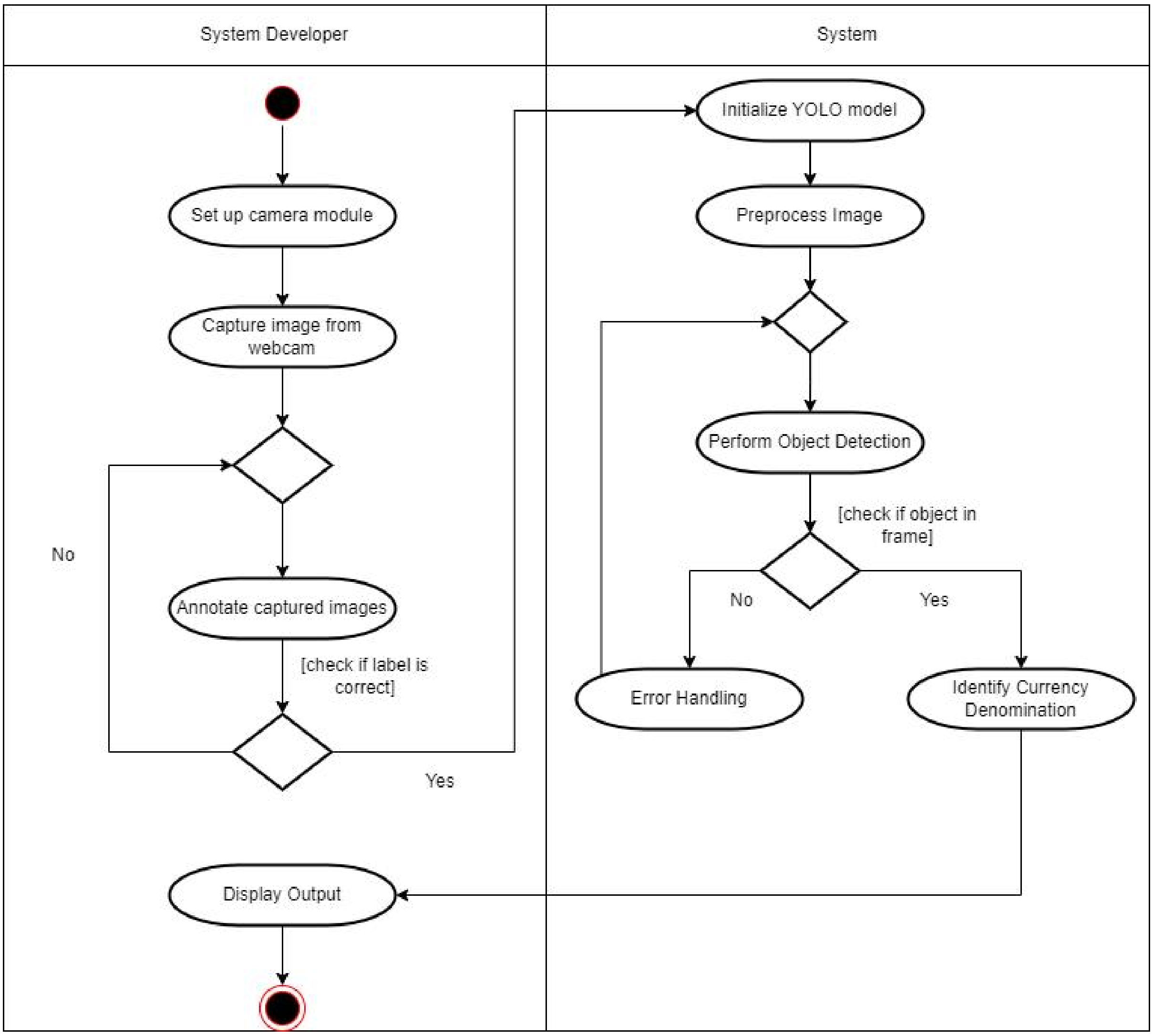


Figure 3.5 Activity diagram from system developer's perspective.

3.4.2 Activity diagram from End User's Perspective

The activity diagram from the End User's Perspective, illustrated in Figure 3.6, outlines the key steps a user follows while interacting with the Cash Reader system. The user begins by turning on the device and positioning the currency in front of the camera. The system captures the image, processes it, identifies the currency denomination, and then displays the result to the user, providing an intuitive and seamless experience.

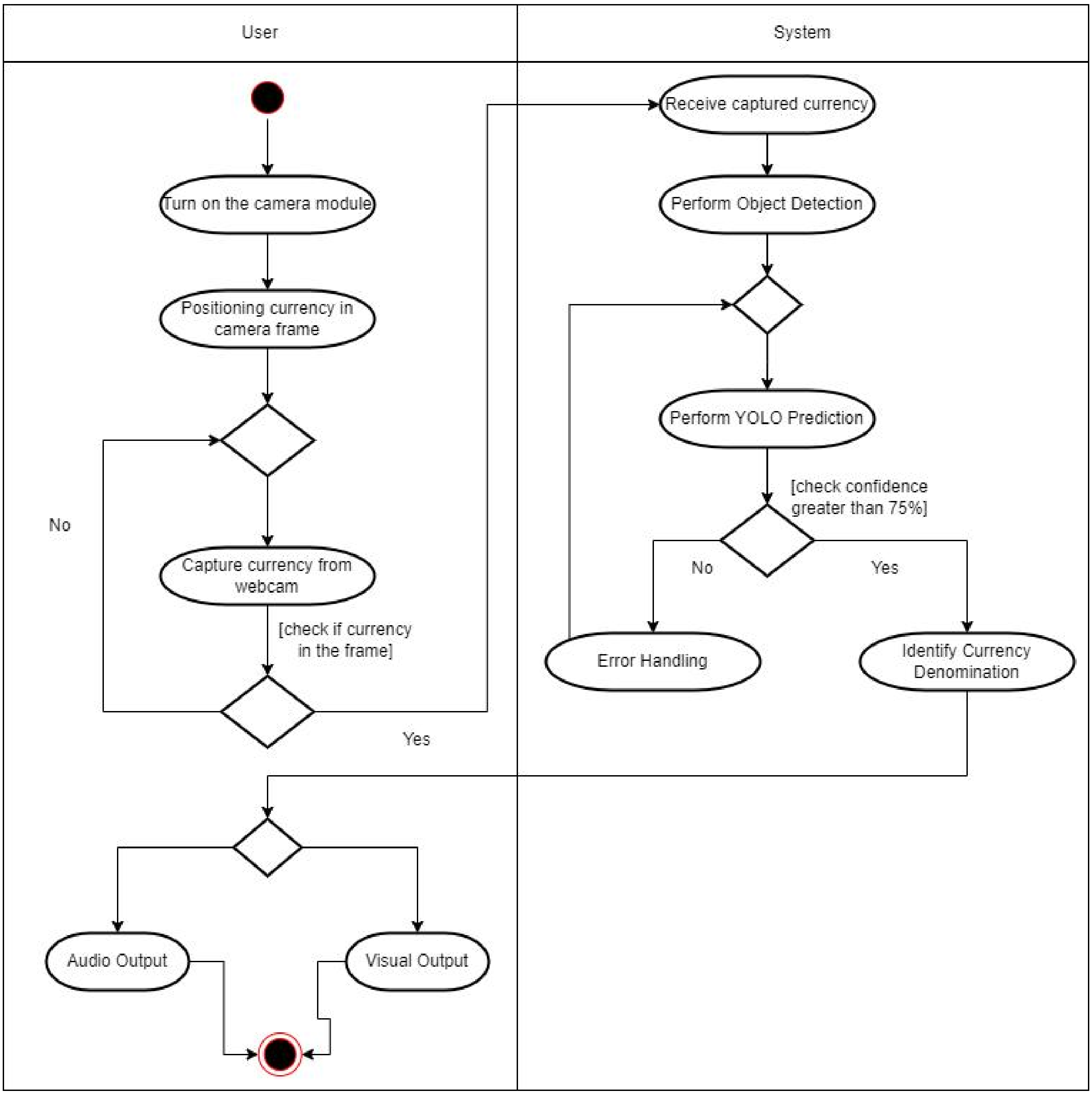

User
System
Turn on the camera module
Positioning currency in camera frame
No
Capture currency from webcam
[check if currency in the frame]
Yes
Receive captured currency
Perform Object Detection
Perform YOLO Prediction
[check confidence greater than 75%]
No
Yes
Error Handling
Identify Currency Denomination
Audio Output
Visual Output

Figure 3.6 Activity diagram from user's perspective.

3.5 System Architecture Diagram

The diagram presents a streamlined workflow for a currency recognition system, leveraging the capabilities of the YOLOv8 model. The process initiates with the **Capture Currency Image** stage, where images of various currency denominations are obtained. These images form the foundational data required for the subsequent stages.

Following image capture, the **Roboflow Labeling Approach** is employed. Roboflow serves as an advanced tool for the precise labeling and annotation of the captured images, ensuring that the dataset is adequately prepared for training. This step is crucial as it enables the model to learn the distinct features of each currency denomination.

Subsequently, the labeled dataset is utilized in the **Train Using YOLOv8 Model** phase. Here, the YOLOv8 model is trained to recognize and detect currency denominations with high accuracy. The model's training is further enhanced through **Data Augmentation**, which introduces variability into the dataset. This step improves the model's generalization ability, enabling it to perform well on new, unseen data.

In the **Object Detection and Prediction** stage, the trained model is deployed to detect and predict currency denominations in real-world images. The system identifies the currency type and its denomination, which is crucial for applications requiring real-time or batch processing of financial data.

Finally, the results are presented in the **Display Output** phase, where the detected currency and its corresponding denomination are displayed to the user. This stage completes the workflow, providing a user-friendly interface for interacting with the system.

This diagram encapsulates the end-to-end process of building and deploying a currency recognition system using state-of-the-art deep learning techniques,

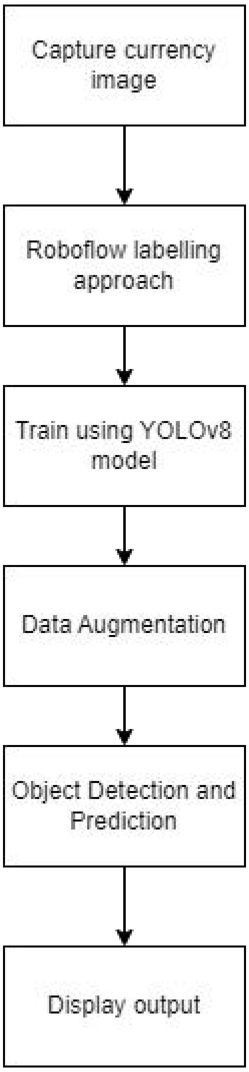
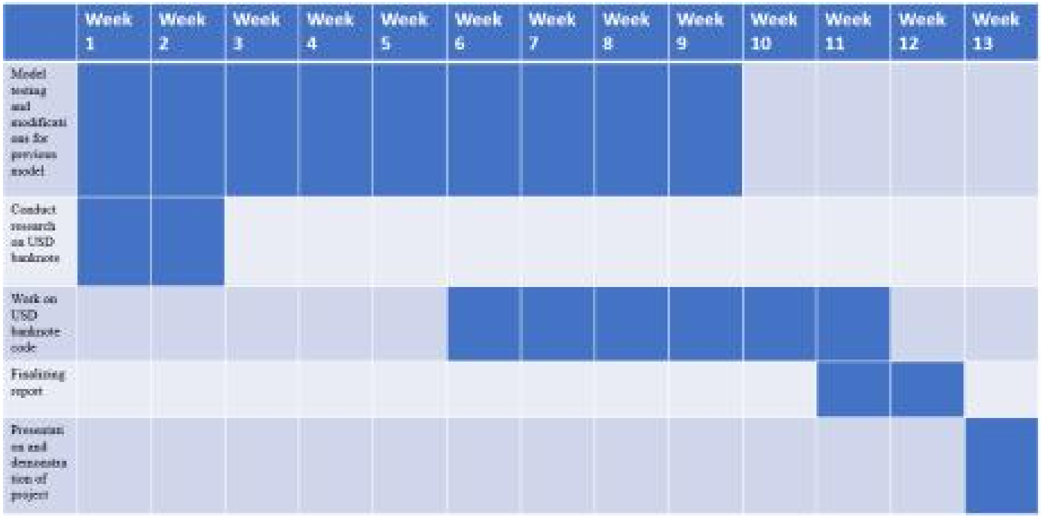


Figure 3.7 Architecture diagram of Cash Reader system.

3.6 Project Timeline

This section highlights the system's development timeline and the progress each week to improve the YOLO model.



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Model testing and modifications for previous model	■	■	■	■	■	■	■	■	■				
Conduct research on USD banknote	■	■											
Work on USD banknote code						■	■	■	■	■	■		
Finalizing report											■	■	
Presentation and demonstration of project													■

Figure 3.8 Project timeline.

Chapter 4

System Implementation

This section offers an overview of the implementation process undertaken in this project. It outlines the key steps and methodologies employed to develop and deploy the system. Each stage of the implementation is discussed in detail, highlighting the strategies, techniques, and technologies utilized. Additionally, insights into the challenges encountered and the solutions devised to overcome them are presented, providing a comprehensive understanding of the system's deployment journey.

4.1 Tools and Technologies Involved

Hardware

The hardware in this project is a laptop, Logitech webcam and iPhone13. The laptop is the main component in this project which acts as the primary tool for developing the Cash Reader system. Additionally, the iPhone 13 was initially used for data collection where currency images are captured. Logitech webcam is used to perform live-demo and test out the capabilities of the YOLOv8 model.

Table 4.1 Specifications of laptop

Description	Specifications
Model	ONYX V TGL
Processor	11th Gen Intel® Core™ i5-11400H @ 2.70GHz
Operating System	Window 11
Graphic	GeForce GTX 16
Memory	8GB DDR4 RAM X 2
Storage	512GB NVMe SSD

Table 4.2 Specifications of Logitech webcam

Description	**Specifications**
Model	Logitech C270 HD Webcam
Max Resolution	720p/30fps
Camera mega pixel	0.9
Focus type	Fixed focus

Table 4.3 Specifications of iPhone 13

Description	**Specification**
Model	iPhone 13
Memory	256GB
Camera	Dual 12MP
Chip	A15 Bionic Chip

4.2 Setting up

4.2.1 Software

Before starting to develop the Cash Reader for Visually Impaired system, 3 software application must be downloaded and installed in my laptop.

1. Logitech Camera Settings

- A software application that is used to setup the webcam and ensure that the laptop allow this external webcam to gain access.

2. Anaconda Navigator (anaconda3)

- A graphical user interface (GUI) included with the Anaconda distribution. It serves as a platform for managing and launching various tools and environments such as Jupyter Notebook.

3. Jupyter Notebook (anaconda 3)

- An open-source web application used as the primary coding platform in this project. It facilitates interactive and collaboration development of code, data analysis and visualization.

4.3 Data Collection using iPhone 13

In this process, an iPhone 13 was utilized to capture images of various currencies, including Malaysian Ringgit (MYR), United States Dollar (USD), Euro (EURO), Singapore Dollar (SGD), and Thai Baht (Baht). The dataset comprises 1,693 images of MYR, 2,519 images of USD, 2,949 images of EURO, 2,035 images of SGD, and 7,285 images of Baht.

To enhance the robustness of the model, the currencies were photographed in diverse backgrounds and environments. This approach ensures the model's ability to recognize currency accurately across different scenarios. The dataset also includes images taken under various lighting conditions, including both extreme brightness and darkness, to address illumination challenges.

To overcome issues related to obstructions, such as unwanted elements on the banknotes, the dataset includes partial images of currencies where the notes are intentionally obscured by fingers or covered with text. This variability helps the model to distinguish currencies even when only a portion of the note is visible.

Including multiple denominations of coins and banknotes in a single frame is another strategy employed to enhance the model’s capability in identifying currency in complex scenarios. This approach ensures that the model can accurately distinguish and recognize different denominations, even when they appear together in a single image.

The dataset also includes currency images captured from various angles, including inverted orientations. This consideration is crucial for visually impaired individuals, as they may not always be aware of the correct orientation of the banknote when holding it. By training the model on images from different perspectives, the system can accurately recognize and identify currencies regardless of their orientation, thereby making it more accessible and user-friendly for the visually impaired.

Thai Baht coins

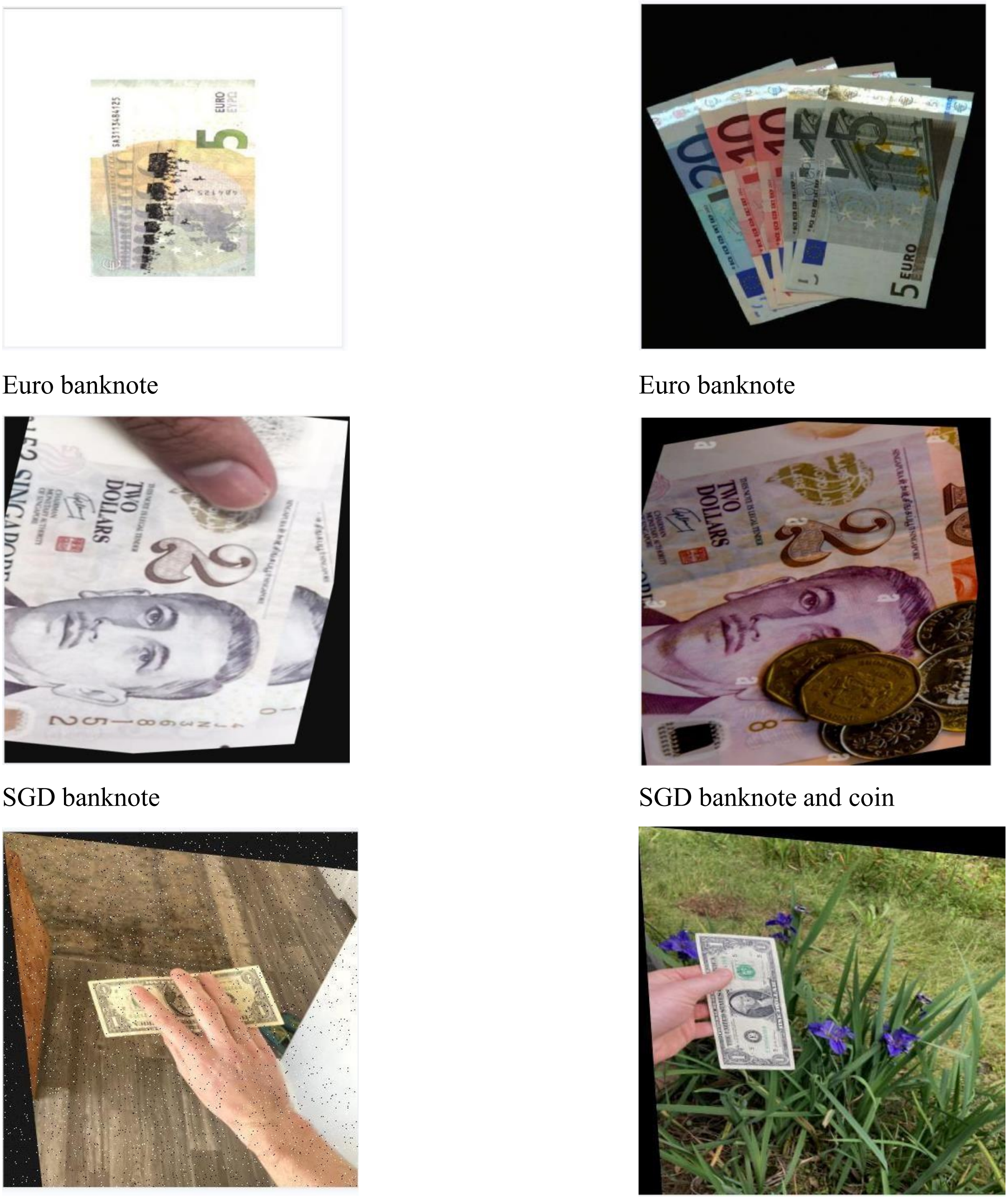
Thai Baht banknotes

MYR coins

MYR coins

MYR banknote

MYR banknote



Euro banknote

Euro banknote

SGD banknote

SGD banknote and coin

USD banknote

USD banknote

Figure 4.1 Samples of currency contained in the dataset.

4.4 Data Annotation and Labelling using Roboflow

To develop an accurate and reliable currency recognition system, creating a well-labeled ground truth dataset is essential. Roboflow, a powerful tool for data annotation and management, has been employed to facilitate this process. By utilizing Roboflow, consistency in labeling and improved accuracy of the model are ensured.

During the annotation process, each currency denomination is treated as a distinct class. For example, the Malaysian Ringgit notes such as RM1, RM5, RM10, and RM50, as well as different coin denominations, are individually annotated with their respective class labels. This allows the model to precisely differentiate between various denominations during training.

The process begins with uploading the collected images into the Roboflow platform. These images, captured under diverse conditions such as varying brightness levels, angles, and partial obstructions, provide a robust dataset for training. Each image is carefully labeled by drawing bounding boxes around the currency and assigning the appropriate class label.

Roboflow's interface simplifies the labeling process with tools that allow for easy adjustment and management of annotations. The ability to zoom, rotate, and adjust the bounding boxes ensures that even the smallest details are captured accurately. Additionally, the tool's version control feature allows for continuous improvement of the dataset, as any necessary adjustments or corrections can be easily implemented.

Once the annotation is completed, the dataset is exported in a format compatible with the YOLOv8 model, ensuring seamless integration with the training pipeline. The consistency in labeling provided by Roboflow enhances the model’s ability to generalize across different scenarios, leading to more accurate and reliable predictions during deployment.

In summary, Roboflow plays a crucial role in creating a high-quality, annotated ground truth dataset for currency recognition. The tool not only streamlines the annotation process but also contributes to the overall accuracy and effectiveness of the deep learning model used in this project. Figure 4.2 is an illustration of the labelling process using Roboflow.

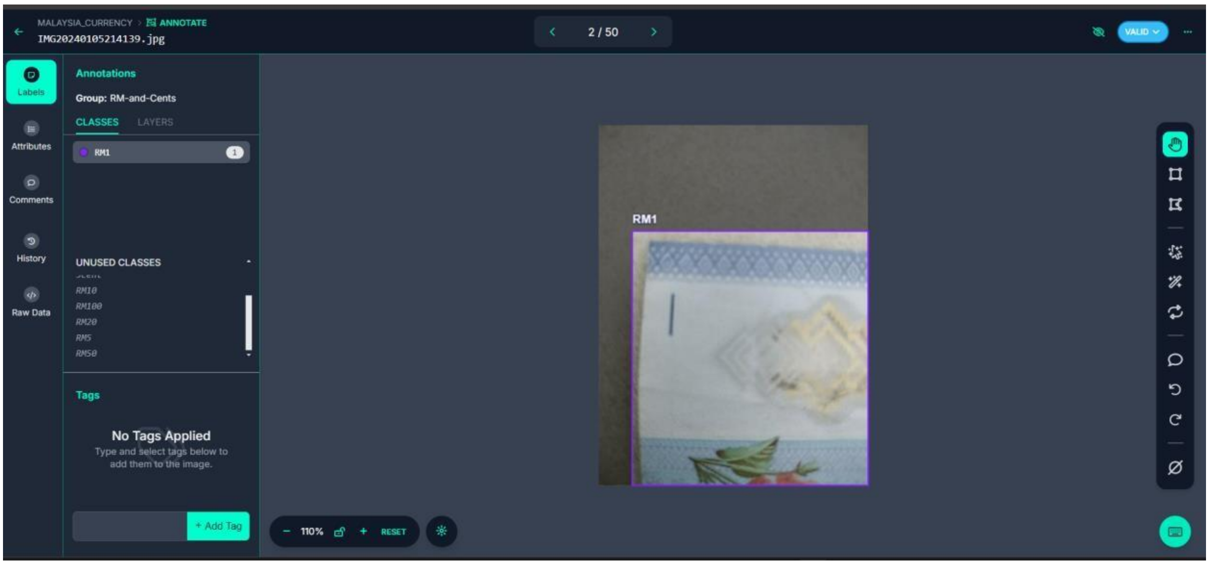


Figure 4.2 Data annotation and labelling using Roboflow.

4.5 Data Augmentation and Preprocessing

To enhance the robustness and accuracy of the currency recognition model, a systematic preprocessing and data augmentation pipeline was applied to the annotated ground truth dataset prior to its utilization in the training phase. These preprocessing and augmentation techniques are critical in preparing the dataset to reflect the diverse conditions encountered in real-world scenarios, thereby improving the model's generalization capabilities.

A. Preprocessing

- **Auto-Orientation:** An auto-orientation step was employed to ensure all images are uniformly aligned. This process corrects any inadvertent misalignment that might have occurred during image acquisition, thereby providing a consistent baseline for subsequent operations.
- **Resizing:** All images were resized to a standardized resolution of 640 × 640 pixels. This resizing not only ensures uniform input dimensions for the model but also optimizes computational efficiency during training.
- **Class Modification:** A class remapping operation was conducted, resulting in five classes being remapped. This step was essential to align the class labels with the model’s classification schema. Notably, no classes were dropped during this process, thereby preserving the dataset's comprehensiveness.

B. Data Augmentation

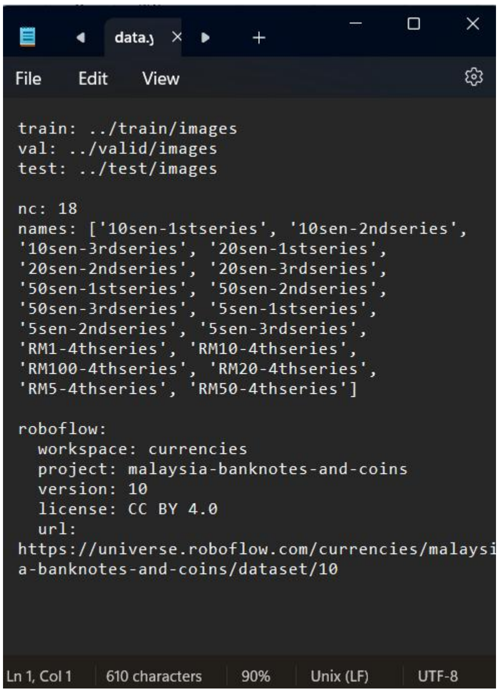
To further augment the diversity of the training data and simulate various real-world conditions, a series of data augmentation techniques were applied. For each training sample, three augmented versions were generated, utilizing the following transformations:

- **90° Rotation:** Images were rotated by 90° increments, including clockwise, counterclockwise, and upside-down orientations. This augmentation ensures that the model remains invariant to the orientation of the currency notes.
- **Cropping:** A cropping transformation was applied, with a zoom range spanning from 0% (no zoom) to 70% maximum zoom. This technique enhances the model's ability to recognize currency notes even when they are partially framed or zoomed in.
- **Rotation:** A random rotation within the range of -15° to +15° was applied to simulate slight angular deviations, thereby training the model to handle scenarios where currency notes are not perfectly aligned.
- **Shearing:** Horizontal and vertical shearing transformations were implemented with maximum shear values of ±29° horizontally and ±24° vertically. This augmentation simulates perspective distortions, challenging the model to maintain accuracy under such conditions.
- **Brightness Adjustment:** The brightness levels of the images were adjusted within the range of -25% to +25%, simulating varying lighting conditions. This ensures that the model can perform effectively across a broad spectrum of illumination scenarios.
- **Cutout:** A cutout augmentation was introduced, where five rectangular sections of each image, covering 50% of the area, were randomly occluded. This technique forces the model to focus on the most salient features of the currency, even when parts of the image are obstructed.

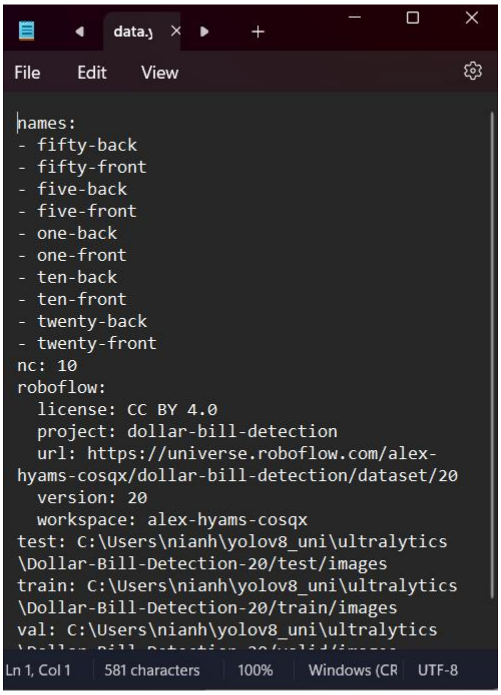
The application of these preprocessing and data augmentation techniques is pivotal in developing a resilient and generalizable currency recognition model. By incorporating these transformations into the training pipeline, the model is better equipped to perform accurately in diverse and uncontrolled environments, thereby improving its practical applicability.

4.6 Creating data.yaml for model

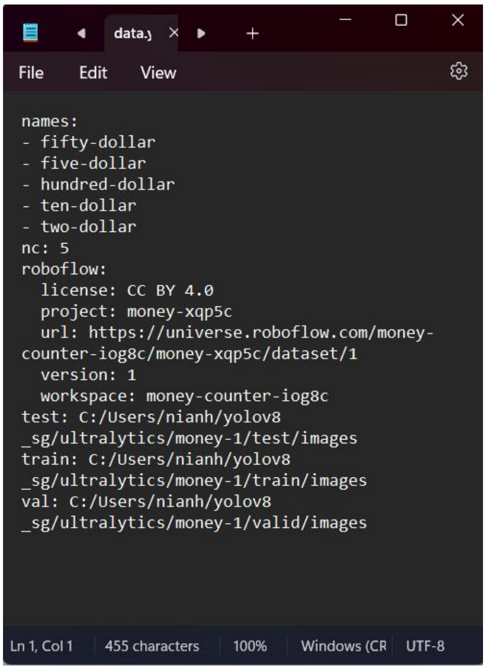
In order to effectively train the model for currency denomination recognition, a configuration file 'data.yaml' was created. This file provedes the essential information about the dataset, including the paths for training, validation and testing images, the number of classes and the names of those classes. Additionally, it contains metadata about the dataset, including the Roboflow project details from which the dataset was annotated and labelled.



