MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT

BY

LEE KYAI LUN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT

Academic Session: 2021/05

I LEE KYAI LUN

(CAPITAL LETTER)

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

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

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
NO 29 JALAN DESA 7/4
BANDAR COUNTRY HOMES
48000 RAWANG, SELANGOR

Dr Manoranjitham a/p Muniandy
Supervisor's name

Date: 19th July 2021

Date: 27th august 2021

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 19th July 2021

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ***LEE KYAI LUN*** (ID No: ***17ACB01383***) has completed this final year project entitled "Malaysia currency recognizer mobile application for visual impairment" under the supervision of Dr Manoranjitham a/p Muniandy (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,

(*LEE KYAI LUN*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" 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 : ___LEE KYAI LUN______

Date : 19th July 2021

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Dr Manoranjitham a/p Muniandy and my moderator, Dr Mogana a/p Vadiveloo who have given me a golden opportunity to involve in the development in artificial intelligence and mobile application field of study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

ABSTRACT

In Malaysia, the prevalence of blindness for all ages has an average of 1.2%. People with vision impairments have difficulty recognizing objects through a vision in daily life, especially for some principal daily used objects such as a banknote. The braille feature on top of the banknotes becomes unusable in its tactile form after a brief usage circulation. Nevertheless, the existing currency recognizer is inefficient and not able to identify Malaysia currency. In this project, a mobile application based on recognizing currency has been developed using deep learning transfer learning techniques. The pre-trained deep learning model, MobileNet V2 utilized to transfer learning on our detection model. One hundred images of each value of Malaysia currency were collected as the dataset for training the model. The output lightweight model from the training process deployed on the mobile application. The mobile application is designed with vision-friendly features embedded into the mobile application through Android studio. Embedded features include vibration notification, real-time scanning on a phone camera, and real-time result voice feedback. Our system reached a mean average accuracy of 97% and average inference time of 0.06 second on detecting the currency. In conclusion, we bring a Malaysia banknotes recognizer mobile application in more accessible and more accurate ways.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADB	Android Debug Bridge
APK	Android Package
BC	Bayesian Classifier
CNN	Convolutional Neutral Network
DTC	Decision Tree Classifier
KNN	k-nearest neighbours' algorithm
OpenCV	Open-Source Computer Vision Library
RM	Ringgit Malaysia (Malaysia currency)
ROI	Region of Interest
RGB	Red Green Blue: RGB
SURF	Speed Up Robust Features: SURF
SIFT	Scale-Invariant Feature Transform
SSD	Single Stage Object Detection
SVM	Support Vector Machine

Chapter 1: Introduction

1.1 Background Information

At least 2.2 billion people worldwide suffer from vision deficiency or blindness, with at least 1 billion of those suffering from a vision impairment that might have been avoided or that has yet to be addressed (World Health Organization, 2019). Disability for a person who loses their vision, also known as vision impairments, is a decreased ability of vision from eyes and, when it comes to worse, will be vision loss, which means the total blind. In real life, people who cannot see through their own eyes face many challenges that we have never imagined.

One of the essential objects that we will be using for our daily life is money. Imagine as an average person when purchasing or buying some stuff will be checking on the correct amount of money before hand out to the receiver. However, for blind people, it might become a challenging task to differentiate the money. In order to let the blind people, recognize the money, some innovations such as a money identification card have been made, but it requires much training to use the tools such like measurement tools (Tom, 2019).

With the evolving of technologies in information technologies, we can make use of it to help the blind people life becomes easier to develop a mobile application with money recognize function. The progress to identified involve of computer vision with deep learning. Our input data will be the image of the fourth series Malaysia dollar, which is RM1, RM5, RM10, RM20, RM50, RM100 (Malaysia Bank Negara, 2020). The output will be the text to speech voice to notify the user what the amount of the currency is.

The algorithm to evaluate the image will be using a deep learning object detection technique to train the model. The process of building a model will be breaking down into few parts, which is getting and pre-processing data, defining the model architecture, selecting the training model, and estimating performance. Accessibility is referring to a way that user's access to the product, services, or devices. It always the concern, especially when the technologies keep evolving. A software application or website which practices inaccessibility will make them usable by many folks as attainable. The idea of building software with user-friendly is essential, but on the other hand, it also needs to consider the users who have a disability such as vision impairments.

1.2 Problem Statement and Motivation

People with vision impairments are having difficulty recognizing banknotes. In daily life, especially for some principal daily used objects such as a banknote, it's hard to let a visual impairment recognize the banknote. Although Malaysia banknote has a braille feature, it is not sufficient for blind people to recognize the banknotes. Many who do not understand braille can have difficulty distinguishing between different Malaysian banknotes. The braille deteriorates at an alarming rate. After only a brief period in circulation, money labelled with Braille becomes unusable in its tactile form (Maurer, 2007).

Recognizing banknotes even challenging for people who not born with blind. An individual born with a vision impairment has an incredible sense of recognizing objects with sensory substitution through their hand might be recognize banknotes through braille feature, but some individuals who are not born with vision impairments might need time to make their sensory substitution for vision (Palm, 2012).

Current banknotes recognition-based application unable to guide blind users in identifying the notes hence, it lacks in terms of effectiveness. mobile applications nowadays were not aware of accessible by visual impairment, leading them out of touch to the environmental changes (J.Thomas, 2019). Developing a mobile application that is user friendly for visual impairment to enable identification of banknotes with ease.

1.3 Project Scope

This project aims to develop a mobile application on android with a function to recognize banknotes. This project involves a deep learning technique used to train a detection model to recognize each banknotes class. A well-trained light model will be developed to deploy on the android application with required lower computational power. Next, the application will build with user-friendly for visual impairments users. The main focus for the prototype application will perform real-time capture and let the machine learning model recognize and feedback the result with voice.

1.4 Project Objectives

RO1: To study the problem of blind users in recognizing currency note in Malaysia

In this project, the first objective is to study the problem of blind users in recognizing currency notes in Malaysia and research regarding the ways blind users acknowledge the banknotes.

RO2: To propose a currency classifier on the mobile application using deep learning technique

The second objective, we proposed a currency classifier on the mobile application using a deep learning model Mobile Nets v2 to recognize the banknote in real-time. We aim to train a well-defined recognition model compatible with Malaysia banknotes properties. This project study on Malaysia banknotes RM1, RM5, RM10, RM20, RM50, and RM100. Hence the detection model will build with a lightweight framework using TensorFlow lite. With TensorFlow, the lite model will have better performance and fewer binary files suitable for mobile applications.

The proposed mobile application will be user-friendly for visual impairment. The design of the architecture of the mobile application will be accessible for visual impairment. Vision impairments are facing the main challenge that cannot have a clear vision when using a smartphone. Physical feeling and hearing are the channels they get in touch with when using a smartphone. The consideration support design, such as voice-feedback, application vibration indicator, voice feedback navigation.

RO3: To validate the proposed application with performance and actual users.

The third objective is to validate the detection performance to the experience of the actual users. The accuracy of the model will be defined at the library within TensorFlow. The proposed application will validate from detection performance to the experience of actual users.

1.5 Impact, significance and contribution

We developed a banknotes recognition application to help visual impairments have an easier way to recognize the banknote from this project. The mobile application will be lightweight and low latency on mobile devices. Nevertheless, accessibility is one of the main points that will focus on this project. In this era, the smartphone has become the most daily used object, and most importantly is the mobile application designed to run on it. We often not aware that those applications are served for the regular users and those who had visual impairments. Hence, this application will be more user-friendly to those with visual impairments to better assess the application.

In real life, people who cannot see through their own eyes face many challenges that we have never imagined. One of the essential objects that we will be using for our daily life is money. This proposed application allows the visual impairments to recognize the banknotes by just moving the camera direct to the banknotes. The application shall feedback the actual value of the banknotes. Nowadays, smartphones are common, and using this application allows users to be the second eye to visual the banknotes.

Chapter 2: Literature Review

In this chapter, past studies on classification techniques as well as existing application on recognizing banknotes are focused on between year 2014 to 2020. We reviewed on the classification techniques that we applied in our project. Deep learning is a subset of machine learning that have more than three layers of neural network. These neural networks aim to imitate the activity of the human brain by allowing it to learn from enormous amounts of data, albeit they fall far short of its capabilities. Deep learning eliminates some of the data pre-processing that machine learning generally entails. These algorithms can ingest and interpret unstructured data such as text and photos, as well as automate feature extraction, which reduces the need for human specialists. (IBM Cloud Education, 2020)

2.1 Literature Review on Bank Notes Classification Approach

A proposed system approach for currency recognition is based on an image processing technique and a ROI extraction method (Semary *et al.*, 2016). In this paper, a general framework has been completed for identifying paper banknotes, which include six main stages. First stages, image acquisition, the images will gather from the digital camera as an RGB image. The images converted into grayscale with a constant intensity value of 1/3. In the second stage, the images going through the pre-processing process by using the gaussian blur technique to remove the noises on the images. Third stages, the images convert to binary vision with black and white intensity values to extract the background of the currency paper. The fourth stage, modified and adjust the brightness of the image by using histogram equalization. Fifth stages, images resized to a dataset height by using ROI (Region-Of-Interest) extraction. The final stages, cross-correlation function, are used to measure the similarity of the images. This proposed framework applies on 120 test images (20 samples for each data) and reaches an average 89% accuracy under the MATLAB system.

