

**DEVELOPMENT AND EVALUATION OF A BANKNOTE READER FOR
VISUALLY IMPAIRED PEOPLE USING DEEP LEARNING-BASED OBJECT
DETECTION**

By
CHAN HONG WAI

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)

JUNE 2023

REPORT STATUS DECLARATION FORM

Title: _____Development and Evaluation of a Banknote Reader For Visually_
_____Impaired People using Deep Learning-Based Object Detection____

Academic Session: ___June 2023___

I _____CHAN HONG WAI_____
(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.



(Author's signature)

Verified by,

Verified by,



(Supervisor's signature)

Address:

___24, Lorong Bertam Indah 11_

___Taman Bertam Indah, 13200___

___Kepala Batas, Penang_____

Ng Hui Fuang

Supervisor's name

Date: ___14th September 2023___

Date: 15/09/2023

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 OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN**

Date: __14th September 2023_____

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that *Chan Hong Wai* (ID No: *19ACB03752*) has completed this final year project entitled “*Development and Evaluation of a Banknote Reader for Visually Impaired People using Deep-Learning Based Object Detection*” under the supervision of *Dr. Ng Hui Fuang* (Supervisor) from the Department of *Computer Science, Faculty of Information and Communication Technology*.

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

Yours truly,




(*Chan Hong Wai*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**DEVELOPMENT AND EVALUATION OF A BANKNOTE READER FOR VISUALLY IMPAIRED PEOPLE USING DEEP LEARNING-BASED OBJECT DETECTION**” 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 : CHAN HONG WAI

Date : 14th September 2023

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my sincerest gratitude to my project supervisor, Dr. Ng Hui Fuang, for his invaluable guidance and support throughout the project. Dr. Ng expertise and knowledge have been instrumental in the successful completion of this project.

I am also thankful to my family and friends for their unwavering support and encouragement, which has been a constant source of inspiration for me. Finally, I would like to express my appreciation to all those who have contributed to this project in one way or another, whether through their feedback or assistance. Their support and contribution have been integral to the development of this project, and I am deeply grateful for their help.

ABSTRACT

Automated currency recognition systems have numerous applications, including banknote counting machines, currency exchange machines, and systems to assist visually impaired individuals. The inability to differentiate between different currencies can lead to financial exploitation of visually impaired individuals, making the need for a reliable currency recognition system even more demanding. In this project, we propose a mobile system for currency recognition that can recognize the Malaysian banknotes and sense of different denominations using deep learning techniques. This project introduces a system using the YOLOv8 architecture for feature extraction and classification, designed to benefit the visually impaired community. The project is structured into two primary phases: the training phase, which encompasses model training and exportation, and the development phase, including model integration and the deployment of the user-friendly application, CashVisionV2. The application serves a dual-system output, providing both text and voice outputs to ensure accessibility. The primary objective of this initiative is to enhance the financial independence of visually impaired individuals by simplifying their day-to-day transactions, particularly in the recognition of various Malaysian banknote denominations.

TABLE OF CONTENTS

TITLE PAGE	I
REPORT STATUS DECLARATION FORM	II
SUBMISSION OF FINAL YEAR PROJECT	III
DECLARATION OF ORIGINALITY	IV
ACKNOWLEDGEMENTS	V
ABSTRACT.....	VI
TABLE OF CONTENTS	VII
LIST OF FIGURES	XI
LIST OF TABLES	XIV
LIST OF ABBREVIATIONS	XV
CHAPTER 1: INTRODUCTION.....	1
1.1 Background Information.....	1
1.2 Problem Statement and Motivation	3
1.2.1 Overpriced existing banknote readers for visually impaired	3
1.2.2 Limited functionality, accuracy and portability for some existing solutions	3
1.3 Motivation.....	4
1.4 Project Scopes.....	5
1.5 Project Objectives	6

1.6	Contributions.....	7
1.7	Report Organization.....	7
CHAPTER 2: LITERATURE REVIEWS		9
2.1	Previous works on banknote reader model	9
2.1.1	Currency Recognition System for Blind people using ORB Algorithm [7].....	9
2.1.2	Image Based Ringgit Banknote Recognition for Visually Impaired ...	12
2.1.3	Malaysian Banknote Reader For Visually Impaired Person [9]	18
2.1.4	Currency Recognition for Visually Impairment People Using Deep Learning Algorithm [10].....	21
2.1.5	Currency Recognition for Visually Impairment People Using Deep Learning Algorithm [11].....	24
2.2	Critical Remarks of Previous Works	30
CHAPTER 3: METHODOLOGY		32
3.1	Methodologies.....	32
3.1.1	Dataset Acquisition.....	32
3.1.2	Dataset Pre-Processing and Data Augmentation	33
3.1.3	Model Training and Testing.....	33
3.1.4	Deployment of trained model into Flutter app.....	34
3.2	Tools	34
3.2.1	Hardware.....	34
3.2.2	Software & Services	35
3.3	Timeline	37

CHAPTER 4: SYSTEM MODEL	39
4.1 Deep Learning-Based Model	39
4.1.1 MobileNetV2 Architecture	39
4.1.2 You Only Look Once V8 (YOLOv8) Architecture	43
4.2 Model Selection	48
CHAPTER 5: SYSTEM DESIGN.....	49
5.1 System Block Diagram	49
5.2 Model Conversion & Inference Package	50
5.3 Input Package.....	51
5.4 Output Package	52
5.5 Use Case Diagram.....	54
5.6 Activity Diagram	56
CHAPTER 6: SYSTEM IMPLEMENTATION	58
6.1 Software Setup	58
6.2 Implementation of Cash Reader Model	58
6.2.1 Datasets Acquisition	59
6.2.2 Data Pre-processing & Augmentation	63
6.2.3 Training of YOLOv8-cla Model	65
6.3 Development of Cash Detection App under Flutter framework.....	70
6.3.1 Model Pre-Setup	70
6.3.2 Model Inference	73
6.3.3 Key Features	74
6.3.4 Deployment of cash reader Application (CashVisionV2)	76
6.4 Implementation Issues and Challenges.....	77

CHAPTER 7: SYSTEM EVALUATION	79
7.1 Performance Metrics	79
7.1.1 Accuracy Score	79
7.1.2 Precision & Recall Score	79
7.1.3 F1 Score	80
7.1.4 Confusion Matrix	81
7.2 Evaluation Results of the Proposed Model	82
7.3 Evaluation Results of the Proposed Application System.....	86
7.4 Objective Evaluation.....	87
CHAPTER 8: CONCLUSION.....	89
8.1 Conclusion	89
8.2 Contributions.....	90
8.3 Future Works	91
REFERENCES.....	92
APPENDIX.....	A-1
POSTER.....	A-10
PLAGIARISM CHECK RESULT	A-11
FYP 2 CHECKLIST	A-13

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	handled cash reader proposed in [4]	4
Figure 1.2	Project Scope Overview	5
Figure 2.1	System Block Diagram of the proposed method 2.1.1	10
Figure 2.2	Equation of Gaussian Blur Effect applied	10
Figure 2.3	Equation of Otsu's Thresholding technique applied	11
Figure 2.4	Pixel Pair Sampling Techniques	11
Figure 2.5	Hamming Distance Measure Equation	12
Figure 2.6	Matching Process using Hamming Distance	12
Figure 2.7	System Block Diagram for current proposed method	13
Figure 2.8	Informative Region and whole region captured	14
Figure 2.9	Main features RB, RG, and RB	15
Figure 2.10	Graph on Finding Optimum DTC (Info Region)	15
Figure 2.11	Graph on Finding Optimum DTC (Whole Region)	16
Figure 2.12	Graph on Finding Optimum DTC (Combined Region) Optimized DTC with 3 levels after minimum leaf size = 21 (Combination Region) applied	16
Figure 2.13	Confusion Matrix for DTC using the whole region	17
Figure 2.14	achieving accuracy of 99.7%	18
Figure 2.15	Architecture Design of the proposed method	19
Figure 2.16	Equation of Discrete Cosine Transformation (DCT)	20
Figure 2.17	Equation of Ideal Percentage P	20
Figure 2.18	System Block Diagram	21
Figure 2.19	The Indian Rupees Dataset	22
Figure 2.20	System Design Proposed in [11]	24
Figure 2.21	The New Zealand Dollars Dataset	25

Figure 2.22	Manual marking using quadrilateral	26
Figure 2.23	Data Augmentation Steps	26
Figure 2.24	SSD (Single Shot Multibox Detector) Model	28
Figure 2.25	LeakyReLU activation function	28
Figure 2.26	Representation of FC Layer	29
Figure 2.27	Loss function	29
Figure 3.1	Gantt chart of project I timeline	38
Figure 3.2	Gantt chart of project II timeline	38
Figure 4.1	MobileNetV2 Model Architecture	39
Figure 4.2	Inverted Bottleneck Residual Layer Architecture	41
Figure 4.3	CSPDarknet53 Architecture	43
Figure 4.4	Conv Layer Architecture	44
Figure 4.5	SiLU Activation Function	44
Figure 4.6	C2f Layer Architecture	45
Figure 4.7	Decoupled Head Architecture	46
Figure 4.8	Spatial Pyramid Pooling Fast (SPPF) Layer Architecture	47
Figure 5.1	System Block Diagram of the development phase	49
	Class Diagram for Inference Package & Model	
Figure 5.2	Conversion	50
Figure 5.3	Class Diagram for the Input package	51
Figure 5.4	Class Diagram for the Output package	52
Figure 5.5	Use Case Diagram of the proposed design	54
Figure 5.6	Activity Diagram	56
Figure 6.1	Types and Samples of image data in the captured dataset	60
Figure 6.2	Branch folders of Captured dataset	60
Figure 6.3	Sample images (RM notes) of the Captured dataset	62
Figure 6.4	Sample images (sen) of the Captured dataset	63
Figure 6.5	Example of Data Augmentation	64
Figure 6.6	Overview of YOLOv8-cls Model	65

Figure 6.7	Overview of YOLOv8-cls Model	66
Figure 6.8	Layer Freezing of the first 5 layers (0-4)	67
Figure 6.9	Samples of 1st batch training	69
Figure 6.10	Structure of the assets folder	70
Figure 6.11	loadModel() Function	70
Figure 6.12	loadCamera() Function	71
Figure 6.13	captureAndRunModel() Function	72
Figure 6.14	runModelOnPhoto() Function	73
Figure 6.15	_getTextColor() Function	74
Figure 6.16	Sample UI of a few real-time predictions	76
Figure 6.17	Github Repo of CashVisionV2	77
Figure 7.1	Formula equation to calculate Accuracy score	79
Figure 7.2	Formula equation to calculate Precision score	79
Figure 7.3	Formula equation to calculate Recall score	80
Figure 7.4	Formula equation to calculate F1 Score	80
Figure 7.5	Confusion Matrix	81
Figure 7.6	Graph plot of train and val loss during training	83
Figure 7.7	Graph plot of val accuracy during training	83
Figure 7.8	Confusion Matrix of YOLOv8-cls model testing	83
	Confusion Matrix of YOLOv8-cls model testing	85
Figure 7.9	(Normalized)	
	Samples of actual and predictions of the proposed	85
Figure 7.10	methodology	

LIST OF TABLES

Table Number	Title	Page
Table 1.1	Examples of banknotes reader models and respective prices range	3
Table 1.2	Reporting Structure	8
Table 2.1	Table of Brief Comparison between each previous study	30
Table 3.1	Desktop Specification	35
Table 3.2	Mobile Device Specification	35
Table 6.1	Training Params Configuration	68
Table 7.1	Data achieve during training and validation of YOLOv8-cls model (20epochs)	82
Table 7.2	Performance of YOLOv8-cls Model on Test Set	84
Table 7.3	Test Case for CashVisionV2 Home Screen	86
Table 7.4	Test Case for CashVisionV2 Settings Screen	87

LIST OF ABBREVIATIONS

<i>WHO</i>	World Health Organization
<i>UN</i>	United Nations
<i>CNN</i>	Convolutional Neural Network
<i>YOLO</i>	You Only Look Once
<i>ResNet</i>	Residual Network
<i>MBA</i>	Malaysian Blind Association
<i>ROI</i>	Region of Interest
<i>ORB</i>	Oriented FAST and BRIEF
<i>RM</i>	Ringgit Malaysia
<i>RGB</i>	RedGreenBlue
<i>DTC</i>	Decision Tree Classifier
<i>k-NN</i>	k-Nearest Neighbours
<i>CMOS</i>	Complementary Metal-Oxide-Semiconductor
<i>CCD</i>	Charge-Coupled Device
<i>UV</i>	Ultraviolet
<i>DCT</i>	Discrete Cosine Transformation
<i>NZD</i>	New Zealand Dollar
<i>GUI</i>	Graphical User Interface
<i>SSD</i>	Single Shot Multibox Detector
<i>DNN</i>	Deep Neural Network
<i>MLP</i>	Multilayer Perceptron
<i>TFLite</i>	TensorflowLite
<i>API</i>	Application Programming Interface

<i>ReLU</i>	Rectified Linear Unit
<i>SiLU</i>	Sigmoid Linear Unit
<i>BN</i>	Batch Normalization
<i>CIoU</i>	Complete Intersection over Union
<i>DFL</i>	Distribution Focal Loss
<i>SPPF</i>	Spatial Pyramid Pooling Fast
<i>ONNX</i>	Open Neural Network Exchange
<i>TTS</i>	TextToSpeech
<i>FC</i>	Fully Connected
<i>c2f</i>	Convolutional to Flatten
<i>GPU</i>	Graphical Processing Unit

CHAPTER 1: INTRODUCTION

1.1 Background Information

Visual impairment is a broad term that refers to a range of conditions that affect a person's ability to see through our “oculus uterque”. Visual impairment can arise from a variety of causes, ranging from genetic disorders, eye diseases, injuries, and aging. It is a significant public health concern, with an estimated 285 million individuals worldwide experiencing some form of visual impairment, with 39 million of them being completely blind, according to the World Health Organization (WHO). Visual impairment can manifest in different ways, ranging from mild to severe loss of vision, which could be leading to blindness in the end. Individuals with visual impairments may encounter difficulties with some basic, yet essential tasks such as reading, writing, and navigating themselves around their environment independently. Plus, visual impairment can impose a significant negative impact on a person's quality of life, affecting them from the aspect of independence, social interactions, and employment opportunities indirectly.

Among these challenges, identifying of the denominations of banknotes can be particularly difficult, if the user does not have access to a reliable source or tool. In normal living, they desperately need the ability to recognize money so that some basic activities like buying groceries, obtaining food, could be performed, else they would not be able to live in current world independently where almost every aspect of life involves monetary element. Unfortunately, there are high chances for them of being cheated by dishonest sellers who take advantage of their lack of visual perception. For example, a seller may charge a visually impaired customer a higher price for a product than they would charge a sighted customer for the same item, when they couldn't recognize the value of the currency they're giving despite of their impairment. It is almost impossible for them to differentiate the denomination only based on the size, texture of the banknotes, since almost every bank notes are identical in terms of both textures and sizes [1]. These types of scams and fraudulent activities are not only unethical, but they can also bring serious consequences for visually impaired individuals, who may be more

CHAPTER 1

vulnerable to these types of schemes. They need a mechanism to protect themselves from such frauds, where they have the right to preserve their own assets, in terms of monetary cash.

Fortunately, there arises many resources and interventions to assist these groups with visual impairments to address this issue. These include banknotes with tactile features, audio assistance, dedicated devices and etcetera. These solutions do, however, have their own drawbacks, such as the requirement for certain hardware, accessibility issues, or the requirement for specialised training. Therefore, a better approach would be adopting deep learning techniques integrated with the use of mobile phone, where the requirement level to entry would be much lesser.

In recent years, deep learning models have achieved outstanding results in object recognition tasks, including banknote detection, in recent years. These methods extract useful properties from images using convolutional neural networks (CNNs), and then utilized those features to identify things. The use of deep learning models for banknote recognition has been investigated in some of the research recently. For example, [2] proposed a deep learning model based on the YOLOv3 architecture that successfully recognised banknotes with a high accuracy rate of 95.71%. Another one would be, [3] that proposed a Convolutional Neural Network (CNN) model based on the Inception-ResNet-v2 architecture for banknote classification and achieved an accuracy of 97.5%.

In this paper, Banknote Reader for the Visually Impaired is proposed, which is a integrated deep learning based model. On a bespoke dataset of Malaysian banknotes and coins, transfer learning is applied to fine-tune the model architecture. The proposed model will be tested on a test dataset for performance evaluation.

Next, to enhance the accessibility and ease of use of the proposed Banknote Reader for the Visually Impaired, it will be incorporated into a mobile application. The development of the application will utilize the Flutter framework, which allows for the seamless integration of TensorFlow Lite, a compact library that enables TensorFlow models to run on mobile devices. This integration will empower visually impaired individuals to utilize the banknote reader on their mobile devices, thereby providing them with a dependable and hassle-free mechanism for currency identification.

1.2 Problem Statement and Motivation

1.2.1 Overpriced existing banknote readers for visually impaired

Individuals who are visually impaired often encounter difficulties in distinguishing between different banknotes due to their physical and sensory impairments. Although some banknote readers are available for purchase, their cost is often prohibitive, and they may not be easily obtained by those in need. Some examples models and their prices are displayed in Table 1.1 below. This lack of accessibility can create barriers for visually impaired individuals, preventing them from conducting basic financial transactions and forcing them to rely on others for assistance. Moreover, this population is at increased risk of financial exploitation and counterfeiting if they are unable to accurately identify banknotes.

Table 1.1: *Examples of banknotes reader models and respective prices range*

Types	Example of model	Price
App	<i>NantMobile Money Reader, IDEAL Currency Identifier</i>	Approximately \$10
Handheld Devices	<i>iBill Talking Banknote Identifier</i>	Approximately \$100 to \$300

1.2.2 Limited functionality, accuracy and portability for some existing solutions

Existing solutions for banknote readers, such as handheld devices, may be limited in their functionality, accuracy, and portability, which can create challenges for visually impaired individuals. Some devices may not accurately detect or recognize banknotes, leading to errors in identification and potential financial losses. Additionally, some solutions may be bulky or require additional accessories, making them less portable and less convenient for everyday use. For example, [4] proposed a prototype handheld device that will be able to differentiate Malaysian notes using CATIA V5R20. Sometimes, these solutions can be expensive and costly, which may make them inaccessible to those who

CHAPTER 1

need them the most. This lack of accessibility can create another level of additional barriers for visually impaired individuals who already face significant challenges in accessing assistive technology. As a result, they may be forced to back to origin, that is to rely on the assistance of others or may be more susceptible to financial exploitation and scams.



Figure 1.1: *handled cash reader proposed in [4]*

1.3 Motivation

The main objective of this project is to enhance the quality of life for visually impaired individuals by proposing an affordable and accessible banknote reader app. According to the World Health Organization (WHO) [5], roughly 2.2 billion people worldwide suffer from blindness or visual impairments, with the number expected to rise in the coming years. Moreover, individuals with disabilities are among the most marginalized groups in society, with lower education and employment rates, and limited access to healthcare, according to the United Nations (UN) [6]. In general, this project aims to help to cater the problems faced by these increasing population of the community.

One of the main challenges faced by visually impaired individuals is the difficulty in identifying banknotes. Although some banknote readers are available on the market, they tend to be costly and not readily available to those in need. This creates significant obstacles for visually impaired individuals in performing routine transactions, leading to potential financial losses due to mistakes in identification or fraudulent activities.

CHAPTER 1

Furthermore, they are susceptible to financial exploitation and scams because they may have trouble recognizing counterfeit banknotes.

To address these issues, this project proposes developing a deep learning-based banknote reader that is affordable, accessible, and precise. By utilizing the power of deep learning models, the banknote reader can accurately identify and recognize banknotes, providing visually impaired individuals with a dependable tool for identifying currency. This can improve their self-reliance and self-determination, allowing them to conduct transactions with confidence without the need for assistance from others.

In addition, this project is motivated by Malaysian Blind Association (MBA), which serves a significant population of visually impaired individuals in Malaysia. By creating a banknote reader that is customized to meet the needs of this population, this project can have a more significant impact on improving their quality of life.

1.4 Project Scopes

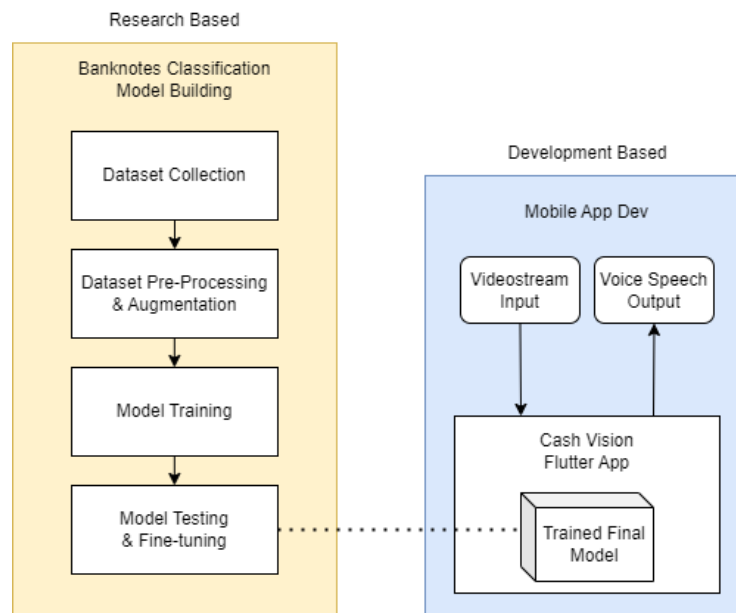


Figure 1.2: *Project Scope Overview*

CHAPTER 1

This project aims to develop a Malaysian banknote and coin reader model that is integrated into a mobile application. The project will be divided into two main phases: research-based and development-based.

In the research-based phase, data will be collected locally since the source dataset for Malaysian banknotes and coins is limited. The dataset will then be pre-processed into batches for efficient training. The training model will then be trained on the pre-processed dataset, and the training will involve various techniques such as data augmentation, fine-tuning, and transfer learning. The trained model will then be evaluated and tested on various metrics involving confusion matrix, accuracy, precision recall and etcetera to ensure its accuracy and reliability. Based on the evaluation results, the model will be fine-tuned until the final optimal model is developed.

In the development-based phase, the final model will be integrated into a mobile application developed under the Flutter framework. The application will accept camera capture as input and generate voice speech based on the classified output, making it more accessible for visually impaired individuals.

1.5 Project Objectives

The objectives of this project are summarized as follows:

1. To develop a deep learning classification model for Malaysian banknotes and coins using transfer learning. The model will be fine-tuned on a bespoke dataset of Malaysian banknotes and coins to improve accuracy and uplift performance.
2. To implement the proposed trained classification model into a mobile application under the Flutter framework. The application will be able to accept video stream input and will output predicted banknotes and coins as voice speech. This integration will provide a convenient (mobile) and accessible (voice output) solution for visually impaired individuals.
3. To develop a methodology that can understand the context of different banknotes and coins scanned in depth. This will enable the system model to extract information and features from the banknotes and coins scanned, primarily the

CHAPTER 1

denominations, as well as any unique features that may distinguish them from counterfeit currency.