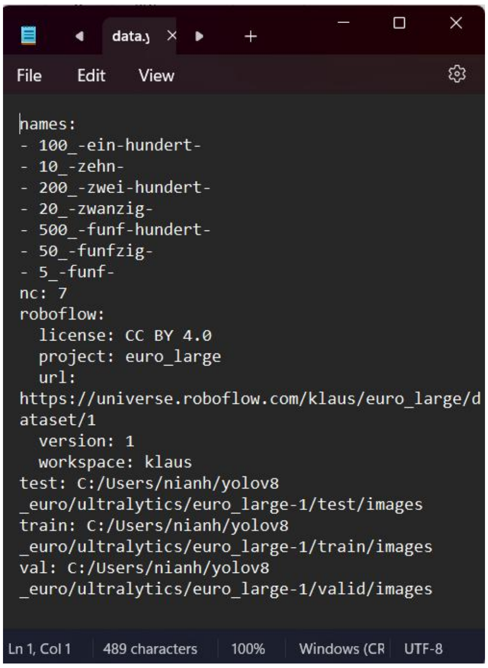
```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 18
names: ['10sen-1stseries', '10sen-2ndseries',
'10sen-3rdseries', '20sen-1stseries',
'20sen-2ndseries', '20sen-3rdseries',
'50sen-1stseries', '50sen-2ndseries',
'50sen-3rdseries', '5sen-1stseries',
'5sen-2ndseries', '5sen-3rdseries',
'RM1-4thseries', 'RM10-4thseries',
'RM100-4thseries', 'RM20-4thseries',
'RM5-4thseries', 'RM50-4thseries']

roboflow:
  workspace: currencies
  project: malaysia-banknotes-and-coins
  version: 10
  license: CC BY 4.0
  url:
https://universe.roboflow.com/currencies/malaysi
a-banknotes-and-coins/dataset/10
```



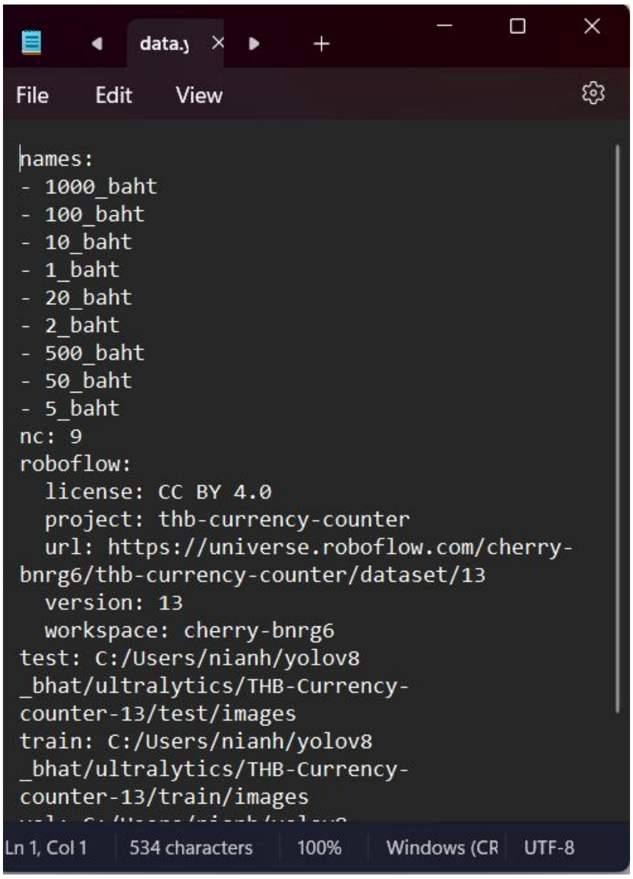
```
names:
- fifty-back
- fifty-front
- five-back
- five-front
- one-back
- one-front
- ten-back
- ten-front
- twenty-back
- twenty-front
nc: 10
roboflow:
  license: CC BY 4.0
  project: dollar-bill-detection
  url: https://universe.roboflow.com/alex-
hyams-cosqx/dollar-bill-detection/dataset/20
  version: 20
  workspace: alex-hyams-cosqx
test: C:\Users\nianh\yolov8_uni\ultralytics
\Dollar-Bill-Detection-20/test/images
train: C:\Users\nianh\yolov8_uni\ultralytics
\Dollar-Bill-Detection-20/train/images
val: C:\Users\nianh\yolov8_uni\ultralytics
```



```
names:
- fifty-dollar
- five-dollar
- hundred-dollar
- ten-dollar
- two-dollar
nc: 5
roboflow:
  license: CC BY 4.0
  project: money-xqp5c
  url: https://universe.roboflow.com/money-
counter-iog8c/money-xqp5c/dataset/1
  version: 1
  workspace: money-counter-iog8c
test: C:/Users/nianh/yolov8
_sg/ultralytics/money-1/test/images
train: C:/Users/nianh/yolov8
_sg/ultralytics/money-1/train/images
val: C:/Users/nianh/yolov8
_sg/ultralytics/money-1/valid/images
```



```
names:
- 100_-ein-hundert-
- 10_-zehn-
- 200_-zwei-hundert-
- 20_-zwanzig-
- 500_-funf-hundert-
- 50_-funfzig-
- 5_-funf-
nc: 7
roboflow:
  license: CC BY 4.0
  project: euro_large
  url:
https://universe.roboflow.com/klaus/euro_large/d
ataset/1
  version: 1
  workspace: klaus
test: C:/Users/nianh/yolov8
_euro/ultralytics/euro_large-1/test/images
train: C:/Users/nianh/yolov8
_euro/ultralytics/euro_large-1/train/images
val: C:/Users/nianh/yolov8
_euro/ultralytics/euro_large-1/valid/images
```



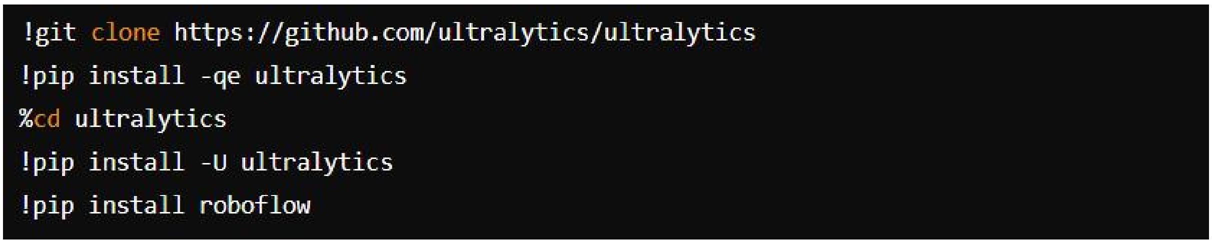
```
names:
- 1000_baht
- 100_baht
- 10_baht
- 1_baht
- 20_baht
- 2_baht
- 500_baht
- 50_baht
- 5_baht
nc: 9
roboflow:
  license: CC BY 4.0
  project: thb-currency-counter
  url: https://universe.roboflow.com/cherry-
bnrg6/thb-currency-counter/dataset/13
  version: 13
  workspace: cherry-bnrg6
test: C:/Users/nianh/yolov8
_bhat/ultralytics/THB-Currency-
counter-13/test/images
train: C:/Users/nianh/yolov8
_bhat/ultralytics/THB-Currency-
counter-13/train/images
```

Figure 4.3 Top left: MYR file, Top right:USD file, Middle left:SGD file, Middle right: Euro file, Bottom: Thai Baht file.

4.7 Implementation of YOLOv8 Model

4.7.1 Environment Setup and Dependencies

The training environment was set up on a system equipped with a GPU to efficiently handle the computational requirements of deep learning. The process began by cloning the Ultralytics YOLOv8 repository from GitHub, ensuring access to the latest features and updates. The required dependencies, including the YOLOv8 framework and the Roboflow package, were installed. The commands used for this setup were as follows:

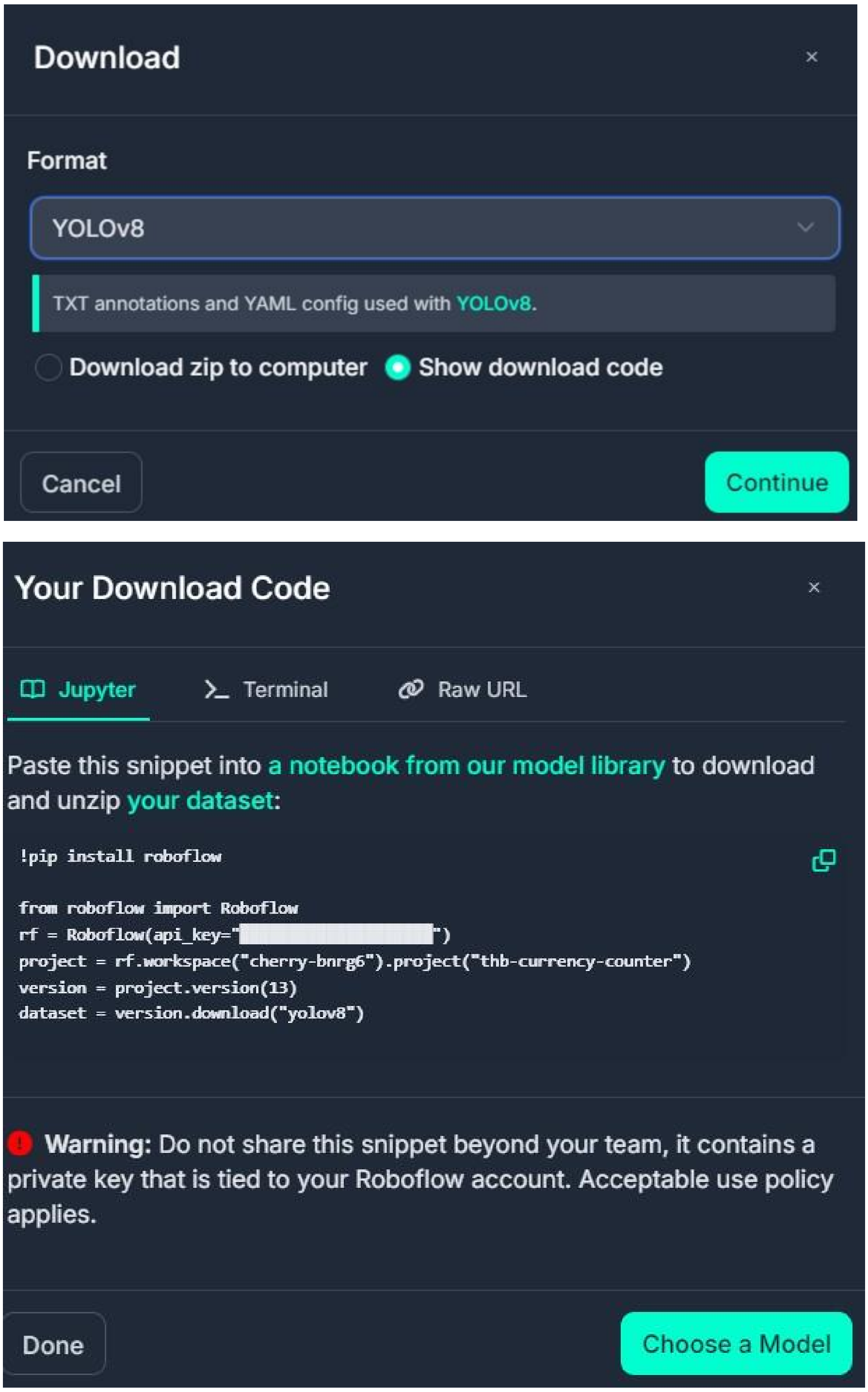


```
!git clone https://github.com/ultralytics/ultralytics
!pip install -qe ultralytics
%cd ultralytics
!pip install -U ultralytics
!pip install roboflow
```

Figure 4.4 Setting up YOLOv8 model.

4.7.2 Dataset Acquisition

The dataset, consisting of Malaysian banknotes and coins across multiple series, was obtained using the Roboflow platform. The Roboflow API was utilized to seamlessly download the dataset, ensuring that it was correctly formatted for YOLOv8 training. The following Python code facilitated this process.



Download

Format

YOLOv8

TXT annotations and YAML config used with YOLOv8.

○ Download zip to computer ● Show download code

Your Download Code

Paste this snippet into a notebook from our model library to download and unzip your dataset:

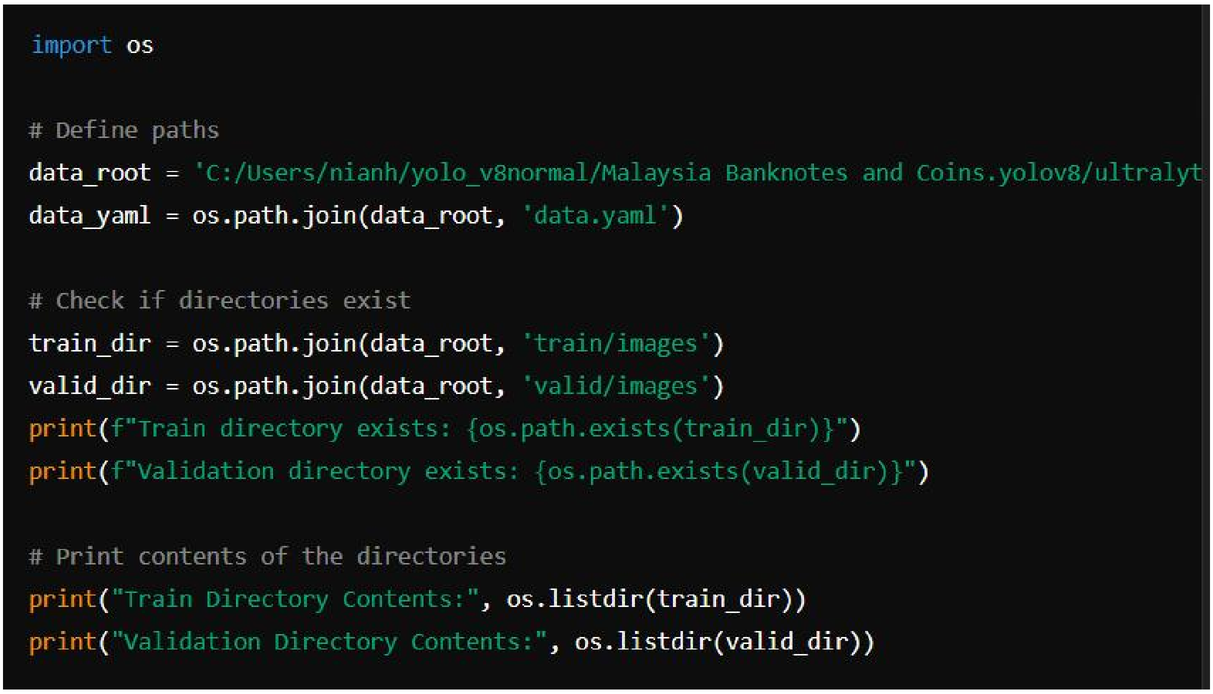
```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="[redacted]")
project = rf.workspace("cherry-bnrg6").project("thb-currency-counter")
version = project.version(13)
dataset = version.download("yolov8")
```

Warning: Do not share this snippet beyond your team, it contains a private key that is tied to your Roboflow account. Acceptable use policy applies.

Figure 4.5 Dataset Acquisition from RoboFlow.

4.7.3 Data Preparation and Directory Validation

To prepare the data for training, the directories containing training and validation images were verified to ensure proper organization and availability. The directory paths were checked, and their contents were listed to confirm that the images were correctly placed.

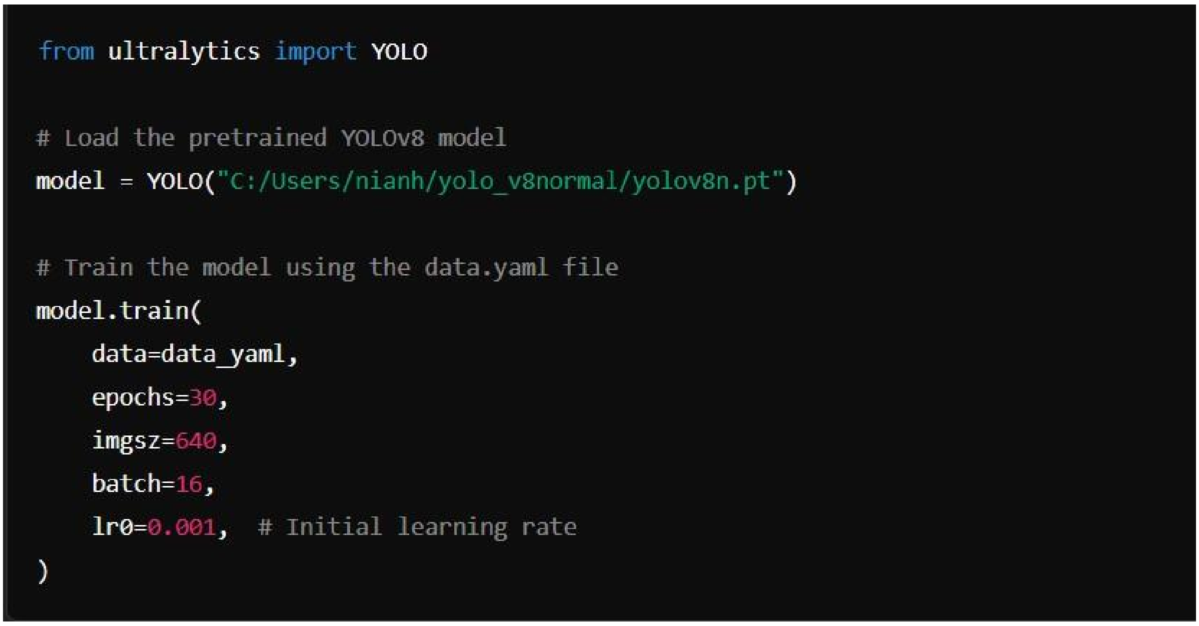


```
import os

# Define paths
data_root = 'C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and Coins.yolov8/ultralyt
data_yaml = os.path.join(data_root, 'data.yaml')

# Check if directories exist
train_dir = os.path.join(data_root, 'train/images')
valid_dir = os.path.join(data_root, 'valid/images')
print(f"Train directory exists: {os.path.exists(train_dir)}")
print(f"Validation directory exists: {os.path.exists(valid_dir)}")

# Print contents of the directories
print("Train Directory Contents:", os.listdir(train_dir))
print("Validation Directory Contents:", os.listdir(valid_dir))
```

Figure 4.6 Data Preparation and Validation.

4.7.4 Model Initialization and Training Configuration

The YOLOv8 model was initialized using pre-trained weights (yolov8n.pt) to leverage the knowledge acquired from a large-scale dataset. This transfer learning approach significantly reduced the training time and improved the model's initial accuracy. The training process was configured with a batch size of 16, an image size of 640x640 pixels, and an initial learning rate of 0.001. The model was trained for 30 epochs, balancing the need for thorough learning with computational efficiency. The training process was initiated using the following commands:



```
from ultralytics import YOLO

# Load the pretrained YOLOv8 model
model = YOLO("C:/Users/nianh/yolo_v8normal/yolov8n.pt")

# Train the model using the data.yaml file
model.train(
    data=data_yaml,
    epochs=30,
    imgsz=640,
    batch=16,
    lr0=0.001,  # Initial learning rate
)
```

Figure 4.7 Configuration to Train YOLOv8 model.

4.8 Prediction Using YOLOv8 Model

After the YOLOv8 model has been successfully trained, the next crucial step involves evaluating its performance on unseen data. This process, known as inference or prediction, tests the model's ability to generalize and accurately identify currency denominations in images that were not part of the training set. The following section details the prediction procedure employed for the Cash Reader system using the YOLOv8 model.

4.8.1 Prediction Setup

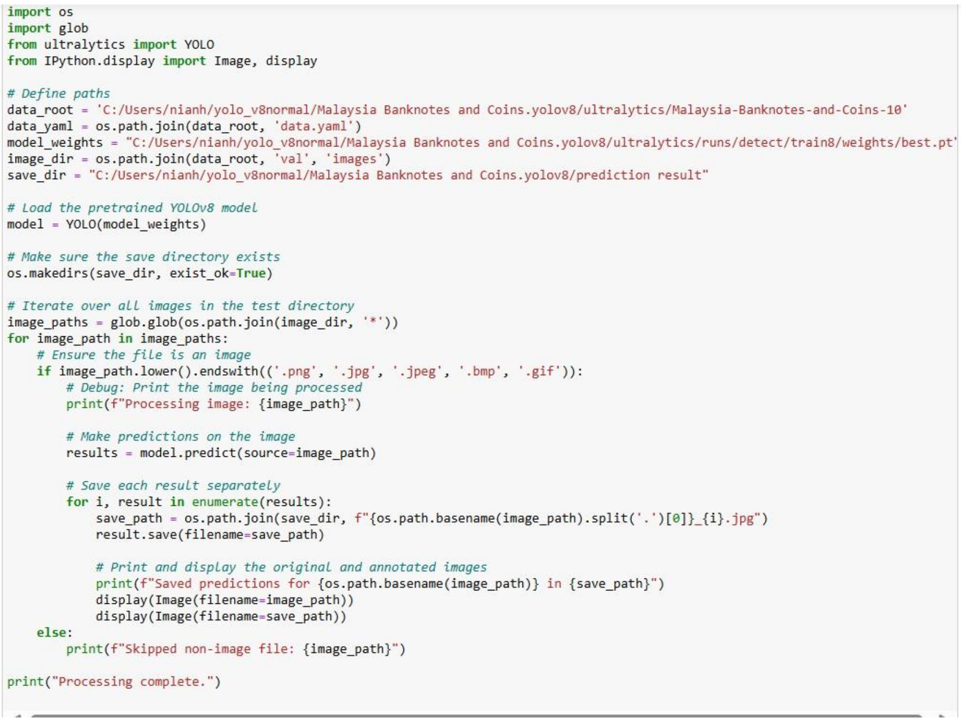
The prediction phase was conducted using the validation dataset. The model used for this purpose was the best-performing model obtained from the training process, denoted as best.pt. This model was loaded into the system, and predictions were made on images contained within the validation directory. The directory paths were carefully configured to ensure seamless access to both the model and the validation images.

The prediction code was executed in a controlled environment, ensuring that each image from the validation set was processed by the model. The results were then saved in a designated directory, ensuring that both the original images and their corresponding annotated predictions were preserved for subsequent analysis.

4.8.2 Prediction Workflow

The prediction workflow is outlined as follows:

1. **Loading the Pretrained Model**: The YOLO class from the ultralytics package was utilized to load the best.pt model. This model had been previously trained and validated on the same dataset and was now being used to predict the classes of objects in new images.
2. **Image Processing and Prediction**: Each image in the validation directory was processed individually. The model performed inference on the image, generating bounding boxes and class labels for detected objects. These predictions were stored and displayed for visual inspection.
3. **Result Storage**: For each image processed, the prediction results, including the annotated images with bounding boxes and labels, were saved in a specified directory. This directory was organized to contain both the original image and the corresponding prediction output, facilitating easy comparison and further evaluation.
4. **Visualization**: To ensure the correctness of the predictions, the images were displayed alongside their corresponding predictions. This step provided an immediate visual check of the model's performance, highlighting its accuracy in identifying and labeling currency denominations.



```
import os
import glob
from ultralytics import YOLO
from IPython.display import Image, display

# Define paths
data_root = 'C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and Coins.yolov8/ultralytics/Malaysia-Banknotes-and-Coins-10'
data_yaml = os.path.join(data_root, 'data.yaml')
model_weights = "C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and Coins.yolov8/ultralytics/runs/detect/train8/weights/best.pt'
image_dir = os.path.join(data_root, 'val', 'images')
save_dir = "C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and Coins.yolov8/prediction result"

# Load the pretrained YOLOv8 model
model = YOLO(model_weights)

# Make sure the save directory exists
os.makedirs(save_dir, exist_ok=True)

# Iterate over all images in the test directory
image_paths = glob.glob(os.path.join(image_dir, '*'))
for image_path in image_paths:
    # Ensure the file is an image
    if image_path.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.gif')):
        # Debug: Print the image being processed
        print(f"Processing image: {image_path}")

        # Make predictions on the image
        results = model.predict(source=image_path)

        # Save each result separately
        for i, result in enumerate(results):
            save_path = os.path.join(save_dir, f"{os.path.basename(image_path).split('.')[0]}_{i}.jpg")
            result.save(filename=save_path)

            # Print and display the original and annotated images
            print(f"Saved predictions for {os.path.basename(image_path)} in {save_path}")
            display(Image(filename=image_path))
            display(Image(filename=save_path))
    else:
        print(f"Skipped non-image file: {image_path}")

print("Processing complete.")
```

Figure 4.8 Prediction script.

4.9 Live Inference using OpenCV

In this section, we describe the implementation of a live inference system for the Cash Reader using the YOLOv8 model and OpenCV. The system is designed to process live video feeds from a webcam, detect and recognize currency denominations, and provide real-time audio feedback for the visually impaired.

Implementation Details

The live inference system was developed using Python, leveraging the YOLOv8 model for object detection and OpenCV for capturing and processing video frames. The implementation also includes the pyttsx3 library to enable text-to-speech functionality, allowing the system to audibly announce the recognized currency denomination when the detection confidence surpasses a predefined threshold.

The following steps outline the core functionality of the live inference system:

1. **Model Initialization**: The YOLOv8 model is initialized using pre-trained weights that were fine-tuned for the currency recognition task. The model is loaded from the specified path, ensuring that it is ready for real-time inference.
2. **Text-to-Speech Engine Setup**: A text-to-speech engine is initialized using pyttsx3. The engine's properties, such as speech rate and volume, are configured to ensure clear and audible announcements.
3. **Video Capture**: The system captures live video feed from an external webcam. The appropriate camera index is selected based on the specific hardware configuration. If the webcam cannot be accessed, the system will output an error message and terminate.
4. **Object Detection and Audio Output**: For each frame captured from the webcam, the YOLOv8 model performs object detection. The system iterates through the detected objects, drawing bounding boxes and labels around recognized currency denominations. If the detection confidence exceeds the threshold of 75%, the system triggers an audio output that announces the name of the recognized currency denomination.
5. **User Interface**: The processed video frames, with annotated bounding boxes and labels, are displayed in real-time. The user can terminate the live inference by pressing the ESC key.

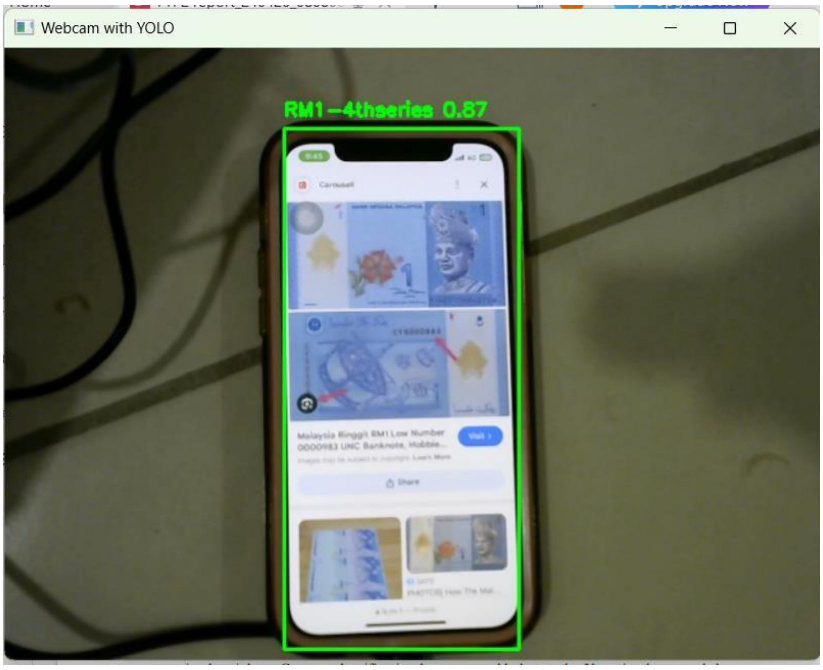


Figure 4.9 Live inference using OpenCV and YOLOv8 model output.

4.10 Implementation Issues and Challenges

One of the notable challenges encountered during the implementation of the Cash Reader system is the detection of small objects, such as coins. YOLOv8, despite its robust performance in various object detection tasks, has demonstrated certain limitations when dealing with small objects. This is particularly relevant in cases where precise detection of small currency denominations, like coins, is crucial. Evidence suggests that object detection models, including YOLOv8, often struggle with small objects due to factors like reduced spatial resolution in deeper layers of the network and the limited number of pixels representing the object in the image. This challenge is well-documented in the literature, where smaller objects are frequently harder to detect and are more prone to being overlooked or inaccurately classified.

Another significant limitation of the Cash Reader system is its inability to detect and differentiate fake or counterfeit currency. The system is designed to recognize and classify legitimate currency denominations based on visual features, but it lacks the capability to verify the authenticity of the money. This poses a challenge in real-world applications where the identification of counterfeit currency is critical. The current model does not incorporate security features such as watermarks, holograms, or other anti-counterfeiting measures that are typically used to distinguish real currency from counterfeit. This limitation highlights the need for additional measures or integrated systems that can handle counterfeit detection, ensuring that the Cash Reader system can be reliably used in all financial transactions.

A third challenge arises with large-volume datasets, such as those involving big data. In the case of manual annotation and labeling with tools like Roboflow, the process can become extremely time-consuming and labor-intensive. For large datasets, the manual effort required to annotate and label each image accurately can significantly delay the project and increase the potential for human error. This challenge is compounded by the need to maintain consistency and accuracy across a vast number of samples.

To overcome this issue, one effective approach is to create and utilize automated scripts for labeling and annotation. Automated data labeling tools can help speed up the process by using algorithms to generate annotations based on predefined criteria or by leveraging semi-supervised learning techniques. Additionally, integrating advanced annotation tools or frameworks that support batch processing and automated labeling can reduce the manual workload and improve efficiency. By implementing these solutions, the time required for data preparation can be significantly reduced, allowing for more rapid development and deployment of the Cash Reader system.

4.11 Concluding Remark

In conclusion, the development and deployment of the Cash Reader system highlight several critical aspects of implementing advanced object detection technologies in real-world applications. The use of YOLOv8 has demonstrated its effectiveness in recognizing various currency denominations, but challenges persist, particularly in detecting small objects like coins, distinguishing counterfeit currency, and managing large datasets. These challenges underscore the need for ongoing research and development to enhance detection accuracy, integrate counterfeit detection capabilities, and streamline data preparation processes. Future work should focus on addressing these limitations by leveraging automated data annotation tools, incorporating additional security features, and refining model architectures to improve performance across all aspects of currency recognition. By addressing these issues, the Cash Reader system can achieve greater accuracy, reliability, and practical utility for visually impaired users in diverse financial transactions.

Chapter 5

System Evaluation And Discussion

5.1 Classification Models Performance Evaluation and Discussion

This section compares the performance of 5 different currency performance using YOLOv8 model. The dataset contains currency from Malaysian Ringgit (MYR), United States Dollar (USD), Euro (EURO), Singapore Dollar (SGD), and Thai Baht (Baht). The dataset comprises 1,693 images of MYR, 2,519 images of USD, 2,949 images of EURO, 2,035 images of SGD, and 7,285 images of Baht.

5.1.1 MYR performance on YOLOv8 model

The graphs presented in Figure 5.1 illustrate the training and validation performance of the model across key metrics, including loss functions and object detection evaluation measures. The analysis is divided into two sections: loss functions and evaluation metrics.

Loss Function Evaluation

The loss functions monitored during training include the **box loss**, **classification loss**, and **distribution focal loss (DFL)** for both training and validation datasets.

- **Training Loss**:
    - The **train/box_loss** graph shows a steady decrease from approximately 0.40 to below 0.25 over the course of the training epochs, indicating that the model is learning to better localize objects within bounding boxes.
    - The **train/cls_loss** follows a similar decreasing trend, starting at around 3.0 and dropping to below 1.0, reflecting the model's improved ability to classify objects accurately.
    - **train/dfl_loss** also exhibits a consistent downward trend, reducing from an initial value of approximately 0.94 to 0.84, signifying enhanced precision in bounding box predictions.
- **Validation Loss**:
    - The **val/box_loss** decreases from 0.625 to around 0.50, albeit with more fluctuations than the training loss, which is expected due to the variable nature of the validation set.

  - Similarly, **val/cls_loss** decreases from 2.2 to approximately 1.2, with periodic spikes that indicate challenges in generalizing over the validation dataset.
  - **val/dfl_loss** fluctuates between 0.98 and 0.88, following a pattern similar to the training loss but with more variance, indicating that the model maintains stability while optimizing bounding box quality on unseen data.

Overall, the alignment between training and validation losses shows minimal signs of overfitting, as both losses decrease steadily over time.

Evaluation Metrics

The model's performance is further evaluated using key object detection metrics, including **precision**, **recall**, and **mean Average Precision (mAP)** at both IoU thresholds of 0.50 (mAP50) and 0.50-0.95 (mAP50-95).

- **Precision**: The **metrics/precision(B)** graph shows an upward trend, though with significant fluctuation. Precision improves from 0.50 to approximately 0.65 by the final epoch, indicating that the model is becoming increasingly accurate in its predictions with fewer false positives.
- **Recall**: The **metrics/recall(B)** curve also demonstrates a positive trend, increasing from around 0.50 to 0.75. This shows the model's growing ability to detect a higher number of true positives, though the fluctuations suggest some inconsistency in recall during certain epochs.
- **mAP50**: The **metrics/mAP50(B)**, which measures the average precision at IoU threshold 0.50, shows a steady rise from 0.60 to 0.85, confirming that the model is effectively learning to detect and classify objects over time.
- **mAP50-95**: The **metrics/mAP50-95(B)** graph shows a gradual increase, reaching around 0.60 at the final epoch. This metric, which accounts for a range of IoU thresholds, highlights the model's robustness across different levels of intersection-over-union (IoU), providing a comprehensive view of performance.