Jegnaw and Yaregal proposed a software and hardware solution for the Ethiopian paper currency recognition system (Fentahun Zeggeye and Assabie, 2016). Image acquisition, denomination, and verification on the currency are the main steps embedded in this system. In the currency image acquisition phase, the image is captured by the scanner and digital camera to extract the features to manual check the genuine of a banknote. Currency denomination is achieved by pre-processing images, extracting characteristic features, and classified based on the extract features. Image resize and enhancement to achieve the desired images before extracting the features from the banknotes. The feature sets extracted from the bank notes are the dominant colour, the dominant colour's distribution, the Hue value, and speeded up robust feature (SURF). The classification is divided into four phases based on the extracted features, which is the dominant colour in RGB format, dominant colour distribution, Hue value definition. The HSV colour model describes a colour space in terms of three constituent components during the hue value phase: hue (colour type), saturation (intensity of particular hue), and value (brightness of the colour). The hue value difference in the Ethiopian paper currency is not more than seventy-five compared to the hue value between the standard images and test images. At the SURF matching phase, SURF features extracted out from the images. Five steps to accomplish the process are finding the interest points, selecting the most vital points, constructing feature vectors or descriptors, matching features, and calculating match percentages. At the currency verification phase to enhance the input banknotes' genuine value after the classification on the banknotes. In this validation, steps are using ROI localization, binarization, morphological closing, morphological area opening, and counting the objects to verify the thin strip and wide strips on the Ethiopian banknotes. This framework uses the MATLAB tool for the development environment and achieves an average domination rate of 90.42%, but the domination rate will decrease between 80% to 90 % for old notes. 100% accuracy to reject the counterfeit banknotes.

Next, a banknote recognition from image based for visually impaired are proposed (Jasmin Sufri *et al.*, 2017). The general frameworks include four steps: data collection, feature extraction, classification and performance evaluation. RGB extraction performs by using MATLAB. The K-NN and DTC were optimized using ten-fold cross-validation, while the most negligible cross-validation loss was considered. A confusion matrix to present the performance of the DTC and K-NN model. The overall result of both models archived 99.7 accuracies. one of the class values of fifty is causing the inaccuracy in the result. Decision Tree Classifier (DTC) and k-Nearest Neighbours (k-NN) for recognizing each class of banknote. The optimized DTC with a height of three and a leaf size of twenty-one.

In this research, Aljutaili had proposed a Speeded up Robust Scale-Invariant Feature Transform (SR-SIFT) currency recognition algorithm to merges the advantages of Speeded up Robust Features (SURF) and Scale Invariant Feature Transform (SIFT) algorithm to enhance the feature detection on the currency (Aljutaili *et al.*, 2018). There are five steps in the SR-

SIFT algorithm: Acquisition the input image, SIFT filter edge, interest point detection, detect SURF features, and locate the matching point. From the algorithm, the performance evaluation by the distribution of the best key points (BKP). The proposed SR-SIFT algorithm gives better performance than the SIFT and SURF algorithms. It improved the distribution of BKP on the currency's surface area and reduced the average response time, particularly with a small and minimum number of BKP. It also improves the precision of the accurate BKP distribution at the currency edge. Although SR-SIFT has a strong power to extract the features from an object but in different banknotes have a different approach.

In Shubham and Shiva study, they proposed a deep learning-based technique for identified Indian currency. Mobile net a deep learning architecture used in their research and achieved an accuracy of 96.6 percent (Mittal and Mittal, 2018). Multiple machine learning approaches were used to test the accuracy of banknote recognition (Sufri *et al.*, 2019). In the first stages, an input image with the focal length being manually clipped in the different sections by TrainningImageLabeler. The features extracted from the images are the colour features RGB with average intensity value. Machine learning models such as k-Nearest Neighbour (KNN), Decision Tree Classifier (DTC), Support Vector Machine (SVM), and Bayesian Classifier (BC) are used to recognize the banknotes. In the second stage, AlexNet model a deep learning approach test with four databases. For both stages, performance evaluation using the cross-validation in a confusion matrix. The SVM and BC's machine learning approach achieves an accuracy of 100%, K-NN and DTC achieve an accuracy of 99.7% accuracies. Alex.net achieve an accuracy of 100% on a different oriented database.

In Alvin, Rhowel, Nino, Honetlet and Estrelita study, they proposed a deep learning object detection technique to detect the Philippine bill and applied it on the Raspberry Pi 4 devices for recognition. The deep learning model used is mobile net v2 quantized. The testing accuracy from the research reaches 86.3% (Alon *et al.*, 2020). In Ali, Mahir, Faisal, Soumik, Marium and Saiful study, they research the lightweight deep learning architecture that recognizes of Bangladeshi Banknotes. The transfer learning techniques used in the study. The finding from this study suggested that the MobileNet reach accuracy of 100% and NASNetMobile reach 97.77% that can easily be deployed on IoT devices for practical use (Linkon *et al.*, 2020). Several deep learning techniques were applied to classified India Banknotes by Suyash. The deep learning model used includes EfficientNet, Xception, VGG 16, ResNet, MobileNet, VGG

19, Inception V3, which achieved 82.03%, 43.18%, 41.19%, 74.51%, 91.87%, 71.86%, 81.43% respectively (Mahesh Bahrani, 2020).

Table 2.1: Comparison between the past research on classification on banknotes.

Author & Year	Study	Findings	Limitations
Semary et al., 2015	The study is about using machine learning approaches with ROI technique on Egyptian banknotes.	Reach average accuracy of 89%	Dataset used are small that will be limited the inference performance and the differences on the dataset.
Fentahun Zeggeye & Assabie, 2016	The study is about using machine learning approach with SURF technique on Ethiopian currency.	Reach average accuracy of 90.42%	Differences on the dataset.
Jasmin Sufri1 et al., 2017	The study is about using machine learning approach with K-NN and DTC model on Malaysia banknotes.	K-NN: Reach accuracy of 99.7% DTC: Reach accuracy of 99.7	The accuracy is high and might facing overfitting issue and not the architecture not suitable deployed on mobile application.
Aljutaili et al., 2018	The study is about using the machine learning approach with speeded up robust scale-invariant feature transform technique.	Better than SIFT	The proposed algorithm applied with not stated which banknotes is focus on.
Shubham et al., 2018	The study is about using deep learning approach with MobileNet on Indian banknotes.	MobileNet: Reach accuracy of 96.6%	The next version of MobileNet introduced and the dataset are different.

Sufri et al., 2019	The study is about using machine learning and deep learning approach on Malaysia banknotes.	SVM, BC: Reach accuracy of 100% K-NN: Reach accuracy of 99.7% DTC: Reach accuracy of 99.7 Alex Net: Reach accuracy of 100%	The 100% accuracy might face overfitting issue and the model not suitable for deployed on mobile application due to a better architecture introduced.
Linkon et al., 2020	The study is about using deep learning approach on Bangla Currency dataset	MobileNet: Reach accuracy of 100% NASNetMobile: 97.77%	Differences on the dataset.
Alon et al., 2020	The study is about using deep learning approach on Philippine Bill.	MobileNet v2: Reach accuracy of 86.3%	Differences on the dataset.
Mahesh Bahrani, 2020	The study is about using deep learning approach on India Banknote.	EfficientNet: Reach accuracy of 82.02% Xception: Reach accuracy of 43.18% VGG16: Reach accuracy of 41.19%	Differences on the dataset.

		ResNet: Reach accuracy of 74.51% MobileNet: Reach accuracy of 91.87% VGG19: Reach accuracy of 71.86 Inception V3: Reach accuracy of 81.43%	

2.2 Review on Existing Mobile Android Application

Cash reader: Bill Identifier an android application which identified banknotes denomination of the largest number of currencies created by Martin Doudera (Doudera, 2020). The application using the phone camera to capture the input image in real-time. User required to point the camera to the banknotes in hand and hear the value feedback. The flashlight will be activated automatically while using the identified banknotes. The app can identify the value of a banknote into vibration and the beep sound while identified banknotes to let users aware the app is active and currently identifying banknotes. The strength of Cash Reader is with largest of database which recognizes world currencies from Europe to Australia, reliable offline application by downloading the specific dataset into mobile and identified in real-time. The weakness is when the identified of banknotes only the side banknotes with a number can be recognized. For example, in Malaysia banknotes the number for both sides are not presentable. Furthermore, the navigation is not providing for vision impairment user and required to install supporting tools such as talkback service for navigation.

The second application reviewed at is MCT Money Reader (MCT Data, 2018). The application used the phone camera and identified the money in real-time. The flashlight will be activated automatically while using the identified banknotes. The applications will be issuing an audible warning tone when no banknote was found in progress. Tic-tac sounds will keep occurring to notify the application is an under-recognition mode. The recognition mode separates two continuous modes and single-mode. In continuous mode, the application will respond when it sees the banknote, and the application will automatically become ready for new recognition. In single mode, users need to start the recognition by touch the screen. The strength of this application is its support offline, embedded with counter function and perform recognition in real-time. The weakness of this application is the feedback sound not workable except the tic-tac sound.

The third application reviewed at is IDEAL U.S Currency Identifier created by IDEAL Group, Inc. Android Development Team Productivity (IDEAL Group, 2014). The application uses the phone camera to identify the money in real-time. The application will load the database on the first time using and able to use it offline on the next time. The vibration mode will appear automatically on when using the application. User required to point the camera to the banknotes in hand and hear the value feedback, the application able to recognize the front and back of the banknotes. The strength is the mobile application are designed in an easy way to let the user use it. For example, when open the application, it direct went into recognizing mode and able to perform recognition within one second. Next, it able to recognize the front and back of the banknotes. The weakness of this application is that it only supports to read the US currency. Next, it does not automatically trigger the flashlight function.