1.6 Contributions

In the end of this project, the deliverables of this project would be an easily accessible and affordable banknote and coin reader model integrated into a mobile application. By providing such banknote reader app, this integration allows visually impaired individuals to confidently perform transactions without relying on the assistance of others, contributing to their overall quality of life. Through this, this project helps to improve the independence and autonomy of visually impaired individuals and reduces their vulnerability to financial exploitation and scams.

Furthermore, The Malaysia Blind Association (MCA) is a primary beneficiary of this project, as it can provide an affordable and accessible solution for the visually impaired individuals they serve. This project's output can ultimately enhance the living quality of visually impaired individuals, allowing them to perform transactions confidently and independently, without the need for assistance from others.

Another significant contribution of this project would be the developed banknote and coin reader model will be deployed as an open-source project. By making the model open source, it can be used by developers, researchers, and organizations to further improve and enhance the model's accuracy and accessibility. This will also increase the availability of affordable and accessible assistive technology for visually impaired individuals in Malaysia and potentially worldwide.

1.7 Report Organization

There are total of 8 chapters in this report, following table listed all the chapters and its description.

Table 1.2: *Reporting Structure*

No	Chapter	Description
1	Introduction	<ul style="list-style-type: none"> • Provides project background. • Define problem statements, and objectives. • Discuss motivation and contribution.
2	Literature Review	<ul style="list-style-type: none"> • Perform preliminary studies on previous works and existing systems • Comparison between different techniques
3	Methodology	<ul style="list-style-type: none"> • Define methods utilized to complete this project • Specify hardware and software requirements • Display project timeline
4	System Model	<ul style="list-style-type: none"> • Study and design Malaysian cash reader model through deep learning approach
5	System Design	<ul style="list-style-type: none"> • Design steps of model deployment to visually impaired • Display system block diagram and discuss key concept of proposed system <p>Emphasizes development part of whole project</p>
6	System Implementation	<ul style="list-style-type: none"> • Document system implementation • Discuss challenges faced
7	System Evaluation	<ul style="list-style-type: none"> • Explain evaluation metrics employed • Perform model evaluation on model and developed apps • Evaluate project based on project objectives
8	Conclusion	<ul style="list-style-type: none"> • Discuss conclusion, novelties, contributions and future works

CHAPTER 2: LITERATURE REVIEWS

In this section, a review of previous studies on banknote reader model will be conducted. Each proposed method will be thoroughly evaluated and analysed.

2.1 Previous works on banknote reader model

2.1.1 Currency Recognition System for Blind people using ORB Algorithm [7]

According to [7], the recognition or detection of denomination of currency emerges as an important needs for Blind and visually impaired people, since they are not equipped with capability to differentiate between currency correctly and precisely, since they have lost a certain percentage of vision, and so it is very easy for them to be cheated by others, whether its untrusted seller, or those with bad intention, which they manipulate the point that they couldn't differentiate the currency value well. This proposed method designed a system that could easily recognize the value of currencies, regardless of internal or external factors like rotation, outdoor illumination, scaling, quality issues and etcetera.

This proposed method proposed a system block diagram as shown below, where the system can be divided into online phase and offline phase, where the online phase serve for capturing and classifying, and the offline phase target to construct dataset and training.

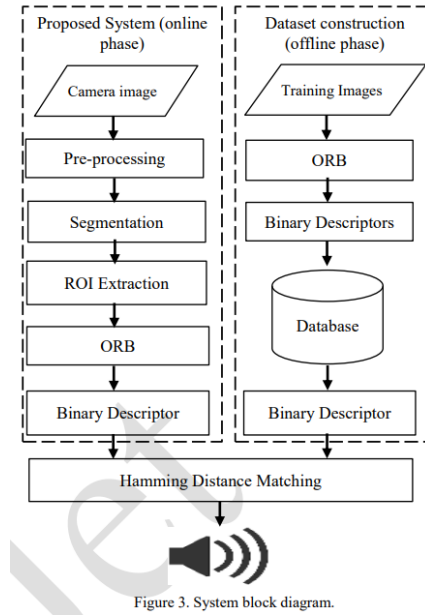


Figure 2.1: *System Block Diagram of the proposed method 2.1.1*

For the online phase, the first step is to obtain camera image, then some pre-processing techniques are employed on the image. Gaussian blurring is first applied to eliminate noise from image, plus sharpen the image to aid in segmentation of the currency for next step. The equation for Gaussian Blurring is as shown below:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Figure 2.2: *Equation of Gaussian Blur Effect applied*

The equation demonstrates that x represents the distance from the origin along the horizontal axis, while y represents the distance along the vertical axis, and is the standard deviation of the Gaussian distribution. The next step will be segmentation, where the captured image is converted into binary image consisting black and white, where Otsu's Thresholding is employed in this case to convert RGB format of original image to binary image consisting only 0 and 1. The equation of Otsu's Thresholding are as shown below:

$$T(x, y) = \begin{cases} 1 & \text{if } T(x, y) \geq th \\ 0 & \text{otherwise} \end{cases}$$

Figure 2.3: Equation of Otsu's Thresholding technique applied

Where in the equation, $T(x, y)$ is the image's density, and th is the threshold value that the Otsu threshold technique modifies. Next stage will then be the Region of Interest (ROI) extracting. In this case, Two-Pass Connected Component Algorithm used to extract currency from the image. In this proposed method, Oriented FAST and BRIEF (ORB) algorithm is employed to perform feature extraction to achieve good results:

1. FAST algorithm is used to identify corners and interest points in the image captured.
2. Harris Corner detector is used to assign scores to each point of interest based on the variation of intensities surrounding the corner.
3. Sorting based on interest points and only takes into account N top areas.
4. Intensity-weighted centroid computed for interest points neighbourhood.
5. Calculates the vector direction and uses the interest point and centroid to assign it as the orientation of the interest point.

After that, main part of ORB is employed after getting the interest points. The feature description is then extracted for these interest points. Utilizing the pixel pair sampling technique, The BRIEF algorithm generates local binary descriptors comprised of 0 and 1 binary digits, regarding a local shape such as a circle or rectangle. Pixel pair sampling consists of a pair of pixels. $P1$ and $P2$ are taken as examples and compared from a 31×31 patch surrounding each interest spot. Put 1 in the description vector as d in the following equation if $P1 > P2$.

$$\tau(P:P1, P2) = \begin{cases} 1 & \text{if } P1 \geq P2 \\ 0 & \text{otherwise} \end{cases}$$

Figure 2.4: Pixel Pair Sampling Techniques

Where τ is the test of point 1 p_1 and point 2 p_2 , and the outcome is either 1 or 0 contingent on the comparison of p_1 and p_2 . The last stage will be matching of the binary descriptor, where Hamming Distance Measure is applied in this case. The binary descriptor newly captured is compared to those previously stored in database. Hamming Distance Measure is chosen since its very fast and efficient for matching binary text.

$$D^{\text{HAD}}(i, j) = \sum_{k=0}^{n-1} (Y_{i,k} \neq Y_{j,k})$$

Figure 2.5: *Hamming Distance Measure Equation*

The hamming distance specifies the number of mismatches between variables paired by k , where i , j , and k are indices of the corresponding variable n . After matching, the largest number of matches with the database indicates the correct identified variety of currency, and the results are played via speaker.

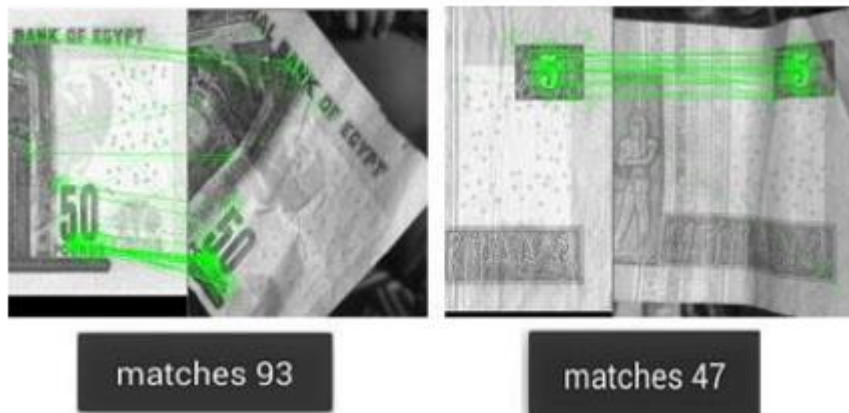


Figure 2.6: *Matching Process using Hamming Distance*

2.1.2 Image Based Ringgit Banknote Recognition for Visually Impaired [8]

Banknote identification system can be divided into sensor-based and vision-based. Due to the complexity of the sensor-based systems' electrical components and their inability to supply sufficient data, the recognition process was hampered and the systems'

accuracy will be inaccurate. In this proposed method, we can see the system block diagram as shown below:

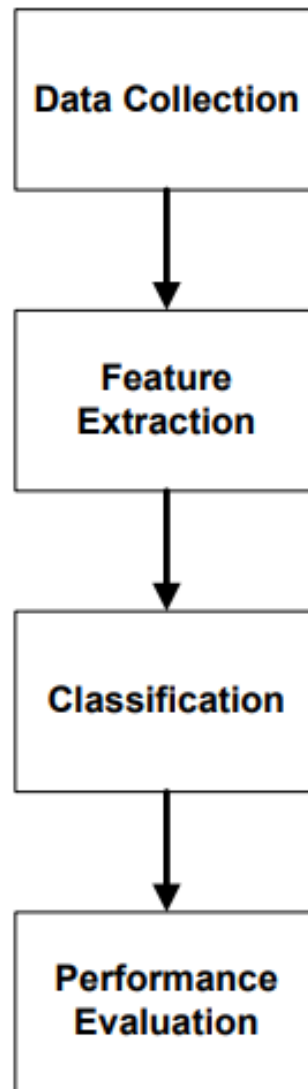


Figure 2.7: *System Block Diagram for current proposed method*

In data collection stage, multiple denomination of Ringgit Malaysia was captured and collected, including RM1, RM5, Rm10, RM20, RM50 and RM100. Then, MATLAB Tool TrainingImageLabeler was employed to extract out the informative region, whole region and combined region.



Figure 2.8: *Informative Region and whole region captured*

The next stage will be feature extraction, where colour feature extraction is applied in this case. Firstly, RB, RG and GB are extracted from both informative region and whole region of captured images as main features.

$$\begin{aligned} RB &= \bar{r} - \bar{b} \\ RG &= \bar{r} - \bar{g} \\ GB &= \bar{g} - \bar{b} \end{aligned}$$

Figure 2.9: Main features RB, RG and RB

Where \bar{r} is the average red channel intensity value, \bar{b} is the average blue channel intensity value, and \bar{g} is the average green channel intensity value of the pixels inside the cropped area.

Next stage will be cross validation, where the main features RB, RG and GB are used for cross-validation with ten-fold to generate optimum model of Decision Tree Classifier (DTC) through varying minimum leaf size. The resultant model is then compared with other classifier model like Naïve Bayes and k-Nearest Neighbour (k-NNs).

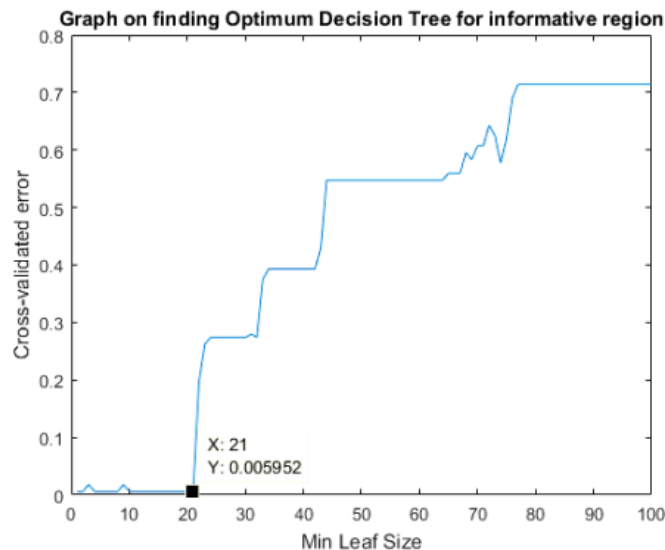


Figure 2.10: Graph on Finding Optimum DTC (Info Region)

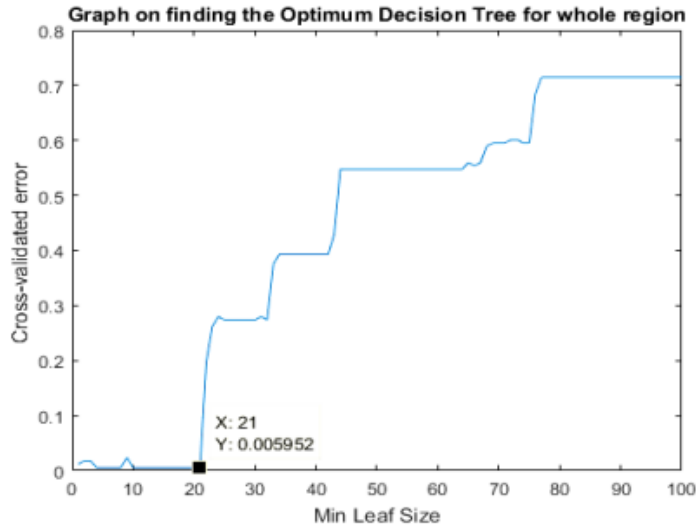


Figure 2.11: *Graph on Finding Optimum DTC (Whole Region)*

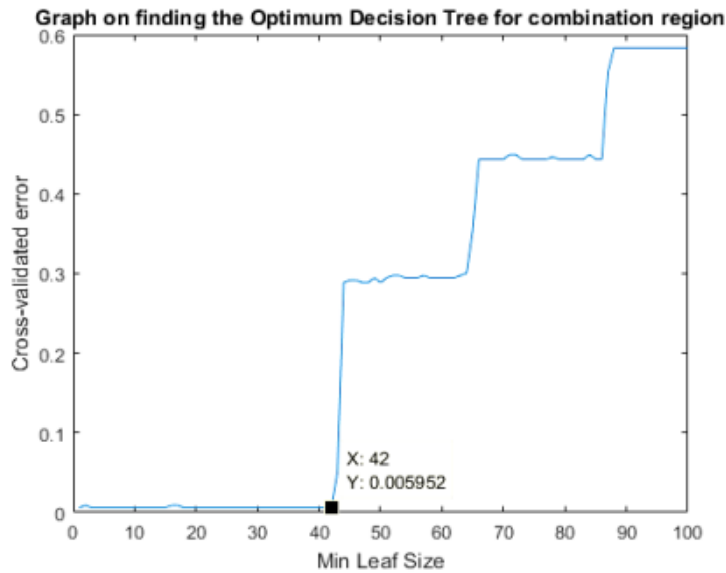


Figure 2.12: *Graph on Finding Optimum DTC (Combined Region)*

In this proposed method, Optimum DTC was chosen based on the varying of the number of minimum leaf size. Through 10-fold cross validation, cross validation losses from multiple different DTC was computed. Minimum leaf size help to determine complexity of produced DTC model, As minimum leaf size increases, the lower the

CHAPTER 2

complexity. In this instance, the minimum number of leaf sizes from 1 to 21 and 1 to 42 result in the lowest cross-validation loss.

The last stage will be evaluation, where Confusion Matrix is generated from the k-NN Classifier and DTC with the optimum parameters. We can conclude that k-NN had achieved better performance on Combination Region as shown below:

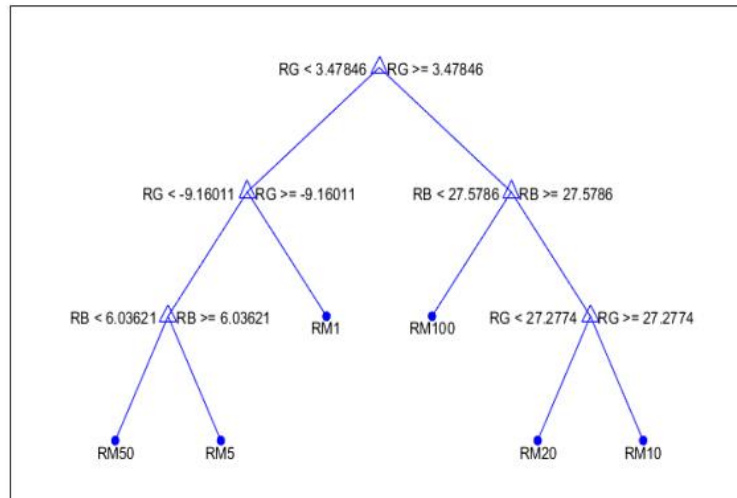


Figure 2.13: *Optimized DTC with 3 levels after minimum leaf size = 21
(Combination Region) applied*

For the DTC, it had achieve an accuracy of 99.7%

Confusion Matrix

1	96 28.6%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	99.0% 1.0%
2	0 0.0%	48 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	48 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	48 14.3%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	47 14.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	48 14.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	97.9% 2.1%	100% 0.0%	99.7% 0.3%
	1	2	3	4	5	6	
	Target Class						

Figure 2.14: *Confusion Matrix for DTC using whole region achieving accuracy of 99.7%*

2.1.3 Malaysian Banknote Reader For Visually Impaired Person [9]

There are 3 types of cash denomination recognition for visually impaired to carry out business activities, consisting of size recognition, colour recognition and image pattern features extraction. [3] stated that size recognition was not suitable since it only suits recognition of object with obvious differ in size, while the size of Ringgit Malaysia has only small difference across each denomination. Colour recognition was also not efficient enough since the HSV/HIS/RGB values of RM10 is similar to RM20, RM1 is similar to RM50, therefore false detection can be easily triggered. Therefore, image pattern recognition is the method that suits Ringgit Malaysia currency recognition the most due to its specialty and characteristics.

The system block diagram of proposed method is as shown below:

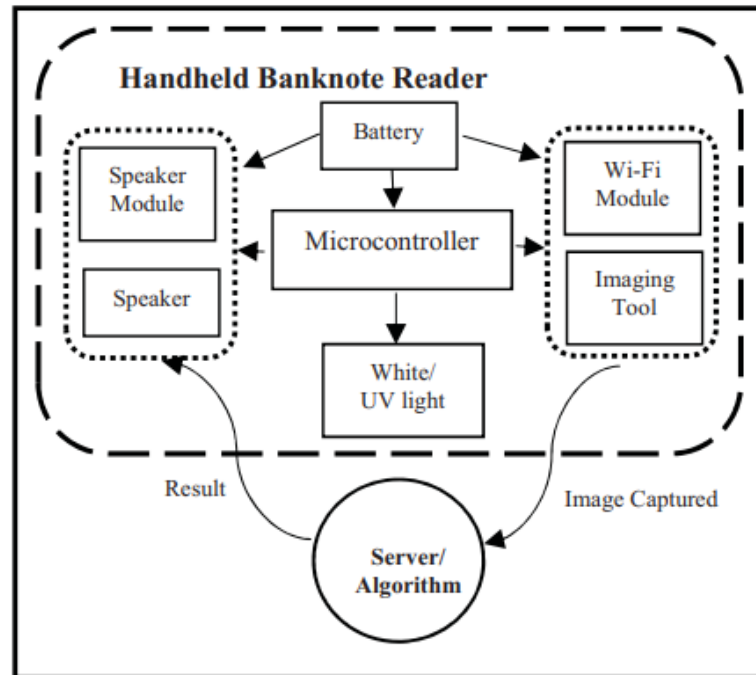


Figure 2.15: *Architecture Design of the proposed method*

The design as shown above includes multiple components, ranging from imaging tool, battery, microcontroller, light, speaker modules, algorithm and etcetera. Imaging tool is needed to capture the image of the bank note to be classified, CMOS camera is employed in this proposed method rather than CCD, since CMOS could capture image with higher resolution and a nice imaging tool is severe as it impacts the accuracy and effectiveness of classification tasks. Next, a small focal length is needed for small focal distance and so ESP-32 based development board by LILYGO is employed. Not only due to the imaging ability, but also ESP-32 comes readily with Wi-Fi modules which is essential in current proposed method.

For the battery part, The system utilised 3.3V and 5V electronic components, with 5V 1A dry-cell batteries, or it could be replaced with 5V 1A power bank. The microcontroller will be ATmega328p from Arduino Uno, which is sufficient for minimum requirements, and it serves to communicate and coordinate between all I/O devices. The lighting conditions is also import for clear images, where the banknote is placed in a small dark environment inside a box device, and so UV light and white light

is both installed for sufficient light source. The speaker modules chosen are DFminiplayer MP3 player, where it helps to translate the classification result through narration of the denomination of the banknote.

The algorithm part involved in few stages. The first stage is to setup database, where the image is captured in devices and transferred, stored in PC. Then, all images in database will be sorted through Perceptual Hashing, where the images will be resized and sorted accordingly with perceptual hash codes. The image input will then be transferred to server. For the Perceptual Hashing algorithm, the images will first be converted into 1024x512px grayscale image, then Gabor Filter is applied, and Discrete Cosine Transformation (DCT) is performed using equation below:

$$\text{DCT: } X_k = \sum_{n=0}^{N-1} 2n * \cos\left(\pi * k * \frac{2n+1}{2N}\right) \forall k \in [0, N] \quad (1)$$

Figure 2.16: Equation of Discrete Cosine Transformation (DCT)

Then, the RGB mean will be extracted and computed. After the mean computed, we now do comparison through Hamming Distance. The hash value y will first be generated, and the input x will be taken to compared with hash value stored in database followed by equation below, where the identical percentage P will be computed:

$$D = \min\{d(x, y) : x, y_n \in C, x \neq y\} \quad (3)$$

Figure 2.17: Equation of Ideal Percentage P

When P achieve a value of 90% to 99%, both images is considered identical. Next, Fuzzy Logic algorithm is applied for the classification. Since there may be more than one image in the database that meets the threshold value, because the perceptual distance between images of banknotes with various orientations that represent the same value is similar to one another, thus 3 rules is followed as shown below:

Rule 1: If either P or $D(x,y_n)$ is greater than 0.90, the input image is identical to one of the images in the database; the resultant output, O equals 1.

Rule 2: If more than three P are within the range $0.8D < D(x,y_n) < 0.9$, then the input image is identical to at least one of the database images; the resultant output, O equals 0.5.

Rule 3: If all of the $D(x,y_n) < 0.8$, then the input image is not identical to any of the database images; the resultant output, O equals 0.