Summary

The evaluation metrics, including precision, recall, and mAP, all exhibit upward trends, reflecting the model's improving accuracy and generalization ability. While the validation losses show slightly more fluctuation than the training losses, this is typical in machine learning models and suggests that the model is not significantly overfitting.

In conclusion, the results indicate that the model has successfully optimized its loss functions while achieving strong precision, recall, and mAP scores. The increasing trends across all metrics demonstrate the model's capability to generalize well to unseen data, making it suitable for deployment in real-world object detection tasks.

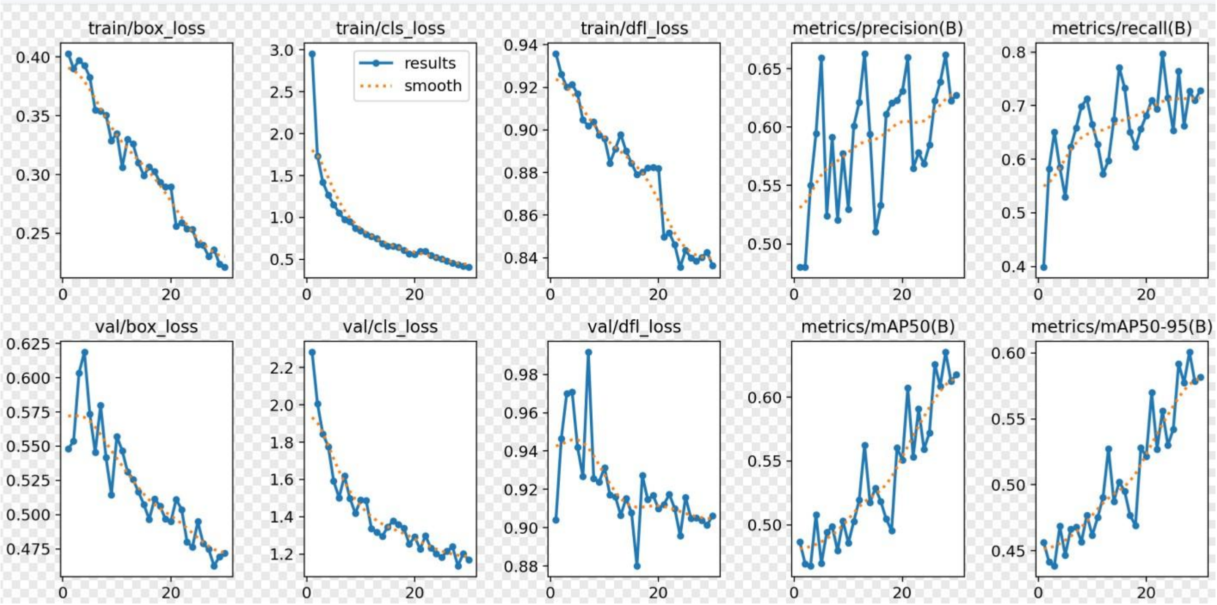


Figure 5.1 Evaluation Metrics for MYR.

The confusion matrix presented in Figure 5.2 provides a comprehensive view of the classification performance across different denominations and series of Malaysian currency. Each row represents the predicted label, while each column represents the true label, with the intensity of color in each cell representing the number of predictions made for that class. The evaluation focuses on interpreting the model's performance across different currency denominations and highlighting any key misclassifications.

Analysis of True Positives

Several classes demonstrate strong true positive rates, indicated by high diagonal values in the matrix, such as:

- **RM50-4thseries**: The model correctly classified this denomination 20 times, with no significant misclassifications into other categories.
- **RM5-4thseries**: The true positive count stands at 14, showing good performance in identifying this denomination.
- **RM100-4thseries**: The model classified this denomination correctly 12 times, with no major confusion with other classes.

This performance suggests that the model is highly effective at distinguishing higher denominations and more recent currency series.

Analysis of Misclassifications

The matrix also highlights several key areas where misclassifications occurred:

- **10sen-1stseries vs 10sen-2ndseries**: A substantial portion of the predictions for **10sen-1stseries** were misclassified as **10sen-2ndseries** (30 instances), while **10sen-3rdseries** was similarly confused with **10sen-1stseries** (29 instances). This indicates challenges in differentiating between the different series of smaller denominations.
- **50sen-3rdseries**: The model misclassified 27 instances of **50sen-3rdseries** as **50sen-2ndseries**, pointing to a difficulty in distinguishing these two closely related denominations.
- **5sen-2ndseries and 5sen-3rdseries**: There was significant confusion between these two classes, with **5sen-2ndseries** being frequently misclassified as **5sen-1stseries** (13 instances) and **5sen-3rdseries** being misclassified as **5sen-2ndseries** (17 instances). This highlights the model's struggles with very small coin denominations across different series.

Background Detection

The **background** class shows high precision, with 75 correct classifications, meaning the model successfully differentiates between currency and non-currency objects in most cases. There is, however, one misclassification where an instance of **50sen-3rdseries** was identified as background, which suggests room for further improvement in identifying all coin types, especially when backgrounds are complex or cluttered.

Summary

The confusion matrix provides valuable insight into the performance of the currency recognition model. While the model performs well in distinguishing larger denominations and newer series, it encounters difficulties with smaller denominations, particularly across different series. Improving the model's ability to differentiate between visually similar coin types, such as **5sen** and **10sen** across various series, is a key area for enhancement. Furthermore, while background detection is largely successful, some confusion remains between smaller coins and background elements, requiring more robust training on various environmental factors.

This detailed evaluation provides a clear understanding of the model's strengths and areas for improvement, critical for refining the currency recognition system's real-world application.

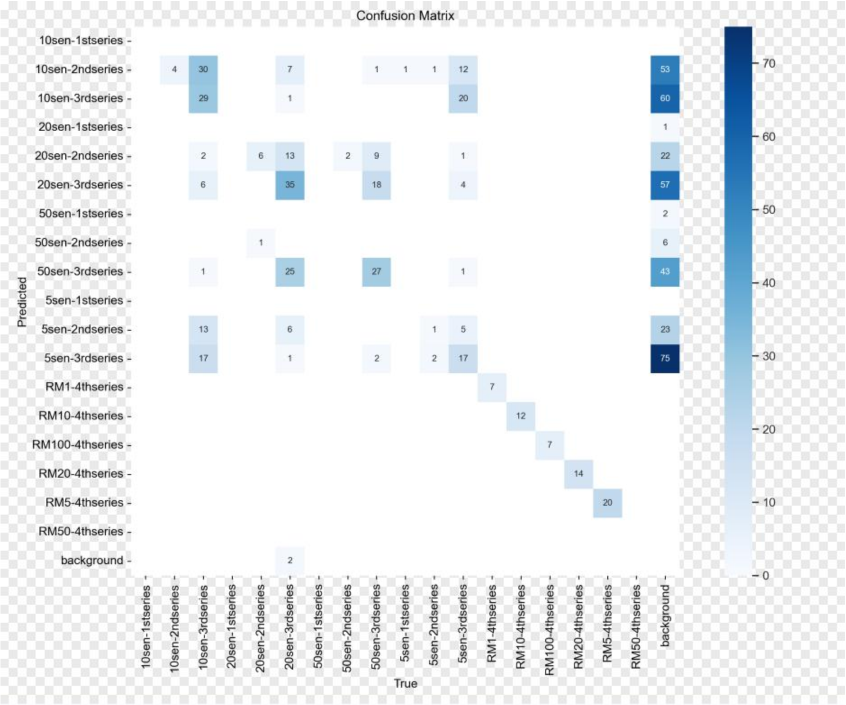


Figure 5.2 Confusion Matrix for MYR.

Based on Figure 5.3, the performance of the YOLOv8 model in detecting Malaysian currency denominations presents notable strengths and challenges. The overall precision (P) of the model stands at 0.662, with a recall (R) of 0.727. The mean average precision at an IoU threshold of 0.5 (mAP@0.5) is 0.634, while the mAP@0.5-0.95 metric, which accounts for varying IoU thresholds, is recorded at 0.600.

Strengths

The model demonstrates excellent performance in recognizing higher-value banknotes from the 4th series. Denominations such as RM1, RM5, RM10, RM20, and RM100 exhibit near-perfect precision and recall values (P ≈ 0.98-0.99, R = 1.00), with corresponding mAP@0.5 values reaching 0.995. These results indicate the model's high efficacy in detecting and classifying these larger denomination notes, suggesting robust generalization across various environmental conditions and angles.

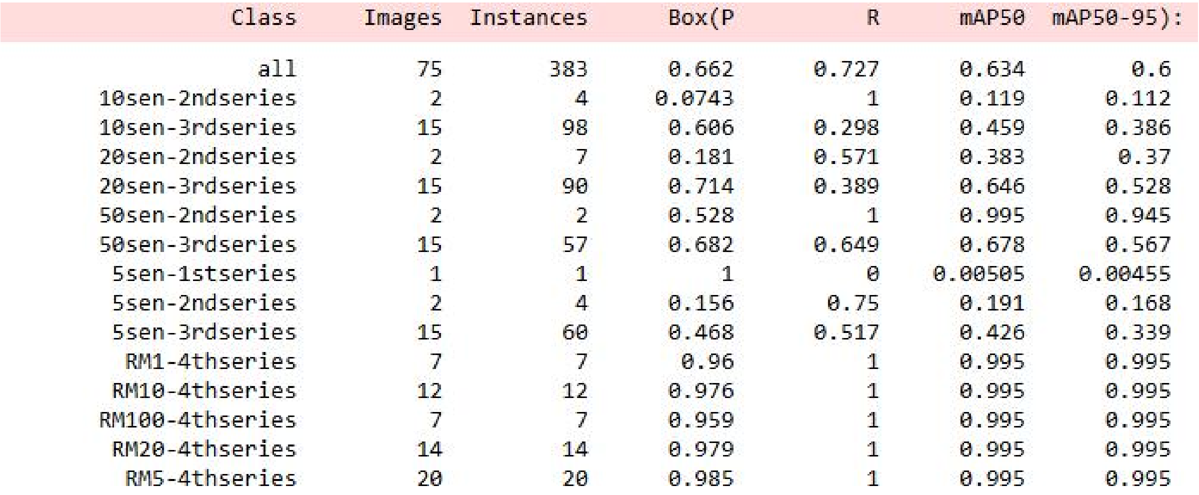
In addition, the 50sen coins from the 2nd and 3rd series show strong detection performance, with precision values ranging from 0.528 to 0.682 and recall values between 0.649 and 1.0.

The high mAP@0.5 values for these coins indicate the model's ability to accurately detect these denominations with consistent performance.

Weaknesses

Conversely, the model's performance is considerably lower for smaller coin denominations. For instance, the 10sen-2ndseries denomination records a low precision (0.0743) despite achieving perfect recall (R = 1.00), indicating a high rate of false positives. Similarly, the 5sen-1stseries achieves perfect precision (P = 1.00) but fails in recall (R = 0.00), indicating that it cannot consistently detect this denomination. These results highlight the challenge of distinguishing smaller objects or objects with subtle differences in features.

Furthermore, variability is observed in the model's ability to detect 10sen and 20sen denominations across different series. For the 10sen-3rdseries and 20sen-3rdseries, recall values are lower, particularly for the 10sen-3rdseries (R = 0.298). The lower precision and recall values for these classes are reflected in their corresponding mAP@0.5-0.95 values, suggesting difficulty in detecting these coins due to either similarity across series or inadequate representation in the training set.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all         75        383      0.662      0.727      0.634        0.6
       10sen-2ndseries          2          4     0.0743          1      0.119      0.112
       10sen-3rdseries         15         98      0.606      0.298      0.459      0.386
       20sen-2ndseries          2          7      0.181      0.571      0.383       0.37
       20sen-3rdseries         15         90      0.714      0.389      0.646      0.528
       50sen-2ndseries          2          2      0.528          1      0.995      0.945
       50sen-3rdseries         15         57      0.682      0.649      0.678      0.567
        5sen-1stseries          1          1          1          0    0.00505    0.00455
        5sen-2ndseries          2          4      0.156       0.75      0.191      0.168
        5sen-3rdseries         15         60      0.468      0.517      0.426      0.339
         RM1-4thseries          7          7       0.96          1      0.995      0.995
        RM10-4thseries         12         12      0.976          1      0.995      0.995
       RM100-4thseries          7          7      0.959          1      0.995      0.995
        RM20-4thseries         14         14      0.979          1      0.995      0.995
         RM5-4thseries         20         20      0.985          1      0.995      0.995
```

Figure 5.3 Accuracy of MYR using YOLOv8 Model.

As illustrated in **Figure 5.4**, the Cash Reader system demonstrates exceptional performance during validation tests, even in challenging scenarios where multiple coins of different denominations appear within the same frame. Despite the complexity of distinguishing between closely similar denominations in such situations, the system accurately detects and classifies the coins into their respective categories.

Detection of Multiple Coins in a Single Frame

A key strength of the Cash Reader system is its ability to handle multiple objects within a single frame. In instances where several coins are present, including various denominations like **10sen**, **20sen**, and **50sen**, the system effectively identifies and assigns them to their correct classes. This indicates the system's robustness and capacity to operate efficiently in real-world conditions where such overlaps are frequent. The strong recall values observed for some classes, such as **50sen-2ndseries** and **RM1-4thseries**, underscore its reliability in detecting and classifying coins in cluttered or complex environments, an important aspect of object detection for assistive technologies.

Handling of Different Denominations in a Single Frame

In situations where different denominations appear together, the Cash Reader system maintains high performance by correctly distinguishing between coins of similar size and color. For example, the system accurately differentiates between coins such as the **50sen-2ndseries** and **50sen-3rdseries**, despite their visual similarities. This demonstrates the system’s ability to generalize effectively, ensuring accurate classification even when handling a mixture of denominations.

Implications for Real-World Applications

The accurate detection and classification of multiple coins in a single frame are particularly beneficial for real-world applications, such as assisting visually impaired individuals. The ability of the Cash Reader system to recognize and classify coins reliably, even when multiple denominations are present, enhances its practical utility. By offering such precision and reliability, the system significantly reduces the likelihood of misidentification, thereby providing users with greater confidence and independence during financial transactions.

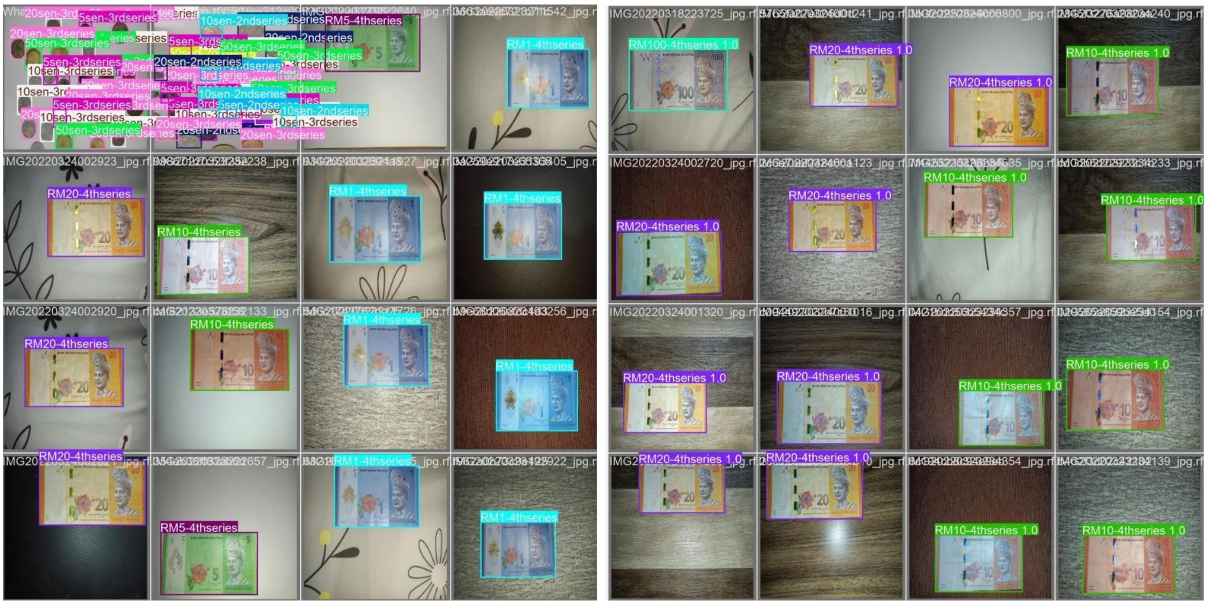


Figure 5.4 Validation Output for MYR.

5.1.2 SGD performance on YOLOv8 model

The graphs provide a comprehensive evaluation of the Cash Reader system trained using the YOLOv8 model with the Stochastic Gradient Descent (SGD) optimizer. Each subgraph provides insights into different aspects of the training and validation process:

Training Loss (Box, Classification, DFL)

- **Train/Box Loss**: The box loss represents the error in bounding box prediction. The continuous decline from an initial value above 0.8 to a value around 0.3 indicates that the model is progressively improving its bounding box predictions as training advances. The smooth curve (orange line) suggests that the training is stable without major fluctuations.
- **Train/Cls Loss**: The classification loss decreases consistently from over 2.5 to around 0.5, signifying the model's enhanced ability to classify objects correctly across the training epochs.
- **Train/DFL Loss**: The distributional focal loss (DFL) starts at 1.4 and drops to around 1.1. A decreasing trend indicates improvement in the accuracy of object localization.

Validation Loss (Box, Classification, DFL)

- **Val/Box Loss**: Similar to the training box loss, the validation box loss decreases from 1.8 to 1.0, indicating improvement in the model’s bounding box prediction on unseen data (validation set). However, the validation loss has more fluctuations, indicating that the model may still face some variance with new data.
- **Val/Cls Loss**: The validation classification loss decreases from approximately 3.5 to 1.0, indicating that the model is better classifying unseen currency images, although the higher initial classification loss reflects more variability at the beginning of the training.
- **Val/DFL Loss**: The validation DFL loss shows a clear downward trend, though with some fluctuations, decreasing from around 2.4 to 1.6. This suggests the model is improving in localizing objects in validation data, but there's still some room for stability.

Metrics (Precision, Recall, mAP50, mAP50-95)

- **Precision (B)**: The precision increases significantly as the training progresses, reaching approximately 0.9, which indicates that the model correctly identifies a high proportion of true positives, minimizing false positives.
- **Recall (B)**: The recall also improves throughout training, reaching around 0.95, meaning the model is correctly identifying almost all relevant objects (high sensitivity) and minimizing false negatives.
- **mAP50 (B)**: The mean Average Precision at an IoU threshold of 0.50 (mAP50) shows a consistent rise, stabilizing around 0.8, indicating strong object detection performance.
- **mAP50-95 (B)**: The mean Average Precision across IoU thresholds from 0.50 to 0.95 shows a steady improvement as well, eventually reaching a value of around 0.7, which suggests the model is performing well across various levels of intersection over union (IoU) thresholds, not just at 0.50.

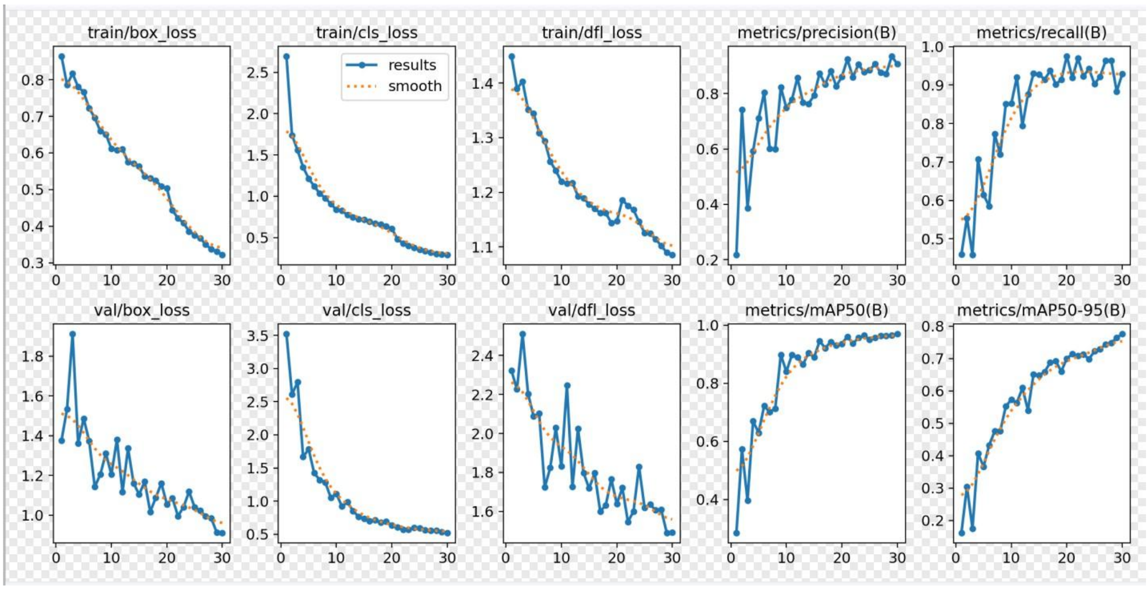


Figure 5.5 Evaluation Metrics for SGD.

The confusion matrix presented in Figure 5.6 provides a detailed overview of the model’s classification performance across various denominations. The rows represent the predicted labels, while the columns represent the true labels, with the color intensity in each cell indicating the number of predictions made for each class. This analysis focuses on the model's performance across different currency denominations, particularly highlighting accurate classifications (true positives) and key misclassifications.

True Positive Analysis

Several currency denominations exhibit strong true positive rates, as indicated by the high values along the diagonal of the confusion matrix:

- **Fifty-dollar**: The model correctly classified 42 instances of fifty-dollar banknotes, showing strong performance in identifying this denomination with minimal confusion into other categories.
- **Five-dollar**: The model accurately predicted 44 instances of five-dollar banknotes, demonstrating good reliability in identifying this denomination.
- **Hundred-dollar**: A total of 34 banknotes were correctly classified as hundred-dollar notes, again suggesting the model's effectiveness in distinguishing this currency denomination.
- **Ten-dollar**: The model identified 39 ten-dollar banknotes correctly, which is another strong indication of reliable classification for this category.
- **Two-dollar**: 6 two-dollar notes were correctly classified, though there were noticeable misclassifications with other denominations.

This performance suggests that the model is highly effective at distinguishing larger denominations, which are often more visually distinct, from other currencies and background elements.

Misclassification Analysis

While the model generally performs well, several misclassifications were observed, particularly in distinguishing smaller denominations and some visually similar currency classes:

- **Ten-dollar vs Two-dollar**: The model confused ten-dollar banknotes with two-dollar banknotes 11 times. This indicates some difficulty in differentiating between these two denominations.
- **Background Misclassification**: A small number of background elements were mistakenly classified as currency notes. Notably, there were 2 instances where the background was classified as a fifty-dollar or five-dollar banknote, highlighting a minor issue with background detection.
- **Two-dollar misclassified as Ten-dollar**: The two-dollar class was incorrectly predicted as ten-dollar 9 times. This highlights a key challenge in differentiating between these lower denominations.

These misclassifications suggest that the model struggles with lower denominations, which might have similar physical characteristics, leading to incorrect predictions.

Background Detection

The model's ability to distinguish between background and actual currency notes also shows some mixed results:

- The background was generally well classified, with only 1 misclassification where a background image was incorrectly identified as a ten-dollar banknote.
- However, 2 instances of actual currency (fifty-dollar and five-dollar) were misclassified as background, indicating that the model occasionally fails to differentiate between currency and non-currency objects in complex visual scenes.

Summary

The confusion matrix provides valuable insights into the model's strengths and areas for improvement:

- **Strengths**: The model performs well in distinguishing larger and more visually distinct denominations such as fifty-dollar, five-dollar, and ten-dollar notes. These strong true positive rates are a positive indicator of the model's robustness in real-world applications, particularly when recognizing larger currency.
- **Challenges**: Misclassifications occur predominantly in distinguishing between visually similar and smaller denominations, such as ten-dollar and two-dollar notes. Additionally, minor background misclassifications indicate a need for further refinement in object detection, especially in complex environments.

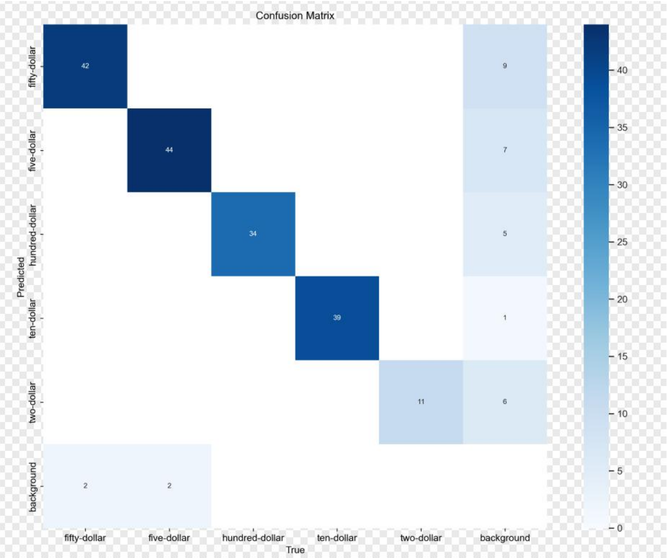


Figure 5.6 Confusion Matrix for SGD.

The Figure 5.7 presented provides a breakdown of the model's performance across five different currency classes: *fifty-dollar, five-dollar, hundred-dollar, ten-dollar,* and *two-dollar*. The metrics used for evaluation are *Precision (P)*, *Recall (R)*, *mean Average Precision (mAP50)*, and *mAP across IoU thresholds (mAP50-95)*. These metrics are essential in assessing the detection accuracy and overall effectiveness of the YOLOv8-based model in real-world scenarios.

Precision (P)

Precision refers to the proportion of correct positive identifications out of all positive identifications made by the model.

- **Overall Precision**: The model achieved an impressive overall precision of **0.905**, indicating that about 90.5% of the model's predictions across all classes were correct.
- **Per-Class Precision**:
  - *Fifty-dollar*: Precision for this class was **0.833**, showing moderate performance. This indicates room for improvement in reducing false positives for fifty-dollar banknotes.
  - *Five-dollar*: Precision was **0.925**, suggesting strong performance with very few incorrect detections.
  - *Hundred-dollar*: Achieved perfect precision (**1.0**), indicating that all predicted hundred-dollar banknotes were correct, with no false positives.
  - *Ten-dollar*: Precision for this class was **0.984**, showing excellent performance.
  - *Two-dollar*: Precision was **0.781**, the lowest among the classes, suggesting the model struggles more with distinguishing two-dollar banknotes, possibly due to their visual similarity with other denominations.

Recall (R)

Recall measures the proportion of actual positive instances that the model correctly identified.

- **Overall Recall**: The overall recall was **0.931**, meaning the model correctly identified 93.1% of all true instances across the dataset.
- **Per-Class Recall**:

- *Fifty-dollar*: Recall was **0.793**, indicating the model identified 79.3% of all true fifty-dollar notes, leaving some under-detection.
- *Five-dollar*: An outstanding recall of **0.957**, meaning the model detected nearly all five-dollar notes.
- *Hundred-dollar*: Perfect recall of **1.0**, with the model identifying every single hundred-dollar note correctly.
- *Ten-dollar*: Another perfect recall of **1.0**, demonstrating the model's effectiveness in detecting all ten-dollar notes.
- *Two-dollar*: Recall was **0.909**, which is strong but leaves room for improvement, as some two-dollar notes were missed.

Mean Average Precision (mAP50)

mAP at IoU=0.5 (mAP50) measures how well the model balances precision and recall at a specific IoU threshold of 0.5, which is commonly used to evaluate object detection models.

- **Overall mAP50**: The overall mAP50 for the model was **0.97**, which indicates that the model's predictions are highly accurate, with excellent performance in detecting objects.
- **Per-Class mAP50**:
    - *Fifty-dollar*: **0.988**, a very high mAP, despite lower precision and recall scores, indicating that when the model is correct, it's highly confident about its predictions.
    - *Five-dollar*: **0.981**, also showing excellent performance, reflecting the high precision and recall values.
    - *Hundred-dollar*: **0.995**, reinforcing that the model is highly effective in detecting hundred-dollar banknotes.
    - *Ten-dollar*: **0.995**, showing that this class is also well-handled by the model.
    - *Two-dollar*: **0.971**, a strong mAP, despite challenges in precision, suggesting that the model's confidence improves when balancing precision and recall.

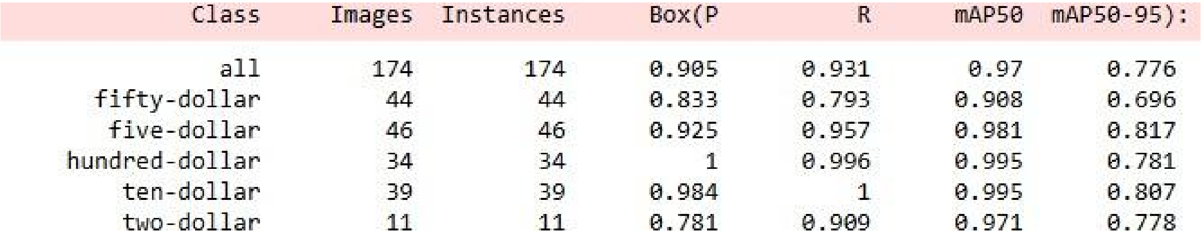
Mean Average Precision Across IoU Thresholds (mAP50-95)

mAP50-95 is a more stringent metric that measures the model's performance across multiple IoU thresholds (ranging from 0.5 to 0.95).

- **Overall mAP50-95**: The overall score was **0.776**, indicating solid performance, though slightly reduced compared to mAP50 due to the more demanding criteria of this metric.
- **Per-Class mAP50-95**:
  - *Fifty-dollar*: **0.696**, the lowest in this category, further reflecting that this class could benefit from additional training or more diverse data.
  - *Five-dollar*: **0.817**, showing strong but slightly reduced performance under stricter IoU thresholds.
  - *Hundred-dollar*: **0.781**, still solid but lower than the perfect performance at mAP50, likely due to the more stringent overlap requirements.
  - *Ten-dollar*: **0.807**, indicating good performance in higher IoU thresholds.
  - *Two-dollar*: **0.778**, reflecting the lower performance on more challenging cases, though still relatively strong.

Summary and Recommendations

- **Strengths**:
  - The model shows excellent performance across larger denominations, such as *hundred-dollar* and *ten-dollar*, with perfect recall and near-perfect precision, mAP50, and strong mAP50-95 values.
  - The overall high scores for precision (0.905), recall (0.931), and mAP50 (0.97) indicate that the model is generally very reliable in detecting and classifying currency denominations, especially in larger denominations.
- **Challenges**:
  - The model's performance with the *two-dollar* and *fifty-dollar* banknotes is slightly weaker, particularly in precision (0.781 for two-dollar and 0.833 for fifty-dollar), indicating a higher rate of false positives for these classes.
  - The mAP50-95 for the *fifty-dollar* class is the lowest (0.696), which suggests that the model struggles with stricter overlap requirements for this denomination.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        174        174      0.905      0.931       0.97      0.776
          fifty-dollar         44         44      0.833      0.793      0.908      0.696
           five-dollar         46         46      0.925      0.957      0.981      0.817
        hundred-dollar         34         34          1      0.996      0.995      0.781
            ten-dollar         39         39      0.984          1      0.995      0.807
            two-dollar         11         11      0.781      0.909      0.971      0.778
```

Figure 5.7 Accuracy of SGD using YOLOv8 model.