The fourth application reviewed at is Cash Recognition for Visually impaired created by Kshitiz Rimal (Rimal, 2019). As an initiative by AI for Development and Intel AI Academy, this app is for the Nepalese visually impaired community, and it will aid them to recognize Nepalese paper currencies without any hassle. By hovering their smartphone over their hand while holding paper currency, this app will recognize the value of the currency and play audio enabling the user to hear and know the value of it. This application recognizes Nepali cash notes and plays the sound signifying. The application allows for offline usage. The application is available with Nepalese and English as audio playback options. The strength of this application is great performance when recognizing the banknotes with any angle during the scan process.

The fifth application reviewed at is Myanmar Money Reader created by Myanmar Assistive Technology (Myanmar Assistive Technology Team, 2020). This application using the phone camera to recognize the Myanmar banknotes. The recognized banknotes will read the value to the user. When nothing found, the application will alert the user with text to speech voice. The strength of this application recognizes accurately of Myanmar banknotes and some voice notification to alert the user. The weakness of the application in the direction of the banknotes must in landscape mode, causing not flexible while using the application. Furthermore, it is not a vision impairment friendly. For example, the notification only alerts the user once when no banknotes were detected, and there is no feedback to the user to inform the current status of the application such like vibration for inform user the application is still running.

Table 2.2: Feature comparison between the existing banknotes mobile application.

Application / feature	Cash Reader	MCT money reader	IDEAL U.S Currency Identifier	Cash Recognition for Visually impaired	Myanmar Money Reader
Detect banknotes in real-time	✓	✓	✓	✓	✓
Automatically trigger flashlight	✓	✓			
Vibration notifier	✓		✓		
Voice feedback	✓		✓		✓
Recognize Malaysia Banknotes	✓				
Vision impaired navigation friendly			✓		✓

2.3 Literature Review on Proposed Mobile Net V2 Network

CNN a deep learning technique, which widely used in image classification. A CNN works by extricating highlights from pictures. This kills the require for manual highlight extraction. The highlights are not prepared. They're learned whereas the arrange trains on a set of pictures. This makes profound learning models greatly precise for computer vision errands. CNNs learn highlight discovery through tens or hundreds of covered up layers. Each layer increments the complexity of the learned features.

In this project, we proposed on using Mobile Net V2 which is a Pre-trained Convolutional Neural Network which objectively designed for mobile vision due to the lower parameters learned during training process. Depthwise Separable Convolution are the technique used in this CNN network. Depthwise Separable Convolution separate to two parts. The first part is depthwise convolution, it performs filtering per input channel. The second part is pointwise convolution uses a 1x1 kernel iterate through computing the linear combination of each layer. This technique results of less computation compare to the regular convolution and make the model lighter. There is three layers for building block structure in this model. The first layer is one times one convolution with ReLU6, second layer represent the depthwise convolution and the last layer will be the one times one convolution. The overall architecture are shows at Table 3.2.2.1, where the t stand for the expansion factor, c stands for the number of outputs channels, n stand for the repeating number and s stand for the stride (Sandler *et al.*, 2018).

Table 2.3: Model architecture of MobileNet v2.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

2.4 Critical Remarks on Previous Works

There exist several applications which providing banknotes classification services. However, exits of several problems within those applications cannot recognize both sides of the banknotes, no proper feedback for the users. There are various classification models to recognize the banknotes, but the performance is not stable due to different design banknotes from different countries. Most importantly, most of the application host the classification model at the server due to high computational power needed. These limitations can be solved using a more effective reorganization method and light weight classification model on the Malaysia banknotes which direct embedded on the application. To have better usage for the vision impairment, some improvement on vision impairment friendly needs to integrate into the mobile application.

Chapter 3: System Methodology

3.1 Incremental Project Developments

We are using incremental development to build our project in our proposed mobile application development for visual impairment to recognize Malaysia banknotes. The requirement and functions needed are identified in the outline description. The specification, development and validation activities are interleaved rather than separate. The documentation that has to be redone will be lesser and at the same time able to rapid delivery and deployment of helpful software. We regular deliverables to measure progress to ensure all activities involved would not lose connection with each other. Figure 3.1 shows the development process flow development for this project.

Details in incremental project development:

Outline Description

The defined problem statement and objective stated corresponding for the reference for the following development. Information collected related and predefined the workflow step to build out the entire system. Overall system diagram builds out for the following phase guide.

Specification

The requirement identified for building the application, hardware and software requirement included. Technology and tools confirmation involved in this project. The details system design for object detection and mobile application builds out.

Development

System implementation and development for the both object detection model and mobile application. The library used are define from the specification stage. Details of the implementation and deployment showed in this phase.

Validation

System testing involved both object detection models in testing the accuracy and test case build-out for specific functions and the general uses on the mobile application.

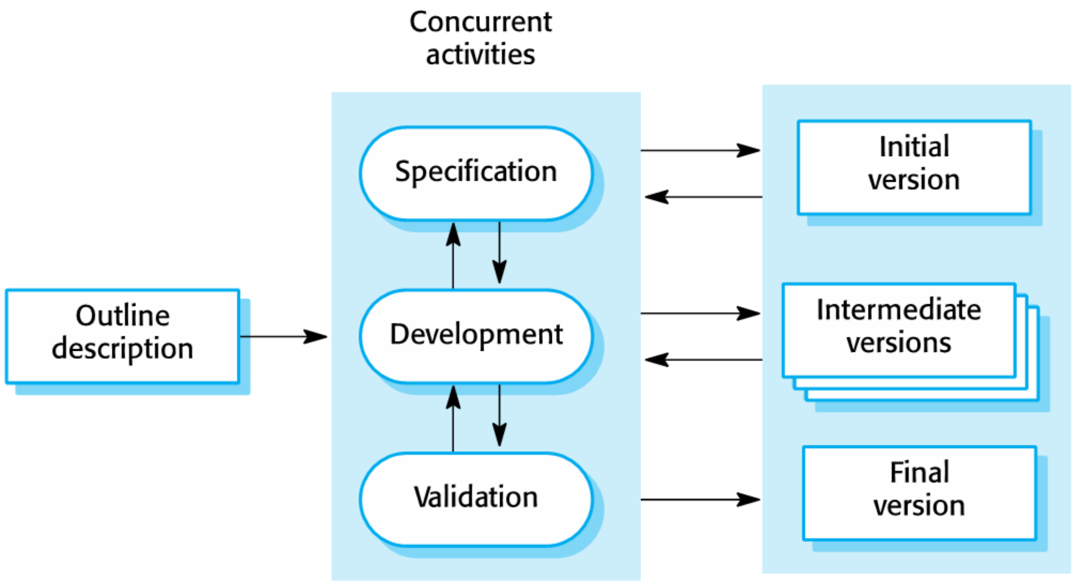


Figure 3.1: Incremental project development illustrate flow(Central Connecticut State University, 2016).

3.2 Tools and Technologies Involved

<u>Software</u>

1. Jupiter Notebook – version 6.4.0

- Open-source web application for integrate python code on building object detection model

2.Android Studio

- Based on JetBrains' IntelliJ IDEA software and developed specifically for Android production, this is the official integrated development environment for Google's Android operating system.

3. Python Anaconda – version 4.0.1

- Simplify package management and deployment of high-level interpreted programming language for scientific computing.

4. CUDA toolkit – version 11.2

- Computer application to enabled the graphics card used to boost up the training process in building object detection model.

5. TensorFlow – version 2.5.0

- Open-source library software for machine learning. In this project we utilize the object detection API from the TensorFlow library.

Coding Languages

1. Python

- programming language for interpreted high-level general used, we using python language to build out the object detection model.

2.Java

- Object oriented programming language which widely used in develop mobile application in android studio.

3. XML

- Markup language for design the interface for the mobile application.

Hardware

1. Laptop (Asus ROG G531G)

Table 3.1: Laptop specification.

Operating System	Windows 10 Home 64-bit
Processor	Intel® Core™ i5-9300H
RAM	DDR4 4G
Graphic Card	NVIDIA® GeForce GTX™ 1650 4GB GDDR5 VRAM

2. Smartphone (Xiaomi Pocophone F1)

Table 3.2: Phone specification

Operating System	MIUI Global 11.0.9
Android Version	10 QKQ1.190828.002
Processor	Qualcomm Snapdragon 845
RAM	6 GB
CPU	Octa-ore Max 2.8GHz

3.3 Requirements

The requirements for the banknotes recognizer mobile application include of functional requirements and non-functional requirements. Functional requirements relate the process of the system on mobile application has to perform. Non-functional requirements relate to the performance and usability on the mobile application.

3.3.1 Functional Requirements

<u>On-boarding guideline for first-time users</u>

- The system will check the first time use of the application.
- The system will proceed to guideline if is first-time used.
- The system will provide voice feedback of the content regard the tutorial.
- The system will notice user to proceed to the main function by sliding the screen left or right.
- The user can swipe left or right to enter the main function.
- The system will not proceed to guideline if it is not first-time used and will direct to main function.

<u>Voice feedback on detected banknotes</u>

- The users can point the phone camera toward the banknotes.
- The system will capture the video stream to the detection model to predict the value.
- The system will provide the voice feedback corresponding to the detected banknotes value.