The output will then be defuzzied, where the captured image matches one or more others images in database if defuzzified $O > 0.5$. While the corresponding image will not be considered if defuzzified $O < 0.5$. The final results will then be transmitted back to the device and narrated out through the speaker to inform user. In order to increase precision and accuracy consistently, the input image are stored to database to train so that unidentified images due to brightness/luminance issues could be minimized

2.1.4 Currency Recognition for Visually Impairment People Using Deep Learning Algorithm [10]

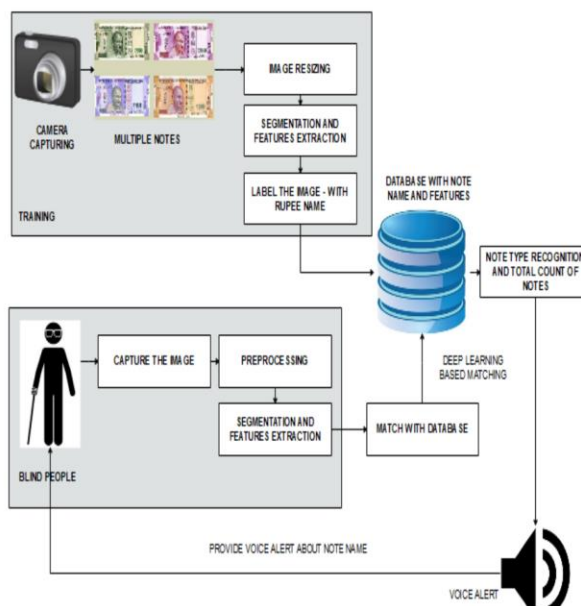


Figure 2.18: *System Block Diagram*

[10] introduces a comprehensive system designed for the recognition of various currencies through the utilization of Convolutional Neural Networks (CNNs). The proposed system block diagram is as shown in Figure 2.18. The methodology proposed initiates with the acquisition of a substantial dataset that encompasses a wide array of currency images. This dataset consisting of a grand total of 6000 images, this dataset covers every denomination of Indian currency notes, encompassing 10, 20, 50, 100, 200, 500, and 2000 rupees, with each denomination being represented by 1000 images. These images were taken under various circumstances, utilizing a range of cameras and different lighting setups, thereby making the dataset exceptionally varied and demanding in terms of complexity. The sample datasets are as shown in Figure 2.19 below.

**Figure 2.19:** *The Indian Rupees Dataset*

Subsequently, a series of preprocessing procedures are implemented. These encompass the removal of unwanted noise from the images, standardizing image sizes, and adjusting pixel values to a uniform range. The pre-processed dataset is then thoughtfully divided into three subsets: one for training the model, another for validation

CHAPTER 2

to assess the model's performance during development, and a final set for testing the model's real-world effectiveness.

To build the core of the system, a CNN architecture is constructed. This architecture comprises multiple convolutional layers, each enhanced by a pooling layer to efficiently process and condense feature maps. Additionally, advanced techniques such as Dense Connection, Multi-Dilation, and Depth-wise Separable Convolution layers are incorporated to elevate the accuracy of the model. In context, Dense Connection enhances CNNs by establishing dense connections between layers for efficient information flow. Multi-Dilation introduces varying dilation rates to capture information at different scales in images. Lastly, Depth-wise Separable Convolution reduces computation by breaking down convolution into two steps.

The convolutional layers output is then channelled through fully connected layers, culminating in the model's final output. Model training leverages backpropagation and gradient descent for weight optimization, and this process integrates data augmentation techniques. In this case, data augmentation not only increases the dataset size but also augments the model's capacity to generalize, aligning with strategies outlined in the paper.

After the training phase, a thorough fine-tuning process ensues. The validation set plays a crucial role in fine-tuning various hyperparameters such as the learning rate, dropout rate, and filter counts. This adjustment of hyperparameters, as outlined in the paper, aims to optimize the model's performance and enhance its accuracy and robustness. Lastly, evaluation phase undergoes. During this evaluation phase, various evaluation metrics, such as accuracy, precision, recall, and F1 score, are employed. Upon successful training and evaluation, the model is poised for deployment in real-time production environments. Its integration potential spans a multitude of applications, including ATMs, cash counting machines, and vending machines.

2.1.5 Currency Recognition for Visually Impairment People Using Deep Learning Algorithm [11]

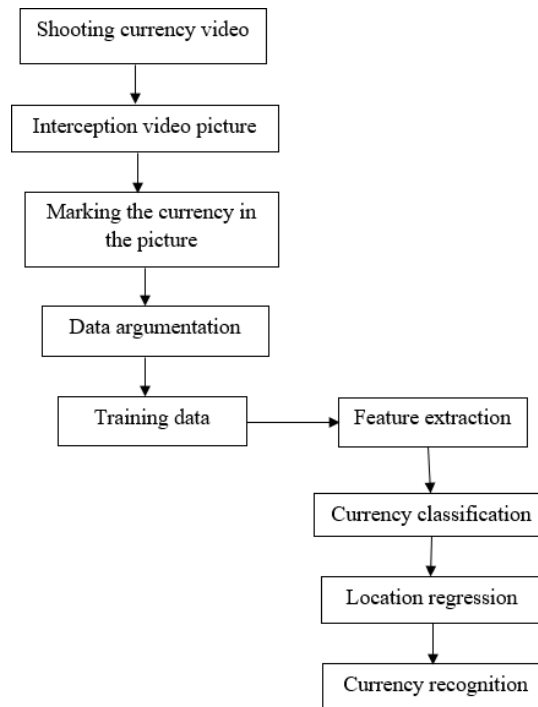


Figure 2.20: *System Design Proposed in [11]*

The system proposed by [11] starts with manual capture of NZD with several denominations, specifically denominations of 5NZD, 10NZD, and 20NZD, each with front and back sides. Videos of these currencies were recorded, with attention to ensuring that the currency was flat, fully visible, and captured from various angles during shooting. Adequate lighting was also maintained to capture currency details clearly. The recorded videos were then organized into six folders, such as "5NZD-F" representing the front side of 5 New Zealand dollars. After video editing to extract clear and complete frames, a raw dataset of 300 images was obtained, each with a resolution of 1280×720 , as shown in Figure 2.21 below.



Figure 2.21: *The New Zealand Dollars Dataset*

In the aforementioned data collection process, [11] chose to curate a dataset comprising 300 currency images. To streamline the manual labelling process, the author revolutionized from the conventional practice of using rectangular annotations. Instead, the author opted for quadrilateral markings, illustrated in Figure 2.22, which offered the advantage of accurately specifying both the currency's position and its orientation.

For the manual labelling task, the author employed the user-friendly MATLAB GUI (Graphical User Interface). Annotation commenced from the upper left corner of the currency image and proceeded clockwise to mark all four corners. What's noteworthy is that this approach allowed for adjustments using movable marker boxes, contributing to improved precision in annotating currency images. Importantly, it also helped mitigate potential inaccuracies caused by environmental factors during the annotation process.

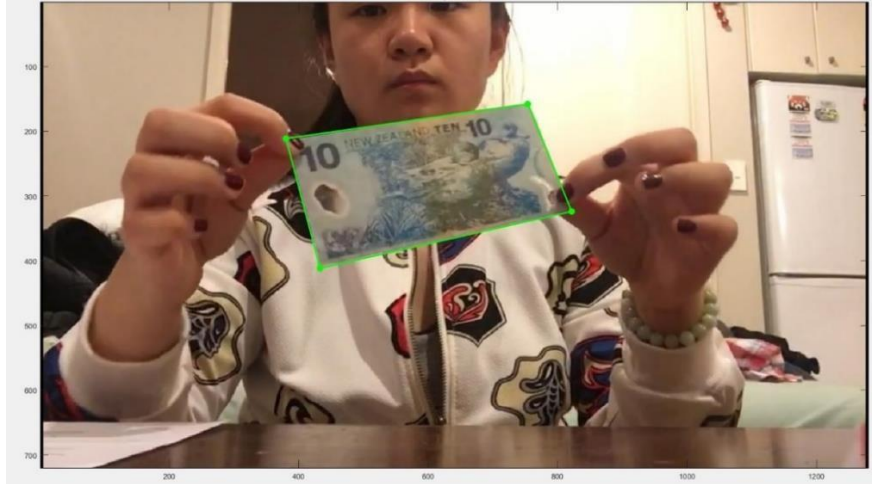


Figure 2.22: *Manual marking using quadrilateral*

Following the initial data collection and manual labelling process, the dataset consisted of 300 valid raw currency images for recognition. Data augmentation was then employed to generate additional dataset.

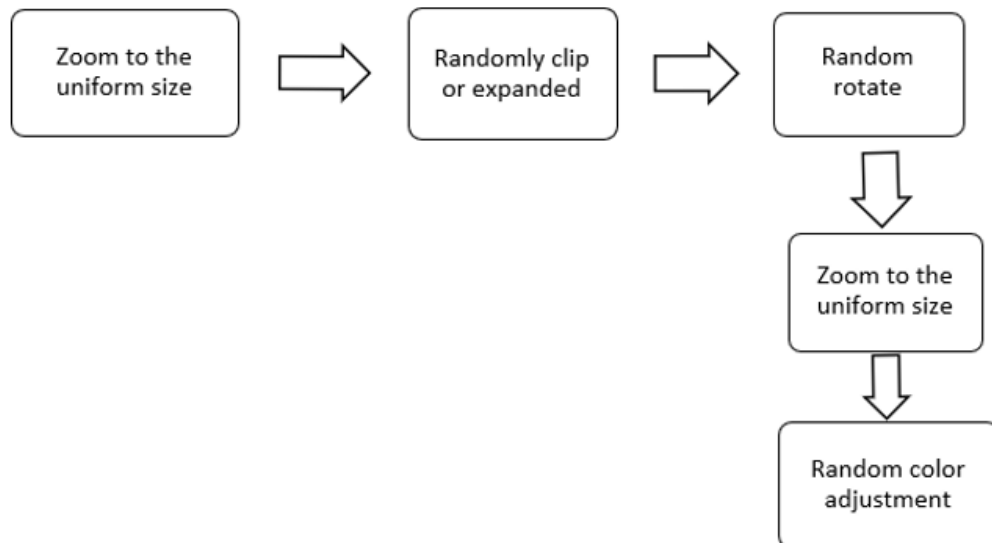


Figure 2.23: *Data Augmentation Steps*

This approach involved five key steps, outlined in Figure 2.23:

1. **Uniform Sizing:** To ensure consistency in model training, the dataset images were resized to a standard resolution of 640×360 pixels, termed nHD, irrespective of variations in size during data collection.
2. **Random Clipping or Expansion:** Given the importance of recognizing currencies in different positions, random clipping or expansion was performed to vary the position and size of the currency within the images. This was achieved by adjusting the area ratio, effectively changing the position of the currency.
3. **Random Rotation:** To introduce diversity in angle and position information, the original currency images were randomly rotated by angles such as 180° , 90° clockwise, and 90° counterclockwise.
4. **Reuniform Sizing:** Following random modifications like cropping, scaling, or rotation, the images were resized again to maintain uniform dimensions, a requirement for training.
5. **Random Color Adjustment:** Given the varying lighting conditions during video capture, random color adjustments were applied by converting RGB (red, blue, green) colors to HSV (hue, saturation, and value). This ensured color consistency across images.

As a result, each original video frame yielded 25 enhanced currency images. In practical terms, the initial 300 raw data images expanded to a dataset of 7,500 images after augmentation.

Next, the proposed method has chosen SSD (Single Shot Multibox Detector) as its currency detecting model, the model can be illustrated in Figure 2.24. The process begins with representing images as matrices of RGB color components, where pixel values range from 0 to 255. To make these images compatible with the continuous nature of Deep Neural Network (DNN) models, they are normalized into the range of $[-1, 1]$.

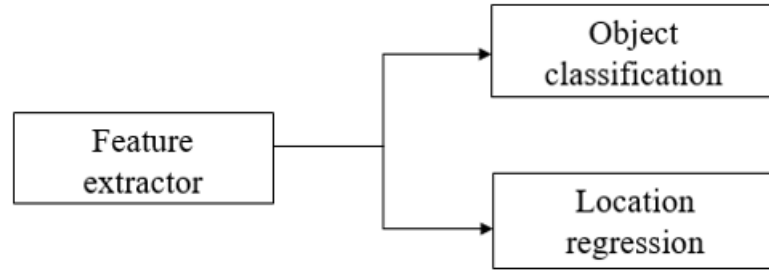


Figure 2.24: *SSD (Single Shot Multibox Detector) Model*

Subsequently, the normalized images go through convolutional layers. These layers perform two critical operations: convolution and pooling. Convolution entails a weighted summation of small portions of the input matrix using convolution templates. This is followed by the application of the LeakyReLU non-linear activation function. The main goal of these layers is to extract essential features from the images.

$$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

Figure 2.25: *LeakyReLU activation function*

Now, the output of convolutional input is in the form of matrices. To prepare this data for subsequent processing, it is flattened into vectors. Additionally, a dropout strategy is employed. This strategy randomly deactivates some nodes, which enhances the model's tolerance to faults without significantly affecting its overall performance.

Classification MLP Layer: The flattened data is then passed through a fully connected layer known as the Classification MLP (Multilayer Perceptron). Here, a vector 'x' serves as input, and the output is another vector 'y', produced using a non-linear activation function, specifically LeakyReLU. In the context of currency recognition, a two-layer full-connection layer configuration is utilized for classification. The output represents a probability vector, with each element denoting the probability of the sample

belonging to a specific class. The FC layer could be illustrated in equation shown in Figure 2.26.

$$y = \sigma (\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$$

Figure 2.26: *Representation of FC Layer*

Apart from classification MLP layer, the proposed model also includes Location MLP layer. Similar to the Classification MLP, the Location MLP layer consists of two fully connected layers. However, its primary purpose is to encode position information. Two types of encodings are employed - one utilizing the original coordinates and another normalizing the coordinates to a range of [0, 1].

Next, prediction layer serves as an integration point for both classification and position outputs. It generates the inferred results, which include vertex coordinates, classification probability (cls), and confidence (conf) in the classification. The classification probability represents the maximum probability among all classes.

Following the prediction, the Validation Layer comes into play. It evaluates the model's accuracy from two perspectives. Firstly, it checks whether the classification is correct by comparing the predicted class to the actual class. Secondly, it assesses the correctness of positioning. Generally, the positioning is considered accurate if the Intersection over Union (IoU) is greater than 0.5.

$$L = L_{cls} + wL_{loc}$$

Figure 2.27: *Loss function*

During the training phase, a weighted sum of classification and positioning loss functions is optimized using the gradient descent algorithm. The loss function, denoted as L , is calculated as the sum of L_{cls} (classification loss) and wL_{loc} (positioning loss), where 'w' represents a weight factor. The model parameters are continually updated through this

process, ensuring that the model learns to recognize currency accurately. This training is carried out within the TensorFlow learning framework.

2.2 Critical Remarks of Previous Works

All of the previous studies reviewed have their very own strengths and weaknesses. I've cautiously reviewed and concluded some main information in the table 2.1 below.

Table 2.1: *Table of brief comparison between each previous study.*

Works	Currency	Model Used	Mobile Integration	Accuracy
[7]	EGP	<i>ORB Algorithm</i>	Yes	96.29 %
[8]	RM	<i>k-NN & DTC Classifier</i>	No	99.70 % (RM50 class)
[9]		<i>Perceptual Hashing Fuzzy Logic Algorithm</i>	No	95.00 %
[10]	INR	<i>CNNs</i>	Yes	•
[11]	NZD	<i>CNNs, SSD Model</i>	No	94.69 %

The first study utilized ORB and FAST algorithms for currency detection, offering the strength of speed and efficiency, which is crucial for real-time recognition. However, they may struggle with complex lighting conditions and variations in currency features, where it might be struggling to detect fine details on currency notes, potentially affecting their ability to distinguish between denominations accurately.

[8] employed k-NN and DTC algorithms for currency recognition, where k-NN classifier is a simple, yet effective pattern recognition method and DTC provides interpretable decision rules, which could be easier to adapt the system for recognizing multiple currency denominations. Third study relied on Perceptual Hashing and Fuzzy Logic algorithms, which are robust to minor alterations, that is useful for currency detection. However, the proposed method employed an external bulky device design, which could be an issue for visually impaired who often rely on mobile devices for assistance.

CHAPTER 2

The next study implemented a Convolutional Neural Network (CNN) for Indian Rupees currency detection. CNNs are adept at learning intricate features and patterns, which is essential for distinguishing between different currency denominations by touch. While [11] adopted the Single Shot MultiBox Detector (SSD) model with a CNN as a feature extractor. SSD's efficiency in localization could enhance system's ability to identify exact orientation and location of the notes. Yet, the study's lack of mobile integration specifics, which may affect its practical utility.

Based on all previous studies of existing currency recognition system, a model to aid in currency detection. The proposed method is capable to understand fine details of the currency, mobile integrable, and visually impaired friendly, through combining of techniques from multiple studies and systems reviewed.

CHAPTER 3: METHODOLOGY

This chapter will firstly outline the applied methodologies and techniques in detail. Then, all hardware, software, libraries, and services requirements utilized and implemented throughout the development of this banknote's classification model will also be summarized.

3.1 Methodologies

The overall approach can be divided into two distinct phases. The initial phase entails the research and development of a currency denomination classification model, where the target deliverables will be a ready to use model to distinguish various banknotes and coins for visually impaired.

Upon the successful creation of this model, the subsequent phase, which is the development stage, will be dedicated to the creation of a mobile currency reading application. During this phase, the subsequent objective will be to craft an accessible cash reader application, leveraging the model developed in the initial research phase. This model will be seamlessly integrated into the core of a Flutter-based application, allowing users to identify and classify different denominations of currency effortlessly and accurately.

3.1.1 Dataset Acquisition

In the context of this study, the focus was placed exclusively on Malaysian Ringgit currency denominations. The process of acquiring the dataset was formidable due to the limited availability of online datasets for Malaysian Ringgit banknotes and coins. To construct a comprehensive dataset for the project, a manual image capture procedure was employed, encompassing various lighting conditions and backgrounds.

Specifically, the dataset was comprised of two primary components: banknotes and coins. For banknotes, six distinct classes were considered, representing the denominations of RM1, RM5, RM10, RM20, RM50, and RM100. Each class featured a collection of 200 images, capturing variations in lighting conditions and backgrounds.

CHAPTER 3

These images were taken from both the front and back perspectives of the Ringgit notes, ensuring comprehensive coverage of the currency's visual attributes.

Similarly, the coin dataset consisted of four classes, corresponding to the denominations of 5 sen, 10 sen, 20 sen, and 50 sen. Within each class, a set of 140 images was meticulously captured, accommodating diverse lighting scenarios and background settings. As with the banknotes, both the front and back views of the coins were included in the dataset. In summary, the datasets captured manually encompasses a total of 1,200 images for banknotes and 560 images for coins was captured, prior to the application of data augmentation techniques.

3.1.2 Dataset Pre-Processing and Data Augmentation

In this project, several essential preprocessing and data augmentation techniques employed to prepare the dataset for effective model training. Initially, the dataset was divided into training, testing, and validation sets, ensuring a systematic approach. Following this, both training and validation datasets were normalized for uniform scaling. Data augmentation techniques were then applied to the training dataset, including image flipping, adjustments to exposure for varied lighting conditions, and the introduction of subtle noise (salt and pepper pixels) to simulate real-world distortions. Additionally, random-sized cropping was used to expose the model to different scales and perspectives. These techniques collectively enriched the training data, enhancing the model's adaptability and robustness without the need for complex embellishments.

3.1.3 Model Training and Testing

In this phase, a transfer learning approach is employed, utilizing a pre-trained model imported from Ultralytics as the base model. To retain the knowledge from the pre-trained model, specific front layers are frozen using callback functions. Subsequently, the pre-processed and augmented data is employed as input for the combined model. Following this, the model's performance is evaluated using various metrics including the F1 score, precision, recall, and ROC analysis. Based on the evaluation results, the model

CHAPTER 3

undergoes a fine-tuning process to enhance its performance. Finally, the model is meticulously tested using an independent test dataset, providing conclusive insights into its performance and effectiveness.

3.1.4 Deployment of trained model into Flutter app

In the deployment phase, the final trained model will be integrated into the Flutter app by initially converting it into a TensorFlow Lite (TFLite) model using the TFLite converter library package. This transformation enables efficient integration into the app through the TFLite Flutter plugin. The app will initially be configured to receive real-time video stream input via the camera library. The predictions generated by the model are then presented on the screen and concurrently relayed as voice speech through the speaker. This dual-output approach ensures the information is not only displayed visually but also audibly, catering to the needs of visually impaired individuals and enhancing accessibility.

3.2 Tools

3.2.1 Hardware

In this project, the hardware involved would be a computer device and a mobile device for testing. These 2 devices are sufficient alone to construct the full model and app, no extra accessories devices are required. In this case, the compute used is a desktop computer, and whole process of coding the model training, testing, and the app will be fully conducted on this desktop computer. Then, the mobile device will be connected through USB debugging for testing purposes.

Table 3.1: *Desktop Specification*

Specification	Description
Processor	Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz
Graphic	NVIDIA RTX 3060 Ti
Memory	16GB
Storage	1TB SSD
Operating System	Windows 11

Table 3.2: *Mobile Device Specification*

Specification	Description
Model	Pocophone F3
Processor	Qualcomm SM8250-AC Snapdragon 870 5G (7 nm)
Memory	8GB
Storage	128GB
Operating System	Android 11

3.2.2 Software & Services

In general, a total of 6 software and services employed to develop the banknote reader model integrated in mobile app in this project. The software and services are summarized as follows:

1. Visual Studio Code

Visual Studio Code is a free, open-source source-code editor that is widely used for software development. It supports a wide range of programming languages and features such as debugging, syntax highlighting, auto-completion, and Git integration. Its popularity is attributed to its ease of use, flexibility, and efficient performance. In this project, it serves as the main code editor to develop Flutter app, integrating the classification model and perform debugging.

2. Jupyter Notebook

Jupyter Notebook is an open-source web application used for interactive data science and scientific computing in Python language. Users can create and share documents containing live code, equations, visualizations, and narrative text within the services provided. It is widely used in the data science community due to its flexibility and ease of use. This will be the main workspace in this project to develop the banknote reader model, where a pipeline is built from dataset collection to model testing and fine-tuning.

3. Keras

Keras is a popular high-level neural networks API written in Python that can operate on top of TensorFlow, Theano, and CNTK. It is designed to simplify the process of creating deep learning models, allowing users to build complex neural networks quickly and easily with just a few lines of code. In this case, TensorFlow library are employed in building and debugging the app model.

4. TFLite

TFLite, short for TensorFlow Lite, is a lightweight framework for deploying machine learning models on mobile and embedded devices. It allows for efficient execution of models on-device, with support for hardware acceleration and optimization for different platforms. TFLite supports a variety of models, including those trained in TensorFlow and other popular machine learning frameworks. The model trained in Jupyter Notebook will be converted and integrated using this plugin.

5. Flutter

Google's Flutter is an open-source mobile application development framework. It enables developers to create natively compiled mobile, web, and desktop applications from a single codebase. Flutter uses the Dart programming language and provides a rich set of pre-built widgets and tools for creating beautiful and responsive user interfaces. Flutter framework is chosen due to its efficiency and fast responsiveness, if compared to another framework like React Native.