In this set of images, various challenging scenarios were introduced to evaluate the robustness of the model in detecting Singaporean banknotes. These scenarios include obstructions like fingers covering parts of the banknotes, multiple banknotes in a single frame, the presence of coins alongside banknotes, and various distortions like sticky tape or drawings on the banknotes. Below is a detailed evaluation of the model's performance based on these scenarios:

Finger Obstruction on Banknotes

Several images in the dataset show portions of banknotes being covered by fingers, simulating a common real-world scenario where users may partially obscure the currency during detection.

- **Model Response**: The model was able to detect the banknotes despite the obstructions. In many cases, the bounding boxes are correctly placed around the visible parts of the notes, and the predictions maintain a high level of confidence (confidence scores displayed in the top-left corners of each bounding box).
- **Performance**: The model demonstrates resilience in identifying partially obscured banknotes, suggesting that it is trained well enough to infer the banknote's identity even when parts are covered.

Multiple Banknotes in a Single Frame

Some images feature multiple banknotes displayed together, which could potentially confuse the model due to overlapping or stacked currency.

- **Model Response**: The model successfully detected all the banknotes present in these scenarios, and the bounding boxes were correctly drawn around each banknote, including those that were stacked or overlapped.
- **Performance**: The predictions for each note maintained a high confidence level, indicating the model's strong capacity to handle multiple instances of the same or different denominations in a single frame.

Coins and Banknotes Together

Coins were introduced alongside banknotes in certain images to see if the model could distinguish between different types of currency.

- **Model Response**: Despite the presence of coins, the model maintained focus on the banknotes and accurately detected and classified them without being confused by the coins.

- **Performance**: The absence of coin-related false positives suggests that the model is specifically tuned to detect banknotes and ignores irrelevant objects like coins.

Overall Model Confidence and Performance

Across all images, the confidence scores associated with the predicted banknotes remain consistently high, indicating that the model is confident in its detections even in challenging conditions. The bounding boxes are well-positioned, and the predictions align accurately with the visual content.

Summary

The evaluation shows that the model effectively handles various complex real-world scenarios, such as obstructions, multiple objects, and altered currency, while maintaining high confidence and detection accuracy across all tests.

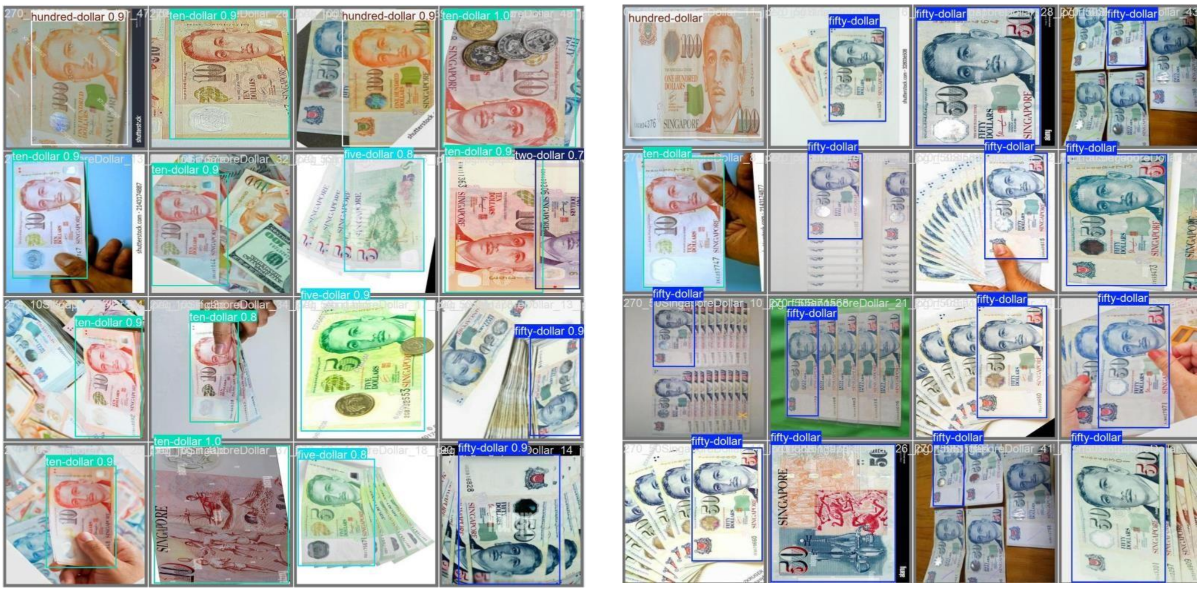


Figure 5.8 Validation Output of SGD.

5.1.3 USD performance on YOLOv8 model

The graphs presented in Figure 5.9 illustrate the training and validation performance of the model on USD currency detection across key metrics, including loss functions and object detection evaluation measures. The analysis is divided into two sections: loss functions and evaluation metrics.

Loss Function Evaluation

The loss functions monitored during training include the box loss, classification loss, and distribution focal loss (DFL) for both training and validation datasets.

- **Training Loss:**
    - The **train/box_loss** graph shows a steady decrease from approximately 0.80 to 0.40 over the course of the training epochs, indicating that the model is learning to better localize USD currency notes within bounding boxes.
    - The **train/cls_loss** follows a similar decreasing trend, starting at around 3.5 and dropping to below 1.5, reflecting the model's improved ability to accurately classify USD notes.
    - The **train/dfl_loss** exhibits a consistent downward trend, reducing from an initial value of approximately 1.2 to 0.9, signifying enhanced precision in bounding box predictions.
- **Validation Loss:**
    - The **val/box_loss** decreases from around 0.70 to 0.50, though with more fluctuations than the training loss. This variation is expected due to the variable nature of the validation set but overall shows good generalization.
    - Similarly, **val/cls_loss** decreases from 3.0 to just above 1.5, with periodic spikes indicating challenges in generalizing classification over the validation dataset.
    - The **val/dfl_loss** fluctuates between 1.0 and 0.90, following a pattern similar to the training loss but with more variance, indicating the model's stability while optimizing bounding box quality on unseen USD data.

Overall, the alignment between training and validation losses shows minimal signs of overfitting, as both losses decrease steadily over time.

Evaluation Metrics

The model's performance is further evaluated using key object detection metrics, including precision, recall, and mean Average Precision (mAP) at both IoU thresholds of 0.50 (mAP50) and 0.50-0.95 (mAP50-95).

- **Precision**: The **metrics/precision(B)** graph shows a high performance, stabilizing close to 1.0. This indicates that the model is achieving a low rate of false positives, demonstrating its ability to accurately identify USD notes.
- **Recall**: The **metrics/recall(B)** curve shows a steady increase from around 0.50 to above 0.90. This improvement shows the model's growing ability to detect most true positives with fewer missed USD notes.
- **mAP50**: The **metrics/mAP50(B)**, which measures the average precision at IoU threshold 0.50, shows a steady rise from 0.60 to 0.95, confirming that the model is effectively learning to detect and classify USD currency over time.
- **mAP50-95**: The **metrics/mAP50-95(B)** graph shows a consistent increase, reaching around 0.80 at the final epoch. This metric, which accounts for a range of IoU thresholds, highlights the model's robustness across different levels of intersection-over-union (IoU), providing a comprehensive view of performance on USD detection.

This evaluation demonstrates that the model is achieving strong performance in both localization and classification of USD currency, with high precision, recall, and mAP scores, as well as consistent loss reduction over the training epochs.

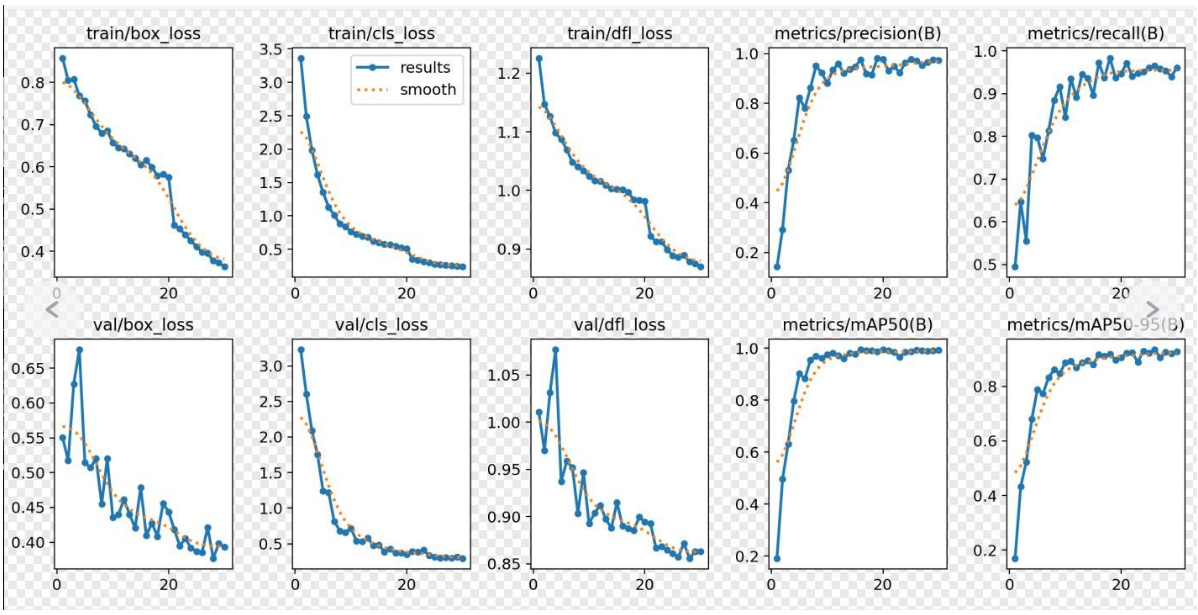


Figure 5.9 Evaluation Metric for USD.

The confusion matrix presented in Figure 5.10 summarizes the model's performance in predicting various classes of USD notes, including both front and back sides of different denominations. The matrix provides insights into the true labels (actual) and predicted labels for each class, allowing for an assessment of misclassifications and overall accuracy.

Class-wise Evaluation:

- **Fifty-back**:
  - Predicted correctly 9 times.
  - One misclassification where the fifty-back note was identified as background.
- **Fifty-front**:
  - Correctly predicted 5 times.
  - One misclassification as the fifty-back note.
- **Five-back**:
  - Correctly predicted 8 times.
  - Two misclassifications: once as five-front and once as ten-front.
- **Five-front**:
  - Accurately predicted 10 times.
  - One instance of misclassification as five-back.
- **One-back**:
  - Predicted correctly 10 times with no misclassifications.
- **One-front**:
  - Predicted correctly 10 times with no misclassifications.
- **Ten-back**:
  - Predicted correctly 10 times with one instance being misclassified as ten-front.
- **Ten-front**:
  - Predicted correctly 7 times.
  - One misclassification as background and one as twenty-back.
- **Twenty-back**:
  - Predicted correctly 8 times.
  - Misclassified 3 times: two as twenty-front and one as ten-front.
- **Twenty-front**:
  - Correctly predicted 13 times.
  - One misclassification as background.

- **Background**:
  - Misclassified once as fifty-back, indicating a false positive.

Overall Insights:

- The model demonstrates strong performance in predicting most classes, particularly for **one-back**, **one-front**, **fifty-back**, **five-front**, and **twenty-front**, with minimal misclassifications.
- A few classes, such as **five-back** and **ten-front**, show minor misclassifications, especially among similar denominations (e.g., five-back being confused with five-front, or ten-back with ten-front).
- **Background** misclassifications are minimal, with only one instance of background being misidentified as fifty-back and two instances where objects were classified as background incorrectly.

This confusion matrix reflects the model's good performance on USD notes, with accurate predictions for most classes and minimal confusion across similar note denominations or backgrounds.

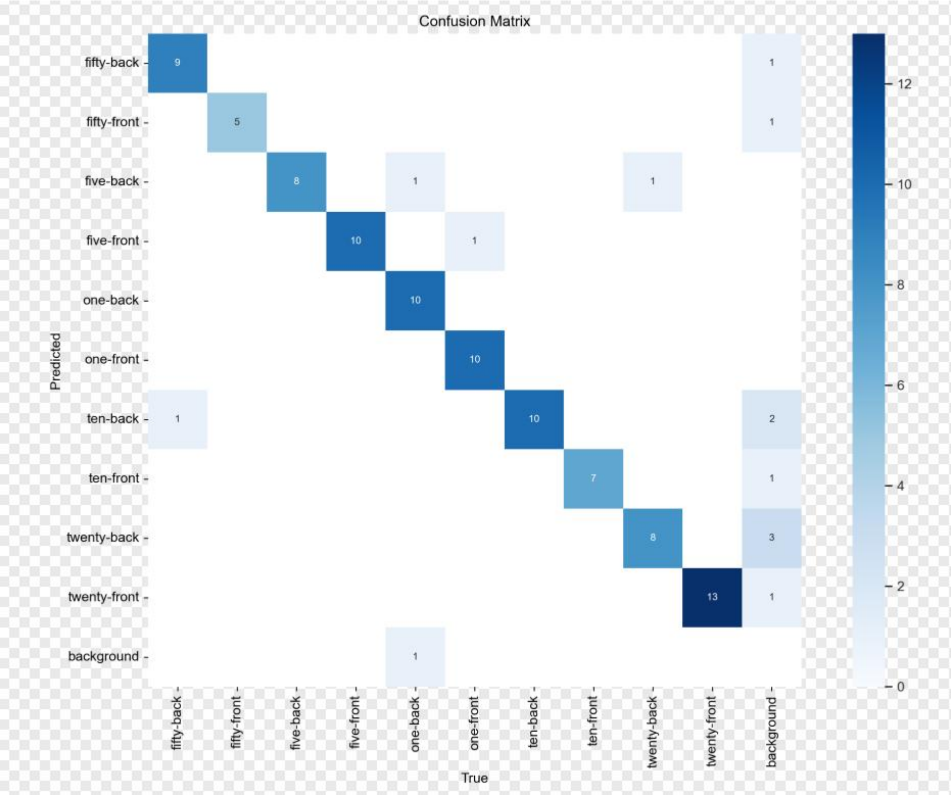


Figure 5.10 Confusion Matrix for USD.

Based on Figure 5.11 provides a detailed evaluation of the model's performance across different classes of USD currency notes, focusing on precision, recall, and mean Average Precision (mAP) metrics. The data includes the number of images, instances, precision (P), recall (R), mAP50, and mAP50-95 values for each class. This analysis highlights the model's detection accuracy and consistency across all classes.

Class-wise Performance:

- **All Classes**:
  - **Images**: 75
  - **Instances**: 95
  - **Box Precision (P)**: 0.972
  - **Recall (R)**: 0.965
  - **mAP50**: 0.993
  - **mAP50-95**: 0.935
  - These overall metrics indicate excellent performance, with very high precision, recall, and mAP scores across all classes.
- **Fifty-back**:
  - **Images**: 10
  - **Instances**: 10
  - **Box Precision (P)**: 0.986
  - **Recall (R)**: 1.0
  - **mAP50**: 0.995
  - **mAP50-95**: 0.963
  - The model shows near-perfect performance in detecting fifty-back notes, with precision, recall, and mAP50 all approaching 1.0.
- **Fifty-front**:
  - **Images**: 5
  - **Instances**: 5
  - **Box Precision (P)**: 0.925
  - **Recall (R)**: 1.0
  - **mAP50**: 0.995
  - **mAP50-95**: 0.935
  - Performance is strong with perfect recall and high mAP50, though precision slightly drops to 0.925.

- **Five-back**:
  - **Images**: 8
  - **Instances**: 8
  - **Box Precision (P)**: 1.0
  - **Recall (R)**: 0.994
  - **mAP50**: 0.995
  - **mAP50-95**: 0.942
  - The model performs excellently on this class with perfect precision and near-perfect recall.
- **Five-front**:
  - **Images**: 10
  - **Instances**: 10
  - **Box Precision (P)**: 0.904
  - **Recall (R)**: 0.948
  - **mAP50**: 0.986
  - **mAP50-95**: 0.911
  - Slightly lower precision and recall compared to other classes, but overall mAP values remain high.
- **One-back**:
  - **Images**: 12
  - **Instances**: 12
  - **Box Precision (P)**: 1.0
  - **Recall (R)**: 0.82
  - **mAP50**: 0.983
  - **mAP50-95**: 0.913
  - While precision is perfect, recall drops to 0.82, suggesting some missed detections for one-back notes.
- **One-front**:
  - **Images**: 11
  - **Instances**: 11
  - **Box Precision (P)**: 1.0
  - **Recall (R)**: 0.892
  - **mAP50**: 0.995

  - **mAP50-95**: 0.888
  - High precision and good recall, though mAP50-95 is slightly lower compared to other classes.
- **Ten-back**:
  - **Images**: 10
  - **Instances**: 10
  - **Box Precision (P)**: 0.949
  - **Recall (R)**: 1.0
  - **mAP50**: 0.995
  - **mAP50-95**: 0.956
  - Strong performance with nearly perfect precision and recall.

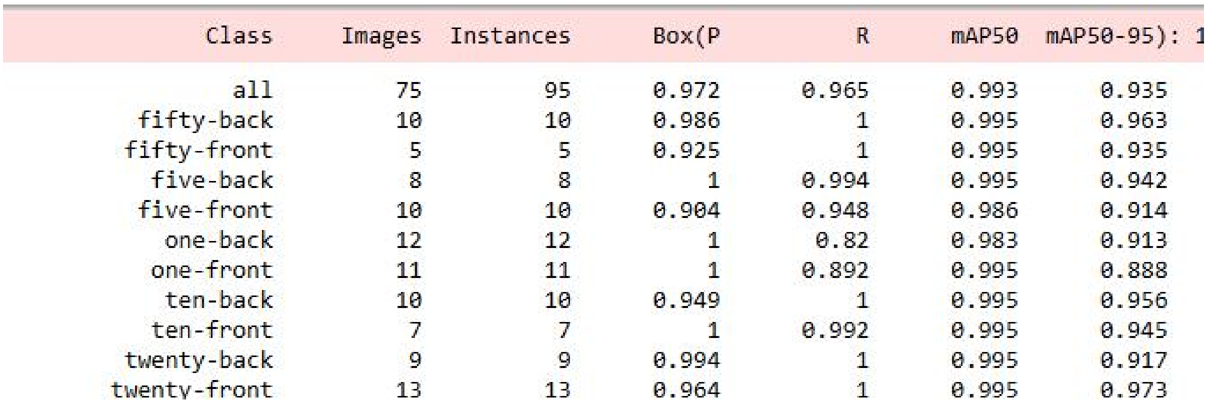
- **Ten-front**:
  - **Images**: 7
  - **Instances**: 7
  - **Box Precision (P)**: 1.0
  - **Recall (R)**: 0.992
  - **mAP50**: 0.995
  - **mAP50-95**: 0.945
  - Excellent results for this class, with very high scores across all metrics.
- **Twenty-back**:
  - **Images**: 9
  - **Instances**: 9
  - **Box Precision (P)**: 0.994
  - **Recall (R)**: 0.991
  - **mAP50**: 0.995
  - **mAP50-95**: 0.917
  - Precision and recall are near perfect, with slightly lower mAP50-95.
- **Twenty-front**:
  - **Images**: 13
  - **Instances**: 13
  - **Box Precision (P)**: 0.964
  - **Recall (R)**: 1.0

  - **mAP50**: 0.995
  - **mAP50-95**: 0.973
  - Outstanding results, with near-perfect scores across all metrics.

Overall Insights:

- The model performs exceptionally well across all USD note classes, with **mAP50** values nearing 1.0 for most classes.
- **Precision (P)** and **Recall (R)** values are consistently high, though there are minor fluctuations in some classes, such as **one-back** and **five-front**.
- The **mAP50-95** metric, which evaluates performance across a range of IoU thresholds, also shows strong results, though it is slightly lower for a few classes compared to mAP50.

This table demonstrates the robustness of the model in accurately detecting and classifying various USD currency notes, with particularly high performance in most categories.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 1
                   all         75         95      0.972      0.965      0.993      0.935
            fifty-back         10         10      0.986          1      0.995      0.963
           fifty-front          5          5      0.925          1      0.995      0.935
             five-back          8          8          1      0.994      0.995      0.942
            five-front         10         10      0.904      0.948      0.986      0.914
              one-back         12         12          1       0.82      0.983      0.913
             one-front         11         11          1      0.892      0.995      0.888
              ten-back         10         10      0.949          1      0.995      0.956
             ten-front          7          7          1      0.992      0.995      0.945
           twenty-back          9          9      0.994          1      0.995      0.917
          twenty-front         13         13      0.964          1      0.995      0.973
```

Figure 5.11 Accuracy of USD using YOLOv8 model.

The image in Figure 5.12 illustrates multiple examples of USD banknotes in various challenging conditions, which serve to evaluate the robustness and generalization of the model. Below are key observations regarding the testing scenarios and the model's ability to handle them:

Evaluation of the Image with USD Notes:

- **Varied Backgrounds**:
  - The image showcases banknotes placed against different surfaces such as wood, outdoor settings, and fabric. The model is still able to detect and classify the

  - banknotes accurately, demonstrating resilience against changes in background complexity.
  - Performance remains consistent across cluttered environments and textured surfaces, highlighting the model's capability to isolate the objects of interest (banknotes) from their surroundings.
- **Different Lighting/Illumination Conditions**:
  - The banknotes are captured under varying lighting conditions, including natural sunlight and indoor artificial lighting. Despite these variations, the model successfully detects and classifies the USD notes, suggesting robustness in handling different lighting situations without a significant drop in performance.
  - Minimal shadows and reflections are present in some instances, but the model's performance remains unaffected, which is a good indicator of its adaptability to real-world environments.
- **Multiple Denominations in a Single Frame**:
  - Several images contain multiple banknotes from different denominations within the same frame (e.g., twenty-front and fifty-back). The model is able to distinguish between the notes and classify them correctly, indicating its capacity to handle multiple objects of interest within a single image.
  - This reinforces the effectiveness of the model in scenarios where mixed currency types may appear together, which is crucial for practical applications like cash reading systems.
- **Partial Obstruction (Finger Blocking)**:
  - In some examples, a user's fingers are partially covering portions of the banknotes, obscuring critical details. Despite this obstruction, the model is still able to detect and classify the banknotes in most cases.
  - The partial occlusion introduces additional complexity by hiding parts of the note, but the model demonstrates resilience in classifying the visible portions accurately. However, this could still pose a challenge when critical details such as denomination or unique identifiers are covered, as it might impact recognition in more extreme cases of occlusion.

Overall Insights:

- The model's ability to detect USD banknotes under varying conditions of background, illumination, and occlusion reflects its robustness and suitability for real-world applications.
- Handling of multiple denominations in a single frame shows that the model can adapt to practical scenarios where banknotes are presented in mixed denominations.
- Minor performance degradation may occur when significant portions of the banknote are obstructed, but overall, the model is capable of maintaining accuracy even under these challenging conditions.

This evaluation highlights the model's strong generalization abilities, making it well-suited for environments where USD banknotes are encountered in varying conditions.

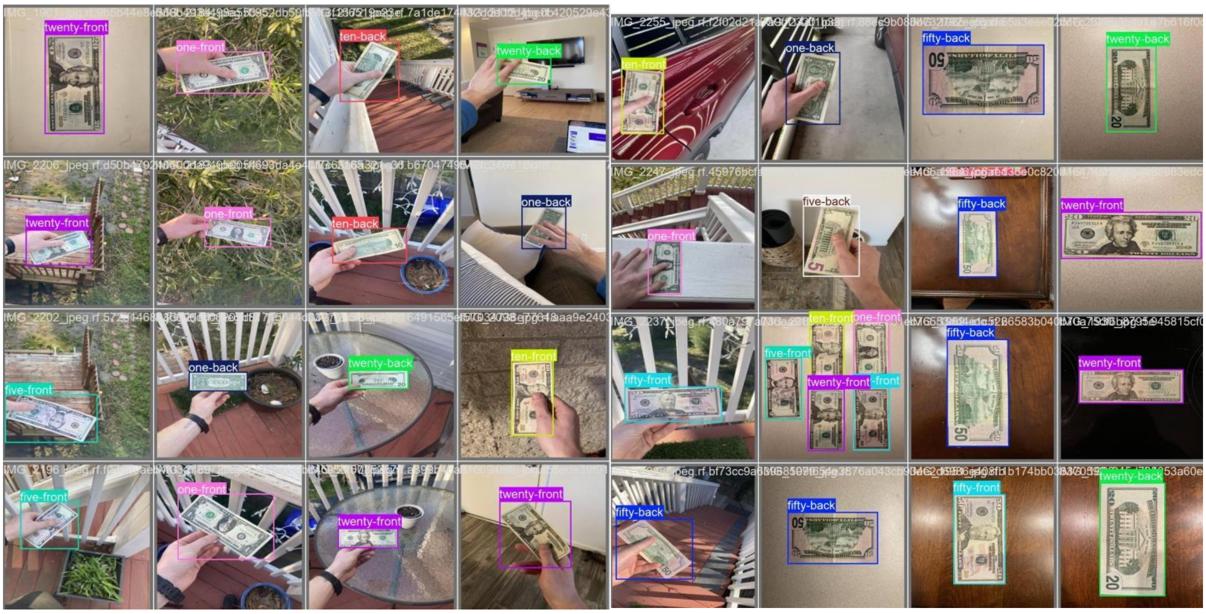


Figure 5.12 Validation Output for USD.

5.1.4 Euro performance on YOLOv8 model

The graphs presented in **Figure 5.13** illustrate the training and validation performance of the model on the EURO dataset. Key metrics evaluated include loss functions and object detection evaluation measures. The analysis is divided into two sections: **loss function evaluation** and **evaluation metrics**.

Loss Function Evaluation

The loss functions tracked during training include box loss, classification loss (cls_loss), and distribution focal loss (DFL) for both training and validation datasets.

Training Loss:

- **train/box_loss**:
    - The box loss decreases from approximately 0.80 to around 0.35 over the course of the training epochs, indicating that the model is learning to better localize objects within bounding boxes for the EURO banknotes.
- **train/cls_loss**:
    - The classification loss starts at around 2.5 and drops to below 1.0, which reflects the model's improved ability to classify the EURO banknotes as the training progresses.
- **train/dfl_loss**:
    - The DFL loss decreases from 1.4 to below 1.1, showing that the model is consistently improving its precision in predicting the bounding boxes over time.

Validation Loss:

- **val/box_loss**:
    - The validation box loss fluctuates at the beginning, starting around 1.1 and decreasing to approximately 0.40. This shows that the model's performance on unseen data improves steadily with some variability in the initial stages.
- **val/cls_loss**:
    - The validation classification loss starts around 2.9 and drops to about 1.0, with several fluctuations. This suggests challenges in generalizing across some validation data but overall improvement as training progresses.
- **val/dfl_loss**:

  - The validation DFL loss shows a similar pattern to the training loss, decreasing from 1.2 to just below 1.0, indicating that the model retains stability in predicting bounding boxes even on unseen data.

Summary of Loss Function Behavior:

- There is alignment between the training and validation losses, with both gradually decreasing over time, though validation shows some fluctuations. This indicates that the model is learning effectively with minimal overfitting signs on the EURO dataset.

Evaluation Metrics

The model's performance is further evaluated using key object detection metrics, including precision, recall, and mean Average Precision (mAP) at IoU thresholds of 0.50 (mAP50) and 0.50-0.95 (mAP50-95).

Precision:

- **metrics/precision(B)**:
  - The precision starts at approximately 0.85 and rises to around 0.90, demonstrating that the model's accuracy in correctly identifying EURO banknotes with fewer false positives improves over time.

Recall:

- **metrics/recall(B)**:
  - The recall improves from around 0.65 to over 0.90 by the end of training, indicating the model's ability to detect true positives improves consistently, although some fluctuations suggest inconsistency in certain epochs.

mAP50:

- **metrics/mAP50(B)**:
  - The mean Average Precision at IoU threshold 0.50 increases from 0.70 to nearly 0.90, confirming that the model is effectively learning to detect and classify EURO banknotes.

mAP50-95:

- **metrics/mAP50-95(B)**:
  - The mAP50-95, which measures performance across multiple IoU thresholds, shows a consistent upward trend, reaching around 0.80 by the end of the

training. This metric highlights the model's robustness across different levels of intersection-over-union (IoU), providing a comprehensive assessment of its detection performance.

Summary of Evaluation Metrics:

- The model performs well on the EURO dataset, with both precision and recall improving steadily. The increase in mAP50 and mAP50-95 indicates that the model is learning to generalize across different IoU thresholds, resulting in robust detection and classification of EURO banknotes.

Overall Insights:

- The model demonstrates strong generalization ability on the EURO dataset, with decreasing loss values and increasing precision, recall, and mAP metrics.
- Some fluctuations in validation loss and recall indicate that there might be challenging cases within the validation set, but the overall performance shows consistent improvement.
- The final mAP50-95 value of 0.80 suggests that the model performs well across various IoU thresholds, making it suitable for detecting EURO banknotes in a variety of conditions.

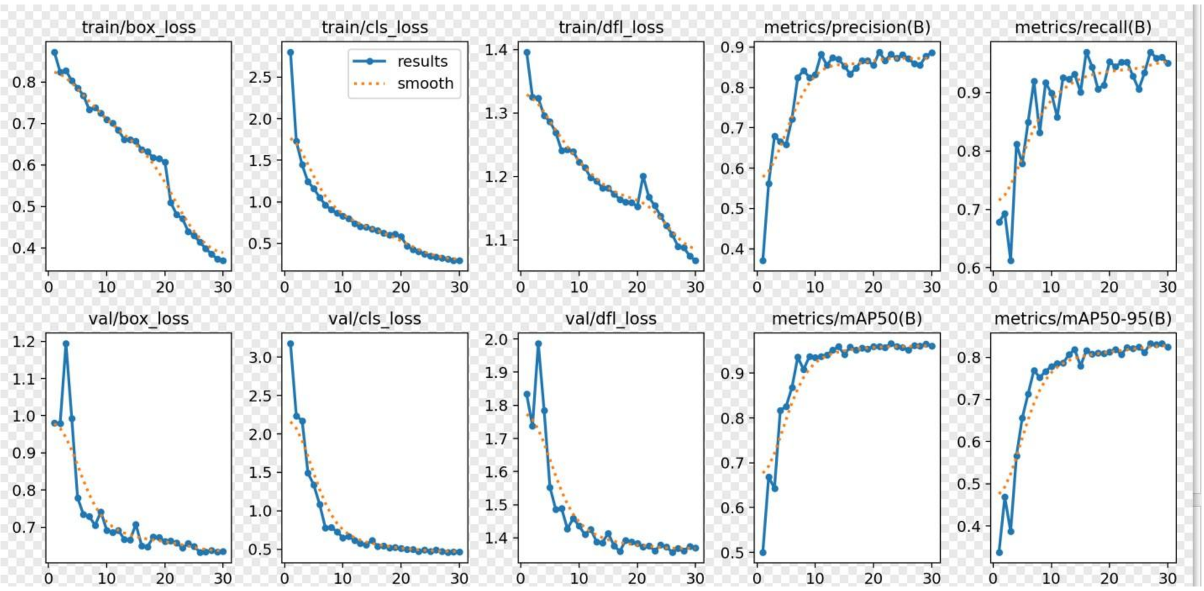


Figure 5.13 Evaluation Metrics for EURO.

The confusion matrix presented in **Figure 5.14** provides a detailed view of the model's performance on the EURO banknote dataset by comparing true labels with predicted labels. The diagonal values represent correctly classified instances, while the off-diagonal values show the misclassifications.

Key Insights:

- **Overall Accuracy**:
    - Most of the predicted labels align with the true labels, as the majority of the values are concentrated along the diagonal. This shows that the model can accurately classify most denominations of EURO banknotes.
- **Correct Classifications**:
    - The model correctly classified:
        - **100 EUR** banknotes 30 times.
        - **10 EUR** banknotes 32 times.
        - **200 EUR** banknotes 30 times.
        - **20 EUR** banknotes 39 times.
        - **500 EUR** banknotes 42 times.
        - **50 EUR** banknotes 35 times.
        - **5 EUR** banknotes 29 times.

These are strong indicators that the model performs well across various EURO denominations, with the highest number of correct classifications for **500 EUR** (42 correct predictions).

- **Misclassifications**:
    - There are some misclassifications, albeit relatively minor:
        - **100 EUR** was misclassified as background twice.
        - **10 EUR** was confused with 5 EUR once.
        - **200 EUR** was misclassified as **500 EUR** three times and once as **50 EUR**.
        - **20 EUR** was misclassified as **5 EUR** six times.
        - **500 EUR** was misclassified as **50 EUR** six times.
        - **50 EUR** was misclassified as **500 EUR** nine times and as **20 EUR** once.
        - **5 EUR** was misclassified as **20 EUR** five times.

The misclassifications tend to occur between similar-sized denominations, such as **50 EUR** and **500 EUR**, which might be due to visual similarities in features or the positioning of the notes in the images.

- **Background Confusion**:
  - There are very few cases where actual notes were classified as "background" and vice versa. This shows that the model is good at distinguishing banknotes from the background.

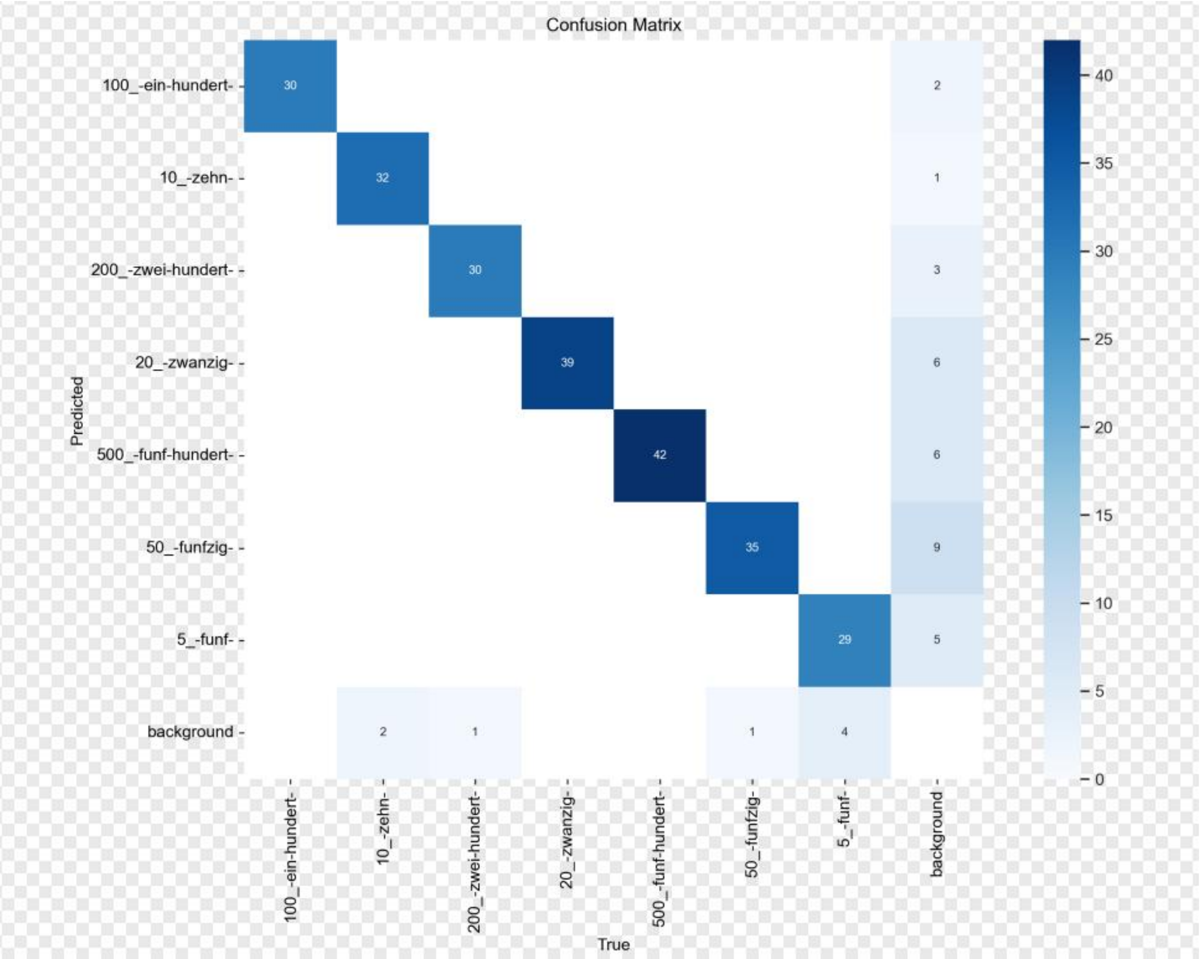


Figure 5.14 Confusion Matrix for EURO.

Based on Figure 5.15, this section provides a detailed evaluation of the model's performance across different classes of Euro currency notes, focusing on precision, recall, and mean Average Precision (mAP) metrics. The analysis highlights the model's detection accuracy and consistency across all classes.

Class-wise Performance:

- **All Classes**:

  - Images: 245
  - Instances: 245
  - Box Precision (P): 0.878
  - Recall (R): 0.961
  - mAP50: 0.965
  - mAP50-95: 0.832
  - These overall metrics indicate strong model performance, with high precision, recall, and mAP values across all classes.

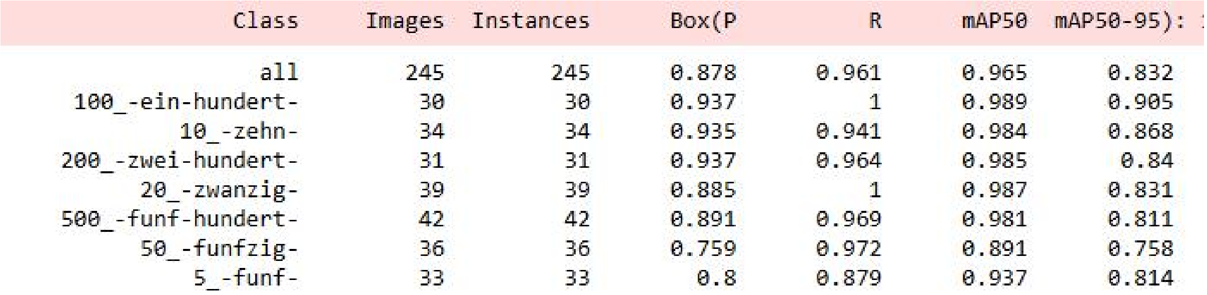
- **100-ein-hundert**:
  - Images: 30
  - Instances: 30
  - Box Precision (P): 0.937
  - Recall (R): 1.0
  - mAP50: 0.989
  - mAP50-95: 0.905
  - The model achieves perfect recall for this class, and its precision and mAP values are notably high.
- **10-zehn**:
  - Images: 34
  - Instances: 34
  - Box Precision (P): 0.935
  - Recall (R): 0.941
  - mAP50: 0.984
  - mAP50-95: 0.868
  - Strong performance in detecting 10 Euro notes, with high precision and recall, though mAP50-95 is slightly lower.
- **200-zwei-hundert**:
  - Images: 31
  - Instances: 31
  - Box Precision (P): 0.937
  - Recall (R): 0.964
  - mAP50: 0.985
  - mAP50-95: 0.84

  - The model shows good performance for 200 Euro notes, with high recall and precision values.
- **20-zwanzig**:
  - Images: 39
  - Instances: 39
  - Box Precision (P): 0.885
  - Recall (R): 1.0
  - mAP50: 0.987
  - mAP50-95: 0.831
  - The model detects all instances perfectly for this class, although the box precision is slightly lower.
- **500-funf-hundert**:
  - Images: 42
  - Instances: 42
  - Box Precision (P): 0.891
  - Recall (R): 0.969
  - mAP50: 0.969
  - mAP50-95: 0.811
  - Precision and recall are strong for this class, with solid mAP metrics.
- **50-funfzig**:
  - Images: 36
  - Instances: 36
  - Box Precision (P): 0.759
  - Recall (R): 0.972
  - mAP50: 0.891
  - mAP50-95: 0.758
  - Precision for 50 Euro notes is lower than other classes, though recall remains high.
- **5-funf**:
  - Images: 33
  - Instances: 33
  - Box Precision (P): 0.8
  - Recall (R): 0.879

- mAP50: 0.937
- mAP50-95: 0.814
- While precision and recall are lower for this class, the mAP metrics still show decent performance.

Overall Insights:

- The model performs well across the majority of Euro note classes, with particularly high precision and recall values in most cases.
- The **mAP50** values, which measure accuracy at a single IoU threshold of 50%, are consistently high across all classes, indicating strong object detection.
- The **mAP50-95** values, which take into account a range of IoU thresholds, are slightly lower, particularly for the 50-Euro class, reflecting more room for improvement in handling varied object sizes and positions.