<u>Vibration feedback</u>

- The system will provide vibration feedback once the detection mode on
- The system will loop the feedbacks every two seconds
- The users can feel the feedback indicate the detection is on going

3.3.2 Non-Functional Requirements

Operational Requirements

- The system will operate in Android environments
- The system will be able to detect the banknotes from the input image stream
- The system will be able to provide voice feedback
- The system will be able to run in no internet connection environment

Performance requirements

- Detection response times will able less than 5 seconds.
- Vibration will able to response to users every 2 seconds

3.4 Project Timeline

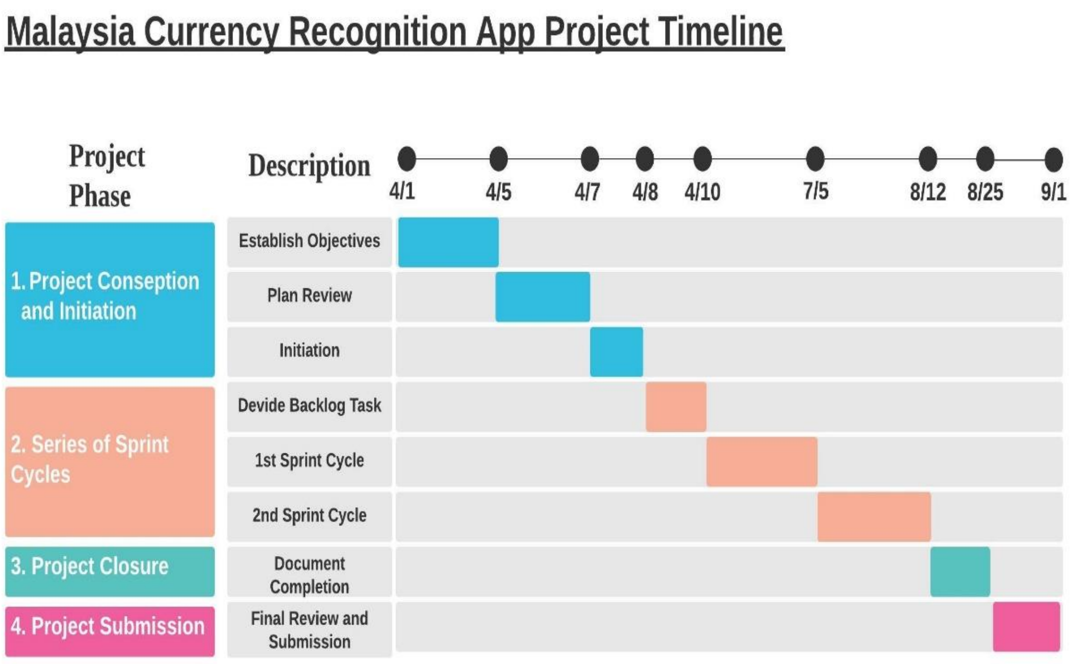


Figure 3.2: Gantt Chart showing timeline of project development progression.

Chapter 4: System Design

4.1 System Block Diagram

In this section includes of the system block diagram for the overall system design and the details for recognition system and mobile application system.

4.1.1 General Project Block Diagram

Malaysia banknotes recognition system organize into three stages: development stage, complete code stage, and output stage. These three stages are the overall top to the down system implemented in the project.

The development stage is divided into mobile application development and object detection model development. We used TensorFlow to train and fine-tune the model through transfer learning in the object detection model development. One hundred each picture of RM1, RM5, RM10, RM20, RM50, and RM100 prepared as the dataset or train and validated purposes for the deep learning model. Converted TensorFlow lite model file exported in this stage and deployed in the mobile application. Detail's system design on object detection model development refers to chapter 4.1.2 object detection model system block diagram.

Next, we designed and refactored a structured code module to adapt the object detection model in mobile application development. The complete trained model deployed into an android mobile application—the features embedded in the android mobile application mainly design for vision impairment. At the first-time usage, the application triggers the usage for this application. When the application starts, a scheduled vibration will act as a physical vibration to indicate the application is running and in recognizing mode. The application feedbacks the banknotes' value while it successfully captures the banknotes that show to the camera. Detail's system design on mobile application development refers to chapter 4.1.3 system design on Malaysia banknotes recognizer application. Secondly, the complete code stage combined the exported model file and the mobile application code. Lastly, the installed application and go through the tested process. The proposed project overall workflow refers to figure 3.1 Malaysia currency recognition project overall workflow.

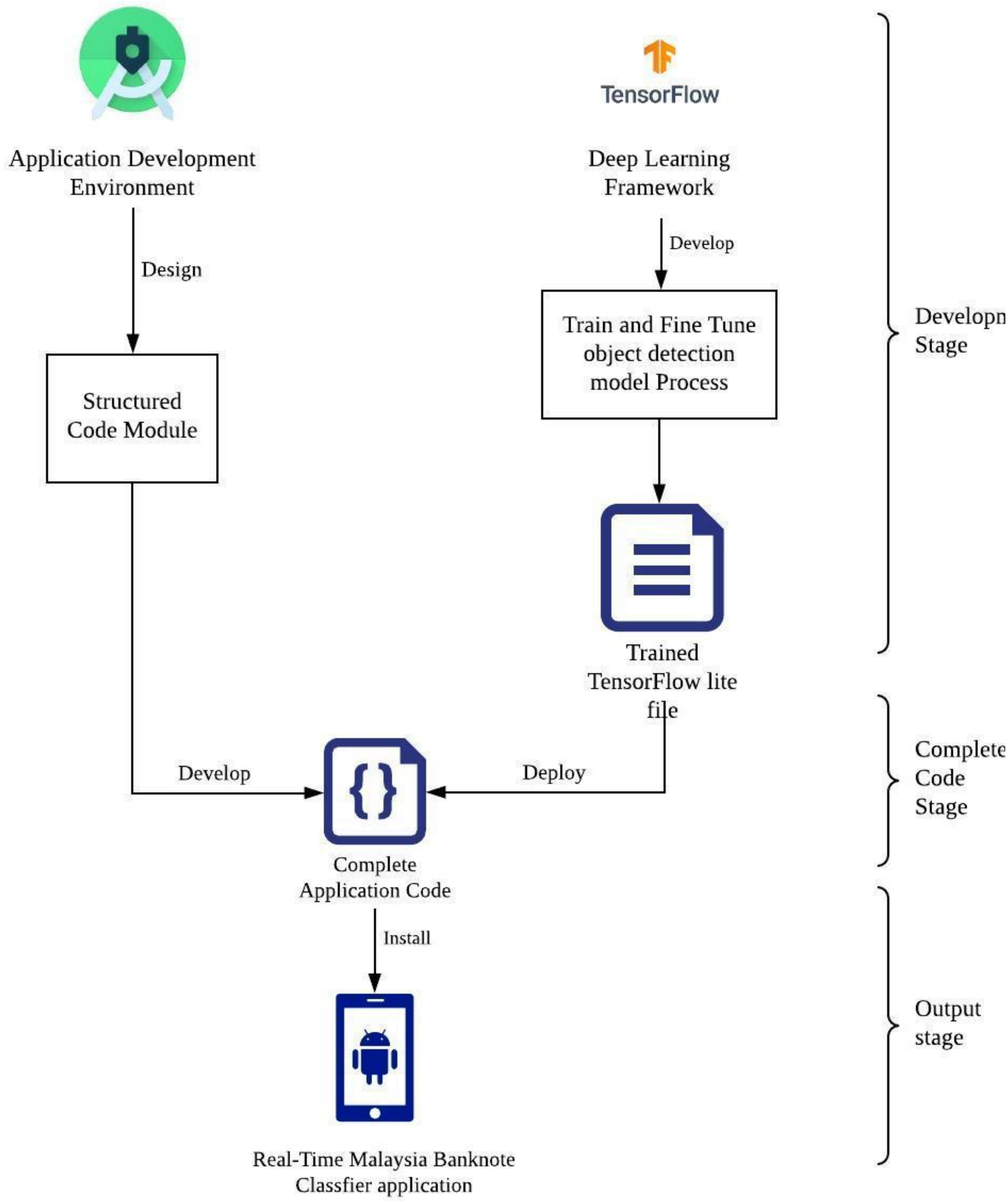


Figure 4.1 Malaysia banknotes recognition project overall workflow.

4.1.2 Object Detection Model System Block Diagram

In training, a deep learning model for banknotes recognition is complete in a system block diagram shown in figure 4.2. The deep learning technique was used in this project to build a model objectively to detect Malaysia banknotes. We apply SSD mobilenet v2, a pre-trained convolutional network designed for deployment on mobile applications.

The details step describes below:

Image Data Preparations and Annotations

One hundred of each value of the Malaysia banknotes collected from phone camera. Images collected with different angle and position. Including horizontal flip and vertical flip. Annotation with Labelling software to produce XML files for the bounding box.

Label Map file

Generate label map file for annotate each class object we detected in the model. Label map file used to label the object we detected in this project with string value.

Import Pre-trained model

We create base model from the pre-trained ssd_mobilenet_v2. The pre-trained model pretrained on coco dataset which consists of 328K images. We utilised the parameters and tuned hyperparameters in this pre-trained model to transfer learning to our project.

Generate tfrecord file

Record files generate from the input of label map, test and train file. These files store the data in the format that recognize by TensorFlow while training.

Configure Pipeline file

We configure the attributes to the adapt on the model while training on the pipeline file. Mainly configure on the number of classes, feature extractor type, batch size and the input path of the pre-trained model, tfrecord file and label map file.

Training Pre-trained model

We start training on the pre-trained model. Training process record at checkpoint to ensure the progress on going. The training process might face external issue lead the training process to stop. Hence the checkpoint will save the latest interrupt process, and enable to continue to train back from the saved checkpoint. The evaluation of the accuracy of the training using Tensor Board. After evaluation we used the last checkpoint file to generate a model file.

Convert to tflite file

We export the SSD frozen tflite graph used for deployed on the embedded devices such like mobile application. Lastly, the saved model converted to tflite file.