6. Ultralytics

Ultralytics is a versatile open-source computer vision framework designed for deep learning research and deployment. It provides a comprehensive suite of tools and utilities for training and deploying state-of-the-art computer vision models, making it an invaluable asset for building robust and accurate image-based applications. In this project, Ultralytics plays a pivotal role in facilitating the development of the currency denomination recognition model, offering an extensive set of functionalities, and streamlined workflows for efficient model training and evaluation.

3.3 Timeline

The project is structured into two distinct phases, referred to as Project I and Project II. Project I stands at 40% completion, while Project II stands at 60% completion. Project I primarily revolve around establishing the foundational architecture of the system, serving as a proof of concept. It involves the training of a fundamental deep learning model for currency denomination recognition. In contrast, Project 2 delves deeper into the study of deep learning models and focuses on refining the currency recognition model to fulfil advanced objectives, including precise detection of currency, satisfying user interface designed for visually impaired. The timeline for both Project I and Project II is illustrated below:

- a) Project I (40%): Concentrating on proof of concept, it encompasses the development of a basic deep learning model tailored exclusively for currency datasets. This model is subsequently integrated into a rudimentary application for prototype testing.
- b) Project II (60%): Emphasizing research and model enhancement, Project 2 extends its scope to encompass a broader array of datasets, including both currency notes and coins. Additionally, it focuses on augmenting features designed to cater to the needs of visually impaired individuals during the application's development.

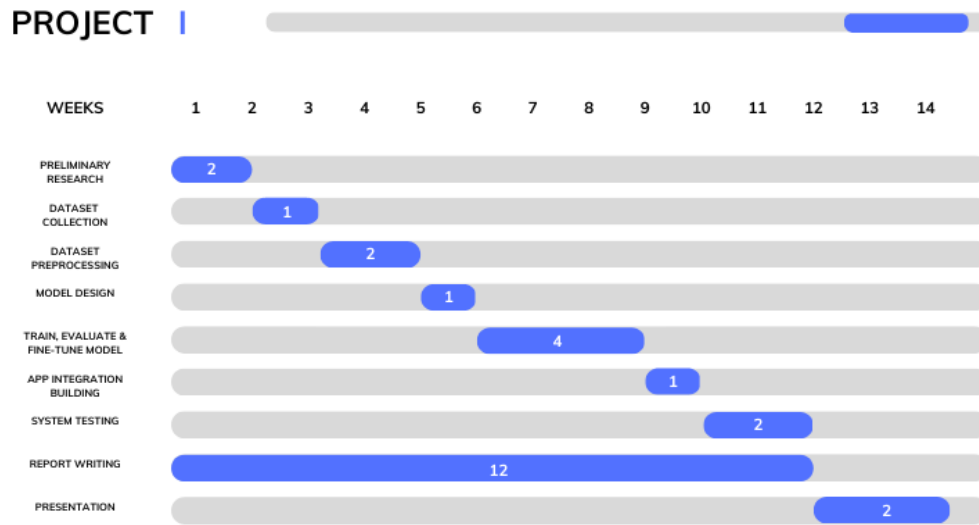


Figure 3.1: Gantt chart of project I timeline

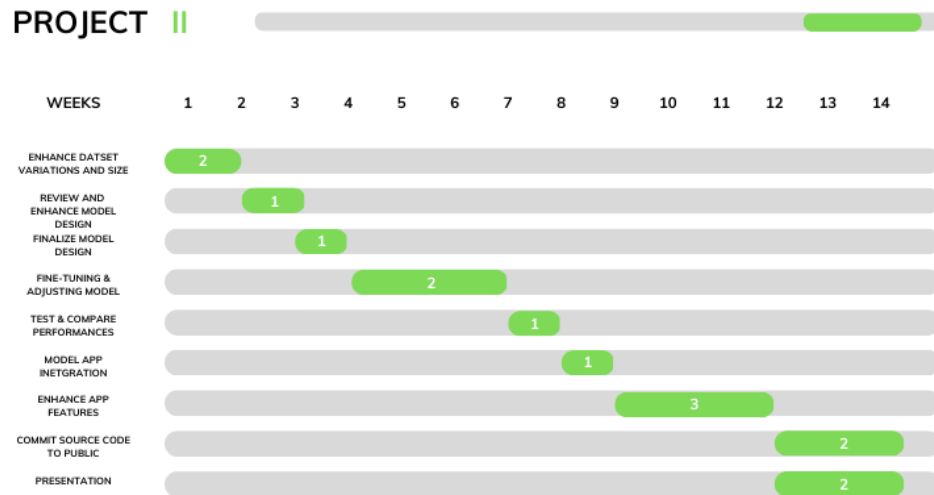


Figure 3.2: Gantt chart of project II timeline

CHAPTER 4: SYSTEM MODEL

As per discussed, the model will be used to detect currency denomination for visually impaired using deep learning approach through classification tasks. There are 2 different models researched in this project, one is MobileNetV2, another would be YOLOV8 (You Only Look Once V8). Based on the information researched and tested overall performance, the one with better performance from all aspects will be chosen and deployed.

4.1 Deep Learning-Based Model

All model researched in this project would be a designed combined deep learning-based model, where both model architecture will be explained and illustrated later, then the top model would be selected to proceed for integration phase.

4.1.1 MobileNetV2 Architecture

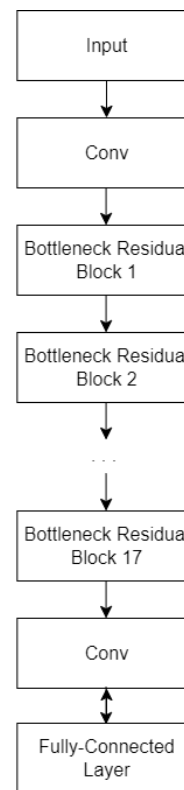


Figure 4.1: *MobileNetV2 Model Architecture*

Figure 4.1 shows the overall implementation of the MobileNetV2 deep learning model architecture. The model starts with the an input layer which accepts image input of shape (224, 224, 3) where 224 as the height and width, followed by 3 as the number of channels, or also known as number of dimension. This layer serves to pass the input image to the subsequent layers of the network, which is the first convolutional (conv) layer. This convolutional layer is 3x3 with a stride of 2, followed by a batch normalization (BN) layer and ReLU activation function. BN layer is applied to normalize the output from the previous layer, through standardizing the activations of previous layer by negating the batch mean and dividing by the batch standard deviation. Through this, the covariate shift, the fluctuations of distribution of input data during training could be minimized. After batch normalization, the ReLU (Rectified Linear Unit) activation function is used, where it provides more complexity and non-linearity to the network model compared to linear function, through setting all negative values to zero, which increased the network's representational power. The input data is now transformed into feature map format.

Next, a new layer architecture called Inverted Bottleneck Residual Layer of MobileNetV2 is introduced which is unseen in MobileNetV1. The layer architecture is designed as followed in Figure 4.2:

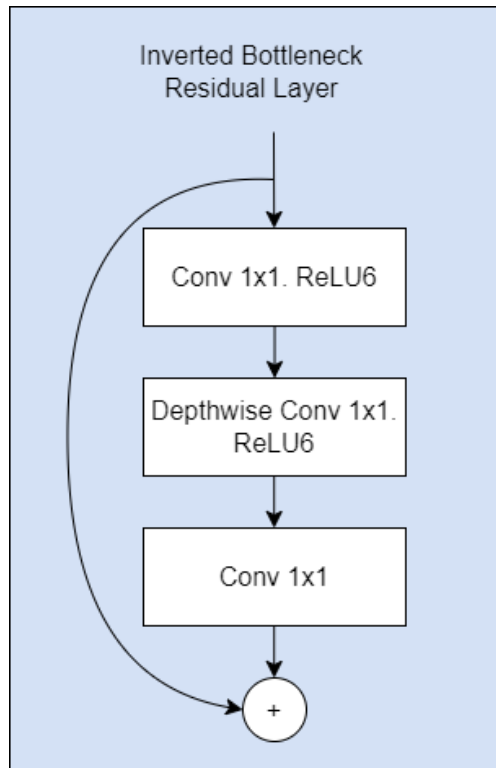


Figure 4.2: *Inverted Bottleneck Residual Layer Architecture*

From Figure 4.2 above, the layers involved are designed to increase the model's efficacy and precision by reducing the number of parameters while maintaining a high level of performance. The Inverted Bottleneck Residual Layer consists of three main components: the depthwise convolution layer, the pointwise convolution layer, and the skip connection.

The Pointwise convolution layer (Conv1 1x1.ReLU6) is the first component of the Inverted Bottleneck Residual Layer, followed by ReLU activation function with max value of 6. This is a 1 x 1 convolutional layer that performs a linear transformation on the input feature maps. It helps to increase the number of output channels while keeping the spatial dimensions the same, to reduce the computational cost of the convolution operation. The output tensor of this layer has a shape of $H \times W \times t$, where H and W are the height and width of the input feature maps, and t is the number of output channels.

The Depthwise convolution layer (Depthwise Conv 1x1.ReLU6) is the second component of the Inverted Bottleneck Residual Layer, followed by ReLU6 activation

CHAPTER 4

function. It serves as a spatial convolution that applies a different filter to each input channel. It helps to reduce the number of parameters and computational complexity while retaining the spatial information. The output tensor of this layer has a shape of $H' \times W' \times t$, where H' and W' are the height and width of the output feature maps, and t is the number of output channels.

After the Depthwise convolution layer, the Inverted Bottleneck Residual Layer will include the final operation, another Pointwise convolution layer, but this time with no ReLU6 activation function. This is due to the number of dimensions in from the output of this Pointwise conv layer will be less. When an output with less number of dimensions is passed to non-linear ReLU6 activation function, it will incur info loss of the Manifold of Interest (MoI) from the model, which will deteriorate the model's performance.

The final step of the Inverted Bottleneck Residual Layer is the addition operation, where the output tensor from the Pointwise Convolution (Conv1 1x1) is added to the input tensor of the Inverted Bottleneck Residual Layer. This is called the residual connection that helps to preserve the information from the input tensor while also allowing for deeper networks without the problem of vanishing gradients. The output tensor of this layer has the same shape as the input tensor, the output tensor can be passed to next Inverted Bottleneck Residual Layer to continue.

The inverted residual block is repeated several times, increasing the number of channels gradually, to help in reducing the computational cost while retaining the performance of the network. The output tensor of this layer has a shape of $7 \times 7 \times 1280$. After the multiple inverted bottleneck residual layers, the fourth layer would be a 1×1 convolutional layer that is used to reduce the number of dimensions. This layer is followed by a global average pooling layer that reduces the spatial dimensions to a single vector of length 1280.

Lastly, the single vector is passed to a fully connected layer with softmax activation function. The output of this layer is a probability distribution over the classes

of the problem being solved. In the case of object classification, the output tensor has a shape of $1 \times 1 \times n$, where n = number of classes.

4.1.2 You Only Look Once V8 (YOLOv8) Architecture

YOLOv8, a recently introduced object detection architecture as of January 10 by Ultralytics, is relatively new in the field. Due to the absence of an official paper, there are limited online resources available. The following discussion of YOLOv8 relies on findings from studying its individual components, allowing for a comprehensive understanding of this architecture.

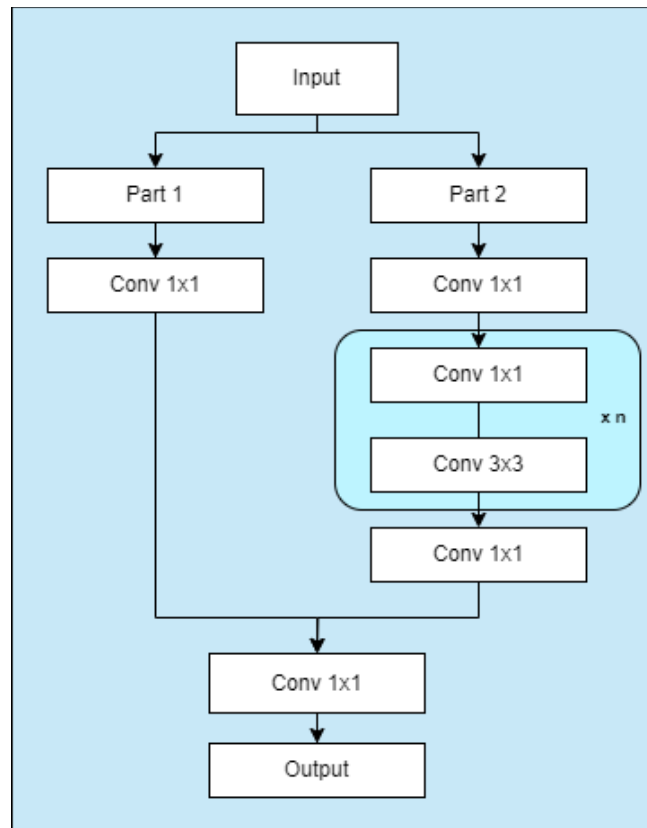


Figure 4.3: CSPDarknet53 Architecture

The YOLOv8 architecture starts with a strong foundation known as the backbone network, which is similar to its predecessor, YOLOv5. This backbone relies on the CSPDarknet53 feature extractor as shown in figure 4.3, specifically designed to extract features from input images. CSPDarknet53 uses a series of layers and operations to

progressively capture features at different levels of detail. This includes both small details and the bigger picture, allowing the network to recognize objects effectively in various scenes.

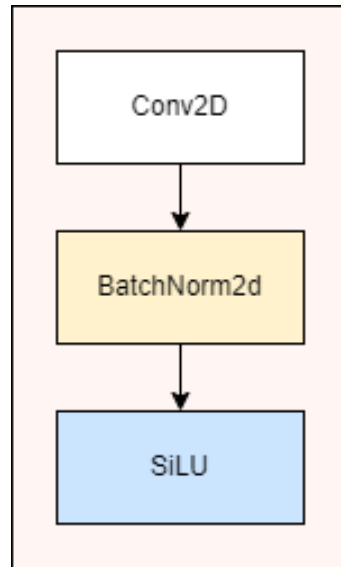


Figure 4.4: *Conv Layer Architecture*

$$\text{silu}(x) = x * \sigma(x)$$

Figure 4.5: *SiLU Activation Function*

Convolutional layers constitute the foundational components extensively integrated throughout the YOLOv8 architecture. Each of these layers is enhanced with two crucial elements: batch normalization layer and SiLU activation function. Batch normalization layer assumes the role of normalizing the activations of each layer, effectively mitigating issues related to covariate shift during the training process. Concurrently, the inclusion of Sigmoid Linear Unit (SiLU) activation function as shown in figure 4.5, which serves as a variant of the ReLU, introduces vital non-linearity into the model's operations. This non-linearity then equips the network with the capacity to adeptly learn and represent intricate data patterns.

To make the model perform well in object detection, YOLOv8 uses specialized loss functions tailored to each task. For accurately pinpointing the location of objects

(bounding box regression), it employs advanced loss functions like Complete Intersection Over Union (CIoU) and Distribution Focal Loss (DFL). In this project, we will employ YOLOv8 for classification task, at this point, YOLOv8 uses binary cross-entropy loss to ensure the model learns to classify objects effectively.

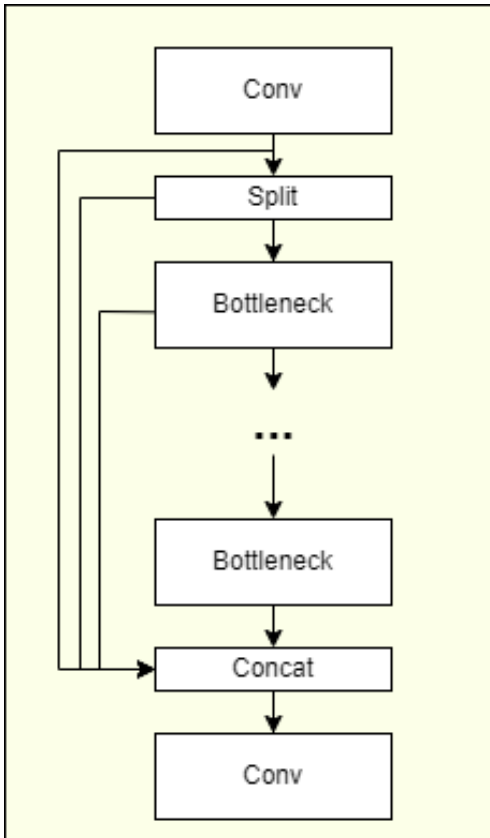


Figure 4.6: *C2f Layer Architecture*

One significant improvement in YOLOv8 is the replacement of the CSP layer in YOLOv5 with the C2f layer as illustrated in figure 4.6. This change plays a vital role in enhancing how the network represents features by combining information from different parts of the network. The C2f module merges important features with context through *concat* layer, increasing model performance at object detection. This shift in architecture eventually helps the model in insight extraction from its feature maps as well.

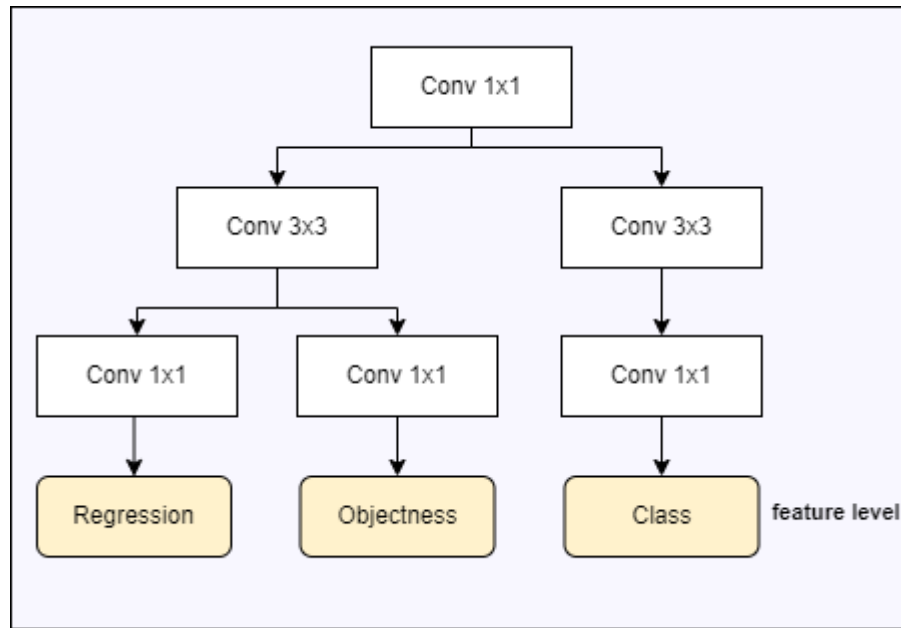


Figure 4.7: *Decoupled Head Architecture*

Additionally, YOLOv8 stands out by using an anchor-free approach instead of the traditional anchor-based method in object detection. In this approach, YOLOv8 uses a decoupled head, illustrated in figure 4.7, which means that it processes three key tasks independently: figuring out if there's an object (*objectness*), classifying the object (*class*), and determining the object's location (*regression*). This separation allows each part of the model to focus on its job, making the model more flexible and potentially better at detecting objects. Getting rid of anchor boxes allows the model to adapt dynamically to different objects in the scene.

In terms of speed and efficiency, YOLOv8 also implements something called Spatial Pyramid Pooling Fast (SPPF) layer as shown in figure 4.6 below:

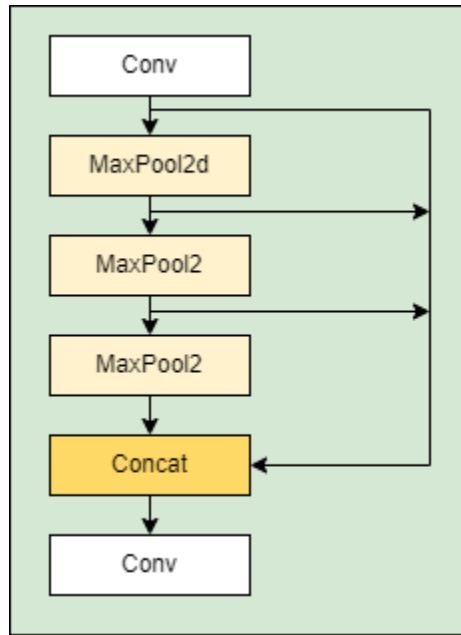


Figure 4.8: *Spatial Pyramid Pooling Fast (SPPF) Layer Architecture*

The SPPF layer includes 3 MaxPool2d layer, where the output will then be pass through a concatenation layer for merging purpose. This full layer aids in speeding up the calculations while making the model work more efficiently. This is achieved by organizing features into a fixed-size map, which makes processing these features faster. This speed boost is crucial, especially in applications where the model needs to make quick decisions in real-time. The SPPF layer's talent for managing features efficiently plays a significant role in making the model faster and more responsive.

In the output layer of YOLOv8, specific activation functions are used for different prediction tasks. Since this project focus on cls model, YOLOv8 uses the softmax function to compute class probabilities, where the highest probabilities among classes would be the predicted classified output.

4.2 Model Selection

Due to resource extensive environment, the selection of an appropriate deep learning model is important in the development of a system catering to the visually impaired community. In this project, two distinct models, MobileNetV2 and YOLOv5, were considered for their potential contributions.

MobileNetV2 is acknowledged for its lightweight and efficient architecture, making it an attractive choice for mobile applications. Its swiftness in delivering inference results is particularly advantageous for real-time applications. Moreover, MobileNetV2 exhibits a commendable attribute by consuming minimal memory resources, ensuring optimal performance even on devices with constrained resources. However, there are trade-offs. While MobileNetV2 excels in efficiency, it does so at the cost of slightly lower accuracy. This compromise in accuracy might affect its performance in tasks requiring intricate classification. Additionally, MobileNetV2's performance have been evaluated in project I, where the speed is efficient, but the result performance of the model is moderately low.

In contrast, YOLOv8 stands out with its remarkable accuracy, especially in tasks such as object detection and classification, making it a pivotal choice for precisely identifying banknotes. It is worth to note that, YOLOv8's versatility is a significant advantage, where it offers a comprehensive model solution that combines accuracy and speed. Its robust object detection capabilities are particularly valuable for distinguishing intricate features on banknotes. Nevertheless, YOLOv8's resource requirements are relatively higher, which may affect its performance on less powerful mobile devices. Additionally, it might exhibit slightly longer inference times when compared to MobileNetV2.

Despite these considerations, the prioritization of accuracy and classification capabilities in this context positions YOLOv8 is chosen as the preferred model for subsequent development phases, aligning more closely with the primary objective of providing visually impaired individuals with reliable assistance in banknote denomination identification.

CHAPTER 5: SYSTEM DESIGN

This chapter documented the subsequent development phase of the whole project. The main deliverable of this phase would be a usable visually impaired user oriented mobile app, which offers easy to access, and high mobile availability for them. By making use the trained model previously, the model will be integrated into the mobile app, which contains three package, which are input package to accept user input in image form, inference package to generate inferenced output and output package to deliver inferenced output to user.