```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): :
                   all        245        245      0.878      0.961      0.965      0.832
    100_-ein-hundert-         30         30      0.937          1      0.989      0.905
            10_-zehn-         34         34      0.935      0.941      0.984      0.868
   200_-zwei-hundert-         31         31      0.937      0.964      0.985       0.84
         20_-zwanzig-         39         39      0.885          1      0.987      0.831
   500_-funf-hundert-         42         42      0.891      0.969      0.981      0.811
         50_-funfzig-         36         36      0.759      0.972      0.891      0.758
             5_-funf-         33         33        0.8      0.879      0.937      0.814
```

Figure 5.15 Accuracy of EURO using YOLOv8 model.

Based on the Figure 5.16, which evaluates the YOLOv8 model's performance on various Euro currency notes, we can assess the model's detection accuracy under different scenarios. The image shows both successful detections (with bounding boxes) and some challenging cases. Here's a detailed evaluation based on the visual results.

Scenarios Evaluated:

1. **Different Denominations**:
    - The evaluation image includes a wide variety of Euro notes, such as 5, 20, 100, 200, and 500 Euros.
    - The model successfully detects and distinguishes between these different denominations with high accuracy, as shown by the colored bounding boxes

around the notes with corresponding class labels (e.g., "5-funf," "20-zwanzig," etc.).

- **Performance**: High detection accuracy across most classes, with correct classification in most cases.

2. **Cluttered Backgrounds**:
   - Some images include cluttered or patterned backgrounds, yet the model is still able to accurately detect and classify the notes.
   - For example, in the left panel, several 5-Euro notes are detected against busy backgrounds.
   - **Performance**: The model maintains high precision and recall even with cluttered backgrounds, suggesting robustness in real-world environments.

3. **Different Angles and Rotations**:
   - The image set includes currency notes presented at various angles and rotations, such as tilted or partially obscured notes.
   - The model still correctly identifies most of the notes, even when rotated or presented at an angle.
   - **Performance**: High, with no noticeable degradation in detection ability despite the changes in orientation.

4. **Partially Obstructed Notes**:
   - Some images feature notes that are partially hidden or stacked, making detection more challenging.
   - The model successfully detects the notes, although there may be a few instances where small portions of the notes go undetected (especially when the obstruction is significant).
   - **Performance**: Good, but performance could slightly drop when the notes are heavily obscured.

5. **Varying Lighting Conditions**:

- The images showcase different lighting conditions, including dimly lit and brightly lit scenes.
- The model appears to perform well in both bright and dark environments, as indicated by the consistent detection of the notes.
- **Performance**: Strong, showing good adaptability to lighting variations.

6. **Multiple Denominations in a Single Frame**:
    - Several frames contain multiple Euro notes, sometimes of different denominations, within a single image.
    - The model is able to correctly detect and classify each note separately, showing good multi-object detection capability.
    - **Performance**: Excellent, with the model accurately identifying multiple notes without confusion.

7. **Different Types of Euro Notes (Front vs. Back)**:
    - The image includes both the front and back sides of various Euro notes.
    - The model successfully differentiates between the front and back views, assigning the correct labels like "funf" (5 Euro front) and "funf-back" (5 Euro back).
    - **Performance**: High, with correct classification for both the front and back of notes.

Overall Evaluation:

The YOLOv8 model demonstrates robust performance across different scenarios, including variations in lighting, note orientation, and background complexity. The detection and classification accuracy is high, and the model performs well even in challenging situations, such as detecting partially obstructed or tilted notes. The bounding boxes are well-placed, and the model’s ability to differentiate between multiple denominations within a single frame is commendable.

Key Strengths:

- **Accurate Multi-Object Detection**: The model effectively handles multiple notes in one frame, assigning the correct class to each detected note.

- **Robust to Background Clutter and Lighting Variations**: Detection remains accurate even with patterned backgrounds and varying lighting conditions.
- **High Precision for Both Front and Back Notes**: The model reliably distinguishes between the front and back of the notes, further enhancing its real-world usability.

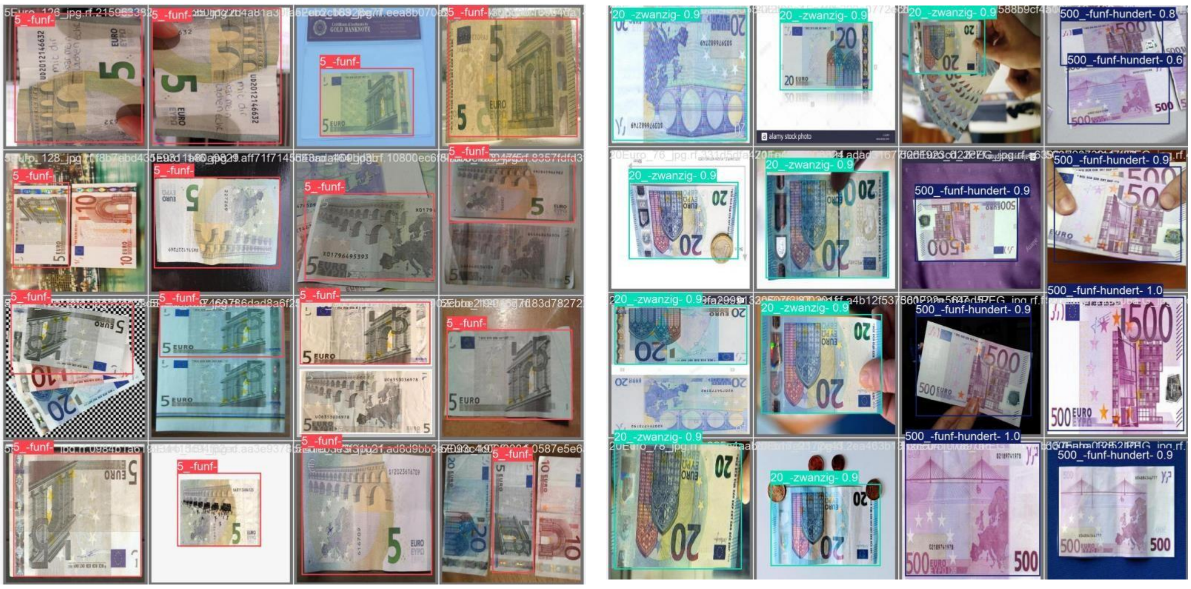


Figure 5.16 Validation Output for EURO.

5.1.5 Thai Baht performance on YOLOv8 model

Figure 5.17 presents an in-depth evaluation of the YOLOv8 model's performance for Thai Baht notes across various metrics, including losses (box, class, and DFL), precision, recall, and mAP50-95. These metrics are plotted across training and validation sets over 30 epochs, showing how well the model generalizes over time.

Training Losses and Validation Losses:

- **Box Loss**:
    - **Training**: Starts at approximately 1.1 and steadily decreases to around 0.6 after 30 epochs.
    - **Validation**: Box loss for validation starts higher than training at around 0.75 but shows similar downward trends, stabilizing just below 0.6.
    - **Interpretation**: The decreasing box loss indicates the model is improving in predicting bounding boxes for Thai Baht notes as training progresses.
- **Class Loss**:
    - **Training**: Starts at approximately 2.5 and decreases smoothly to below 0.75.
    - **Validation**: Follows a similar trend, beginning at 1.75 and reaching about 0.65 by the end.
    - **Interpretation**: The class loss reduction suggests that the model is becoming more accurate in classifying different denominations of Thai Baht notes.
- **DFL (Distribution Focal Loss)**:
    - **Training**: Begins at around 1.3 and reduces to approximately 0.85.
    - **Validation**: Similar to training, starting near 1.1 and dropping to below 0.85.
    - **Interpretation**: This indicates an improvement in the model's ability to refine the objectness score and localize the bounding boxes.

Metrics - Precision, Recall, and mAP:

- **Precision**:
    - **Training**: The precision metric steadily increases from 0.7 to over 0.95 during training.
    - **Validation**: Similarly, the precision in validation starts at about 0.75 and rises to over 0.95, indicating the model's excellent ability to minimize false positives while detecting Thai Baht notes.
- **Recall**:

    - **Training**: The recall metric improves from 0.75 to 0.95 over the course of training.
    - **Validation**: Follows a similar upward trend, reaching above 0.95, demonstrating the model's ability to minimize false negatives and detect nearly all relevant objects in the validation set.
- **mAP50**:
    - **Training**: The mAP50 starts at 0.75 and climbs above 0.95 by the end of training.
    - **Validation**: The validation set mAP50 similarly rises to over 0.95, suggesting that the model performs exceptionally well when using the 50% IoU (Intersection over Union) threshold for object detection.
- **mAP50-95**:
    - **Training**: The mAP50-95 metric, which is more stringent, starts at around 0.65 and reaches approximately 0.85.
    - **Validation**: Similarly, the validation set mAP50-95 begins at about 0.7 and rises to just below 0.85, indicating strong performance even across a range of IoU thresholds.

Overall Insights for Thai Baht Notes:

- The YOLOv8 model demonstrates consistent improvement in detecting Thai Baht notes as training progresses.
- Precision and recall metrics both surpass 0.95, indicating strong performance in accurately detecting and classifying objects without significant overfitting.
- The mAP metrics (both mAP50 and mAP50-95) indicate that the model achieves high detection accuracy and robustness across different IoU thresholds.
- Overall, the model shows excellent generalization from training to validation, with minimal divergence between the two, confirming its strong performance in detecting and recognizing Thai Baht currency notes.

This evaluation highlights the YOLOv8 model's effectiveness in recognizing Thai Baht denominations with high precision, recall, and mAP scores, showcasing its suitability for currency recognition tasks.

4o

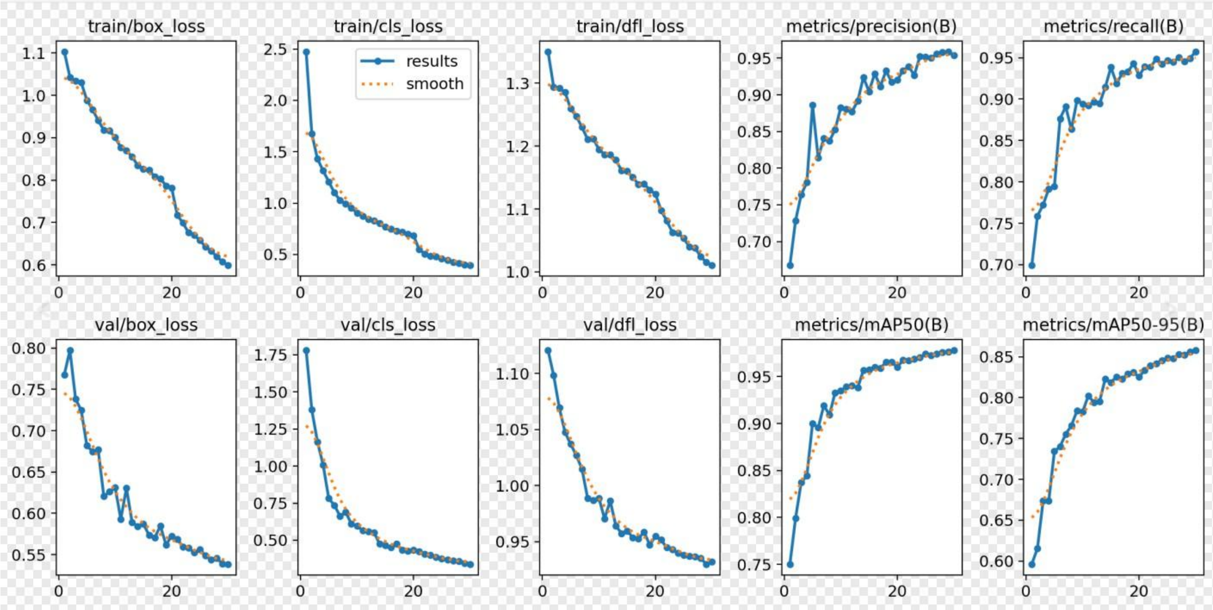


Figure 5.17 Evaluation Metric for Thai Baht.

The confusion matrix in Figure 5.18 provides a detailed breakdown of the YOLOv8 model's performance for detecting various Thai Baht denominations. Each row represents the predicted class, while each column represents the actual (true) class. The diagonal values indicate correct predictions, while off-diagonal values represent misclassifications.

Key Observations from the Confusion Matrix:

1. **1000 Baht**:
   - Correct Predictions: 219
   - Misclassifications: 3 instances predicted as 100 Baht, 11 as background.
   - **Evaluation**: The model performs exceptionally well in identifying 1000 Baht notes with minimal confusion.
2. **100 Baht**:
   - Correct Predictions: 198
   - Misclassifications: 1 instance predicted as 1000 Baht, 2 as background, 6 as other categories.
   - **Evaluation**: Strong performance with some minor confusion with background and other classes.
3. **10 Baht**:
   - Correct Predictions: 216
   - Misclassifications: 1 instance predicted as 20 Baht, 11 as background.

  - **Evaluation**: The model is highly accurate in detecting 10 Baht, with few background misclassifications.

4. **1 Baht**:
   - Correct Predictions: 168
   - Misclassifications: 3 instances predicted as 2 Baht, 42 as background.
   - **Evaluation**: The 1 Baht coin has a noticeable number of misclassifications, especially with background, indicating some challenge in differentiating it.
5. **20 Baht**:
   - Correct Predictions: 232
   - Misclassifications: 1 instance predicted as 2 Baht, 6 as background.
   - **Evaluation**: Excellent performance, with very few errors.
6. **2 Baht**:
   - Correct Predictions: 191
   - Misclassifications: 12 instances predicted as 1 Baht, 16 as background.
   - **Evaluation**: Strong performance with occasional confusion with the 1 Baht coin and the background.
7. **500 Baht**:
   - Correct Predictions: 228
   - Misclassifications: 2 instances predicted as 50 Baht, 11 as background.
   - **Evaluation**: The model performs exceptionally well, with very few errors in detecting 500 Baht notes.
8. **50 Baht**:
   - Correct Predictions: 231
   - Misclassifications: 2 instances predicted as 500 Baht, 8 as background.
   - **Evaluation**: Strong detection accuracy with minimal confusion.
9. **5 Baht**:
   - Correct Predictions: 212
   - Misclassifications: 1 instance predicted as 50 Baht, 29 as background.
   - **Evaluation**: Some challenges in distinguishing the 5 Baht coin from the background, leading to a higher misclassification rate.
10. **Background**:

- ○ Correct Predictions: Most correctly identified as background, with a few misclassifications into other classes such as 1000 Baht, 100 Baht, and lower denominations.
- ○ **Evaluation**: Background detection is generally reliable, but some instances are incorrectly classified as currency.

Overall Evaluation:

- **High Accuracy for Large Denominations**: The model performs best for higher-value notes such as 1000 Baht, 500 Baht, and 100 Baht, with accuracy rates close to 100%.
- **Slight Confusion for Smaller Denominations**: There is some confusion between smaller denominations, particularly with 1 Baht and 2 Baht coins, and background.
- **Background Classification**: The model occasionally struggles to differentiate small coins like 1 Baht or 5 Baht from background noise, which could be due to similarities in shape or size.

Summary:

The confusion matrix indicates that YOLOv8 performs exceptionally well for detecting and classifying larger currency notes but faces minor challenges with small denominations and distinguishing coins from the background.

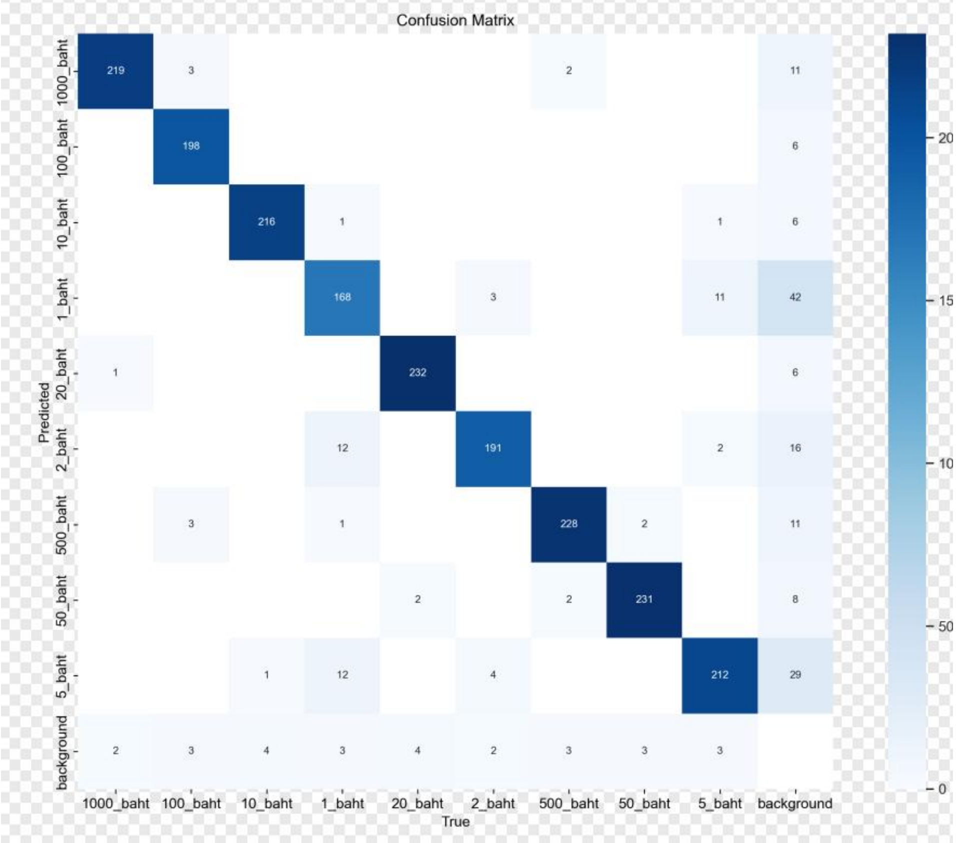


Figure 5.18 Confusion Matrix for Baht.

This Figure 5.19 summarizes the performance of the YOLOv8 model for detecting various Thai Baht denominations. The metrics included are:

- **P** (Precision): The fraction of true positive detections out of all positive detections.
- **R** (Recall): The fraction of true positive detections out of all actual positive instances.
- **mAP50**: Mean Average Precision at 50% IoU threshold, measuring the model's ability to detect objects.
- **mAP50-95**: Mean Average Precision across IoU thresholds from 50% to 95%, providing a more comprehensive evaluation of model performance.

Key Metrics and Evaluation:

1. **Overall Performance (All Classes)**:
   - **Precision (P)**: 0.954
   - **Recall (R)**: 0.958
   - **mAP50**: 0.978
   - **mAP50-95**: 0.858
   - **Evaluation**: The overall model performance is strong with high precision and recall, and excellent mAP values, indicating accurate and reliable detection across all classes.
2. **1000 Baht**:
   - **Precision (P)**: 0.968
   - **Recall (R)**: 0.991
   - **mAP50**: 0.991
   - **mAP50-95**: 0.886
   - **Evaluation**: The model performs excellently for detecting 1000 Baht notes with near-perfect precision and recall. The mAP50-95 of 0.886 reflects a very high detection capability at varying IoU thresholds.
3. **100 Baht**:
   - **Precision (P)**: 0.990
   - **Recall (R)**: 0.965
   - **mAP50**: 0.988
   - **mAP50-95**: 0.873
   - **Evaluation**: The model's performance is near-perfect for 100 Baht notes, with excellent precision and recall values.
4. **10 Baht**:

- **Precision (P)**: 0.977
- **Recall (R)**: 0.982
- **mAP50**: 0.979
- **mAP50-95**: 0.857
- **Evaluation**: The model accurately detects 10 Baht coins with high precision and recall, though the mAP50-95 value is slightly lower than higher denominations, reflecting more difficulty in detecting coins at tighter IoU thresholds.

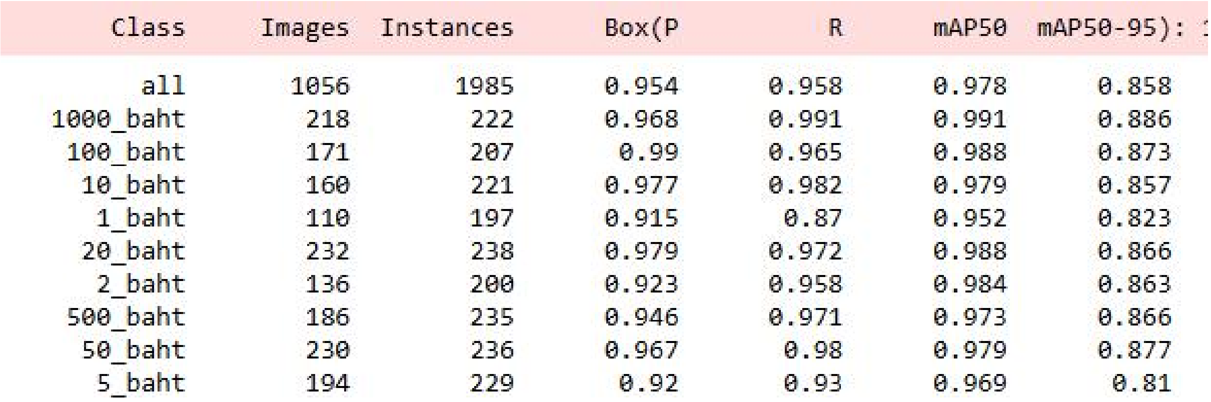
5. **1 Baht**:
   - **Precision (P)**: 0.915
   - **Recall (R)**: 0.870
   - **mAP50**: 0.952
   - **mAP50-95**: 0.823
   - **Evaluation**: The detection performance for 1 Baht coins is lower compared to higher denominations, with a noticeable drop in recall (R). This suggests the model has more difficulty correctly identifying 1 Baht coins, likely due to their smaller size or similarity to background noise.
6. **20 Baht**:
   - **Precision (P)**: 0.979
   - **Recall (R)**: 0.972
   - **mAP50**: 0.988
   - **mAP50-95**: 0.866
   - **Evaluation**: The model performs extremely well in detecting 20 Baht notes with strong precision and recall values.
7. **2 Baht**:
   - **Precision (P)**: 0.923
   - **Recall (R)**: 0.958
   - **mAP50**: 0.984
   - **mAP50-95**: 0.863
   - **Evaluation**: The model shows good performance for 2 Baht coins with slightly lower precision but strong recall.
8. **500 Baht**:
   - **Precision (P)**: 0.946

  - **Recall (R)**: 0.971
  - **mAP50**: 0.973
  - **mAP50-95**: 0.860
  - **Evaluation**: Detection of 500 Baht notes is reliable, with good precision and recall metrics.

9. **50 Baht**:
   - **Precision (P)**: 0.967
   - **Recall (R)**: 0.980
   - **mAP50**: 0.979
   - **mAP50-95**: 0.877
   - **Evaluation**: The model performs very well for 50 Baht notes with a high mAP50-95 score.
10. **5 Baht**:
    - **Precision (P)**: 0.920
    - **Recall (R)**: 0.930
    - **mAP50**: 0.969
    - **mAP50-95**: 0.810
    - **Evaluation**: Detection of 5 Baht coins is more challenging for the model, as indicated by slightly lower precision, recall, and mAP values compared to larger denominations.

Overall Summary:

- The model achieves **high precision and recall** for most Thai Baht denominations, particularly for higher-value notes like 1000, 100, 500, and 20 Baht, with **mAP50 values close to or above 0.98**.
- There are **slightly more challenges in detecting smaller coins**, especially 1 Baht and 5 Baht, where both precision and recall drop, and the **mAP50-95 values are lower** compared to other classes.
- **Model performance is robust** across most categories, with **precision and recall metrics above 0.9** for all classes, ensuring reliable detection of Thai Baht in most scenarios.