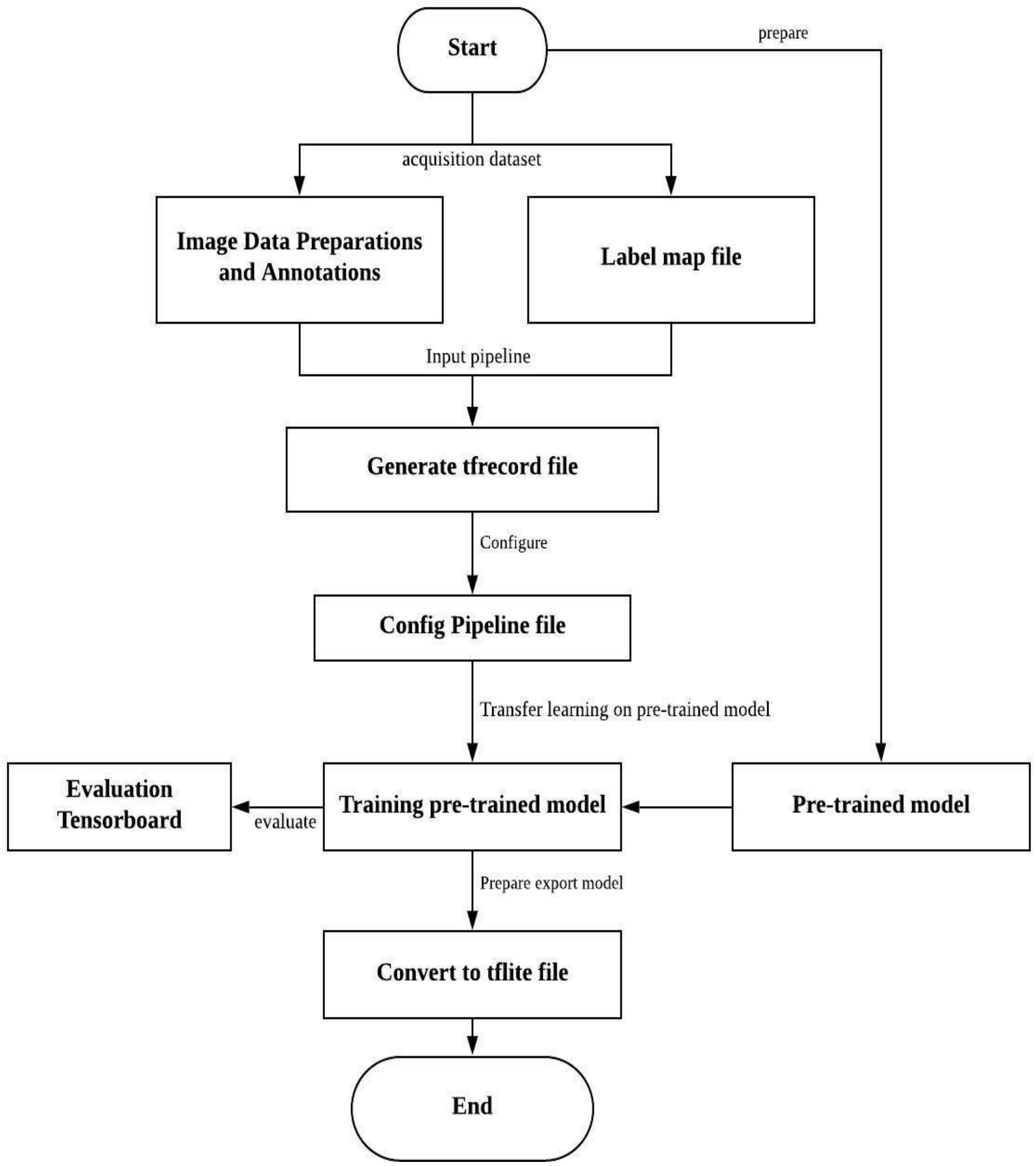


Figure 4.2 Object Detection Model System Block Diagram.

4.1.3 System Design on Malaysia Banknotes Recognizer Mobile Application

The features embedded in the android mobile application mainly design for vision impairment. At the first-time usage, the application will trigger the first-time use guideline in audio format. When the application starts up, a scheduled vibration will act as a physical notification for the users to know the application is running and recognizing mode. The application feedbacks the voice with the corresponding value of the banknotes while it successfully captures the banknotes that show to the camera.

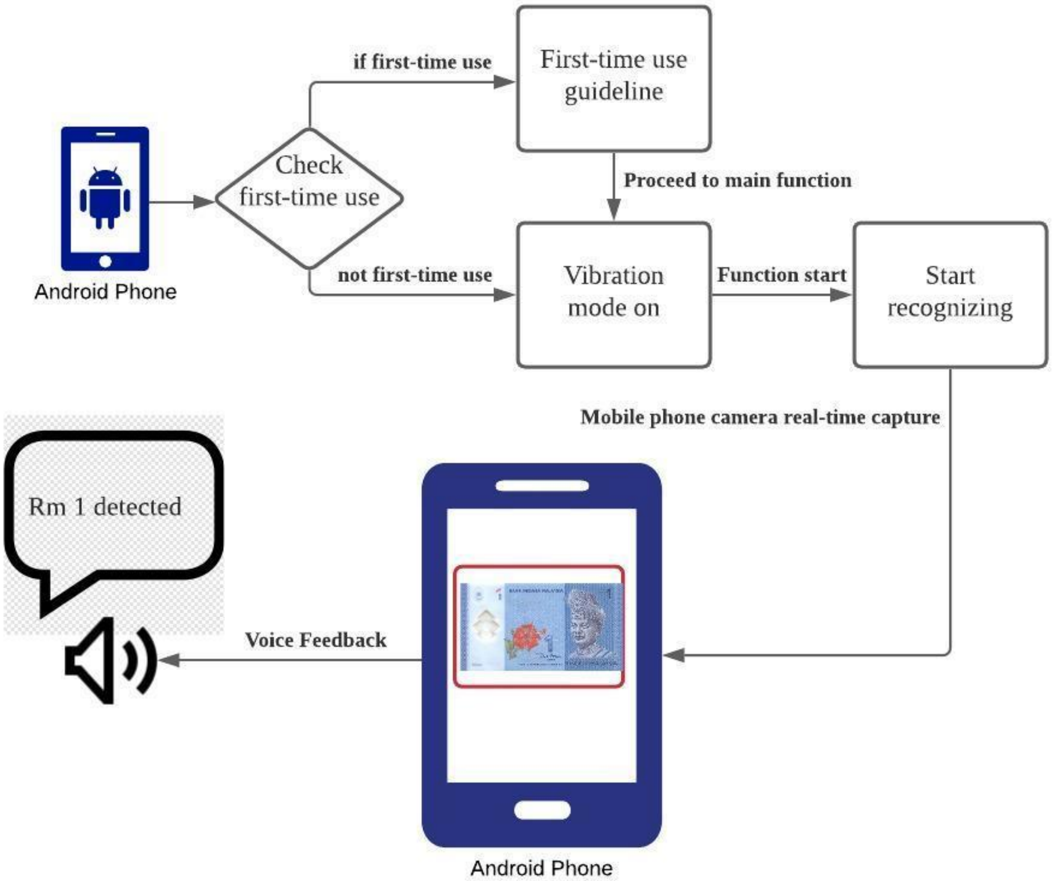


Figure 4.3: System design on Malaysia banknotes recognizer mobile application.

4.2 Use Case Design

We captured the requirements to illustrates between the system and the environment. We visualize the interact within user and the external environment that interacts with the system itself.