5.1 System Block Diagram

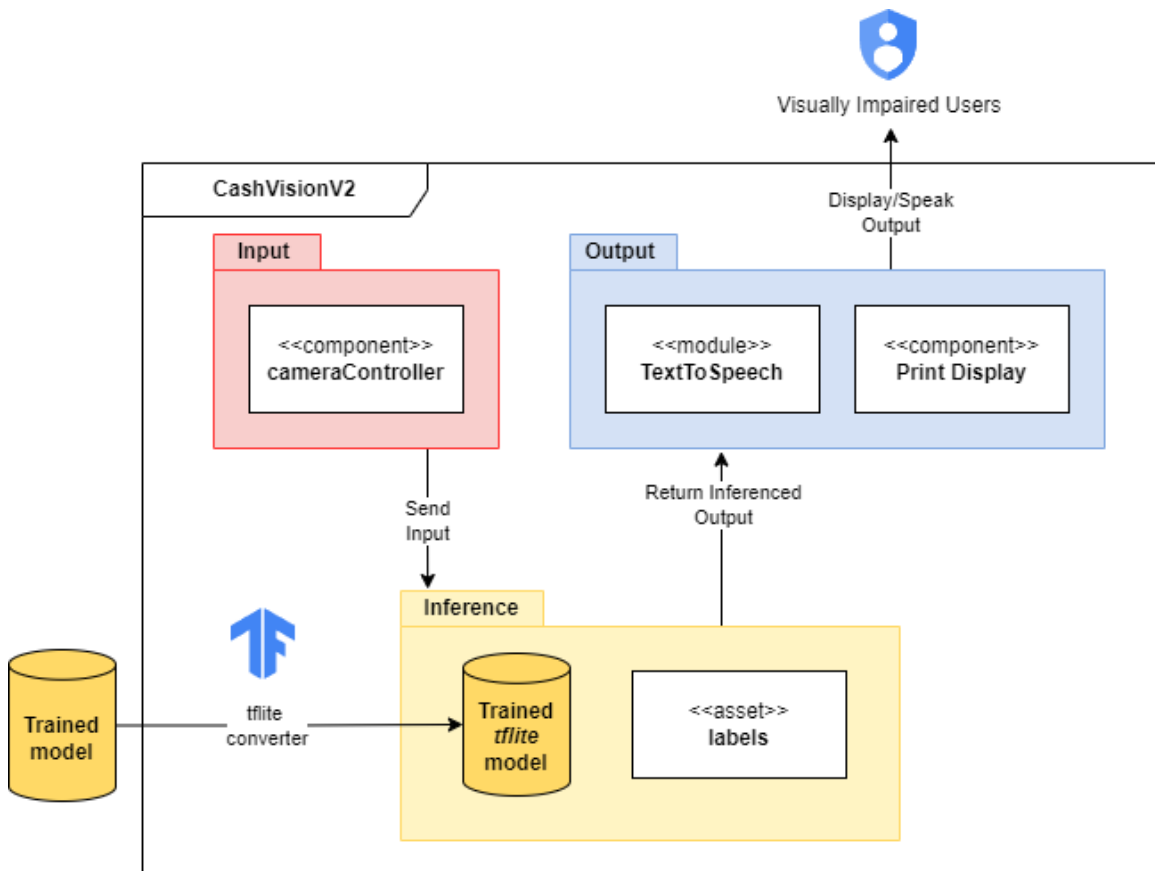


Figure 5.1: System Block Diagram of development phase

To start with the development phase, figure 5.1 shows system block diagram for integration of trained model into a mobile app namely "CashvisionV2" to cater to

visually impaired users. It starts with converting a trained model into a mobile-compatible TFLite format for efficient use. The system is deployed using the Flutter framework, making it compatible with both Android and iOS devices, making it cross-platform compatible.

User interaction begins with the Input package, which contains CameraController component that displays camera preview for user to capture cash notes or coins to detect. These images are then passed to the Inference Package, which contains the Converted TFLite Model and necessary labels for quick inference. After all, the subsequent Output package consists of two components: the Text-to-Speech Module and the On-Screen Display. These modules work together to convey the predicted currency identification to visually impaired users in real-time, through voice feedback and text feedback, making it a dual-output system.

5.2 Model Conversion & Inference Package

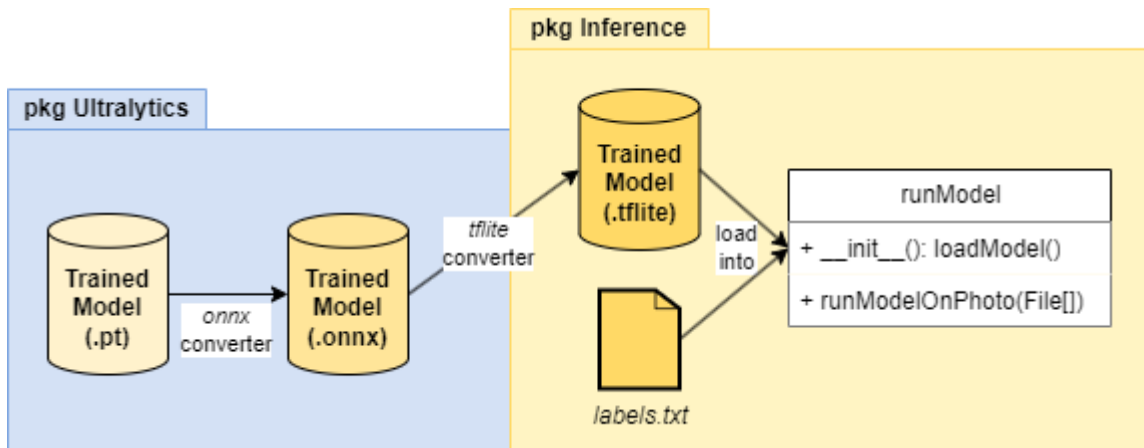


Figure 5.2: Class Diagram for Inference Package & Model Conversion

The inference package design for CashVisionV2 application is as shown in figure 5.2, starting from model conversion. Utilizing Ultralytics' service, the PyTorch model is transformed from its native .pt format into ONNX format and subsequently into the TensorFlow Lite (.tflite) model, which is optimized for integration. Within this package,

the 'runModel' class is pivotal, featuring an constructor function ('init'), which is responsible for loading and preparing the TensorFlow Lite model during application startup, ensuring its full availability. Another critical function, 'runModelOnPhoto()', is severe as well by accepting image file input and initiating inference with the loaded model. Following inference, the system will returns inferenced results, which will be passed to next step.

5.3 Input Package

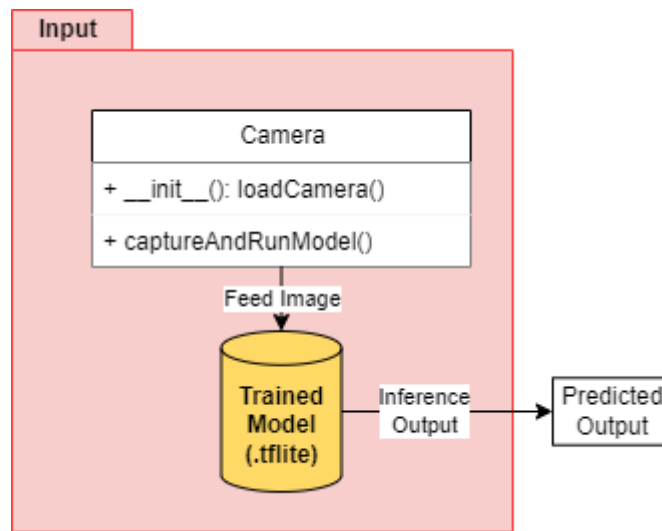


Figure 5.3: *Class Diagram for Input package*

In the design of the input package, we adhere to a structured approach, as illustrated in Figure 5.3. This package commences with the 'camera' class, a fundamental component responsible for enabling image capture. Within the 'camera' class, the 'loadCamera()' function, situated in the constructor ('_init_'), ensures the initialization of the camera capture service. This pivotal function establishes the connection to the native Android/iOS camera, initiating the camera preview for seamless user interaction. The image, once captured by the user, is then routed into the trained TensorFlow Lite (tflite) model for inference processing. This integrated workflow ensures a streamlined pipeline for input processing, culminating in the prompt return of inference results for further processing.

5.4 Output Package

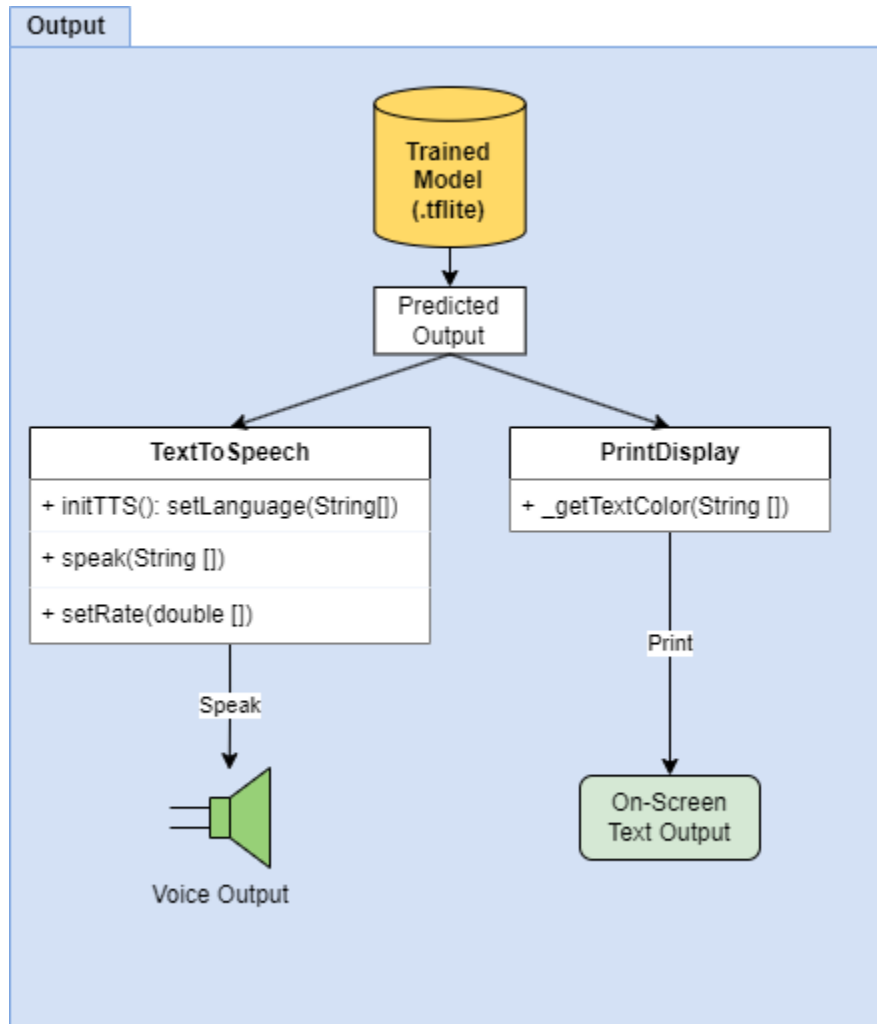


Figure 5.4: *Class Diagram for Output package*

The design of the output package in the CashVisionV2 application, illustrated in Figure 5.4, facilitates two distinct mechanisms for conveying inference results to the user. The primary method involves the utilization of the 'TextToSpeech' class, an integral component endowed with three essential functions. Within this class, the 'setLanguage()' function, positioned in the constructor ('initTTS'), empowers users to switch between different voice languages, ensuring linguistic versatility. Subsequently, the 'speak()' function plays the important role, serving as the mechanism through which the application audibly conveys the desired output to the user via the device's speaker.

CHAPTER 5

Additionally, the 'setRate()' function offers users the flexibility to customize the speed of the spoken voice, aligning with individual preferences.

The second method of conveying output involves the 'PrintDisplay' class, which efficiently manages the visual display of results. This class ensures users receive comprehensive feedback through the on-screen display of textual content. Notably, the class incorporates the '_getTextColor' function, enhancing user engagement by dynamically altering the displayed text's color based on the detected currency. This dynamic visual feedback not only provides essential information but also augments the user interface's intuitiveness, facilitating visually impaired to read the output through colour distinction.

5.5 Use Case Diagram

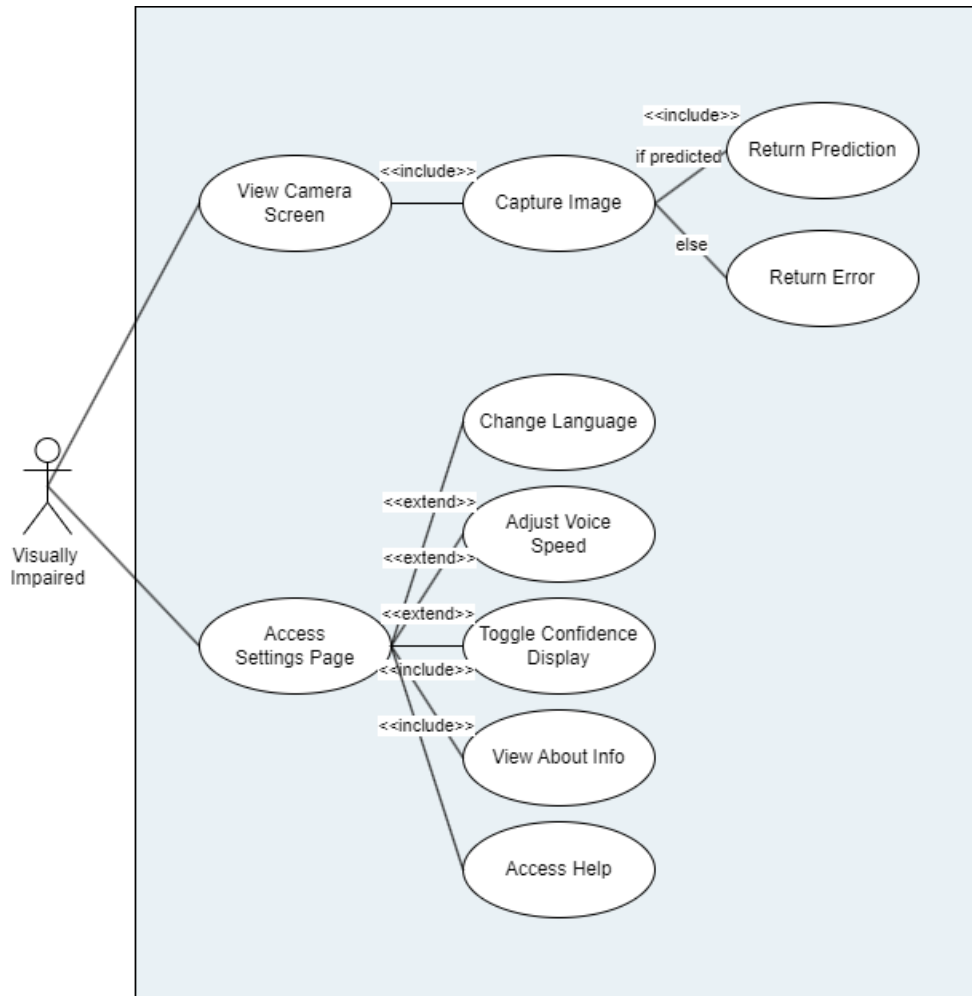


Figure 5.5: Use Case Diagram of proposed design

The use case diagram for the CashVisionV2 application outlines how visually impaired users interact with the app's features. At its core is the visually impaired user, the primary app user. Their journey begins with accessing the camera screen preview, which is critical for images capturing.

In this home interface, two essential actions are defined. The first action is capturing an image, which can lead to receiving predictions if an object is detected in the image. This process ensures that the app promptly recognizes and conveys predicted objects. The second action, also stemming from image capture, deals with scenarios

CHAPTER 5

where no objects are detected. This error feedback mechanism ensures that users remain informed even when detections fail.

While exploring the app, users can access the settings page, where various options are available. The first option allows users to easily change the language, making the app more versatile. The next option is to adjust the voice speed, ensuring that auditory feedback matches user preferences.

Furthermore, users can choose to toggle confidence display, providing options to access insights of the every detection. The "view about info" feature allows users to access essential app information and current version. Lastly, users can access help for guidance when using the app, creating a user-centric experience. Together, these interactions result in an app design that empowers visually impaired users to effectively use CashVisionV2.

5.6 Activity Diagram

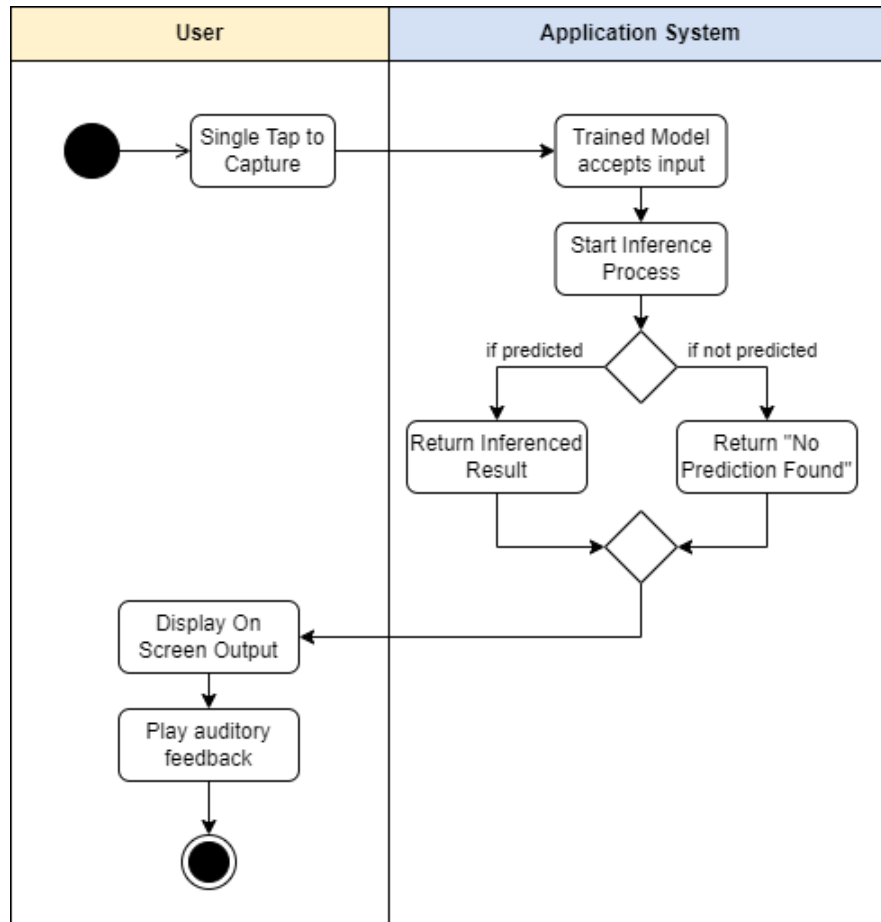


Figure 5.6: Activity Diagram

Figure 5.6 above presents the activity diagram that encapsulates the user interaction flow between user and application interface within CashVisionV2, designed for effortless control by visually impaired individuals. The process commences on the home page, where a single tap action initiates the image capture sequence, ensuring user-friendliness.

The captured image swiftly advances to the application interface, accepted by the trained model. Subsequently, the model initiates the real-time inference process, utilizing the trained model to generate predictions based on the input's contents. In cases where predictions are successfully generated, the respective results are promptly returned to the user.

CHAPTER 5

However, if no predictions are found, the system tactfully communicates this with the user through a prompt stating, "No prediction found." This transparent feedback mechanism ensures that users remain informed even in the absence of predictions. Following prediction generation, the output embarks on a dual output system. On one front, the output is visually displayed on the screen through text, facilitating easy access for visually impaired users. Simultaneously, auditory feedback of the output is provided. This system is particularly beneficial for the visually impaired community which greatly enhance their accessibility to the application.

CHAPTER 6: SYSTEM IMPLEMENTATION

In this chapter, we will delve into the implementation details of the cash denomination detection system. To provide a structured overview, our implementation can be distinctly categorized into two key sections. The initial section is dedicated to the experimental exploration of the cash denomination detection deep learning model, as elaborated upon in Chapter 4. Subsequently, the second section focuses on the integration and deployment of the trained model into a mobile application developed under the Flutter framework, as discussed in Chapter 5. This approach underscores our commitment to realizing an efficient cash denomination detection system.

6.1 Software Setup

Prior to the system model implementation, the environment has to be defined first on the local machine. Firstly, following software, libraries or extensions would be installed at start:

1. Python 3.10
2. Visual Studio Code + Python Extension
3. Jupyter Notebook
4. Flutter Framework
5. Android Studio
6. Pytorch
7. Ultralytics Service

6.2 Implementation of Cash Reader Model

Based on the model design outlined in Chapter 4, the selection process led to the adoption of the YOLOv8 architecture, which was primarily released in January 2023. This choice was made carefully due to its effectiveness and suitability for the cash denomination detection system. In the subsequent phases of implementation, the plan is

to configure and train this selected model in the notebook (.ipynb) format on a local machine using Jupyter Notebook. The results of this process will be recorded as a crucial part of the overall system development.

6.2.1 Datasets Acquisition

In the data acquisition phase of this study, we encountered the challenge of limited available datasets for Malaysian Ringgit Notes and Sen on the internet. To address this constraint, a manual data capture approach was employed. This method involved capturing all datasets using a mobile phone camera, the specification of the camera would be 48MP Sony IMX 582 Quad-Bayer 1/2" sensor with 0.8 μ m pixels, 25mm f/1.8 lens, ensuring a comprehensive representation of the target currency classes.

The captured dataset consists of six classes of Ringgit notes (RM1, RM5, RM10, RM20, RM50, RM100) and four classes of Sen (5 sen, 10 sen, 20 sen, and 50 sen). During the data collection process, images were captured from various angles to simulate real-life conditions. This diverse set of angles was chosen to enhance the model's adaptability and learning capabilities.

Type	Example
<i>White Background</i>	
<i>Dark Background</i>	
<i>Partially Occluded</i>	
<i>Folded</i>	
<i>Noisy Background</i>	

Figure 6.1: *Types and Samples of image data in captured dataset*

To further diversify the dataset and replicate real-world scenarios, six different types of conditions were considered during image capture. These conditions included images with light backgrounds, dark backgrounds, partially occluded notes, folded notes, and notes placed against complex and noisy backgrounds, as illustrated in Figure 6.1. Notably, the coin dataset shares similarities with the note dataset, except that it does not include the folded condition.

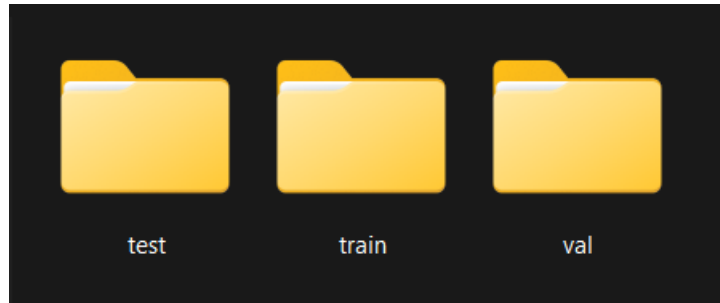


Figure 6.2: *Branch folders of Captured dataset*

Both faces of the currency notes and coins were captured to provide a holistic dataset for the model's training and evaluation. The datasets were organized and stored in three distinct branch folders, namely 'train' for training, 'val' for validation, and 'test' for final testing, as depicted in Figure 6.2.