```
      Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
        all       1056       1985      0.954      0.958      0.978      0.858
  1000_baht        218        222      0.968      0.991      0.991      0.886
   100_baht        171        207       0.99      0.965      0.988      0.873
    10_baht        160        221      0.977      0.982      0.979      0.857
     1_baht        110        197      0.915       0.87      0.952      0.823
    20_baht        232        238      0.979      0.972      0.988      0.866
     2_baht        136        200      0.923      0.958      0.984      0.863
   500_baht        186        235      0.946      0.971      0.973      0.866
    50_baht        230        236      0.967       0.98      0.979      0.877
     5_baht        194        229       0.92       0.93      0.969       0.81
```

Figure 5.19 Performance of Baht using YOLOv8 model.

The Figure 5.20 showcases the model's predictions on images containing Baht currency (both banknotes and coins). Several conditions were employed to increase the complexity of the task, which helps in assessing the robustness of the model's performance.

Conditions to Increase Complexity:

1. **Multiple Currency Types and Denominations**:
   - Both **banknotes** and **coins** of various denominations (1000, 500, 100, 50, 20 Baht notes, and 10 Baht coins) are included in the images, making the detection task more complex.
2. **Mixed Denominations**:
   - Different denominations are placed together in close proximity within the same frame, which increases the challenge for the model to distinguish and correctly classify each note and coin.
3. **Varied Backgrounds**:
   - The images display diverse backgrounds, including plain surfaces, reflective surfaces, and human hands, adding to the difficulty of detecting objects accurately.
   - Objects like cups, pens, and mobile devices also appear in the background, which introduces potential distractions for the model.
4. **Different Illumination Conditions**:
   - The images exhibit varying lighting conditions, from well-lit environments to dimmer settings, which test the model's ability to maintain detection accuracy under different lighting.

5. **Obstructed Objects**:
   - Some of the Baht coins are partially blocked by fingers, making it harder for the model to recognize them accurately.
   - Coins and banknotes overlap in some images, which presents an additional challenge for object detection.
6. **Rotations and Angles**:
   - Baht banknotes are captured at different angles and orientations, further testing the model's capability to handle spatial transformations.
7. **Presence of Foreign Objects**:
   - The presence of unrelated objects (e.g., camera lens, finger) introduces potential noise that could confuse the model during classification.

Model's Performance:

- **Correct Predictions**:
  - The model has successfully detected and classified most of the Baht denominations correctly, with bounding boxes highlighting the denominations (e.g., **100 Baht**, **500 Baht**, **10 Baht coin**).
  - The bounding boxes and labels indicate that the model is capable of differentiating between various notes and coins, even when multiple are present in a single image.

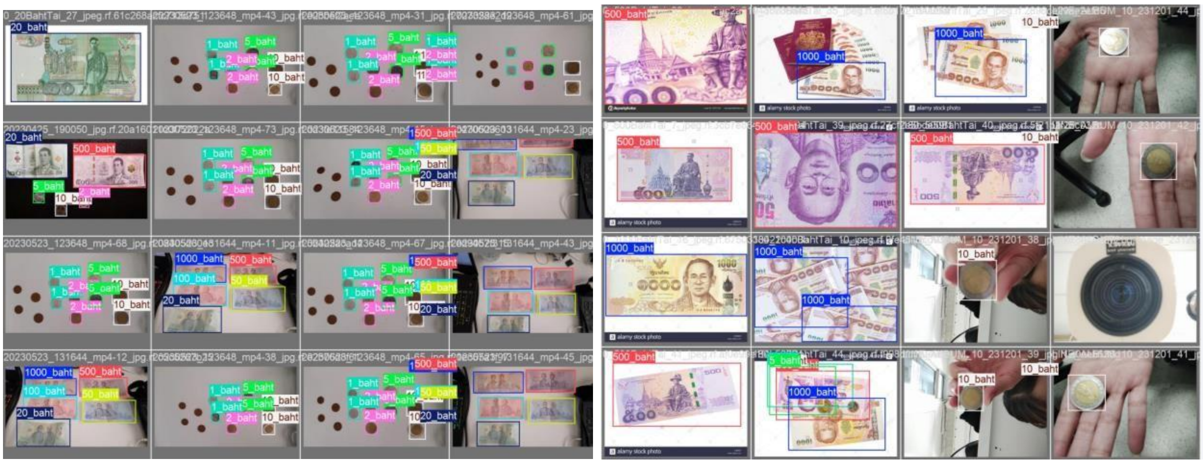


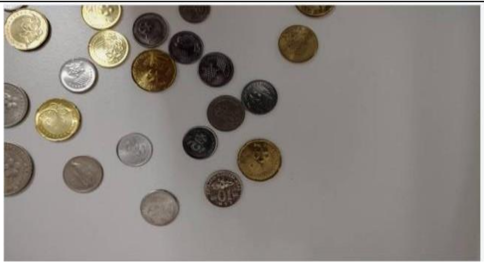
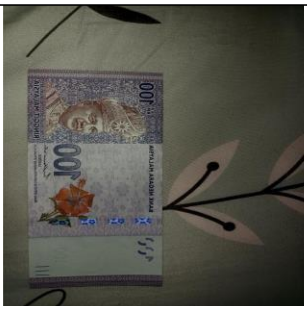
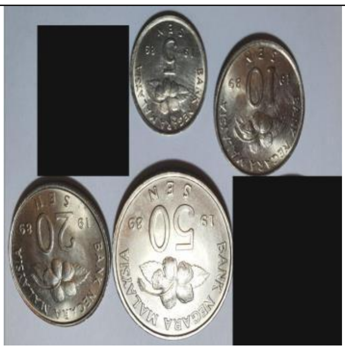
Figure 5.20 Output for Baht.

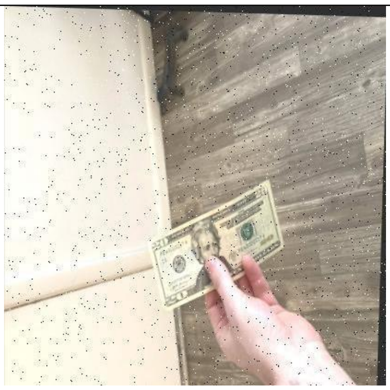
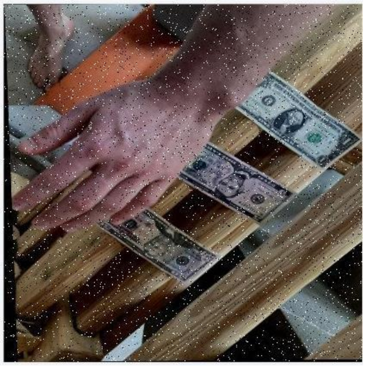
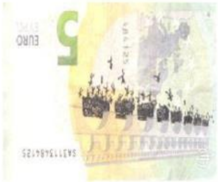
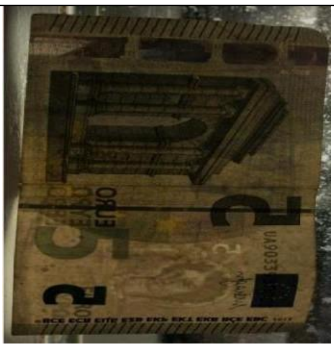
5.2 Data Collection Process Evaluation

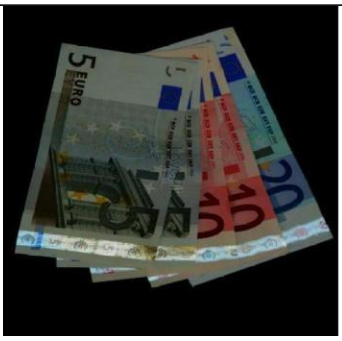
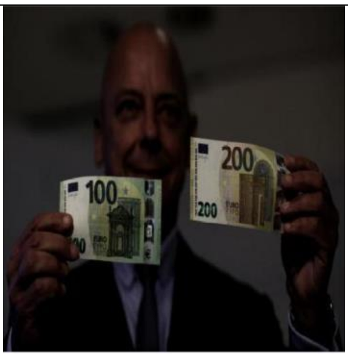
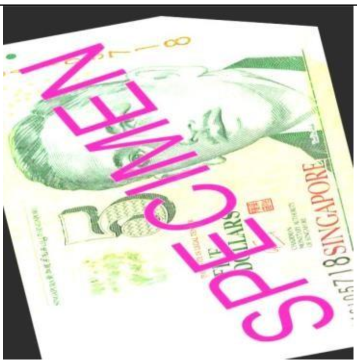
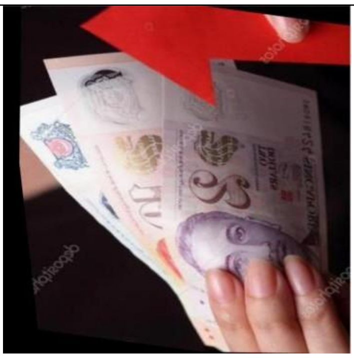
The dataset consists of various currency types, simulating real-world scenarios where different currencies are encountered. The inclusion of multiple currencies ensures that the system can effectively detect and classify a wide range of banknotes, which is crucial for the Cash Reader system.

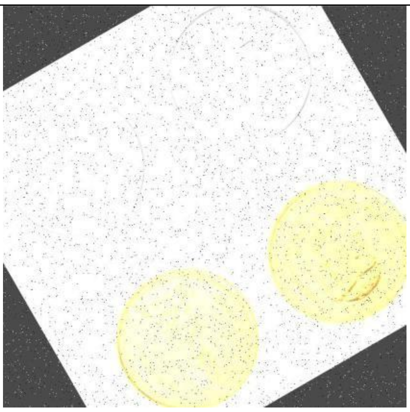
- Malaysian Ringgit (MYR)
- United States Dollar (USD)
- Euro (EUR)
- Singapore Dollar (SGD)
- Thai Baht (THB)

Table 5.1 Example of currency image contained in the dataset.

No.	Currency and Denominations	Currency Image	Discussion
1	MYR Coins		-Different denominations of coins in a single frame
2	MYR Banknote		-Illustartion of highly illuminated scenarios -Different rotation and angles
3	MYR Coins		-Using sticky tape to create presence of foreign object -Different background

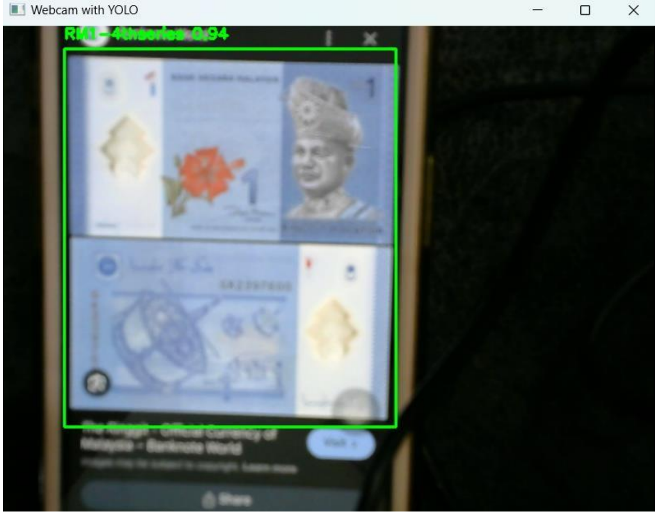
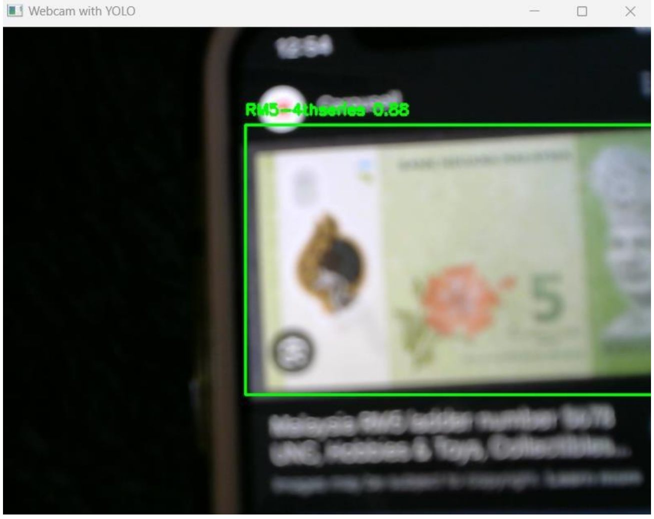
4	USD Banknote		-Create obstruction using finger to block the partial information of the banknote -Highly illuminated -Salt and pepper noise
5	USD Banknote		-Different backgrond -Create obstruction -Multiple denominations in a single frame -Salt and pepper noise
6	EURO Banknote		-Highly illuminated -Different background
7	EURO Banknote		-Highly illuminated -Folded banknote -Worn/old banknote used

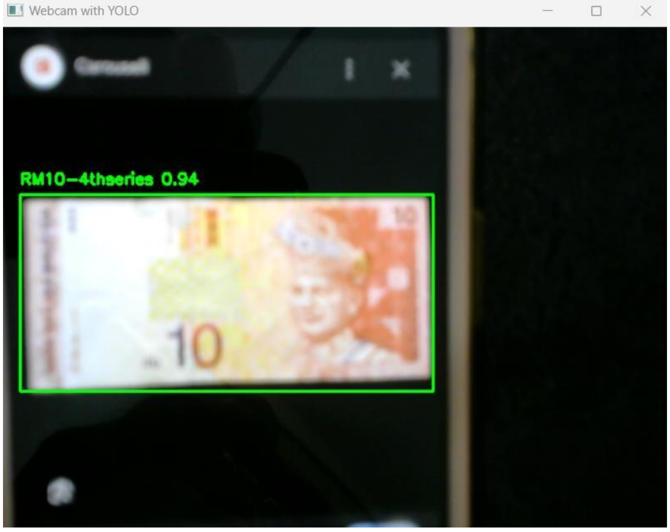
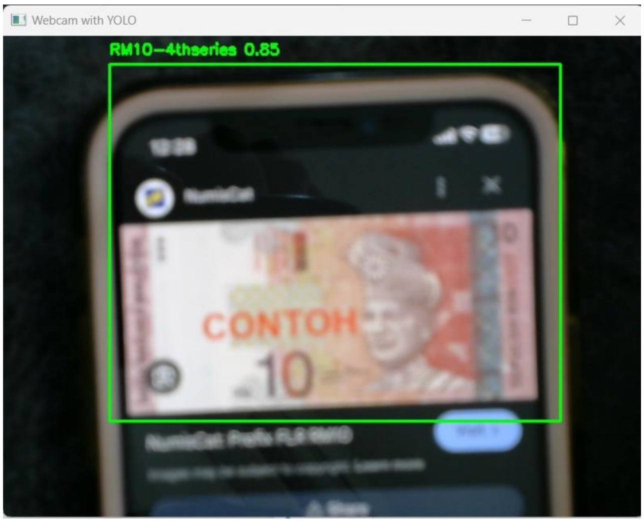
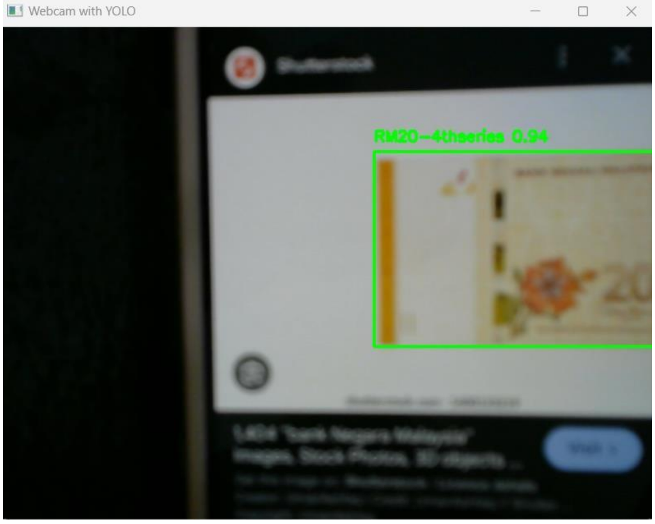
10	EURO Banknote		-Highly illuminated -Multiple denominations in a single frame -Captured in different angle
11	EURO Banknote		-Highly illuminated -Reduce color contrast -Applying live scenario of holding banknote
12	SGD Banknote		-Wording on banknote to create obstruction -Partially captured banknote -Highly illuminated
13	SGD Banknote		-Multiple denomination in a single frame -Highly illuminated -Obstruction by using finger to block -Arrow is used as foreign object

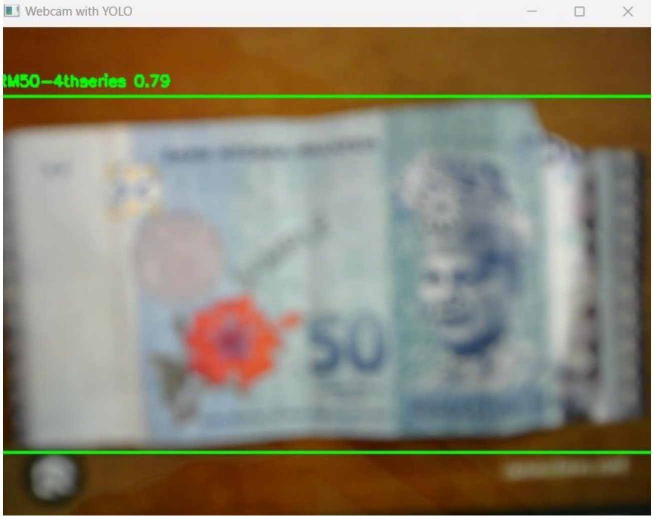
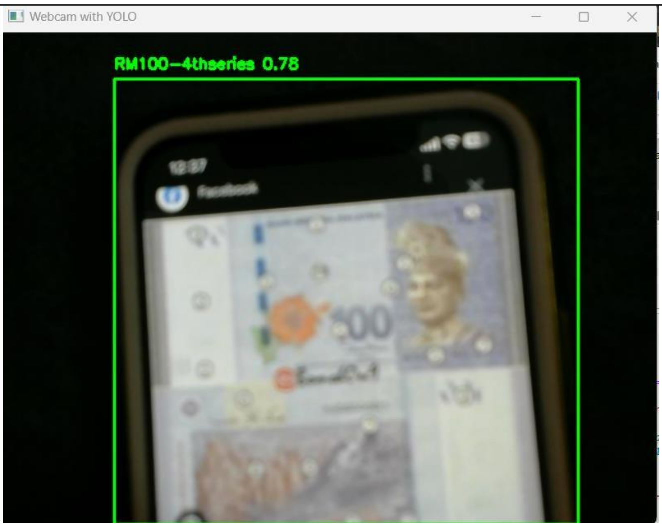
14	SGD Banknote		-Blur -Folded banknote -Finger as obstruction -Partially captured banknote
15	Baht Coins		-Highly illuminated -Multiple denomination in a single frame -Salt and pepper noise
16	Baht Banknote		-Folded banknote -Create obstruction by finger blocking important informations
17	Baht Banknote		-Different background -Folded banknote

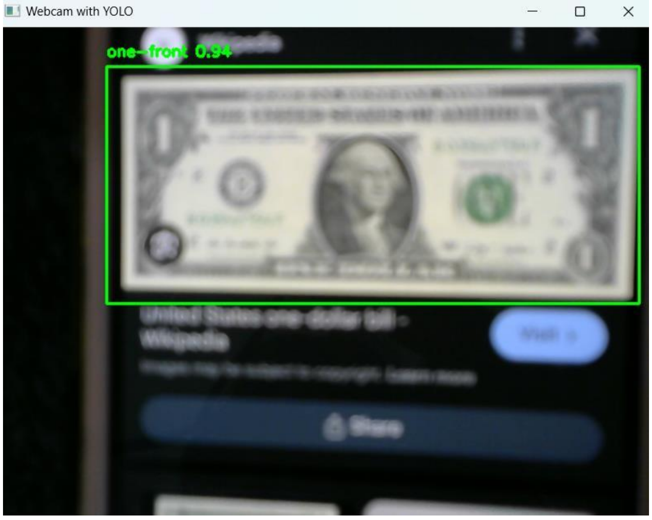
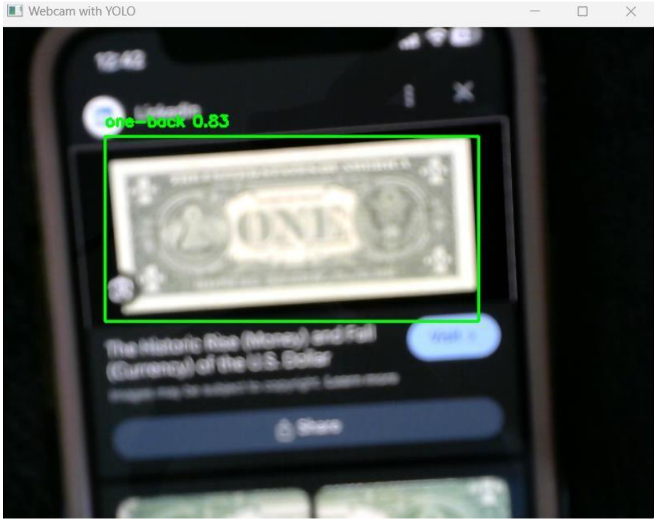
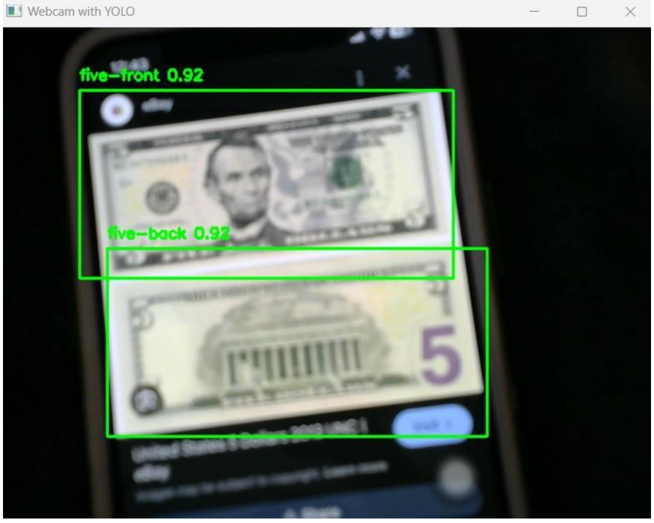
5.3 Test Case for all Currency Types

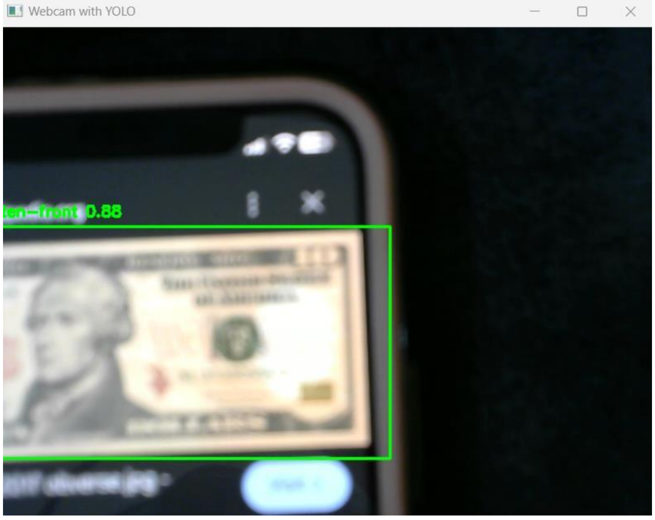
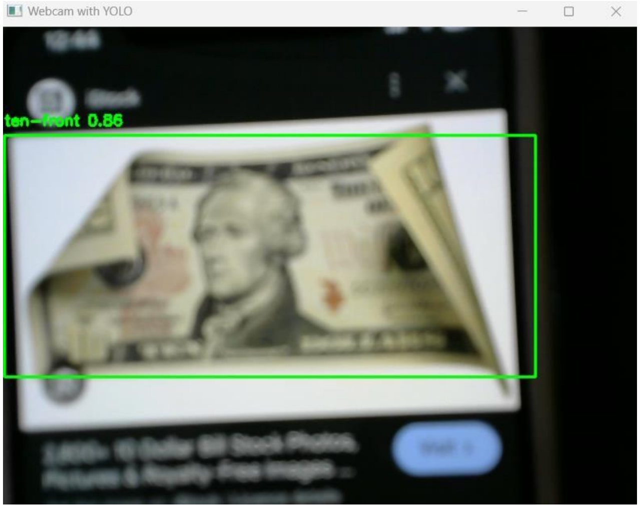
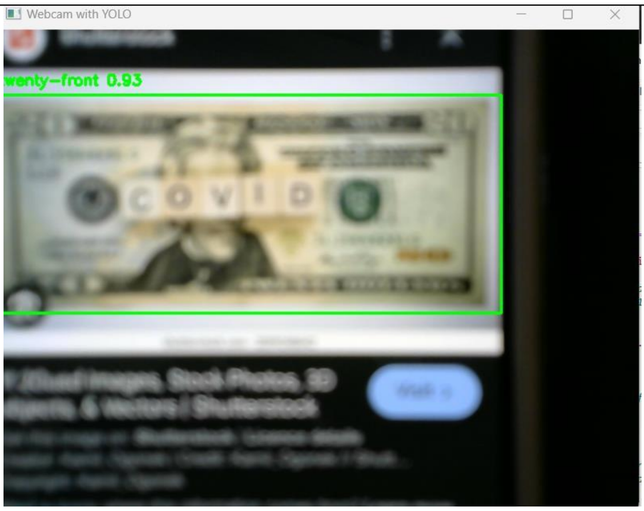
Table 5.2 Test Case for all Currency Types and Different Scenarios.

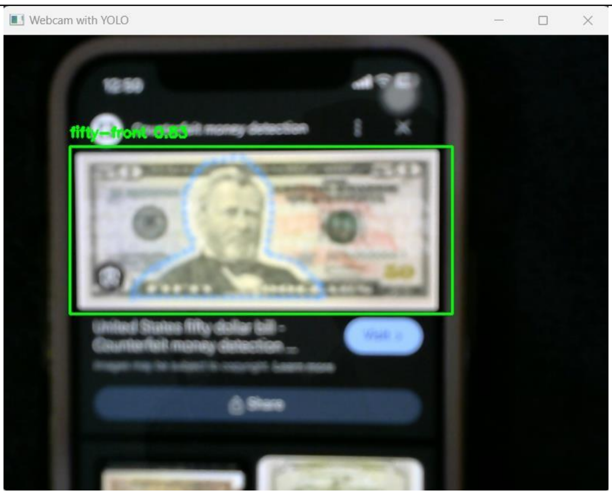
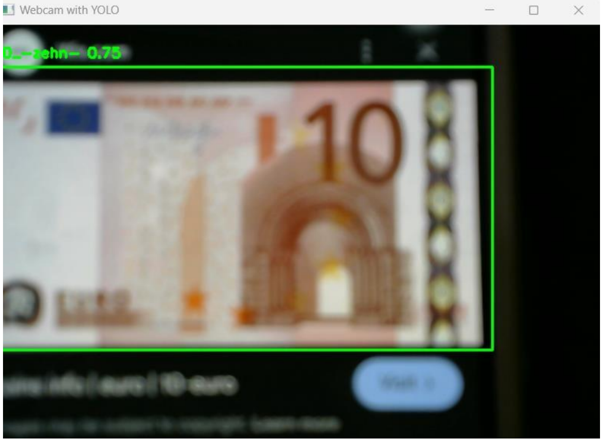
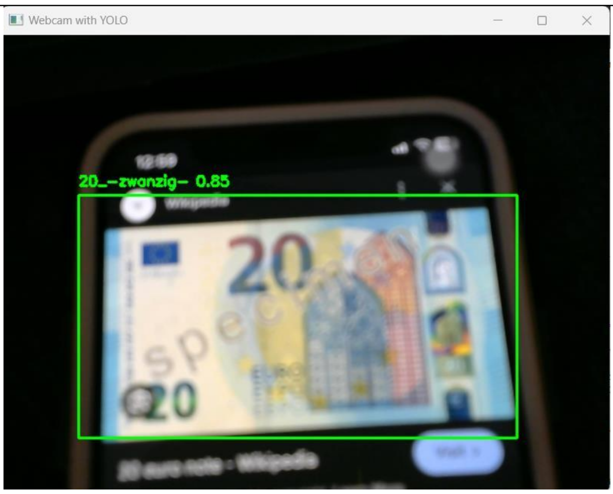
Currency Type	Currency Image	Discussion
RM1		-The front and back of RM1 can be detected
RM5		-Partially captured RM5 -Can be detected at low resolution or blur

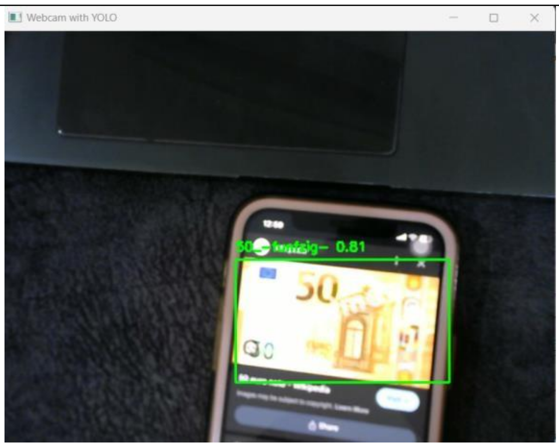
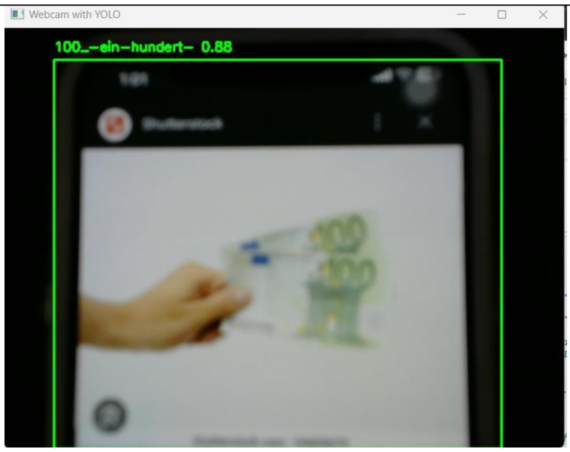
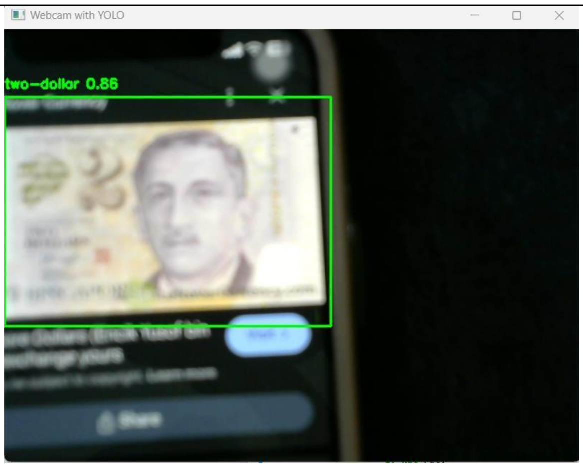
RM10	 	-CONTOH word acts as obstruction -Can be detected at low resolution or blur
RM20		-Partially captured RM20 -Can be detected at low resolution or blur

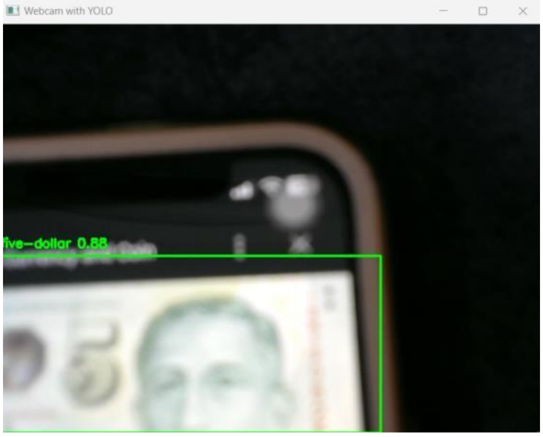
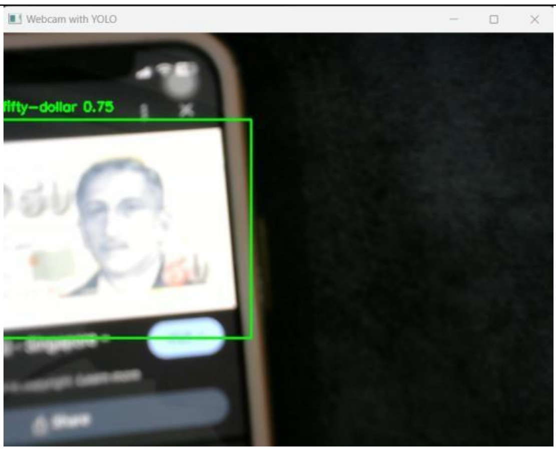
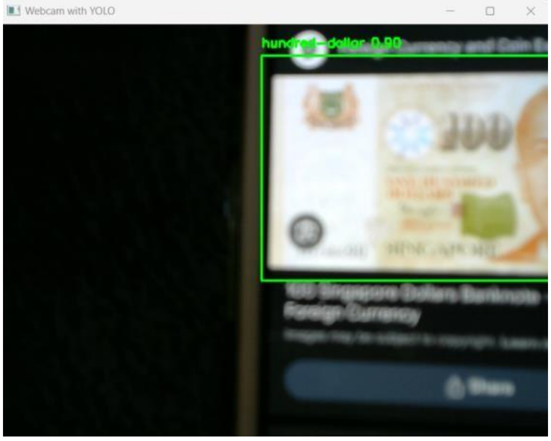
RM50	 	-Torn RM50 -Can be detected at low resolution or blur
RM100	 	-Back and front of RM100 can be detected at low resolution or blur

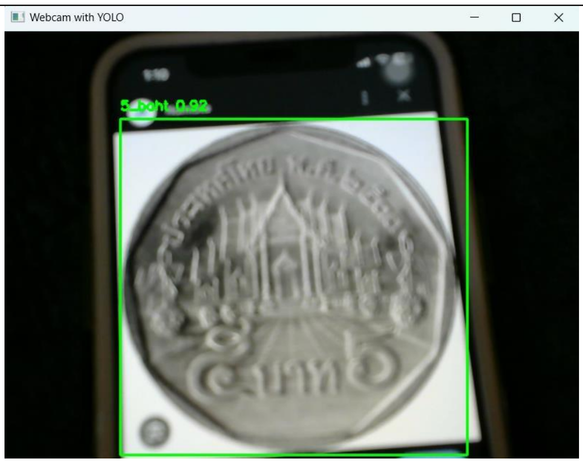
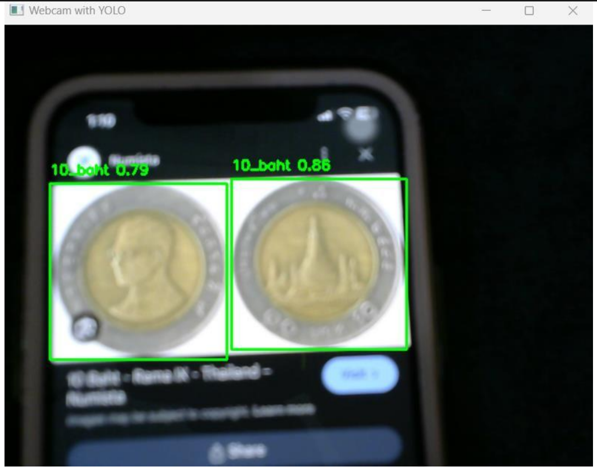
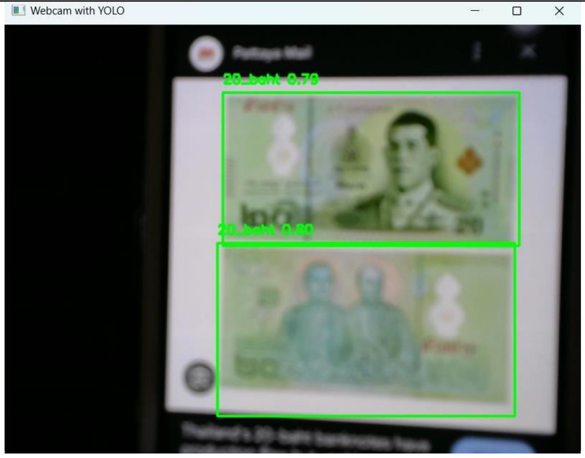
1USD front	 Webcam with YOLO one-front 0.94 United States one-dollar bill - Wikipedia  Webcam with YOLO 12:42 one-back 0.83 The Historic Rise (Money) and Fall (Currency) of the U.S. Dollar	-1USD front and back can be detected and have different class names
5USD		-5USD can be detected front and back with different class names -Can be detected at low resolution or blur

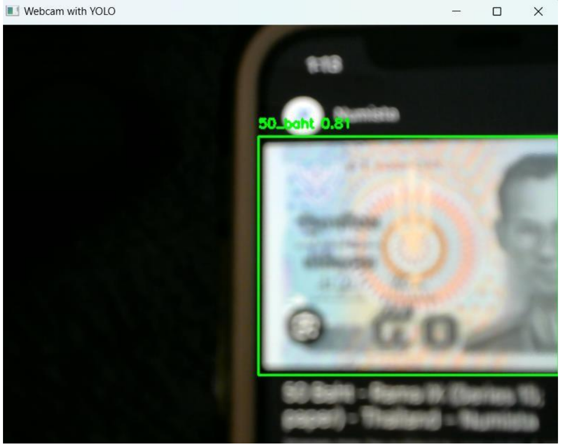
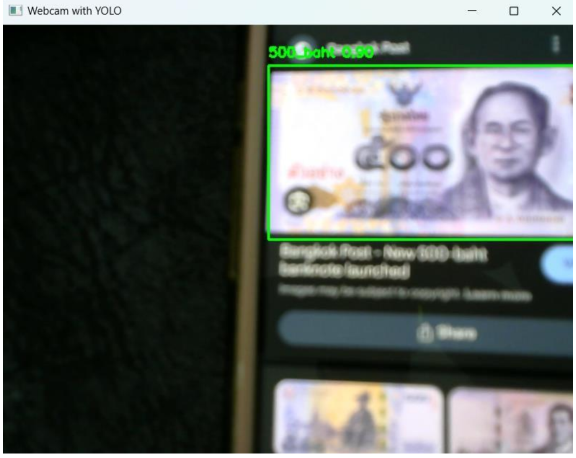
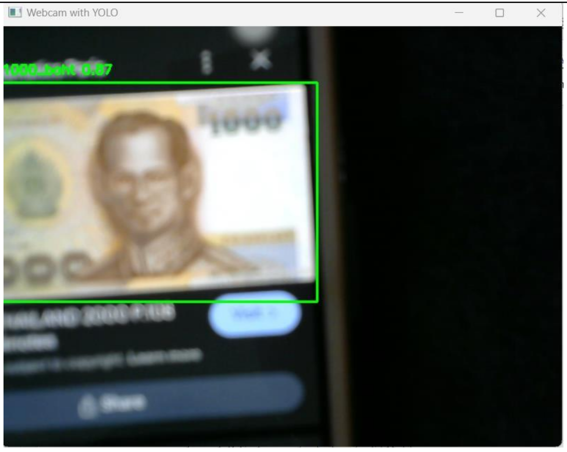
10USD	 	-Partially captured 10USD -Folded 10USD
20USD		-COVID word acts as obstruction -Can be detected at low resolution or blur

50USD		-Blue color shades added around the contents of 50USD banknote
10EURO		-Partially capture 10Euro -Can be detected at low resolution or blur
20EURO		-Specimen word acts as obstruction

50EURO		-Long distance detection
100EURO		-Holding multiple 100Euro banknotes
2SGD		-Partial captured of 2SGD -Can be detected at low resolution or blur

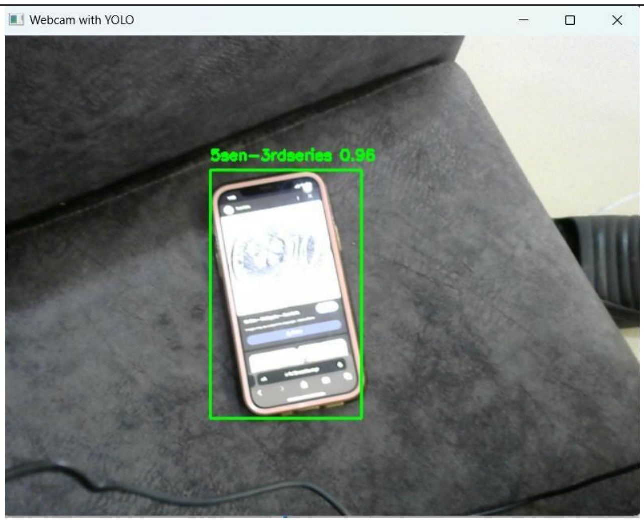
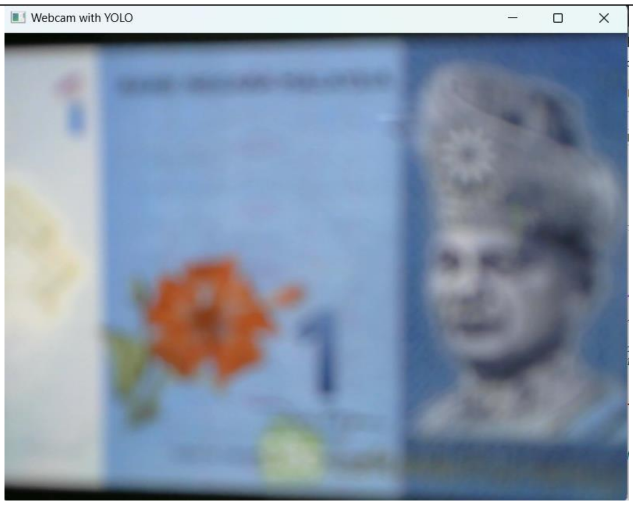
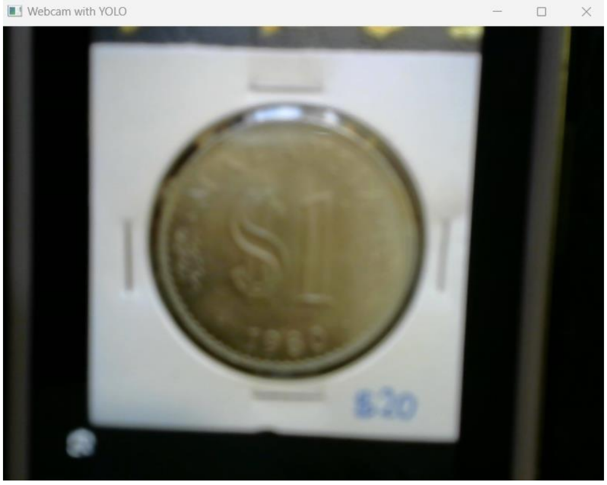
5SGD		-Caputring only a corner of 5USD
50SGD		-Highly illuminated scenario -Partial capturing of 50SGD
100SGD		-Partial caputring of 100SGD -Can be detected at low resolution or blur

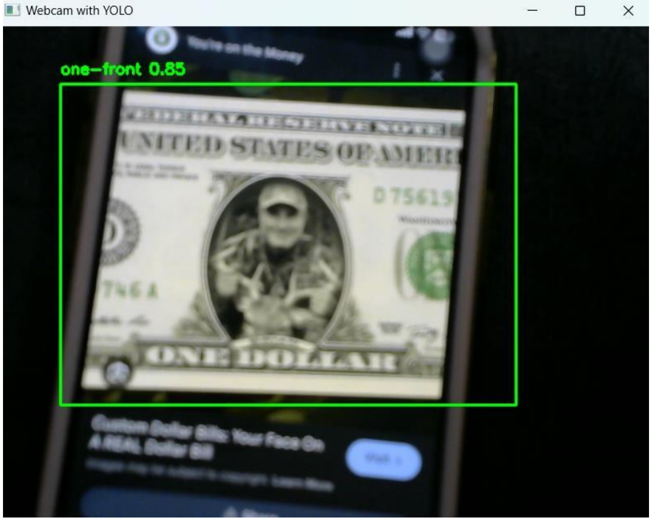
5baht		-inverted 5baht coin
10baht		-front and back of 10 baht coin
20baht		-front and back of 20baht banknote -Can be detected at low resolution or blur

50baht		-partially captured 50baht -Can be detected at low resolution or blur
500baht		-partially captured 500baht -Can be detected at low resolution or blur
1000baht		-partially capture 1000baht -Can be detected at low resolution or blur

5.4 Error Analysis

Table 5.3 Error Analysis.

Scenarios	Output	Discussion
Highly illuminated and capturing from a far distance		-Poor object detection due to the object being too small in the frame, leading to insufficient feature points for recognition
Distance of webcam and currency is too close		-When currency is too close to the webcam, image may become blurred or out of focus which negatively impacts the detection and recognition capabilities
Old currency		-The model is unfamiliar with older currency designs due to a lack of training data that includes these variations

Fake currency		-Fake currencies have subtle differences in design, paper quality or texture where the model is not trained for counterfiet recognition.

In this analysis of the Cash Reader system's performance, several key issues have been identified impacting its ability to accurately recognize and classify currency under varying conditions:

1. **Distance Issues**: The Cash Reader struggles with both very close and very distant currency captures. At too close a range, the image quality deteriorates due to blurring, whereas at too far a distance, the currency becomes too small for effective recognition. Ensuring proper distance from the camera and optimizing the system for different zoom levels are crucial for accurate detection.
2. **Old Currency Recognition**: The model's performance is compromised when encountering older versions of currency. This is due to a lack of exposure to these older designs during the training phase. Expanding the training dataset to include a diverse range of currency notes, including older versions, can enhance the system's recognition capabilities.
3. **Fake Currency**: The system also faces challenges in detecting and correctly classifying counterfeit currency. This is often due to the model not being trained on examples of fake currency or differences in the design features that may not match those of genuine notes. Incorporating data on counterfeit notes and implementing additional checks for authenticity could improve the system's robustness.

5.5 Classification Currency Comparison

This section will assess and contrast the performance of each currency using YOLOv8 model. The table beow will display the key metrics of each model on the testing dataset.

Table 5.4 Performance metrics for 5 currency using YOLOv8 model.

Currency	Precision	Confidence	F1-score	Recall
MYR	0.980	0.930	0.910	0.937
USD	0.948	1.000	0.970	0.993
EURO	0.965	0.961	0.920	0.990
SGD	0.961	0.965	0.920	0.990
BAHT	0.978	0.966	0.960	0.99

The table presents the performance metrics for the YOLOv8 model in detecting and recognizing five different currencies (MYR, USD, EURO, SGD, and BAHT). The key metrics shown are Precision, Confidence, F1-score, and Recall. Here's an evaluation of these results:

1. **Precision**:
   - Precision measures the accuracy of the model in identifying true positives (correct classifications) out of all predicted positives.
   - The precision values for all currencies are high, ranging from 0.948 (USD) to 0.980 (MYR). This indicates that the model is excellent at minimizing false positives, meaning it doesn't mistakenly classify one currency as another very often.
2. **Confidence**:
   - Confidence reflects the certainty of the model in its predictions. A value of 1.000 for USD suggests that the model is fully confident in predicting USD, while other currencies have slightly lower confidence levels. Despite the slight differences, confidence levels are generally high across the board.
3. **F1-score**:
   - The F1-score is the harmonic mean of Precision and Recall, providing a balance between them.
   - The F1-scores range from 0.910 (MYR) to 0.970 (USD), showing that the model performs well across all currencies, with the highest performance for USD.

4. **Recall**:
   - Recall measures how well the model identifies all true positives (correct classifications out of all actual positives).
   - The recall values are consistently high, with USD having the highest (0.993) and MYR the lowest (0.937). This indicates that the model has a strong ability to detect currencies when they are present, especially USD.

Overall Evaluation:

- **Model Strengths**: The model excels in detecting USD, with the highest scores across most metrics (especially confidence and recall). High precision, recall, and F1-scores for all currencies show that the model is effective at both detecting and correctly identifying currencies, making it reliable across a diverse set of currencies.
- **Areas for Improvement**: While the overall performance is impressive, MYR has the lowest F1-score (0.910) and recall (0.937). This suggests that the model may occasionally miss true positives when identifying MYR or could improve in terms of making more consistent predictions for this currency.

The table suggests that YOLOv8 is robust and accurate in this currency detection task, but slight improvements could be made for MYR to ensure balanced performance across all currencies.

5.6 Concluding Remark

To wrap up this chapter, the evaluation provides valuable insights into the Cash Reader capability. While each aspect demonstrated promising results, there are notable areas for improvement in improving the data annotation and labelling process as well as robustness of the predictive model.

Chapter 6
Conclusion And Recommendation

6.1 Conclusion

In conclusion, this project has successfully developed a Cash Reader system tailored specifically for visually impaired individuals, addressing a critical need for greater financial independence and accessibility. Leveraging the power of YOLOv8 for real-time currency recognition, the system is capable of accurately detecting and classifying various denominations across multiple currencies, including Malaysian Ringgit, USD, Euro, SGD, and Thai Baht. The integration of state-of-the-art deep learning techniques enables rapid and reliable identification, making the system suitable for real-world use.

However, the project faced challenges in certain areas, particularly in detecting smaller objects like coins, where the system struggled with precision due to the reduced spatial resolution and minimal visual differences between denominations. Another limitation was the system's inability to detect counterfeit currency. Since the current model focuses solely on identifying legitimate notes based on their visual features, additional work is needed to incorporate security features like watermarks or holograms that could help in distinguishing counterfeit notes.

A key strength of this project lies in the use of Roboflow for data annotation, which significantly improved the quality and consistency of the training data. The precision in labeling and bounding box creation helped the YOLOv8 model to learn more effectively, leading to improved detection accuracy. By including a wide variety of conditions, such as different angles, lighting, and backgrounds, the dataset used for training enhanced the system’s ability to generalize across diverse real-world scenarios.

Overall, the Cash Reader system demonstrates significant potential to empower visually impaired individuals by enabling them to independently recognize and handle currency. While further improvements are necessary to address current limitations, the project has laid a solid foundation for future work that could focus on expanding the model’s capabilities, improving coin detection, and integrating counterfeit detection features for enhanced security and reliability.

6.2 Future Work

Future work will prioritize enhancing the detection of small objects, such as coins, by optimizing the YOLOv8 model's architecture, potentially incorporating multi-scale feature extraction to better capture fine details. Additionally, the use of higher-resolution images and more targeted data collection will be explored to improve the system's accuracy in recognizing smaller denominations, which have subtle visual differences. Techniques such as advanced data augmentation and training the model with specialized datasets focused on small objects can also be implemented to further enhance the model’s robustness and performance in real-world scenarios.

REFERENCES

[1] L. Dandona and R. Dandona, “What is the global burden of visual impairment?,” BMC Med., vol. 4, Mar. 2006, doi: 10.1186/1741-7015-4-6.

[2] “Dr Noor Hisham: Prevalence of blindness is 1.2% of country’s population,” The Star. [Online]. Available: https://www.thestar.com.my/news/nation/2022/10/13/dr-noor-hisham-prevalence-of-blindness-is-12-of-country039s-population. [Accessed: Sep. 09, 2023].

[3] “Eye health is crucial for overall wellness,” [Online]. Available: https://www.nst.com.my/lifestyle/heal/2022/05/794334/eye-health-crucial-overall-wellness. [Accessed: Sep. 09, 2023].

[4] C. M. Correia Da Costa, “Multiview banknote recognition with component and shape analysis,” ScalABLE 4.0 View project Robotic Bin Picking of Entangled Tubes View project,” doi: 10.13140/RG.2.1.2254.3443.

[5] Y. Mitsukura, M. Fukumi, and N. Akamatsu, “Design and Evaluation of Neural Networks for Coin Recognition by Using GA and SA,” 2000.

[6] A. Ali and M. Manzoor, “Recognition system for Pakistani paper currency,” World Appl. Sci. J., vol. 28, no. 12, pp. 2069–2075, 2013, doi: 10.5829/idosi.wasj.2013.28.12.300.

[7] “Blind and Partially Sighted,” Bank of Canada. [Online]. Available: https://www.bankofcanada.ca/banknotes/audience-specific-resources/blind-and-partially-sighted/. [Accessed: Sep. 09, 2023].

[8] “Euro banknote tactile features,” European Central Bank. [Online]. Available: https://www.ecb.europa.eu/euro/banknotes/tactile/html/index.en.html. [Accessed: Sep. 09, 2023].

[9] “Tactile Features for Banknotes,” United States Currency Education Program. [Online]. Available: https://www.uscurrency.gov/learn/tactile-features-banknotes. [Accessed: Sep. 09, 2023].

[10] “Malaysia's tactile features on banknotes,” Bank Negara Malaysia. [Online]. Available: https://www.bnm.gov.my/tactile-features-on-banknotes. [Accessed: Sep. 09, 2023].

[11] A. CS, D. M, H. V. R, and A. Anju, “Currency Recognition For The Visually Impaired People,” in 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 2022, pp. 1-3, doi: 10.1109/DELCON54057.2022.9753373.

[12] A. A. S. S. Bharkad, “Survey of Currency Recognition System Using Image Processing,” International Journal of Computational Engineering Research (IJCER), pp. 36, 2013.

[13] “Ultralytics Documentation,” [Online]. Available: https://docs.ultralytics.com/. [Accessed: Sep. 09, 2023].

[14] S. Karshiev, S. Bang, S. Ryue, and H. Jung, “Real-Time Object Detection and Face Recognition Application for the Visually Impaired,” Computers, Materials and Continua, vol. 79, no. 3, pp. 3569-3583, 2024, doi: 10.32604/cmc.2024.048312. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1546221824000055. [Accessed: Sep. 09, 2023].

[15] Sagar Dev Achar, C. Shankar Singh, CS Sumanth Rao, K. Pavana Narayana, Ashwini Dasare, “Indian Currency Recognition System Using CNN And Comparison With YOLOv5,” IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 3-4 December 2021. DOI: 10.1109/ICMNWC52512.2021.9688513.

[16] I. Abu Doush and S. AL-Btoush, ‘Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms’, Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 4, pp. 484–492, Oct. 2017, doi: 10.1016/j.jksuci.2016.06.003

[17] H. Hassanpour and P. M. Farahabadi, “Using Hidden Markov Models for paper currency recognition,” Expert Systems with Applications, vol. 36, no. 6, pp. 10105–10111, Aug. 2009, doi: 10.1016/j.eswa.2009.01.057.

[18] M. Sinthuja, S. Pravinthraja, S. R. ManiSekhar, and V. Karthik, “Currency Recognition App for Blind People,” in Human-Centric Smart Computing, ICHCSC 2023, S. Bhattacharyya, J. S. Banerjee, and M. Köppen, Eds., vol. 376, Singapore: Springer, 2024, doi: 10.1007/978-981-99-7711-6_6.

[19] R. C. Joshi, S. Yadav, and M. K. Dutta, “YOLO-v3 Based Currency Detection and Recognition System for Visually Impaired Persons,” in 2020 International Conference on Contemporary Computing and Applications (IC3A), Lucknow, India, 2020, pp. 280-285, doi: 10.1109/IC3A48958.2020.233314.

[20] R. Maganti, J. Gutla, Y. Mallimpalli, and S. Yellisetti, “Innovative Currency Identifier for the Blind through Audio Output using Deep Learning,” in 2024 International Conference on Automation and Computation (AUTOCOM), Dehradun, India, 2024, pp. 358-363, doi: 10.1109/AUTOCOM60220.2024.10486196.

[21] M. Nahid, M. R. Islam, P. K. Roy, and R. Runti, "A Feature Based Classifier for Bangla Currency Using Deep Learning," in 2024 6th International Conference on Electrical

Engineering and Information & Communication Technology (ICEEICT), Dhaka, Bangladesh, 2024, pp. 1413-1417, doi: 10.1109/ICEEICT62016.2024.10534568.

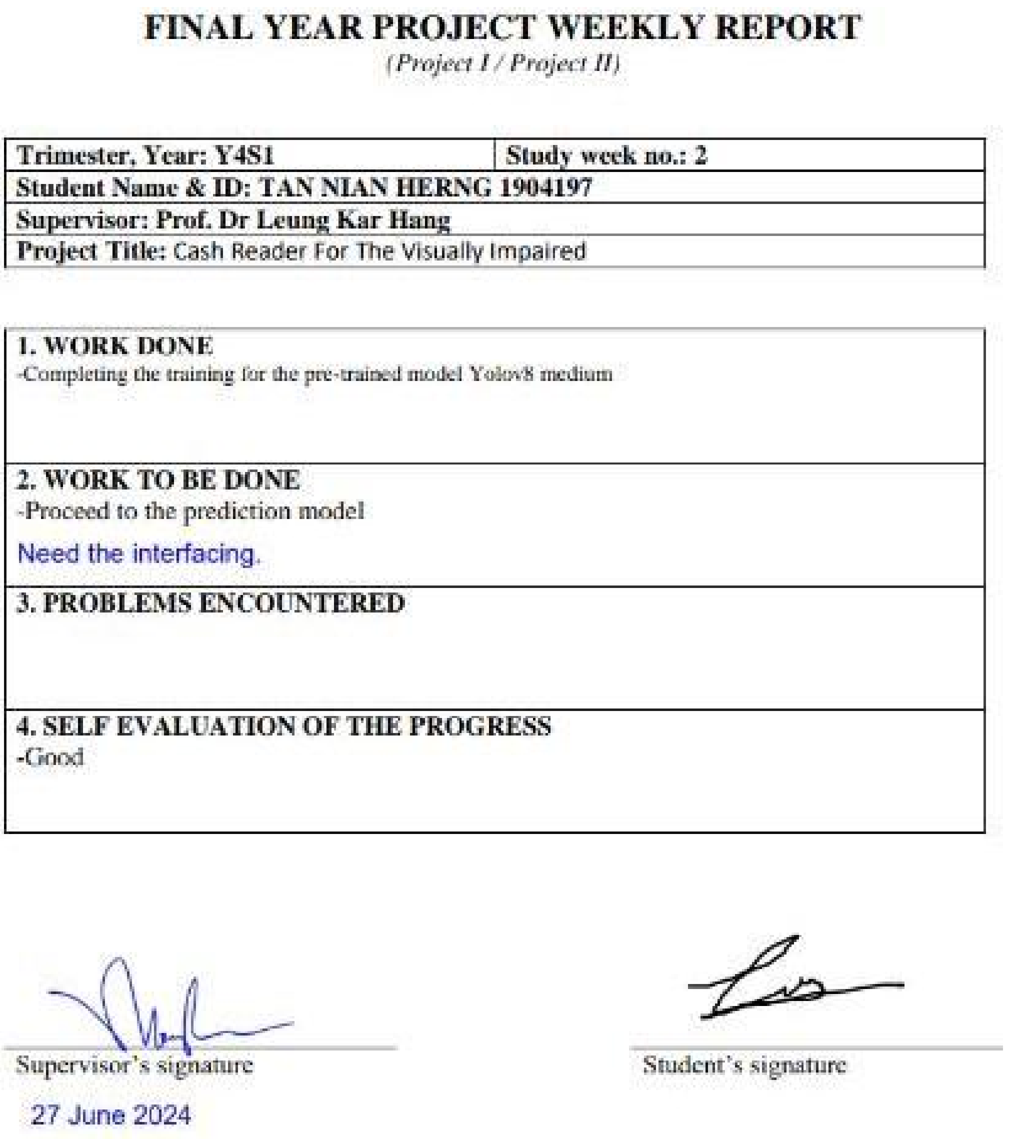
[22] S. W. Ambler, "UML Use-Case Diagrams," The Elements of UMLTM 2.0 Style, pp. 33–46, Jul. 2005, doi: 10.1017/CBO9780511817533.005

APPENDICES