4.2.1 Use Case Diagram

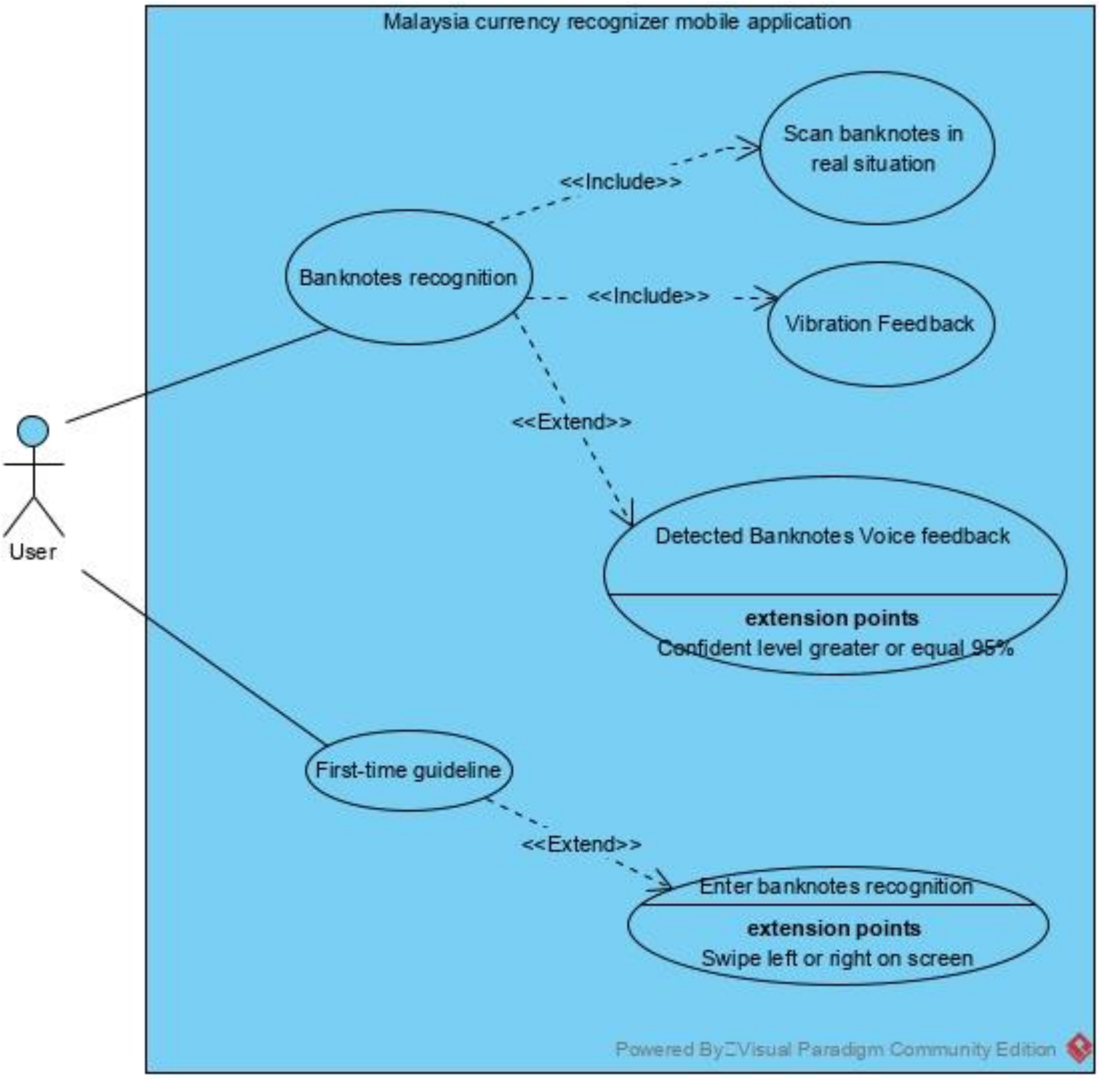


Figure 4.4: Use case diagram for Malaysia banknotes recognizer mobile application.

4.2.2 Use Case Description

Table 4.1: Use case description – Banknote's recognition.

Use case ID:	1
Use case name:	Banknote's recognition
Actor:	User
Brief Description:	Malaysia currency recognizer mobile application start up
Trigger:	User click on the application icons in mobile
Relationships:	Include - Scan banknotes in real situation - Vibration feedback Extend - Detect banknotes voice feedback
Normal Flow of Events:	1. The user clicks on the application icons 2. Application running/start-up
Sub flows:	-
Alternate Flows:	-

Table 4.2: Use case description – Scan banknotes in real situation.

Use case ID:	2
Use case name:	Scan banknotes in real situation.
Actor:	User
Brief Description:	User in action to scan the banknotes in real-time.
Trigger:	User move around the mobile toward the banknote's direction.
Relationships:	-
Normal Flow of Events:	1. The user moves the mobile camera to the direction of the banknotes. 2. The system detects the exist of the banknotes. 3. The system creates a bounding box on the banknotes detected and show on the screen of the mobile application
Sub flows:	-
Alternate Flows:	1. The system cannot detect the exist of the banknotes. 2. The system would not create bounding boxes.

Table 4.3: Use case description – Vibration feedback.

Use case ID:	3
Use case name:	Vibration feedback
Actor:	User
Brief Description:	User receive vibration feedback while the application running.
Trigger:	Banknote's recognition activity start.
Relationships:	-
Normal Flow of Events:	1. The system banknote's recognition activity start. 2. The user receives the vibration feedback.
Sub flows:	-
Alternate Flows:	1. The application shut down. 2. The user not receives vibration feedback 3. The use case close.

Table 4.4: Use case description – Detected banknotes voice feedback.

Use case ID:	4
Use case name:	Detected banknotes voice feedback.
Actor:	User
Brief Description:	The system provides voice feedback once banknotes detected.
Trigger:	Banknotes successful detected with confident level over or equal ninety-five percent.
Relationships:	-
Normal Flow of Events:	1. The system successfully detected banknotes. 2. The system confident level on the detected banknotes over or equal ninety-five percent. 3. The system provides voice feedback corresponding to the value of the detected banknotes. 4. The user received the voice feedback.
Sub flows:	-
Alternate Flows:	1. The system not detected banknotes. 2. The system detected banknotes but confident level not over or equal ninety-five percent. 3. The system not providing voice feedback. 4. The use case end.

Table 4.5: Use case description – First-time guideline.

Use case ID:	5
Use case name:	First-time guideline
Actor:	User
Brief Description:	Provide on-boarding guideline for first-time users.
Trigger:	User enter the application for the first-time.
Relationships:	Include - Enter banknotes application
Normal Flow of Events:	1. The users enter the application for the first times. 2. The system provides the guideline/on-boarding screen to user. 3. The system provides voice feedback on guideline. 4. The user received the voice feedback on guideline
Sub flows:	-
Alternate Flows:	1. The user not enter the application for the first times. 2. The system not providing the guidelines. 3. The system enters the banknotes recognition.

Table 4.6: Use case description – Enter banknotes recognition.

Use case ID:	6
Use case name:	Enter banknotes recognition
Actor:	User
Brief Description:	The user enters the banknotes recognition mode.
Trigger:	The user swipe left or right on the screen.
Relationships:	-
Normal Flow of Events:	1. The users swipe left or right. 2. The system enters the banknotes recognition mode.
Sub flows:	-
Alternate Flows:	-

4.3 Activity Diagram

The activity diagram provides the model processes in the system. The workflows are shown in this diagram to shows the activity top to down flow while activate the banknotes recognition application.

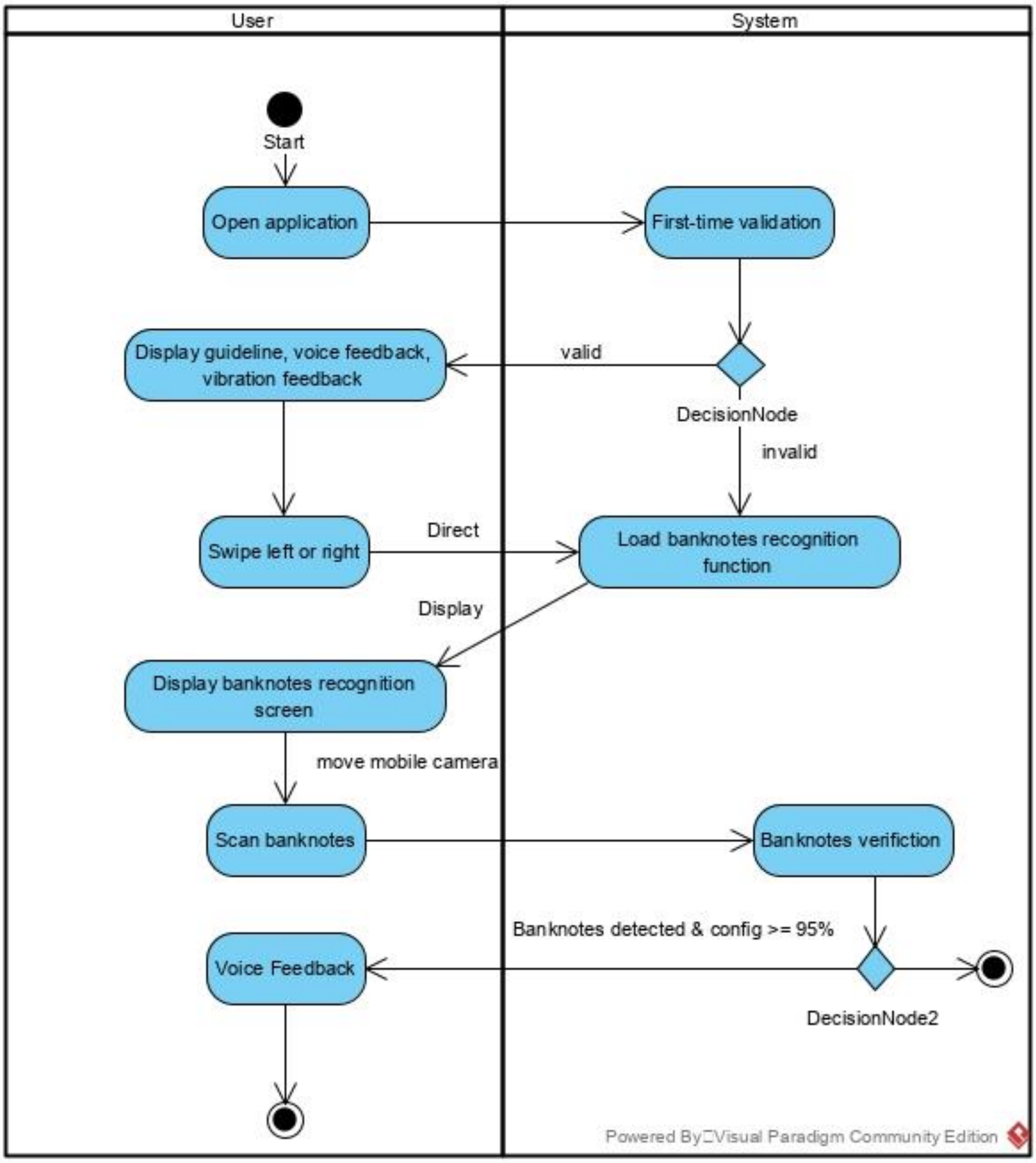


Figure 4.5: Activity diagram for Malaysia banknotes recognizer mobile application.

4.4 User Interface Design

4.4.1 User Interface Design on Malaysia Banknotes Recognition Mobile Application

The user interface design mainly separates into two activities in the application. The first activities are the first-time guideline screen providing the guideline tutorial for the first-time users with voice feedback refer to figure 4.6. The first part of the screen is the title, the next part is the guideline content, and the last part is the swipe notification. The all-context guideline display on the user interface will be provided with voice feedback by the system. The screen is enabled with audio and vibration feedback.