In total, each class of Ringgit notes was represented by 200 images, each captured under various conditions. For the Sen currency, 140 images were captured for each class. A selection of example images from this diverse dataset is presented in Figure 6.3 and Figure 6.4 below, providing a visual overview of the acquired dataset's richness and complexity.

CHAPTER 6

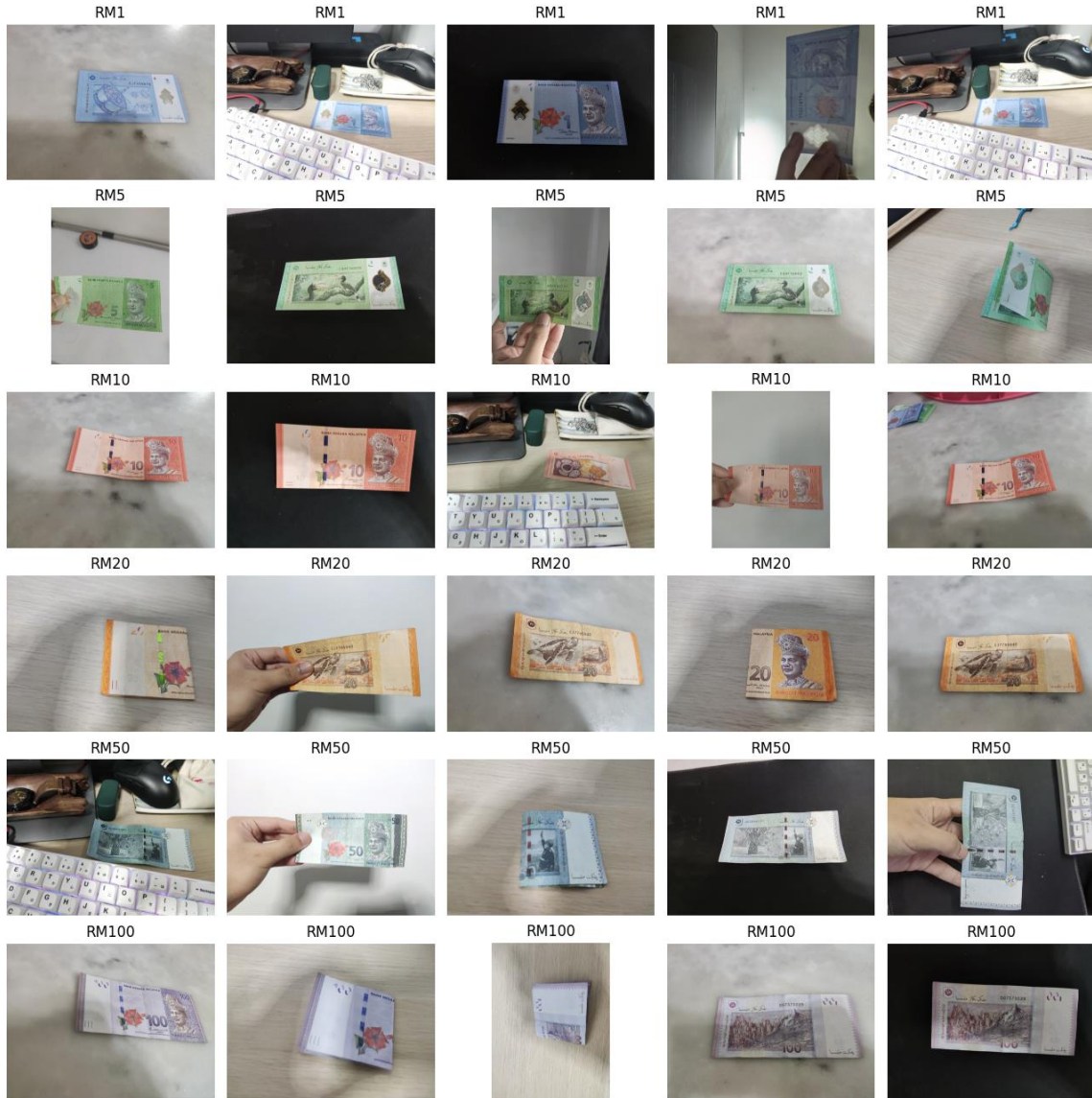


Figure 6.3: Sample images (RM notes) of Captured dataset

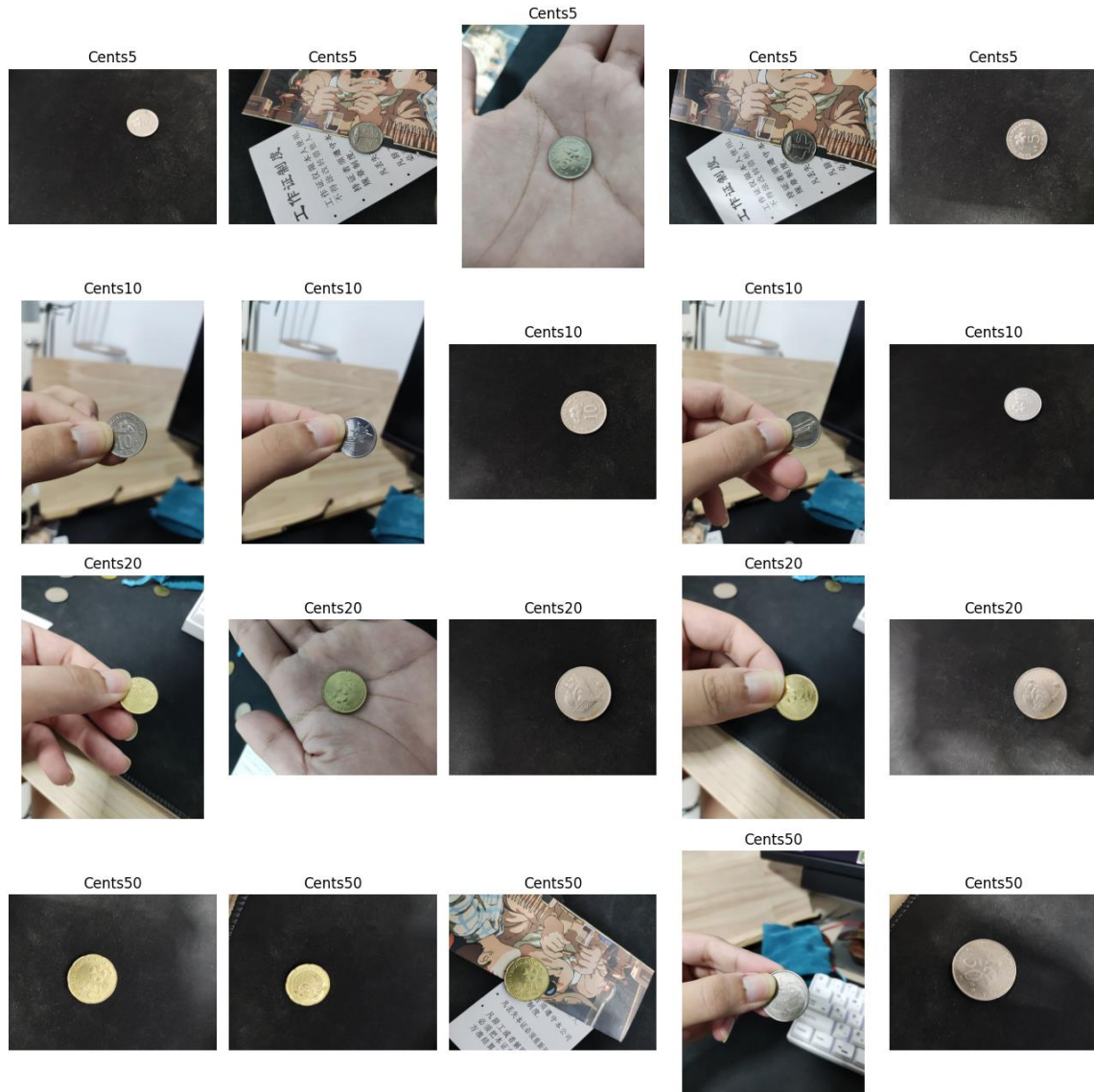


Figure 6.4: *Sample images (sen) of Captured dataset*

6.2.2 Data Pre-processing & Augmentation

In the data preprocessing and augmentation phase, our primary objective was to expand the initially captured dataset to improve the model's variability and adaptability. To achieve this, a series of data augmentation techniques were applied to the existing dataset, enhancing its diversity and robustness. The initial step in this process involved resizing all images to a standardized format of 224x224 pixels. This standardization not

only ensured uniformity across the dataset but also facilitated compatibility with the model architecture for training purposes.

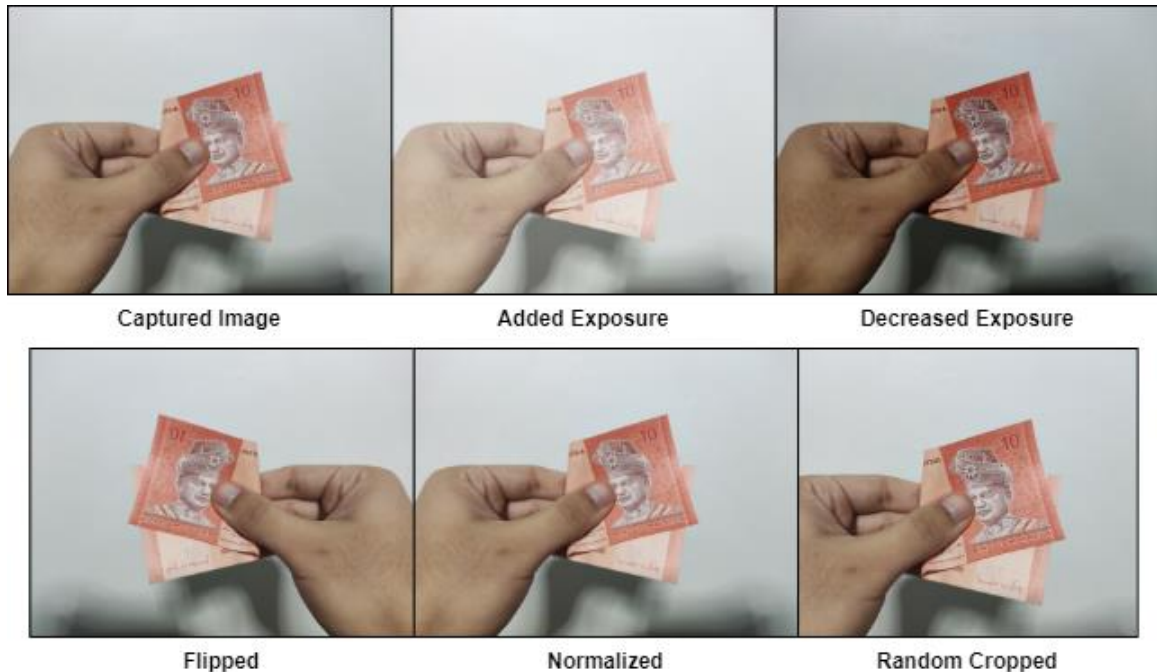


Figure 6.5: *Example of Data Augmentation*

Subsequently, six distinct augmentation techniques as shown in Figure 6.5 were employed to augment the dataset:

- 1) **Flipping:** The 'fliplr()' function from the NumPy library is utilized to randomly flip images horizontally. This augmentation method effectively created mirrored versions of the original images, mitigating the model's sensitivity to orientation changes that can occur in real-world scenarios.
- 2) **Exposure Variation:** Given the constant variability in lighting conditions, it was crucial to expose the model to a diverse range of lighting scenarios. To achieve this, versions of images with increased and decreased exposure are produced, ensuring that the model became familiar with various lighting conditions.
- 3) **Salt and Pepper Noise:** This augmentation involved introducing minor disruptions to the images by randomly adding small clusters of 1x1 or 2x2 pixel

regions in different shades. This simulated noisy input, enhancing the model's ability to handle background noise effectively.

- 4) **Normalization:** To enhance image contrast and maintain pixel value consistency, all photos were normalized to a range of 0 to 255. This normalization process played a pivotal role in optimizing image contrast and feature extraction.
- 5) **Random Size Crop:** Images were subjected to random size cropping, introducing variability in the dataset. This augmentation technique greatly improved the model's generalization capabilities, making it more robust and less susceptible to overfitting. Additionally, it rendered the model invariant to translations occurring within the input frame.

Following the completion of augmentation and pre-processing, each class of Ringgit notes now comprises 1,400 images, while every Sen class includes 980 images. This augmentation process has substantially expanded the dataset, resulting in a total of 12,320 images. The resultant dataset is now ready and may proceed to next step.

6.2.3 Training of YOLOv8-cls Model

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	7360	ultralytics.nn.modules.block.C2f	[32, 32, 1, True]
3	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
4	-1	2	49664	ultralytics.nn.modules.block.C2f	[64, 64, 2, True]
5	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
6	-1	2	197632	ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	460288	ultralytics.nn.modules.block.C2f	[256, 256, 1, True]
9	-1	1	335364	ultralytics.nn.modules.head.Classify	[256, 4]

YOLOv8n-cls summary: 99 layers, 1443412 parameters, 1443412 gradients

Figure 6.6: Overview of YOLOv8-cls Model

In this phase of the project, we employed the transfer learning approach, specifically utilizing the YOLOv8-cls model, where "cls" denotes a classification model. The choice of YOLOv8-cls aligns with its suitability for our tasks. Since the YOLOv8 model was recently introduced by Ultralytics, the YOLO package from Ultralytics is imported for this training. A concise overview of the YOLOv8-cls model is presented in

Figure 6.6, while the detailed YOLOv8-cls architecture is available in the report's appendix.

The model comprises a total of 99 layers, with convolutional layers (conv), convolutional-to-flatten layers (c2f), and the final classification layer being the primary layer types, as previously discussed in Chapter 4. These layers collectively contribute to a model with 1,443,412 parameters. In our training process, conventional layer freezing methods, such as freezing the front layers and altering the output Fully Connected (FC) layer, proved ineffective due to our reliance on Ultralytics' training services.

```
def freeze_layer(trainer):
    model = trainer.model
    num_freeze = 5
    print(f"Freezing {num_freeze} layers")
    freeze = [f'model.{x}.' for x in range(num_freeze)] # layers to
freeze
    for k, v in model.named_parameters():
        v.requires_grad = True # train all layers
        if any(x in k for x in freeze):
            print(f'freezing {k}')
            v.requires_grad = False
    print(f"{num_freeze} layers are frozen.")
```

Figure 6.7: *Overview of YOLOv8-cls Model*

To address this challenge, we adopted a layer freezing approach through a callback function, as depicted in Figure 6.7. Specifically, we froze the first 5 layers (0-4) of the model. During training, the freeze_layer function iterates through named parameters, identifying and setting the requires_grad parameter to False for the specified layers to reduce the training load and streamline the optimization process. This approach ensures the model's focus on learning discriminative features relevant to our specific tasks, yielding more efficient training outcomes. In results, Figure 6.8 below provides a visual representation of the first 5 layers (0-4) marked for freezing.

```
Freezing 5 layers
freezing model.0.conv.weight
freezing model.0.bn.weight
freezing model.0.bn.bias
freezing model.1.conv.weight
freezing model.1.bn.weight
freezing model.1.bn.bias
freezing model.2.cv1.conv.weight
freezing model.2.cv1.bn.weight
freezing model.2.cv1.bn.bias
freezing model.2.cv2.conv.weight
freezing model.2.cv2.bn.weight
freezing model.2.cv2.bn.bias
freezing model.2.m.0.cv1.conv.weight
freezing model.2.m.0.cv1.bn.weight
freezing model.2.m.0.cv1.bn.bias
freezing model.2.m.0.cv2.conv.weight
freezing model.2.m.0.cv2.bn.weight
freezing model.2.m.0.cv2.bn.bias
freezing model.3.conv.weight
freezing model.3.bn.weight
freezing model.3.bn.bias
freezing model.4.cv1.conv.weight
freezing model.4.cv1.bn.weight
freezing model.4.cv1.bn.bias
freezing model.4.cv2.conv.weight
freezing model.4.cv2.bn.weight
freezing model.4.cv2.bn.bias
freezing model.4.m.0.cv1.conv.weight
freezing model.4.m.0.cv1.bn.weight
freezing model.4.m.0.cv1.bn.bias
freezing model.4.m.0.cv2.conv.weight
freezing model.4.m.0.cv2.bn.weight
freezing model.4.m.0.cv2.bn.bias
freezing model.4.m.1.cv1.conv.weight
freezing model.4.m.1.cv1.bn.weight
freezing model.4.m.1.cv1.bn.bias
freezing model.4.m.1.cv2.conv.weight
freezing model.4.m.1.cv2.bn.weight
freezing model.4.m.1.cv2.bn.bias
5 layers are frozen.
```

Figure 6.8: *Layer Freezing of the first 5 layers (0-4)*

Table 6.1: *Training Params Configuration*

Parameter	Config
<i>epochs</i>	20
<i>batch</i>	16
<i>workers</i>	8
<i>optimizer</i>	Adam
<i>learning rate</i>	0.01
<i>verbose</i>	True
<i>momentum</i>	0.937
<i>weight decay</i>	0.0005

With the model architecture and data preparation in place, we are now prepared to commence the training process. Table 6.1 outlines key training parameters that play important roles in shaping the training process and subsequently the model's performance.

- i. **Epochs** (Epochs = 20): The term "epochs" refers to the number of times the entire training dataset is passed forward and backward through the neural network. In our case, we set this parameter to 20, implying that our training process will iterate over the entire dataset 20 times.
- ii. **Batch** (Batch = 16): The batch size signifies the number of training examples utilized in one forward and backward pass. A batch size of 16 shows that during each training iteration, 16 images are processed.
- iii. **Workers** (Workers = 8): The workers parameter defines the number of CPU cores or processes allocated for data loading during training. With a value of 8, our training process can efficiently utilize 8 CPU cores at the same time.
- iv. **Optimizer** (Optimizer = Adam): The optimizer, in this context, refers to the optimization algorithm used to update the model's weights during training. Adam optimizer is chosen in this case, which is a popular optimizer algorithm due to its adaptability and effectiveness in optimizing deep neural networks.

CHAPTER 6

- v. **Learning Rate** (Learning Rate = 0.01): The learning rate determines the step size at which the optimizer adjusts the model's weights during training. A learning rate of 0.01 indicates moderate step sizes, striking a balance between convergence speed and stability.
- vi. **Verbose** (Verbose = True): The "verbose" parameter, set to "True," enables the model to display training progress and metrics during the training process. This real-time feedback aids in monitoring the model's performance and convergence.
- vii. **Momentum** (Momentum = 0.937): Momentum is a parameter that influences the optimizer's ability to escape local minima during training. The momentum value at this case is set to 0.937.
- viii. **Weight Decay** (Weight Decay = 0.0005): Weight decay, also known as L2 regularization, is a regularization technique that discourages large weight values in the neural network. A weight decay of 0.0005 introduces a mild regularization effect, helping to prevent overfitting.

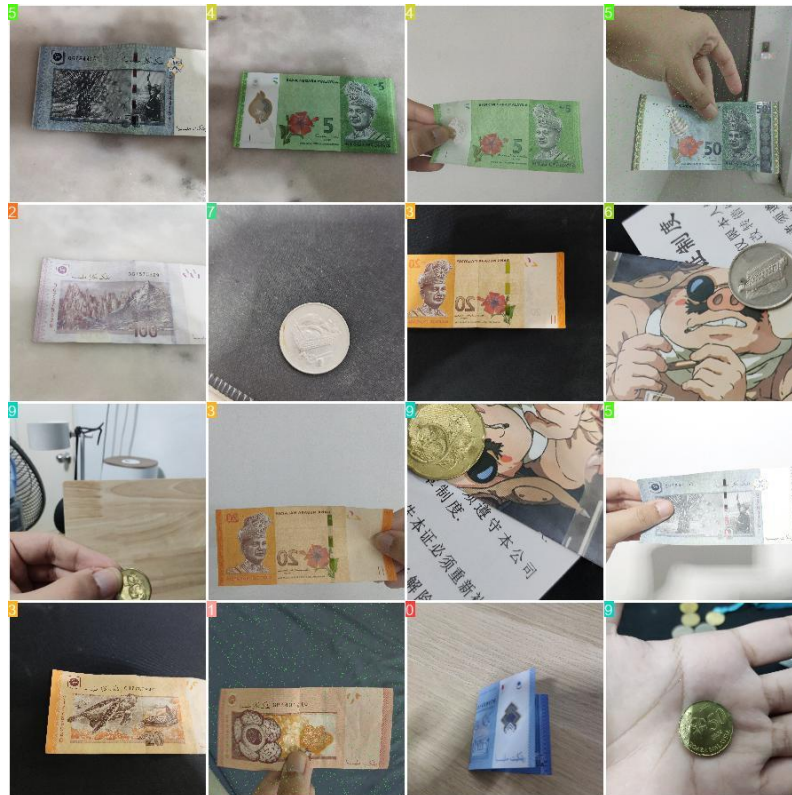


Figure 6.9: Samples of 1st batch training

CHAPTER 6

The full training process, executed on an RTX 3060Ti graphics card, typically concludes in approximately 1.5 hours, resulting in a well-trained model. Figure 6.9 visually presents few samples of the first batch training, offering insights into the model's learning progress. Before training phase ends, an `.export()` function from Ultralytics is employed to convert the trained model from pytorch extension to ONNX extension, finally to tflite extension, so that it could compatible for mobile integration. With the successful completion of the training phase, we are able to proceed into the development phase, where the trained model will be integrated into the application for real-world use.

6.3 Development of Cash Detection App under Flutter framework

In the development phase of our project, our primary goal at this point is to integrate and deploy the previously trained deep learning model into a mobile application, designed to be easily accessible to visually impaired users. This approach make use of the ubiquity and user-friendly nature of mobile phones as an ideal platform for our Malaysian cash denomination detection system. In this project, the cash reader will be developed under Flutter framework, and the programming language used will be Dart language.