```
!git clone https://github.com/ultralytics/ultralytics
!pip install -qe ultralytics
%cd ultralytics
!pip install -U ultralytics
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="xWdumS8fiuro0iJtsTrE")
project = rf.workspace("currencies").project("malaysia-banknotes-and-coins")
version = project.version(10)
dataset = version.download("yolov8")
import torch
from torch.utils.data import DataLoader
from torchvision import transforms, datasets
import torch.nn as nn
import os
from ultralytics import YOLO

# Define paths
data_root = 'C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/Malaysia-Banknotes-and-Coins-10'
data_yaml = os.path.join(data_root, 'data.yaml')

# Check if directories exist
train_dir = os.path.join(data_root,  'train/images')
valid_dir = os.path.join(data_root, 'valid/images')
print(f"Train directory exists: {os.path.exists(train_dir)}")
print(f"Validation directory exists: {os.path.exists(valid_dir)}")

# Print contents of the directories
print("Train Directory Contents:", os.listdir(train_dir))
print("Validation Directory Contents:", os.listdir(valid_dir))
```

```
# Load the pretrained YOLOv8 model
model = YOLO("C:/Users/nianh/yolo_v8normal/yolov8n.pt")
from ultralytics import YOLO
# Train the model using the data.yaml file
model.train(
    data=data_yaml,
    epochs=30,
    imgsz=640,
    batch=16,
    lr0=0.001,  # Initial learning rate
)

# Validate on the validation set
metrics = model.val(data=data_yaml)

# Print metrics
print(metrics)
# Define paths
data_root = 'C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/Malaysia-Banknotes-and-Coins-10'
data_yaml = os.path.join(data_root, 'data.yaml')
model_weights = "C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/runs/detect/train8/weights/best.pt"

# Load the pretrained YOLOv8 model for validation
model = YOLO(model_weights)

# Validate the model on the validation set
metrics = model.val(data=data_yaml)

# Print metrics
print(metrics)
```

```
import os
import glob
from ultralytics import YOLO
from IPython.display import Image, display

# Define paths
data_root = 'C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/Malaysia-Banknotes-and-Coins-10'
data_yaml = os.path.join(data_root, 'data.yaml')
model_weights = "C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/runs/detect/train8/weights/best.pt"
image_dir = os.path.join(data_root, 'val', 'images')
save_dir = "C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and Coins.yolov8/prediction
result"

# Load the pretrained YOLOv8 model
model = YOLO(model_weights)

# Make sure the save directory exists
os.makedirs(save_dir, exist_ok=True)

# Iterate over all images in the test directory
image_paths = glob.glob(os.path.join(image_dir, '*'))
for image_path in image_paths:
    # Ensure the file is an image
    if image_path.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp', '.gif')):
        # Debug: Print the image being processed
        print(f"Processing image: {image_path}")

        # Make predictions on the image
        results = model.predict(source=image_path)

        # Save each result separately
        for i, result in enumerate(results):
            save_path = os.path.join(save_dir,
f"{os.path.basename(image_path).split('.')[0]}_{i}.jpg")
            result.save(filename=save_path)

            # Print and display the original and annotated images
            print(f"Saved predictions for {os.path.basename(image_path)} in {save_path}")
            display(Image(filename=image_path))
            display(Image(filename=save_path))
    else:
        print(f"Skipped non-image file: {image_path}")

print("Processing complete.")
import cv2
from ultralytics import YOLO

# Function to display live webcam feed with object detection
def display_webcam_with_yolo(conf_threshold=0.75):
    # Initialize YOLO model
    model = YOLO(r"C:/Users/nianh/yolo_v8normal/Malaysia Banknotes and
Coins.yolov8/ultralytics/runs/detect/train8/weights/best.pt")