Figure 4.6: User interface design – guideline screen.

The next screen is the system starting the banknotes recognition activities refer to figure 4.7. The top right corner is designed with the application name banner. The middle section of the screen includes the detection frame where the recognition activities will occur. Inside, the detection frame will have the bounding box while banknotes detected, and the banknotes value and config value will show on top of the bounding box. The screen is enabled with audio and vibration feedback.

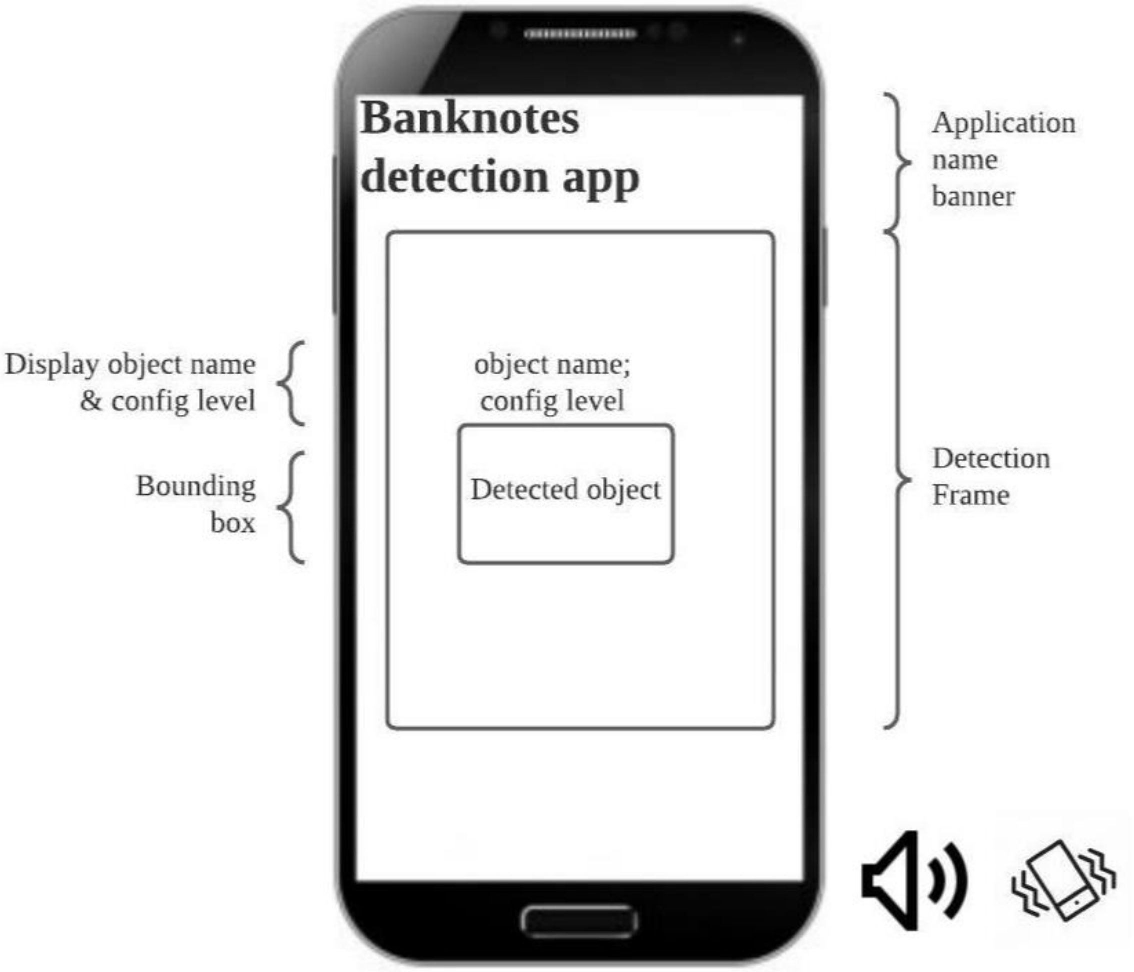


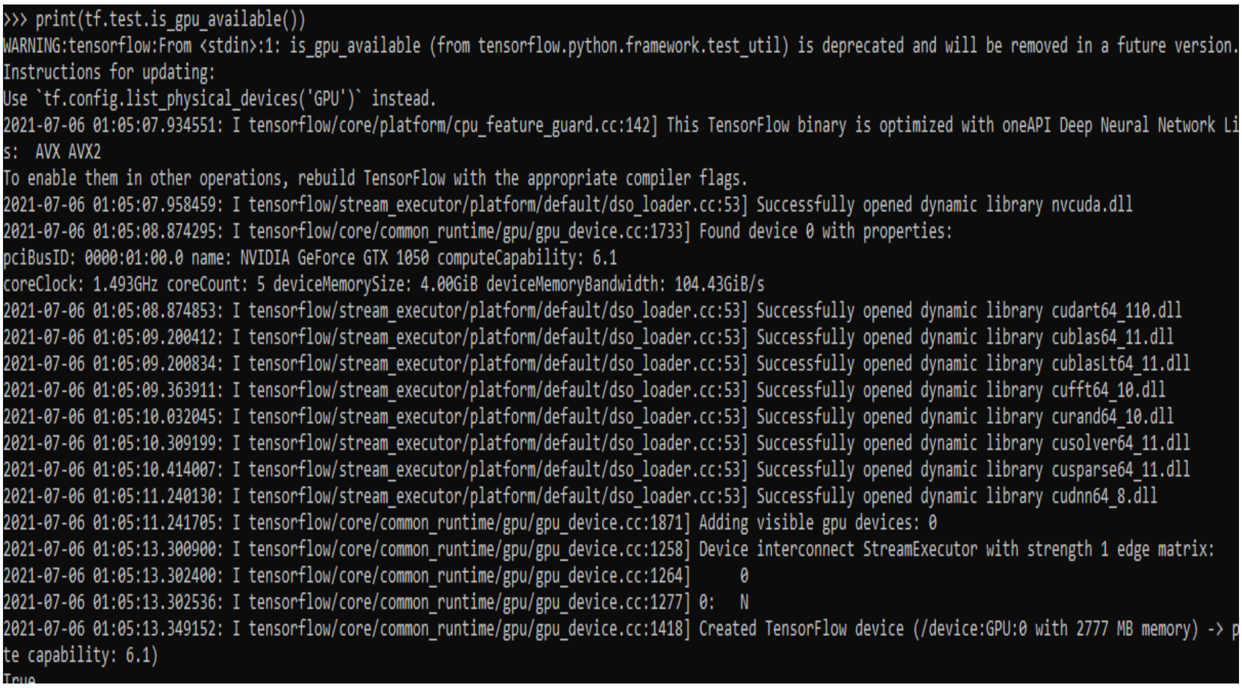
Figure 4.7: User interface design – recognition screen.

Chapter 5: System Implementation

5.1 Banknotes Recognition Model System Implementation

5.1.1 Pre-settings for Training Environment

CUDA toolkit version 11.2 installed compatible with TensorFlow version 2.5.0 and python version 3.9. CUDA toolkit utilizes the GPU power on the laptop and increases the speed while training the recognition model. Figure 5.1 show the CUDA toolkit successfully utilize with .is_gpu_available() command.



```
>>> print(tf.test.is_gpu_available())
WARNING:tensorflow:From <stdin>:1: is_gpu_available (from tensorflow.python.framework.test_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.config.list_physical_devices('GPU')` instead.
2021-07-06 01:05:07.934551: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Li
s:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-07-06 01:05:07.958459: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library nvcuda.dll
2021-07-06 01:05:08.874295: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1733] Found device 0 with properties:
pciBusID: 0000:01:00.0 name: NVIDIA GeForce GTX 1050 computeCapability: 6.1
coreClock: 1.493GHz coreCount: 5 deviceMemorySize: 4.00GiB deviceMemoryBandwidth: 104.43GiB/s
2021-07-06 01:05:08.874853: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudart64_110.dll
2021-07-06 01:05:09.200412: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cublas64_11.dll
2021-07-06 01:05:09.200834: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cublasLt64_11.dll
2021-07-06 01:05:09.363911: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cufft64_10.dll
2021-07-06 01:05:10.032045: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library curand64_10.dll
2021-07-06 01:05:10.309199: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cusolver64_11.dll
2021-07-06 01:05:10.414007: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cusparse64_11.dll
2021-07-06 01:05:11.240130: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudnn64_8.dll
2021-07-06 01:05:11.241705: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1871] Adding visible gpu devices: 0
2021-07-06 01:05:13.300900: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1258] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-07-06 01:05:13.302400: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1264]      0
2021-07-06 01:05:13.302536: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1277] 0:   N
2021-07-06 01:05:13.349152: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1418] Created TensorFlow device (/device:GPU:0 with 2777 MB memory) -> p
te capability: 6.1)
True
```

Figure 5.1: Check environment utilizing GPU power.

Anaconda environment was created for training the model from the computer system. Jupiter notebook and TensorFlow installed in the environment. TensorFlow model garden, we utilize the state-of-the-art models that help develop the recognition model (Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang Li, Frederick Liu, Jaeyoun Kim, 2020). COCO API was installed in the environment for later training the object detection model. Figure 5.2 below shows the directory path set up for the training environment. The annotations folder stores the label map file, test and train record files. Next, in the images folder stores train and test images with corresponding XML files. Moreover, the model files store the checkpoint while training the model and also output directory of the final exported model. Lastly, the pre-trained -model stores the pre-trained SSD mobilenet V2 architecture model from the coco image net.

```
(tfod) C:\RealTimeObjectDetection\Tensorflow\workspace>tree
Folder PATH listing for volume OS
Volume serial number is F895-B107
C:.
├───annotations
│   └───.ipynb_checkpoints
├───images
│   ├───test
│   └───train
├───models
│   ├───my_ssd_mobnet
│   │   └───train
│   └───my_ssd_mobnet_2
│       ├───eval
│       ├───exported_model
│       │   ├───checkpoint
│       │   └───saved_model
│       │       ├───assets
│       │       └───variables
│       ├───exported_model_2
│       │   └───saved_model
│       │       ├───assets
│       │       └───variables
│       ├───tflite
│       └───train
└───pre-trained-models
    └───ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8
        ├───checkpoint
        └───saved_model
            └───variables
```

Figure 5.2: Training environment directory tree.