6.3.1 Model Pre-Setup

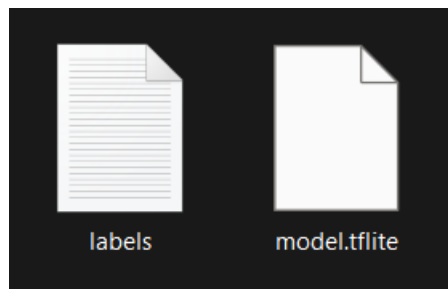


Figure 6.10: *Structure of assets folder*

```
loadModel() async {  
  await Tflite.loadModel(  
    model: "assets/model.tflite",  
    labels: "assets/labels.txt",  
  );  
}
```

Figure 6.11: *loadModel()* Function

The mobile app development begins with the creation of a new Flutter project named "cashvisionv2." Within this project structure, the trained model and its associated labels are placed under the "assets" folder, as illustrated in Figure 6.10. Both files shall also be declared in pubspec.yaml dart file before use. This setup ensures that the model is readily available to the application during runtime started. With the model files in place, executing "flutter pub get" prepares the environment for further integration. The "loadModel()" function, imported from the "tflite" package (as seen in Figure 6.11), plays a crucial role in initializing and preparing the model for inference.

```

loadModel() {
  cameraController = CameraController(
    cameras![0],
    ResolutionPreset.medium,
    imageFormatGroup: ImageFormatGroup.yuv420,
  );
  cameraController!.initialize().then((value) {
    if (!mounted) {
      return;
    } else {
      setState(() {
        cameraController!.startImageStream((imageStream) {
          // Unused camera stream
        });
      });
    }
  });
}
}

```

Figure 6.12: *loadCamera()* Function

To utilize the power of real-time image capture, the "loadCamera()" function as depicted in Figure 6.12 is employed. This function is integrated into the application's constructor (__init__), ensuring that the camera module is ready for action, where a camera preview will appear in home screen as soon as the app is launched.

```

void captureAndRunModel() async {
  if (cameraController != null && cameraController!.value.isInitialized) {
    setState(() {
      capturedImage = null;
      isProcessing = true;
      output = '';
    });

    await cameraController!.setFlashMode(FlashMode.auto); // Turn off flash
    await cameraController!
      .setFocusMode(FocusMode.locked); // Set locked focus mode
    await cameraController!
      .setExposureMode(ExposureMode.auto); // Set auto exposure mode

    final XFile? photo = await cameraController!.takePicture();
    if (photo != null) {
      runModelOnPhoto(File(photo.path));
    }
  }
}

```

Figure 6.13: *captureAndRunModel() Function*

The next function of this application lies in the "captureAndRunModel()" function (Figure 6.13). This function serves as the trigger for the image capture process, delivering the captured image to the "runModelOnPhoto()" function (Figure 6.14) for subsequent inference. During this phase, the camera controller undergoes necessary configurations, including setting the flash mode to auto, the focus mode to locked, and the exposure mode to auto, which is prepared for input capturing.

6.3.2 Model Inference

```

Future<void> runModelOnPhoto(File photoFile) async {
    final predictions = await Tflite.runModelOnImage(
        path: photoFile.path,
        imageMean: 127,
        imageStd: 1,
        numResults: 2,
        threshold: 0.4,
    );

    setState(() {
        capturedImage = photoFile;
        isProcessing = false;

        if (predictions != null && predictions.isNotEmpty) {
            output = predictions[0]['label'];
            double confidence =
                predictions[0]['confidence']; // Extract confidence level

            if (displayConfidence) {
                output += '\nConfidence: ${(confidence * 100).toStringAsFixed(2)}%';
            }

            TextToSpeech.speak(output);

            if (displayConfidence) {
                HapticFeedback.vibrate();
            }
        } else {
            output = 'No prediction found';
            TextToSpeech.speak(output);
        }
    });
}

```

Figure 6.14: *runModelOnPhoto()* Function

In "runModelOnPhoto()" function, the "runModelOnImage()" function from the "tflite" package is utilized. Its parameters, such as "imageMean" set to 127 and "imageStd" set to 1, standardizing the input image. The "numResults" parameter specifies the expected number of inference outcomes (in this case, 2), and the "threshold" parameter ensures that only predictions with confidence values exceeding 0.4 are considered.

Upon generating inference results, our application identifies the prediction with the highest confidence value. The predicted output is then displayed on the screen, and the "speak()" function from the "TextToSpeech" module is employed to provide auditory feedback through the device's speaker, catering to visually impaired users. Furthermore, haptic feedback is also triggered to enhance the user experience, which will be vibration in this case. In cases where no prediction is found, the application handles the situation by presenting an "No prediction output" alert to the user.

```

Color _getTextColor(String output) {
    if (output.contains('1 Ringgit Malaysia')) {
        return AppColors.c1;
    } else if (output.contains('10 Ringgit Malaysia')) {
        return AppColors.c10;
    } else if (output.contains('100 Ringgit Malaysia')) {
        return AppColors.c100;
    } else if (output.contains('5 Ringgit Malaysia')) {
        return AppColors.c5;
    } else if (output.contains('50 Ringgit Malaysia')) {
        return AppColors.c50;
    } else if (output.contains('20 Ringgit Malaysia')) {
        return AppColors.c20;
    } else {
        return Colors.white70;
    }
}

```

Figure 6.15: *_getTextColor() Function*

For text display output, a noteworthy function is "_getTextColor" (Figure 6.15). This function dynamically determines the color of the notes based on the predicted denomination. This feature greatly enhances the accessibility of the output, particularly for users with visual impairments or those dealing with blurry vision.

6.3.3 Key Features

Additionally, several primary features are introduced to enhance user experience and accessibility, since the target user of this project requires additional attention. Following are some of the key features:

- a) **Single Tap Capture:** This intuitive feature allows users to initiate the capture and inference process with a single tap anywhere on the screen. This simplicity ensures that users can effortlessly interact with the app, promoting a no hesitant interaction with the app.
- b) **Voice Language Selection:** Recognizing the linguistic diversity of our users, we offer a choice of four languages: Chinese, Malay, English, and Tamil. This selection empowers users to interact with the app in their preferred language.
- c) **Speech Rate Control:** It is understandable that each user has unique preferences when it comes to voice feedback speed. To accommodate this, we provide three distinct rates: slow, moderate, and fast. Users can select the rate that best aligns with their preferences, optimizing their interaction with the app.
- d) **Feedback Customization:** User-centric design is at the core of our approach. With the option to include or exclude confidence levels in the displayed output, users can tailor their experience to meet their specific needs.
- e) **TalkBack & VoiceOver Compatibility:** This app is highly designed to ensure compatibility with Android TalkBack and VoiceOver on iOS. This commitment guarantees that visually impaired users with Android or iOS devices can seamlessly integrate the app into their assistive technology ecosystems.
- f) **Haptic Feedback:** Users can rely on haptic feedback, which serves as a subtle vibration, to receive notifications during successful predictions. This tactile feedback reinforces user interactions, enhancing their experience.
- g) **About:** The "About" section provides users with detailed project information, including app version, app details and etcetera. Users can access essential details about the app, project objectives, and more.
- h) **Help and Support:** We recognize that users may encounter challenges or have inquiries. To address this, we offer a "Help and Support" feature. Users can easily access contact information and receive guidance when navigating the app. This support system ensures that users have assistance readily available whenever they need it.

CHAPTER 6

After all, the development phase could be considered completed, and the sample app is also designed to fits target user's needs. Figure 6.16 below shows some sample predictions through the system UI developed.

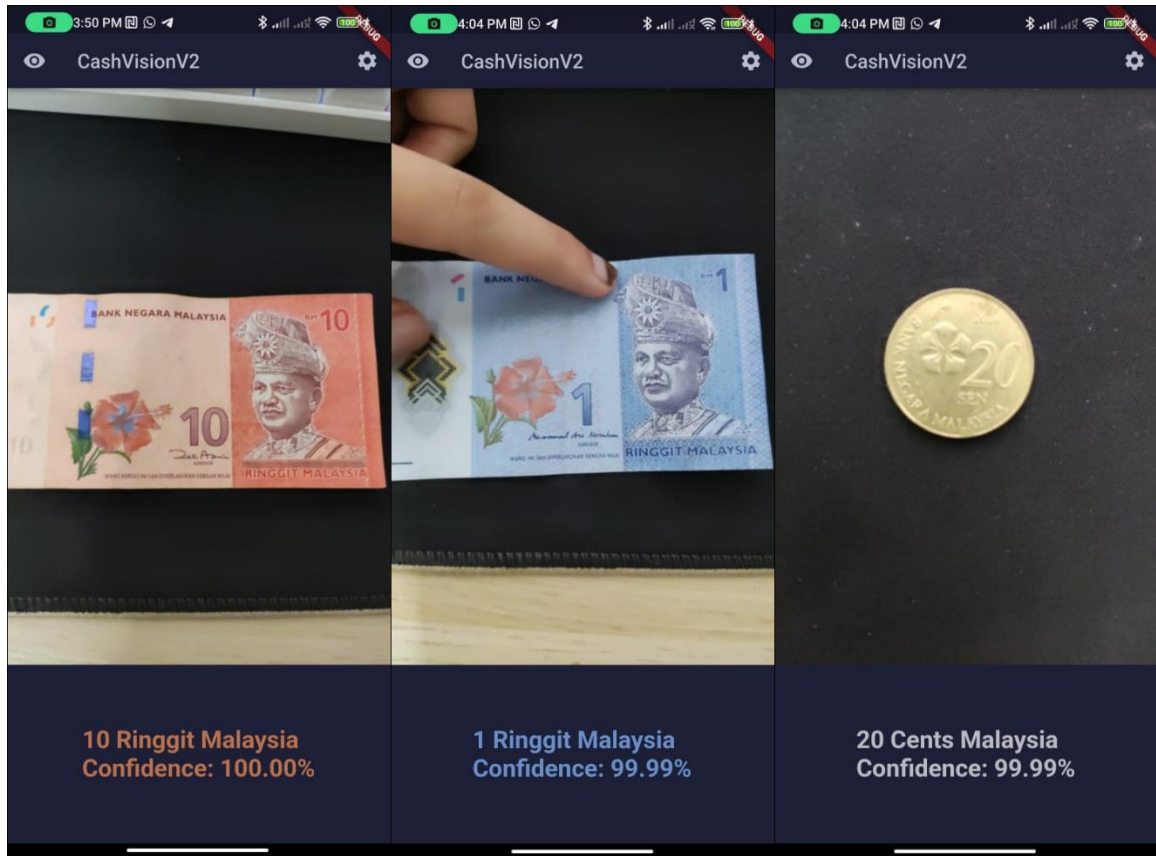


Figure 6.16: *Sample UI of few real time predictions*

6.3.4 Deployment of cash reader Application (CashVisionV2)

In following section, the documentation of the CashVisionV2 development will be recorded, where the full source code will be published publicly on Github. The full project source code can be found in following link, the sample image is displayed in Figure 6.17.

- <https://github.com/Kevinchan9989/CashVisionV2>

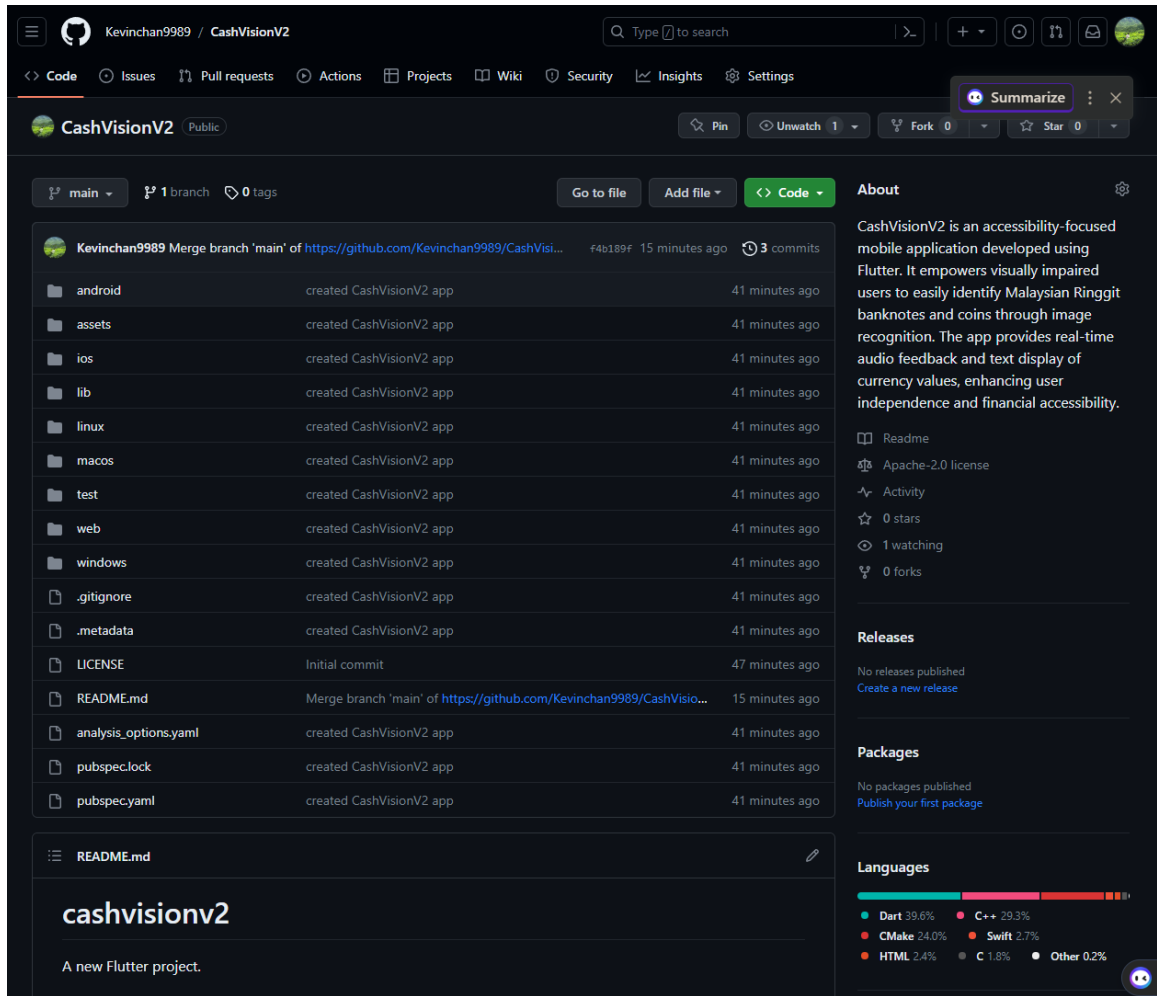


Figure 6.17: Github Repo of CashVisionV2

6.4 Implementation Issues and Challenges

In the implementation phase of our project, we encountered several notable issues and challenges. These challenges required careful consideration and problem-solving to ensure the successful development of our cash detection application.

Firstly, one of the key challenges we faced was the high similarity between the coins dataset. For instance, sen coins like the new pattern and size of the 20 sen and 50 sen coins bear an approximate correspondence. This similarity posed a significant barrier for our model, as it demanded a deep understanding of the context to differentiate between these visually similar denominations accurately.

Another challenge we encountered was related to computing power limitations during the model training phase. Specifically, training the YOLOv8 classification (YOLOv8-cls) model, even after layer freezing, demanded substantial computational resources. Our local machine, equipped with an RTX 3060Ti GPU, faced considerable processing demands during the training process. This challenge extended the time required for model training, as each epoch of training for a moderately small dataset of approximately 10,000 images took around 1.5 hours to complete. This time-intensive aspect of the training process imposed notable time costs, possibly during the experimental tuning and testing phases of our project.

The third challenge we encountered was the requirement for a compulsory paid service—the Google Play Console developer account. This account was necessary for uploading our application to the Google Play Store. However, the associated cost of USD 25 (equivalent to approximately RM 120 after currency conversion) posed a financial issue for students and individuals with limited budgets. This expense highlighted a practical challenge for those looking to publish their applications on a widely accessible platform.

CHAPTER 7: SYSTEM EVALUATION

The following chapter recorded all testing and evaluation of the proposed model, which will be mainly focused on the performance of the cash reader classification model, only few will be tested on the developed mobile application side.

7.1 Performance Metrics

7.1.1 Accuracy Score

$$\textit{Accuracy} = \frac{\textit{True Positive} + \textit{True Negative}}{\textit{All samples}}$$

Figure 7.1: *Formula equation to calculate Accuracy score*

Accuracy is a fundamental metric in the scope of classification tasks, serving as a robust indicator of a model's overall performance. This metric is calculated by dividing number of correctly classified instances, which would be TP and TN in Figure 7.1, by total samples. However, it exhibits certain limitations when faced with imbalanced datasets, a common scenario in real-world applications. In such situations, where one class heavily outweighs the others, accuracy may misleadingly emphasize the majority class while downplaying misclassifications in the minority class, causing imbalanced results. Therefore, it could only be served as an brief overview of the model's performance.

7.1.2 Precision & Recall Score

$$\textit{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

Figure 7.2: *Formula equation to calculate Precision score*

Precision signifies the ability of the classification model to accurately identify positive instances among those it has classified as positive. It is calculated by dividing the number of true positives (correctly identified positive instances) by the sum of true positives and false positives (instances incorrectly identified as positive). Precision is valuable when the cost of false positives is high, as it provides insights into the model's capability to avoid false alarms. However, an excessively high precision may be accompanied by low recall, as the model becomes overly cautious in labeling positive instances.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Figure 7.3: *Formula equation to calculate Recall score*

Recall, on the other hand, reflects the model's capacity to identify all positive instances within the dataset. It is computed by dividing the number of true positives by the sum of true positives and false negatives (positive instances erroneously classified as negative). Recall is particularly crucial in scenarios where failing to detect positive cases could have adverse consequences. Nonetheless, a heightened recall may result in a lowered precision, as the model becomes more inclusive and tends to produce more false positives.

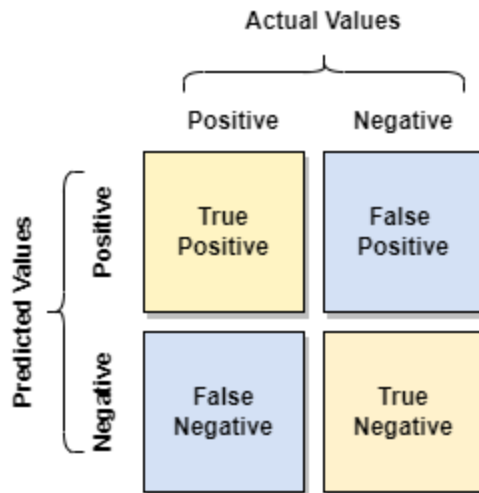
7.1.3 F1 Score

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

Figure 7.4: *Formula equation to calculate F1 Score*

The F1 score, often referred to as the F1 measure or F1 score, averages precision and recall into a single metric, striking a balance between these two conflicting objectives. It is calculated as the mean of precision and recall, thus providing a unified evaluation of a model's classification performance. The F1 score becomes especially useful when dealing with imbalanced datasets, where precision and recall may diverge substantially. By incorporating both aspects, the F1 score offers a comprehensive assessment of a model's effectiveness in balancing false positives and false negatives.

7.1.4 Confusion Matrix

**Figure 7.5:** *Confusion Matrix*

The confusion matrix serves as the foundation for calculating various performance metrics, including accuracy, precision, recall, and the F1 score. It provides a more nuanced understanding of a model's strengths and weaknesses, particularly in scenarios with imbalanced datasets, by explicitly delineating different types of prediction outcomes. As a result, the confusion matrix is an indispensable tool for fine-tuning and optimizing machine learning models to meet specific classification objectives.

7.2 Evaluation Results of the Proposed Model

The model trained chosen to integrate in this project would be YOLOv8-cls, where every aspect of performance results will be discussed in this section. Firstly, some data achieved during training phase for 20 epochs is recorded in Table 7.1 below.

Table 7.1: Data achieve during training and validation of YOLOv8-cls model (20epochs)

epoch	train/loss	val/loss	accuracy	lr
0	0.18664	0.73669	0.97944	0.00023763
1	0.01107	0.72449	0.99611	0.00045208
2	0.0098	0.72366	0.99778	0.00064298
3	0.00389	0.72374	0.99667	0.00060797
4	0.00358	0.7228	0.99722	0.00060797
5	0.00133	0.72158	0.99889	0.00057263
6	0.00031	0.72155	0.99833	0.00053728
7	0.00075	0.7219	0.99889	0.00050194
8	0.00359	0.72219	0.99722	0.0004666
9	0.00045	0.72224	0.99722	0.00043126
10	0.00036	0.7222	0.99722	0.00039591
11	0.00035	0.7219	0.99778	0.00036057
12	0.00027	0.72176	0.99889	0.00032523
13	7.00E-05	0.72143	0.99944	0.00028988
14	0.00012	0.72151	0.99944	0.00025454
15	7.00E-05	0.72156	0.99944	0.0002192
16	5.00E-05	0.72164	0.99889	0.00018385
17	5.00E-05	0.72171	0.99944	0.00014851
18	3.00E-05	0.72159	0.99944	0.00011317
19	7.00E-05	0.72158	0.99889	7.78E-05

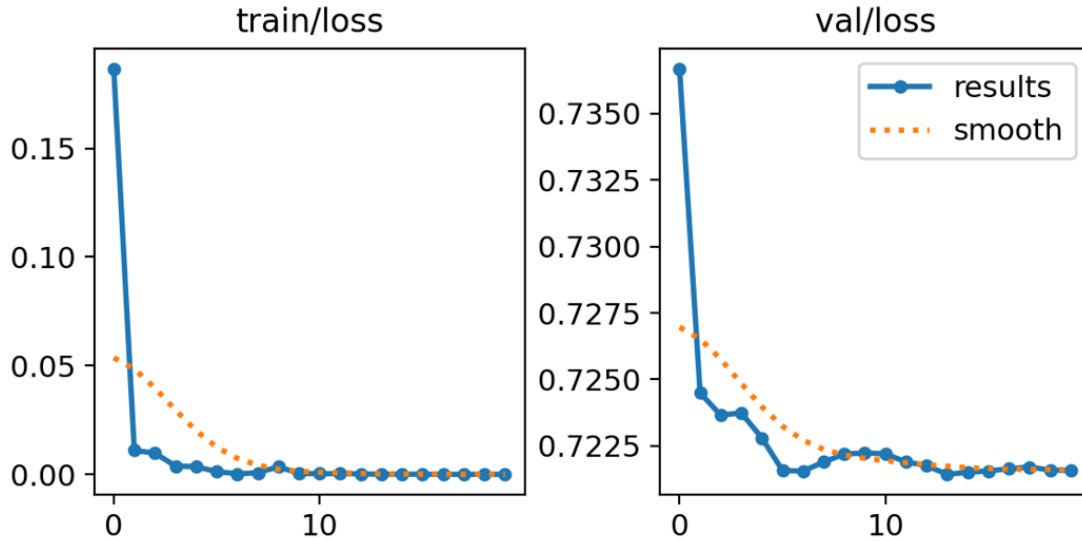


Figure 7.6: Graph plot of train and val loss during training

In this section, the performance of our model is analyzed based on the observations derived from Table 7.1 and Figure 7.6. Throughout the training process, we noticed a consistent decrease in both training and validation loss. This decline in loss values signifies that the model is gradually learning and adapting to the underlying data patterns. It's noteworthy that the model appears to have converged at approximately epoch 10, indicating a stable and well-adjusted state.