    # Change the index based on your external webcam's position
    cap = cv2.VideoCapture(1)  # Use index 1 or 2, etc., depending on your setup

    if not cap.isOpened():
        print("Error: Could not open webcam.")
        return

    while True:
        ret, frame = cap.read() # Capture frame-by-frame
        if not ret:
            print("Error: Failed to capture frame.")
            break

        # Perform object detection on the frame
        results = model(frame)  # model.predict() is called internally

        # Iterate through each detection
        for result in results:
            boxes = result.boxes
            for box in boxes:
                xmin, ymin, xmax, ymax = map(int, box.xyxy[0])
                conf, cls = box.conf[0], int(box.cls[0])

                if conf >= conf_threshold:
                    # Draw bounding box and label on the frame
                    cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)
                    label = f'{model.names[cls]} {conf:.2f}'
                    cv2.putText(frame, label, (xmin, ymin - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 255, 0), 2)

        cv2.imshow('Webcam with YOLO', frame)  # Display the frame with YOLO detection

        if cv2.waitKey(1) == 27: # Exit on ESC key press
            break

    cap.release() # Release the capture
    cv2.destroyAllWindows()  # Close all OpenCV windows

# Call the function to display live webcam feed with YOLO object detection
display_webcam_with_yolo()
```



FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y4S1	**Study week no.: 2**
Student Name & ID: TAN NIAN HERNG 1904197	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Cash Reader For The Visually Impaired	

1. WORK DONE

-Completing the training for the pre-trained model Yolov8 medium

2. WORK TO BE DONE

-Proceed to the prediction model

Need the interfacing.

3. PROBLEMS ENCOUNTERED

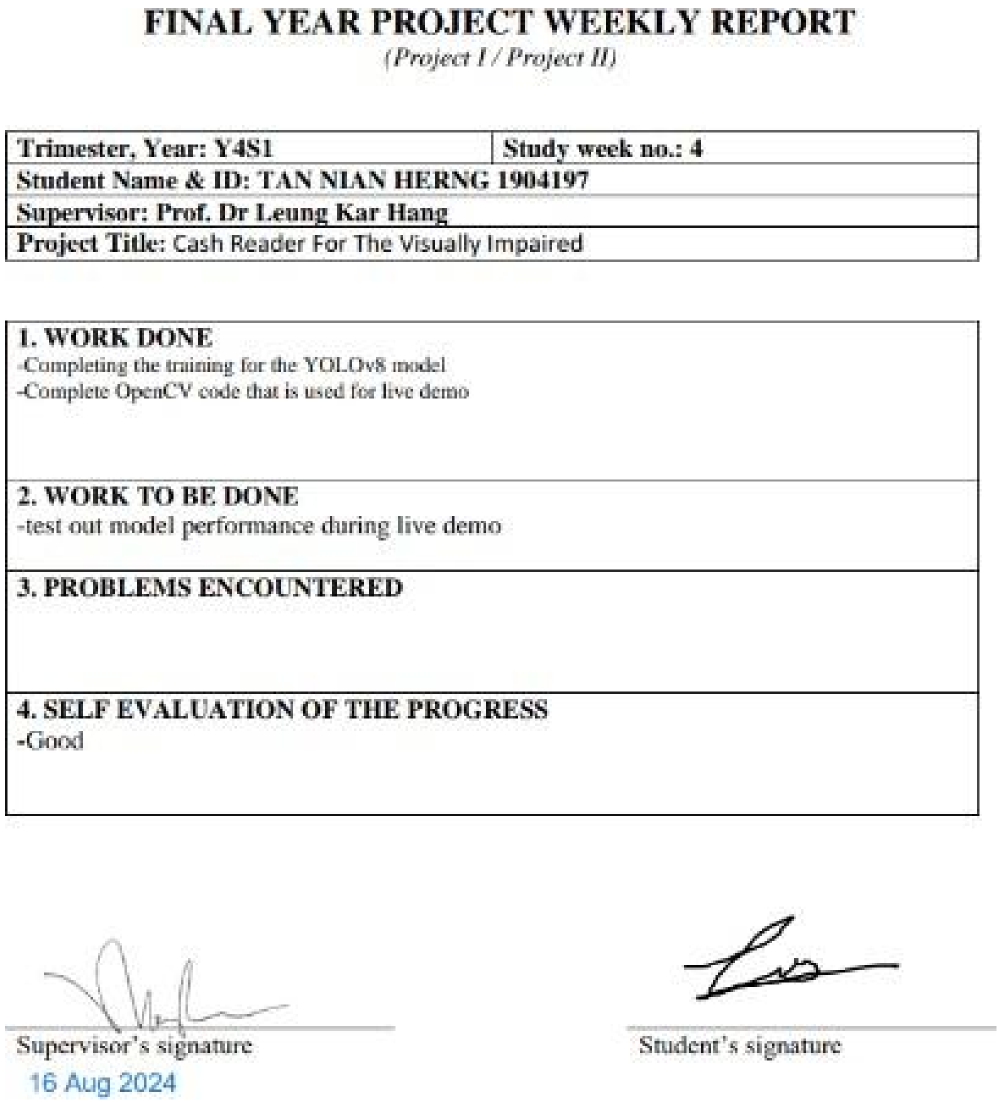
4. SELF EVALUATION OF THE PROGRESS

-Good

Supervisor's signature

27 June 2024

Student's signature



FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y4S1	**Study week no.: 4**
Student Name & ID: TAN NIAN HERNG 1904197	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Cash Reader For The Visually Impaired	

1. WORK DONE

-Completing the training for the YOLOv8 model
-Complete OpenCV code that is used for live demo

2. WORK TO BE DONE

-test out model performance during live demo

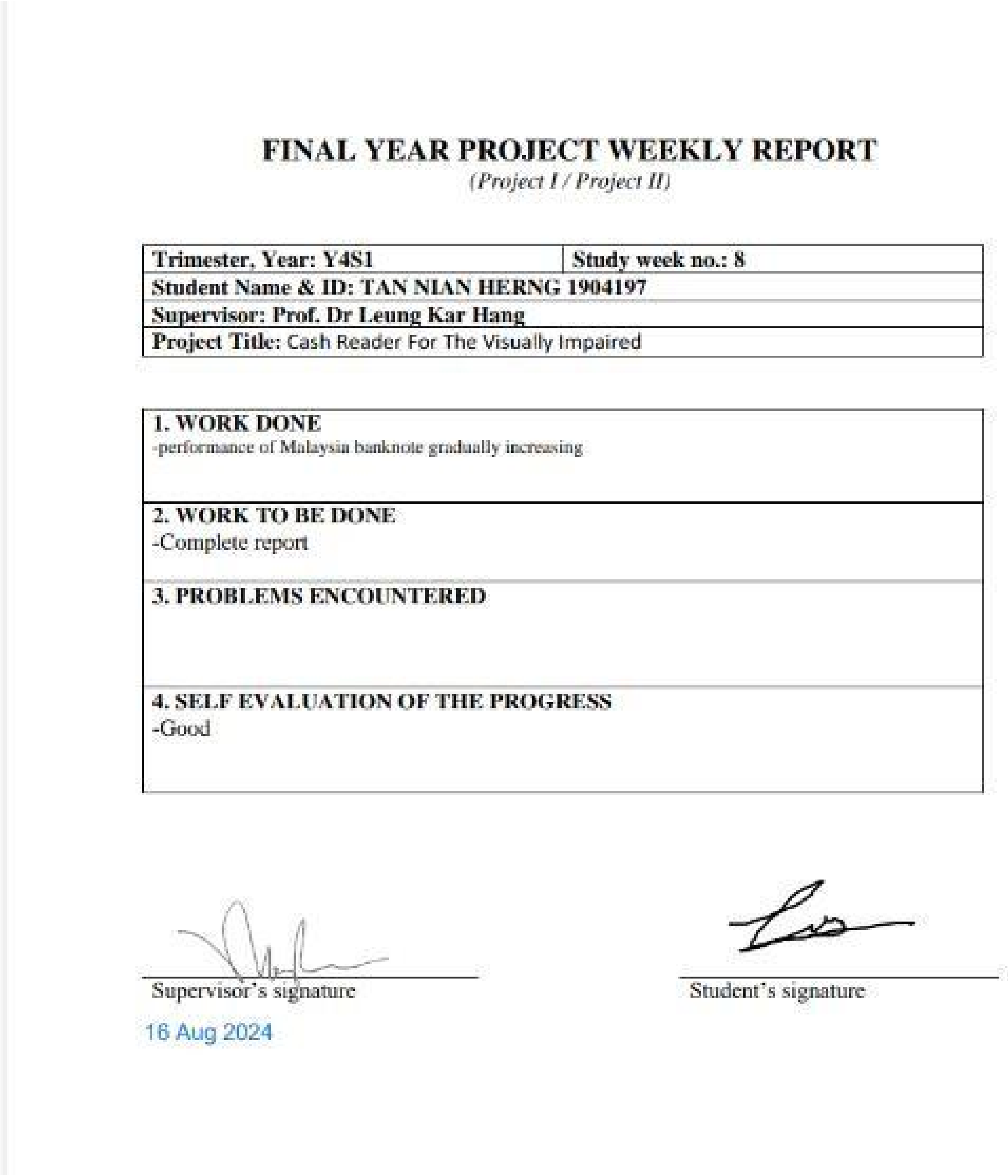
3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

-Good

Supervisor's signature
16 Aug 2024

Student's signature



FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y4S1	**Study week no.: 8**
Student Name & ID: TAN NIAN HERNG 1904197	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Cash Reader For The Visually Impaired	

1. WORK DONE

-performance of Malaysia banknote gradually increasing

2. WORK TO BE DONE

-Complete report

3. PROBLEMS ENCOUNTERED

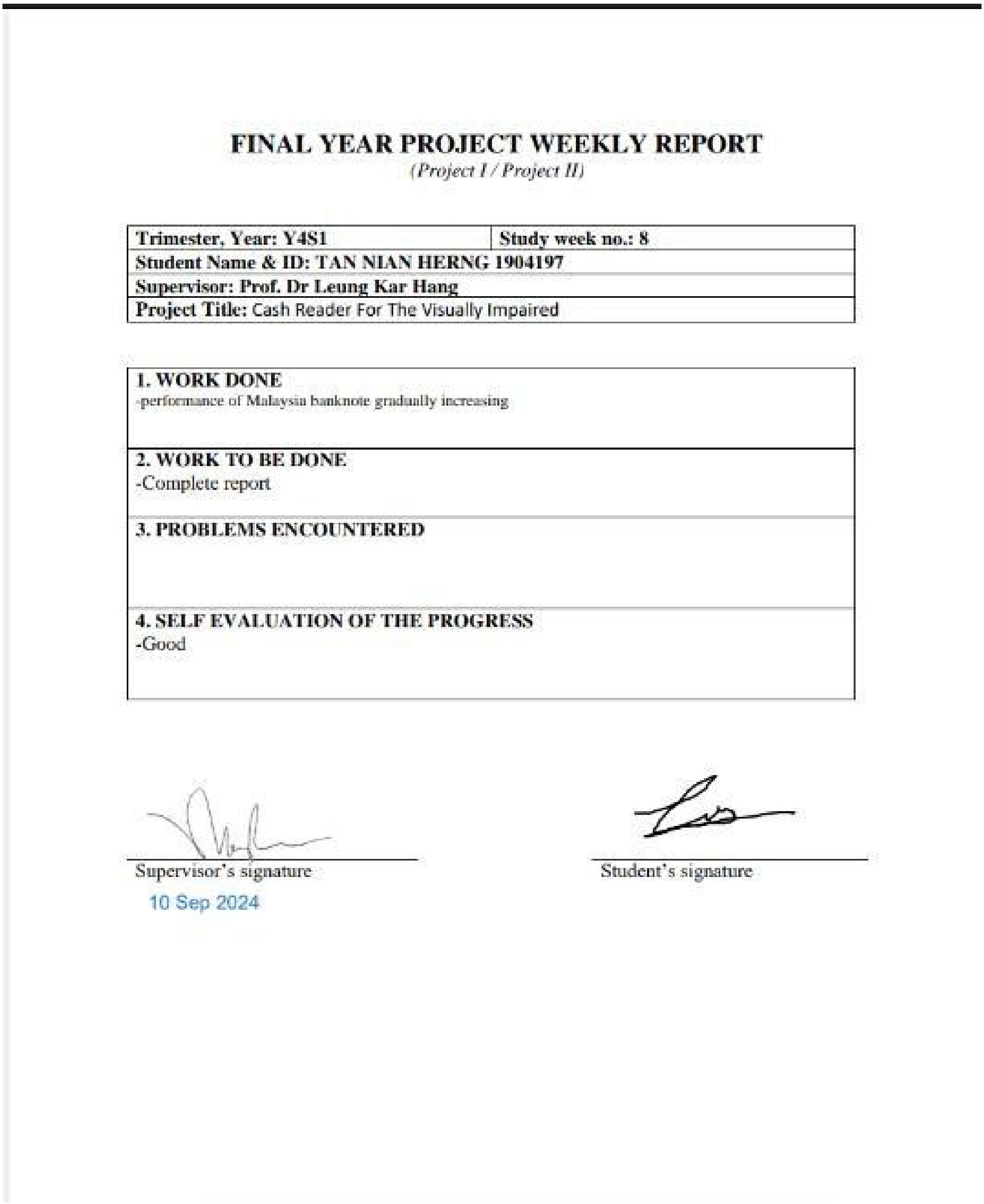
4. SELF EVALUATION OF THE PROGRESS

-Good

Supervisor's signature

16 Aug 2024

Student's signature



FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y4S1	**Study week no.: 8**
Student Name & ID: TAN NIAN HERNG 1904197	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Cash Reader For The Visually Impaired	

1. WORK DONE

-performance of Malaysia banknote gradually increasing

2. WORK TO BE DONE

-Complete report

3. PROBLEMS ENCOUNTERED

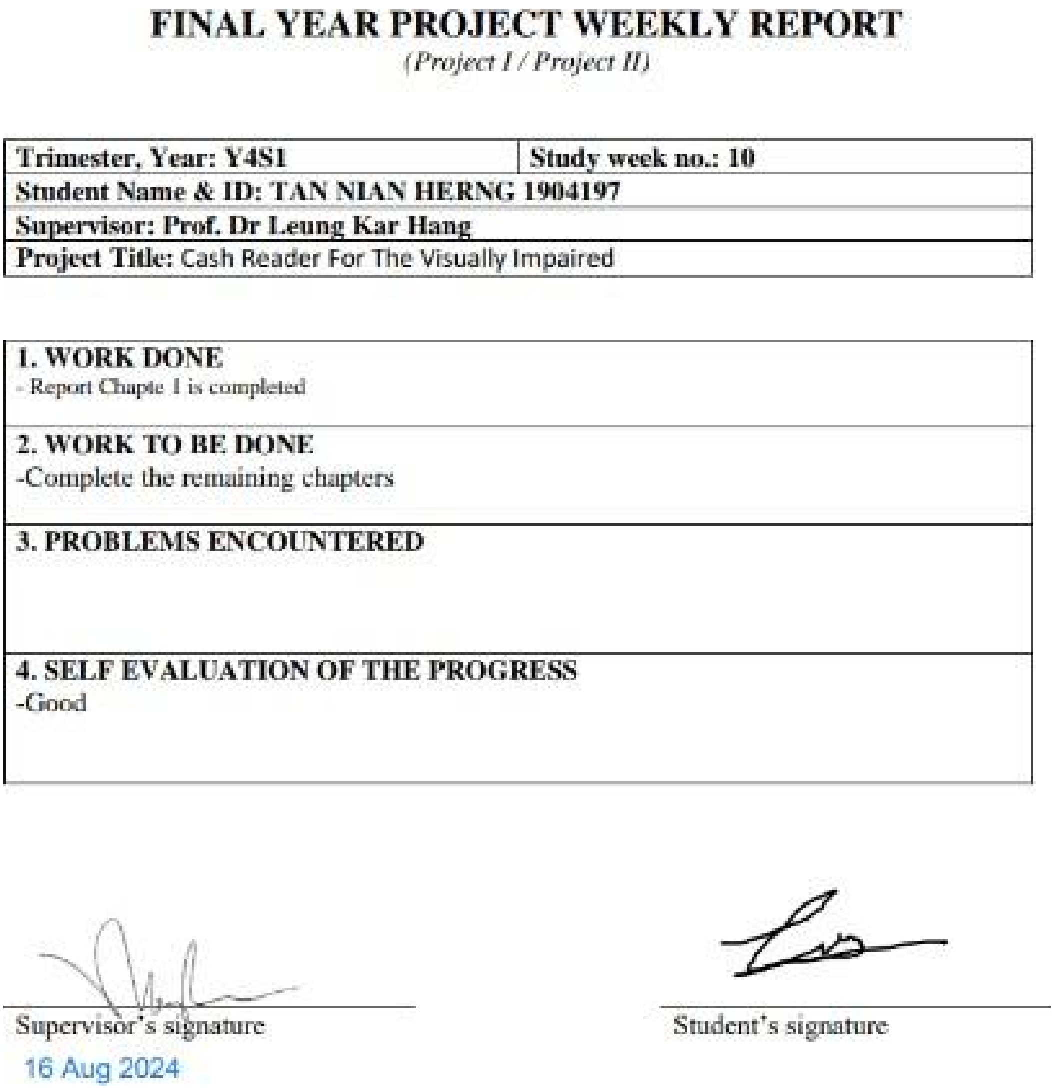
4. SELF EVALUATION OF THE PROGRESS

-Good

Supervisor's signature

10 Sep 2024

Student's signature



FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y4S1	**Study week no.:** 10
Student Name & ID: TAN NIAN HERNG 1904197	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Cash Reader For The Visually Impaired	

1. WORK DONE

- Report Chapte 1 is completed

2. WORK TO BE DONE

-Complete the remaining chapters

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

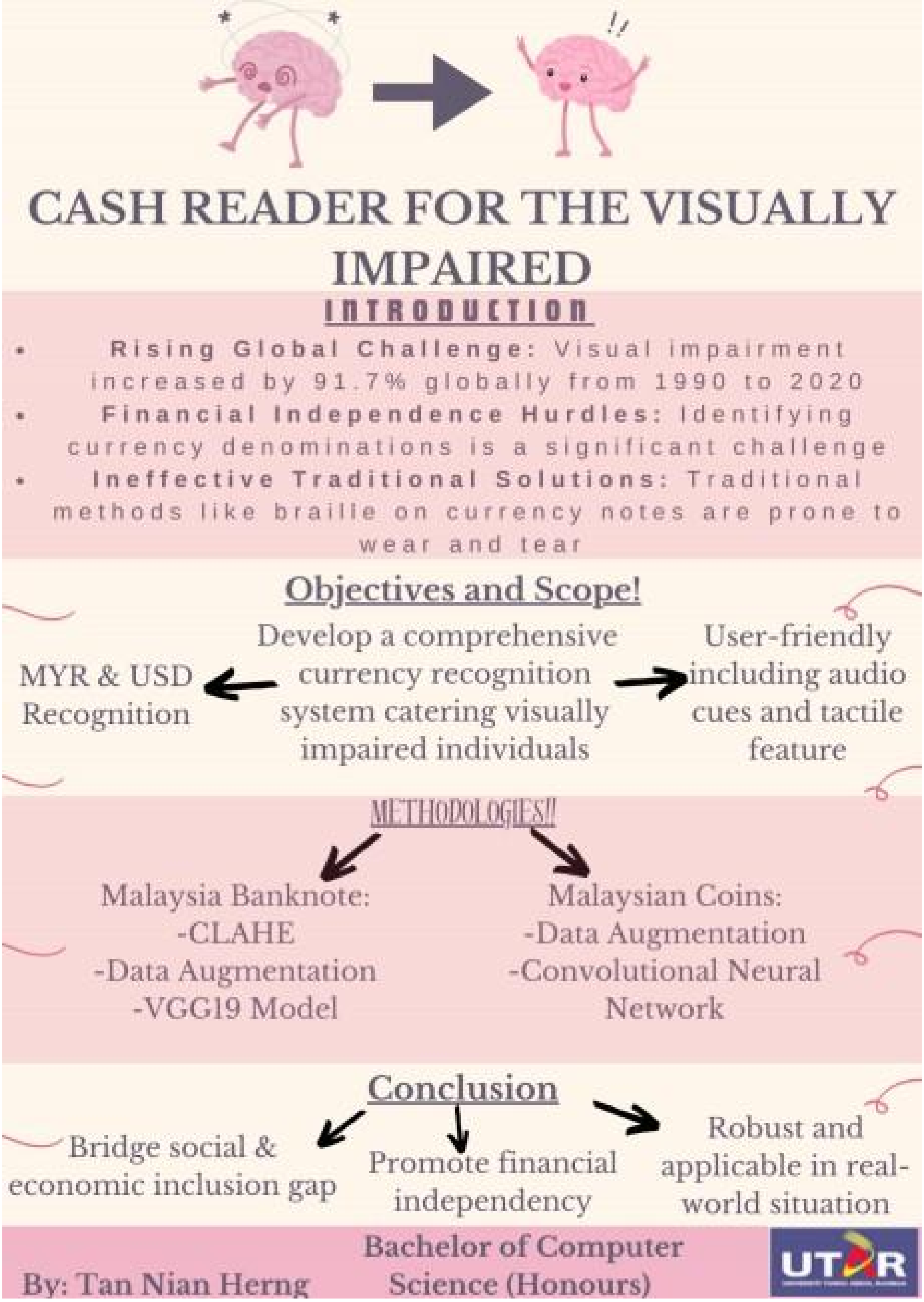
-Good

Supervisor's signature

16 Aug 2024

Student's signature

POSTER



CASH READER FOR THE VISUALLY IMPAIRED

INTRODUCTION

- **Rising Global Challenge:** Visual impairment increased by 91.7% globally from 1990 to 2020
- **Financial Independence Hurdles:** Identifying currency denominations is a significant challenge
- **Ineffective Traditional Solutions:** Traditional methods like braille on currency notes are prone to wear and tear

Objectives and Scope!

MYR & USD Recognition ← Develop a comprehensive currency recognition system catering visually impaired individuals → User-friendly including audio cues and tactile feature

METHODOLOGIES!!

Malaysia Banknote:
-CLAHE
-Data Augmentation
-VGG19 Model

Malaysian Coins:
-Data Augmentation
-Convolutional Neural Network

Conclusion

Bridge social & economic inclusion gap

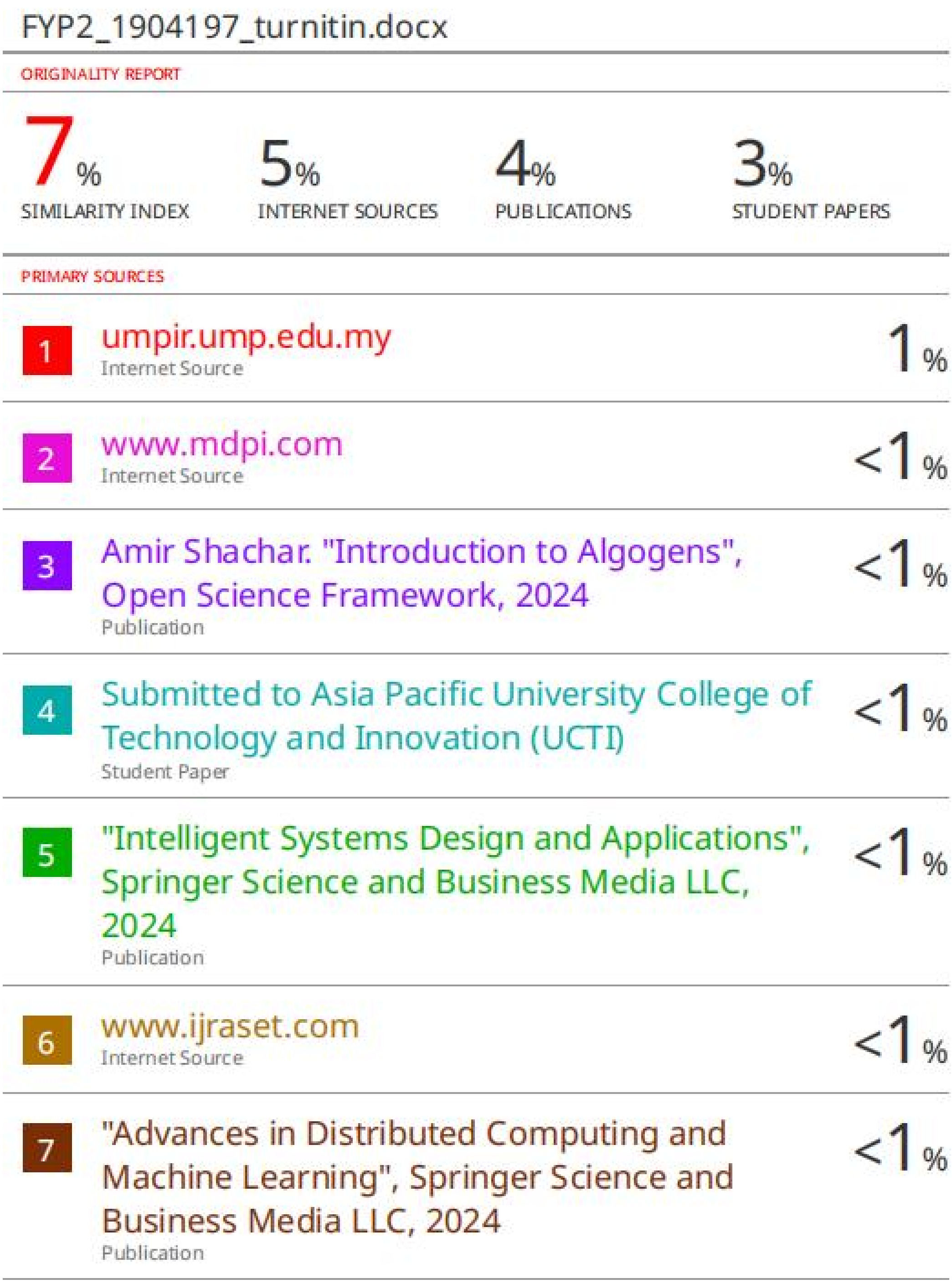
Promote financial independency

Robust and applicable in real-world situation

By: Tan Nian Herng

Bachelor of Computer Science (Honours)

PLAGIARISM CHECK RESULT



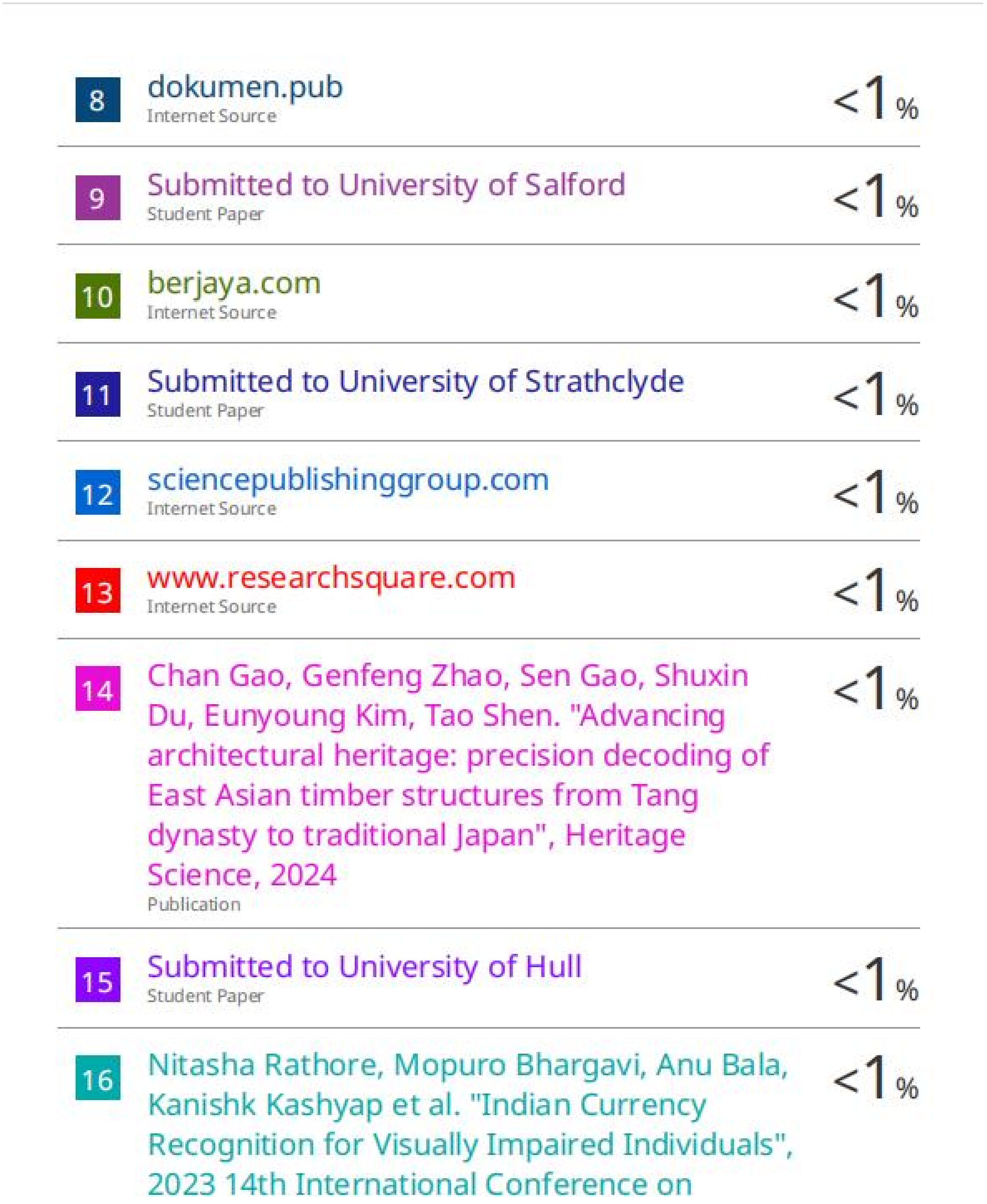
FYP2_1904197_turnitin.docx

ORIGINALITY REPORT

7%	5%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	umpir.ump.edu.my Internet Source	1%
2	www.mdpi.com Internet Source	<1%
3	Amir Shachar. "Introduction to Algogens", Open Science Framework, 2024 Publication	<1%
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
5	"Intelligent Systems Design and Applications", Springer Science and Business Media LLC, 2024 Publication	<1%
6	www.ijraset.com Internet Source	<1%
7	"Advances in Distributed Computing and Machine Learning", Springer Science and Business Media LLC, 2024 Publication	<1%



8	dokumen.pub Internet Source	<1 %
9	Submitted to University of Salford Student Paper	<1 %
10	berjaya.com Internet Source	<1 %
11	Submitted to University of Strathclyde Student Paper	<1 %
12	sciencepublishinggroup.com Internet Source	<1 %
13	www.researchsquare.com Internet Source	<1 %
14	Chan Gao, Genfeng Zhao, Sen Gao, Shuxin Du, Eunyoung Kim, Tao Shen. "Advancing architectural heritage: precision decoding of East Asian timber structures from Tang dynasty to traditional Japan", Heritage Science, 2024 Publication	<1 %
15	Submitted to University of Hull Student Paper	<1 %
16	Nitasha Rathore, Mopuro Bhargavi, Anu Bala, Kanishk Kashyap et al. "Indian Currency Recognition for Visually Impaired Individuals", 2023 14th International Conference on	<1 %

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	TAN NIAN HERNG
ID Number(s)	19ACB04197
Programme / Course	CS / UCCC3596 PROJECT II
Title of Final Year Project	CASH READER FOR THE VISUALLY IMPAIRED

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 7 **%** **Similarity by source** Internet Sources: 5 % Publications: 4 % Student Papers: 3 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Leung Kar Hang

Date: 12 Sep 2024

Signature of Co-Supervisor

Name:

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB04197
Student Name	TAN NIAN HERNG
Supervisor Name	Prof. Dr Leung Kar Hang

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
NA	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 10/9/2024