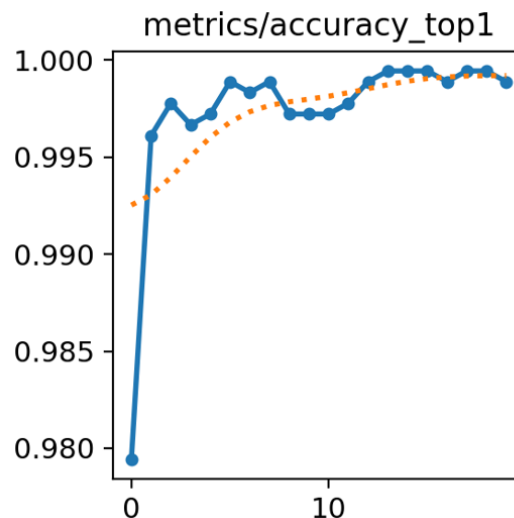


Figure 7.7: Graph plot of val accuracy during training

In terms of validation accuracy, our model exhibits strong performance. The training and validation accuracy scores consistently demonstrate high values. Figure 7.7 illustrates the accuracy curve, which aligns closely with our accuracy metrics. One notable that, the curve shows a level of stability with minor fluctuations. This suggests that the model maintains its high accuracy across epochs, further validating its robustness in classifying the data.

Table 7.2: *Performance of YOLOv8-cls Model on Test Set*

Metric	YOLOv8-cls Model
Accuracy	1.00
Precision	0.9987
Recall	0.9987
F1 Score	0.9987

In the results evaluation phase, the YOLOv8-cls model underwent testing phase on the test set, as discussed earlier. As presented in Table 7.2, the model's performance metrics reveal its remarkable capabilities. Notably that, the overall accuracy achieves a perfect score of 1.00. However, it is essential to recognize that accuracy alone provides a broad overview of the model's performance, lacking granularity. Explore deeper, the precision and recall scores both attain a score of 0.9987, signifying the model's proficiency in minimizing false alarms and adeptly capturing all data points. Moreover, the F1 scores, serving as a balanced metric that averages precision and recall, also record a commendable 0.9987, which is more accountable to represent the full model performance.

Turning our attention to the confusion matrices depicted in Figures 7.8 and 7.9 shown below, the confusion matrix generated reveals minimal flaws during testing. The only flaw lies in 2 model's misclassification of 5 sen as 10 sen, this is possible due to the highly similar size of both coins. Nonetheless, it is shown that the model's performance across all other categories remains near-perfect.

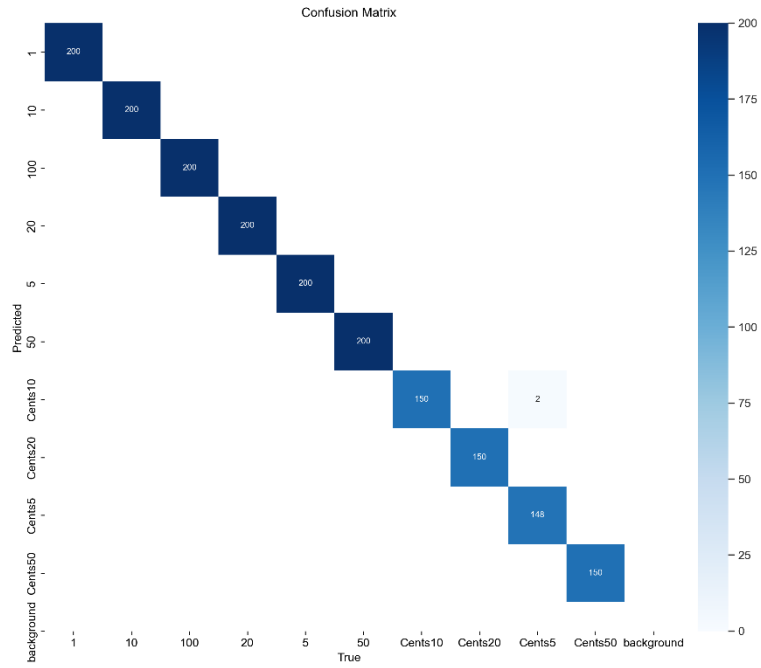


Figure 7.8: Confusion Matrix of YOLOv8-clс model testing

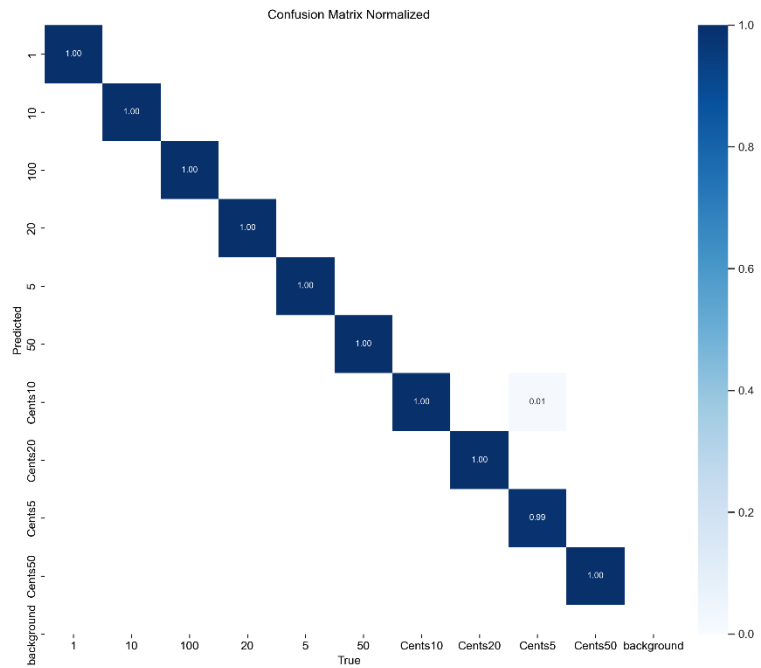


Figure 7.9: Confusion Matrix of YOLOv8-clс model testing (Normalized)

7.3 Evaluation Results of the Proposed Application System (CashVisionV2)

In this section, after evaluates our model, it is essential as well to construct a set of exhaustive module tests. to mitigate the risk of encountering issues such as application crashes, unresponsive functions, or inaccurate detection. Few tests below as recorded in table 7.3 and table 7.4 are designed to evaluate the application across diverse scenarios, ultimately ensuring peak performance and an optimal user experience for visually impaired.

Table 7.3: *Test Case for CashVisionV2 Home Screen*

No.	Test Action	Expected Result	Actual Result
1.	Launch CashVisionV2	Able to launch and enter home page	Launched Successfully
2.	Camera Preview display	Able to display camera preview	Displayed Successfully
3.	Single Tap Camera Capture	Able to capture through single tap	Captured Successfully
4.	Generate Model Inference	Able to send input to model inference and receive prediction	Generated Successfully
5.	Prediction display	Able to display predicted output	Displayed Successfully
6.	No Prediction alert display	Able to display no prediction alert	Displayed Successfully

Table 7.4: *Test Case for CashVisionV2 Settings Screen*

No.	Test Action	Expected Result	Actual Result
1.	Change Language	Able to change voice language between English, Malay, Chinese, Tamil	Changed Successfully
2.	Change Voice Speed	Able to change voice speed between Slow, Default, Fast	Changed Successfully
3.	Toggle Display Confidence	Able to choose to show confidence level or not showing confidence level	Switched Successfully
4.	About	Able to display about dialog box when triggered	Displayed Successfully
5.	Help	Able to display help dialog box when triggered	Displayed Successfully

7.4 Objective Evaluation

Overall, all of the objectives initiate in this project have been successfully achieved. Firstly, the primary project objective was to create a functional deep learning classification model for Malaysian banknotes and coins through transfer learning approach. This goal has been effectively realized through the development of the proposed YOLOv8-cls model, which adeptly classifies various Malaysian banknotes and coins with remarkable precision.

Next, the second objective focused on integrating the developed model into a mobile application within the Flutter framework. This objective has been fulfilled with the creation of the CashVisionV2 application. This user-friendly mobile app not only accepts image input but also provides output predictions both visually on the screen and audibly through voice speech successfully. This feature ensures convenience and accessibility for individuals with visual impairments.



Figure 7.10: *Samples of actual and predictions of proposed methodology*

The third objective revolved around crafting a methodology capable of comprehending the contextual complexity of diverse banknotes and coins. Remarkably, this objective has been met as well. The proposed methodology exhibits the ability to discriminate between different denominations of cash notes and coins with remarkable accuracy, even when faced with highly similar patterns as shown in Figure 7.8.

CHAPTER 8: CONCLUSION

8.1 Conclusion

The daily struggles faced by visually impaired individuals, particularly in the aspect of financial transactions, are marked by significant challenges. One of the most pressing issues founds in their ability to recognize and differentiate banknotes and coins. The absence of visual cues deprives them of the means to independently assess the value of currency, making simple yet vital tasks like payment or receiving change a challenging situation. In such a scenario, the application of computer vision emerges as a beacon of hope, offering a potential solution to alleviate this predicament.

However, it's essential to recognize that existing solutions, such as dedicated cash readers for the visually impaired, come with their own set of limitations. Many of these devices are considerably expensive, making them inaccessible to a significant portion of the visually impaired population. Furthermore, the bulkiness and lack of mobility associated with some of these readers impede their practicality in real-world scenarios. These shortcomings underscore the desperate need for an alternative approach that combines affordability and convenience, ultimately placing the power of financial independence back into the hands of the visually impaired.

In pursuit of this, our project has successfully developed a mobile application, CashVisionV2, integrated with a state-of-the-art trained deep learning model, YOLOv8-cls. This proposed solution empowers visually impaired users to easily classify complex banknotes and coins with intricate and highly similar patterns. The vision was to create a one-stop, user-friendly application that streamlines the entire process, from input to output, with utmost consideration for the challenges faced by our target users.

The results of this model as evaluated before is near-perfect, with high overall performance indicating the success of our model. The potential impact of this technology on the lives of visually impaired individuals is immeasurable, offering them a newfound level of independence and confidence in handling financial transactions. As we move forward, the project researcher's hope is that this solution will continue to enhance the

real-life experiences of the visually impaired community, reaffirming our commitment to making a meaningful difference in their lives.

8.2 Contributions

The contribution part of this project is mainly from the deliverables obtained from development phase. This project's applications are readily accessible on GitHub, as mentioned earlier, embracing the ethics of open-source development. This accessibility opens the door for fellow researchers and developers to contribute, refine, and innovate upon the existing system. By fostering a collaborative environment, this project aims to continually improve its technology, addressing evolving challenges and enhancing the experience of visually impaired community in the future.

One of the core contributions of this project is its ability to instil confidence in visually impaired individuals during financial transactions involving physical cash and coins. By providing an accurate and reliable means of identifying currency, the system empowers users to confidently engage in financial activities, improving their autonomy without fully relying on others.

Another crucial aspect of this project's contribution is its commitment to providing free services to visually impaired individuals. In a world where accessibility solutions often come at a cost, this project stands out as an exception. Users can utilize the proposed tools without any financial burden, ensuring that the technology remains inclusive and within reach for all those who require its assistance in their daily lives.

In short, this project aligns its efforts with the overarching goal of establishing a secure and supportive transactional environment for the visually impaired community. Every aspect of this effort has been guided by the singular mission of assisting those in need. As the positive impact of these tools on the lives of visually impaired individuals grows, the intrinsic value that this project holds is constantly reminded, touching and improving the experiences of many.

8.3 Future Works

Future efforts will primarily focus on enhancing the model's accuracy and usability. This includes exploring newer deep learning model architectures and optimizing the app's usability based on user feedback. The aim is to continually improve the tool's performance and accessibility for visually impaired individuals in their daily financial transactions.

REFERENCES

- [1] T. S. Larissa Theodore, "Shopping blind: Assistants help, but currency doesn't," *Beaver County Times*, 07-Dec-2008. [Online]. Available: <https://www.timesonline.com/story/news/2008/12/07/shopping-blind-assistants-help-but/18306317007/>. [Accessed: 20-Apr-2023].
- [2] R. C. Joshi, S. Yadav and M. K. Dutta, "YOLO-v3 Based Currency Detection and Recognition System for Visually Impaired Persons," 2020 International Conference on Contemporary Computing and Applications (IC3A), Lucknow, India, 2020, pp. 280-285, doi: 10.1109/IC3A48958.2020.233314.
- [3] N. A. J. Sufri, N. A. Rahmad, N. F. Ghazali, N. Shahar and M. A. As'ari, "Vision Based System for Banknote Recognition Using Different Machine Learning and Deep Learning Approach," 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2019, pp. 5-8, doi: 10.1109/ICSGRC.2019.8837068.
- [4] N. bin Hidzir Pauzi and S. A. Abdullah, "Journal of Mechanical Engineering," *Currency Identification Assistant For Visually Impaired*, 2018.
- [5] World Health Organization, "Vision Impairment and blindness," *World Health Organization*. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>. [Accessed: 20-Apr-2023].
- [6] United Nations, "International Day of Persons with Disabilities," *United Nations*, 2022. [Online]. Available: <https://www.un.org/en/observances/day-of-persons-with-disabilities>. [Accessed: 20-Apr-2023].
- [7] A. Yousry, M. Taha, and M. Selim, 'Currency Recognition System for Blind people using ORB Algorithm', *International Arab Journal of Information Technology*, vol. 5, pp. 34–40, 01 2018.

REFERENCES

- [8] N. A. J. Sufri et al., 'Image Based Ringgit Banknote Recognition for Visually Impaired', 11 2020.
- [9] C. J. Tan, W. K. Wong and T. S. Min, "Malaysian Banknote Reader For Visually Impaired Person," 2020 IEEE Student Conference on Research and Development (SCOREd), Batu Pahat, Malaysia, 2020, pp. 74-79, doi: 10.1109/SCOREd50371.2020.9250979.
- [10] L. Chinmay and H. S. Gururaja, "Currency Detection for Visually Challenged People using Convolutional Neural Network Model," 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), Bengaluru, India, 2021, pp. 1-4, doi: 10.1109/FABS52071.2021.9702694.
- [11] Q. Zhang, 'Currency Recognition Using Deep Learning', 2018. [Online]. Available: <https://www.timesonline.com/story/news/2008/12/07/shopping-blind-assistants-help-but/18306317007/>. [Accessed: 1-Sep-2023].

APPENDIX

YOLOv8-cls Layer Architecture

```

ClassificationModel(
  (model): Sequential(
    (0): Conv(
      (conv): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (act): SiLU()
    )
    (1): Conv(
      (conv): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (act): SiLU()
    )
    (2): C2f(
      (cv1): Conv(
        (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (act): SiLU()
      )
      (cv2): Conv(
        (conv): Conv2d(48, 32, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (act): SiLU()
      )
    (m): ModuleList(
      (0): Bottleneck(
        (cv1): Conv(
          (conv): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
          (bn): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
          (act): SiLU()
        )
        (cv2): Conv(

```

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR

APPENDIX

```
        (conv): Conv2d(16, 16, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
        (bn): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
        (act): SiLU()
    )
)
)
)
(3): Conv(
  (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
  (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (act): SiLU()
)
(4): C2f(
  (cv1): Conv(
    (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (act): SiLU()
  )
  (cv2): Conv(
    (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (act): SiLU()
  )
)
(m): ModuleList(
  (0-1): 2 x Bottleneck(
    (cv1): Conv(
      (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      (bn): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (act): SiLU()
    )
    (cv2): Conv(
      (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
```


APPENDIX

```
    )
  )
)
)
(7): Conv(
  (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1), bias=False)
  (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (act): SiLU()
)
(8): C2f(
  (cv1): Conv(
    (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (act): SiLU()
  )
  (cv2): Conv(
    (conv): Conv2d(384, 256, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (act): SiLU()
  )
)
(m): ModuleList(
  (0): Bottleneck(
    (cv1): Conv(
      (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      (bn): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (act): SiLU()
    )
    (cv2): Conv(
      (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      (bn): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (act): SiLU()
    )
  )
)
)
)
```

APPENDIX

```
)
(9): Classify(
  (conv): Conv(
    (conv): Conv2d(256, 1280, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (bn): BatchNorm2d(1280, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (act): SiLU()
  )
  (pool): AdaptiveAvgPool2d(output_size=1)
  (drop): Dropout(p=0.0, inplace=True)
  (linear): Linear(in_features=1280, out_features=10, bias=True)
)
)
```

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S3	Study week no.: 2
Student Name & ID: Chan Hong Wai, 19ACB03752	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development And Evaluation Of A Banknote Reader For Visually Impaired People Using Deep Learning-based Object Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Study and finding research papers and resources relate to theme work.

2. WORK TO BE DONE

Review on previous similar cash reader approach

Identify project design and tools to be used

3. PROBLEMS ENCOUNTERED

Slow progress due to limited resources

4. SELF EVALUATION OF THE PROGRESS

Discussion with supervisor



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S3	Study week no.: 6
Student Name & ID: Chan Hong Wai, 19ACB03752	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development And Evaluation Of A Banknote Reader For Visually Impaired People Using Deep Learning-based Object Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Literature review on previously done work under same core theme.
Prototype design and project development environment identified.

2. WORK TO BE DONE

Prototype development
Model training
Flutter framework app design

3. PROBLEMS ENCOUNTERED

Limited model dataset on Internet

4. SELF EVALUATION OF THE PROGRESS

Need to be more dedicated on the model training.
Self capture the dataset.



Supervisor's signature

Bachelor of Computer Science (Honours)

Faculty of Information and Communication Technology (Kampar Campus), UTAR



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S4	Study week no.: 4
Student Name & ID: Chan Hong Wai, 19ACB03752	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development And Evaluation Of A Banknote Reader For Visually Impaired People Using Deep Learning-based Object Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

More literature reviews added
features designing

2. WORK TO BE DONE

Prototype completion
Model training
Application development

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

More effort needed for app development



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S4	Study week no.: 8
Student Name & ID: Chan Hong Wai, 19ACB03752	
Supervisor: Dr. Ng Hui Fuang	
Project Title: Development And Evaluation Of A Banknote Reader For Visually Impaired People Using Deep Learning-based Object Detection	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Prototype development done
Model training done

2. WORK TO BE DONE

Model Evaluation
App development
Project Report

3. PROBLEMS ENCOUNTERED

Integration issues of model into mobile app

4. SELF EVALUATION OF THE PROGRESS

Need research more to solve integration problems




Supervisor's signature



Student's signature

POSTER



ENABLING INDEPENDENCE FOR VISUALLY IMPAIRED

DEVELOPMENT AND EVALUATION OF A BANKNOTE READER FOR VISUALLY IMPAIRED PEOPLE USING DEEP LEARNING-BASED OBJECT DETECTION

PROJECT DEVELOPER: CHAN HONG WAI
PROJECT SUPERVISOR: DR. NG HUI FUANG

INTRODUCTION

This project aims to develop a banknote reader using deep learning-based object detection for visually impaired people. The system will be able to recognize the denomination of banknotes and provide audio output to the user.



MOTIVATION

The project's objective is to create an affordable, precise, and accessible banknote reader using deep learning to enhance financial independence for visually impaired individuals, driven by the Malaysian Blind Association's mission to improve their quality of life.

OBJECTIVE

The project aims to develop a deep learning model for Malaysian banknotes and coins, integrate it into a mobile app for visually impaired users, and create a method to understand and classify scanned currency.

PROBLEMS

This project addresses the challenges faced by visually impaired individuals in recognizing banknotes due to expensive and limited existing solutions.

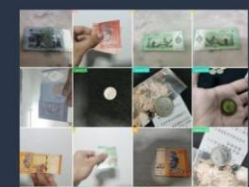
PROPOSED SOLUTION

The proposed solution entails training a YOLOv8-cls model on a dataset of around 10,000 Malaysian banknotes and coins, followed by its integration into a user-friendly Flutter mobile application to enhance accessibility and mobility for visually impaired users.

RESULTS & DISCUSSION

The proposed method, trained for 20 epochs, demonstrates its feasibility and showcases near-perfect performance, with overall accuracy reaching 100%, and precision, recall, and F1 score consistently at 99.87%.

Metric	Score
Precision	99.87
Recall	99.87
F1	99.87



PLAGIARISM CHECK RESULT

PLAGIARISM CHECK RESULT

Turnitin Originality Report

Document Viewer

Processed on: 14-Sep-2023 10:27 PM CDT
ID: 2166582961
Word Count: 16258
Submitted: 1

DEVELOPMENT AND EVALUATION OF A
BANKNOTE READ... By chanbg21 1

Similarity by Source	
Similarity Index	9%
Internet Sources:	7%
Publications:	5%
Student Papers:	N/A

include quoted	include bibliography	excluding matches < 8 words	mode: quickview (classic) report	print
download				
1% match (Internet from 02-Sep-2013) http://docs9.chomikuj.pl				
1% match () Sufri, N. A., Jasmin, Rahmad, N. A., As'ari, M. A., Zakaria, N. A., Jamaludin, M. N., Ismail, L. H., Mahmood, N. H., "Image Based Ringgit Banknote Recognition for Visually Impaired", Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 2017				
<1% match (Internet from 23-Jun-2023) https://www.mdpi.com/2071-1050/15/13/9878				
<1% match (Internet from 06-Sep-2023) https://www.mdpi.com/2075-4418/13/18/2867				
<1% match (Internet from 29-Aug-2023) https://www.mdpi.com/2072-4292/15/17/4213				
<1% match (Internet from 10-May-2023) https://www.mdpi.com/2077-0472/13/5/1023				
<1% match (Internet from 21-Mar-2023) https://www.mdpi.com/2072-4292/15/6/1607				
<1% match (Internet from 21-Jul-2023) https://www.ijres.org/papers/Volume-11/Issue-4/1104732738.pdf				
<1% match (Internet from 29-Jul-2023) https://journals.bilpubgroup.com/index.php/aia/article/download/3249/4474				
<1% match (Internet from 30-Jul-2023) https://viso.ai/computer-vision/model-performance/				
<1% match ("Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019) "Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)", Springer Science and Business Media LLC, 2019				
<1% match () Wong, Wai Kit, Tan, Chi Jie, Min, Thu Soe, Wong, Eng Kiong. "Fuzzy Logic Based Perceptual Image Hashing Algorithm in Malaysian Banknotes Detection System for the Visually Impaired", 'Bilingual Publishing Co.', 2021				

PLAGIARISM CHECK RESULT

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.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	CHAN HONG WAI
ID Number(s)	19ACB03752
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	DEVELOPMENT AND EVALUATION OF A BANKNOTE READER FOR VISUALLY IMPAIRED PEOPLE USING DEEP LEARNING-BASED OBJECT DETECTION

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)
Overall similarity index: <u>9</u> % Similarity by source Internet Sources: <u>7</u> % Publications: <u>5</u> % Student Papers: <u>N/A</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
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 <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

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: Ng Hui Fuang

Date: 15/09/2023

Signature of Co-Supervisor

Name: _____

Date: _____

FYP 2 CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB03752
Student Name	CHAN HONG WAI
Supervisor Name	Dr. Ng Hui Fuang

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)
	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: 14/9/2023