5.1.2 Data Preparation for Deep Learning Model

The input data are essential to achieve a good model because they will directly affect the output result. Hence, we prepared Malaysian Ringgit to vary from one Ringgit to hundred Ringgit and take the image from a mobile device. Figure 4.1 shows the types of Malaysian Ringgit taken. Total of six hundred banknotes captured and dividing one hundred according to the value itself. Each image is resized to reach the desired size of between 100kb to 200kb.

Figure 5.3: Types of Malaysia banknotes.

The images split to 9:1 ratio to train and test folder respectively. LabelImg software is used to annotate the images with the bounding boxes and label up the corresponding value (Tzutalin, 2015). XML file for each image generated and store the position of the object bounded in the images. The XML file is stored in the corresponding folder with the images folder. Figure 5.4 shows the label on images with LabelImg.

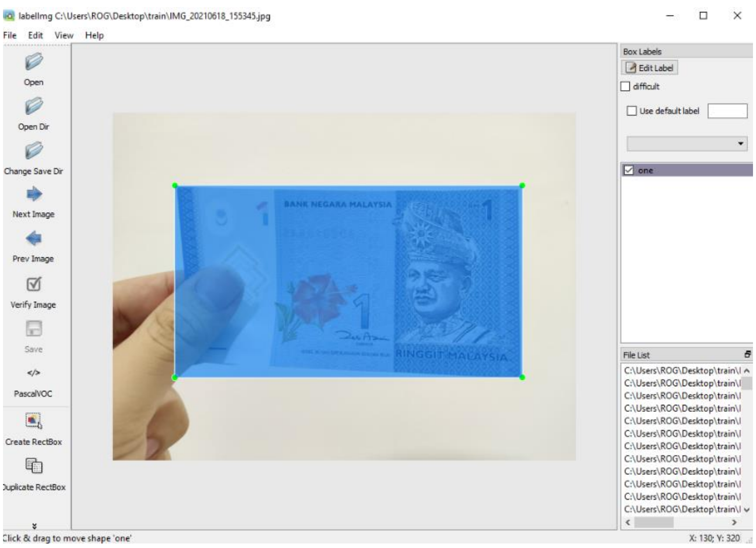


Figure 5.4: Label images with LabelImg software.

5.1.3 Pre-Preparation for Training Model

Label Map created for TensorFlow to map the classes and the ID that going to detect in the model. Six items corresponding to each Malaysia banknotes written in this file refer to figure 5.5.

```
item {
    id:1
    name:'ten'
}
item {
    id:2
    name:'five'
}
item {
    id:3
    name:'one'
}
item {
    id:4
    name:'twenty'
}
item {
    id:5
    name:'fifty'
}
item {
    id:6
    name:'hundred'
}
```

Figure 5.5: Label Map file.

TFRecord train and test file generated with input of train folder, test folder and Label Map file. TFRecord file stores the input data as sequence of binary strings which recognize by TensorFlow framework while used in training stage. Figure 5.6 shows the snippet code for generate the TFRecord file.

```
[ ] !python {SCRIPTS_PATH + '/generate_tfrecord.py'} -x {IMAGE_PATH + '/train'} -l
    {ANNOTATION_PATH + '/label_map.pbtxt'} -o {ANNOTATION_PATH + '/train.record'}

    !python {SCRIPTS_PATH + '/generate_tfrecord.py'} -x {IMAGE_PATH + '/test'} -l
    {ANNOTATION_PATH + '/label_map.pbtxt'} -o {ANNOTATION_PATH + '/test.record'}

    Successfully created the TFRecord file: Tensorflow/workspace/annotations/train.record
    Successfully created the TFRecord file: Tensorflow/workspace/annotations/test.record
```

Figure 5.6: Generate TFRecord file.

The TensorFlow 2 Detection Model Zoo, which provided a various pre-trained models for object detection purposes. The pre-trained SSD MobileNet V2 FPNLite 320x320 model on the COCO dataset used as the transfer learning model which required lower computational power and suitable deployed for mobile application. Depthwise Separable Convolution are the technique used in this CNN network. Depthwise Separable Convolution separate to two parts. The first part is depthwise convolution, it performs filtering per input channel. The second part is pointwise convolution uses a 1x1 kernel iterate through computing the linear combination of each layer. This technique results of less computation compare to the regular convolution and make the model lighter (Sandler *et al.*, 2018).

The configuration for the pipeline config file is mandatory to fit our desire model. The main configuration refers to table 5.1 that implemented in the pipeline config file. The more details that remain in the configuration pipeline file, refer to appendix A-1.

Table 5.1: Configuration in pipeline config file.

Attribute in Pipeline Config	**Value**	**Description**
Num_classes	6	Number of classes to recognize
Batch_size	10	Batch size for training the model
Fine_tune_checkpoint	'/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8/checkpoint/ckpt-0'	Folder path to store the checkpoint while training, to achieve recover if training process interrupt.
Fine_tune_checkpoint_type	detection	Detection to recognition the banknotes
Label_map_path	'/label_map.pbtxt'	Refer to the generated label map file
Tf_record_input_reader	'/train.record' 'test.record'	Refer to the generated record file

5.1.4 Training for Recognition Model

The command to start the training process refer to figure 5.7. This model_main_tf2.py file executes the training process with the input of the pipeline config file that prepared in the previous cycle and outputs the checkpoint at the checkpoint directory refer to figure 5.8. The number of the training steps set to twenty thousand.

```
python Tensorflow/models/research/object_detection/model_main_tf2.py
--model_dir=Tensorflow/workspace/models/my_ssd_mobnet_2
--pipeline_config_path=Tensorflow/workspace/models/my_ssd_mobnet_2/pipeline.config --num_train_steps=20000
```

Figure 5.7: Train execute model command.

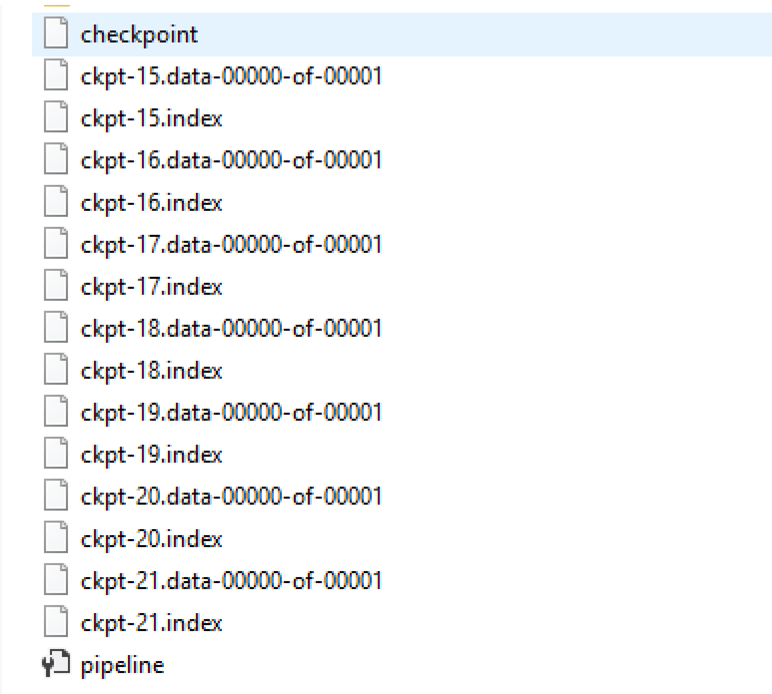


Figure 5.8: Output checkpoint from the training process.

5.1.5 TensorFlow Lite File Conversion and Exportation

Export_tflite_graph_tf2 python execution file export the inference graph from the training process refer to figure 5.9 shows the command to export the tflite inference graph file. Executing a TensorFlow Lite model on-device to create predictions based on input data is referred to as inference. An interpreter is required to run an inference using a TensorFlow Lite model. The TensorFlow Lite interpreter is lightweight and quick. To ensure minimum memory usage, the interpreter utilises a static graph ordering and a specialised (less-dynamic) memory allocator. The output file generated at exported_model_2 folder, refer to figure 5.10. Export_tflite_graph_tf2 python execution file export the inference graph from the training process. Refer to figure 5.9 that shows the command to export the tflite inference graph file. The output file generated at exported_model_2 folder, refer to figure 5.10.

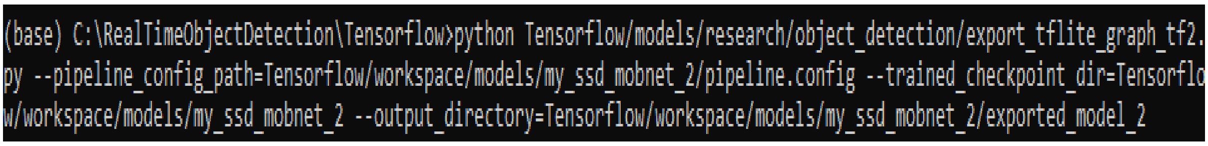


Figure 5.9: Generate save_model.pb file command.

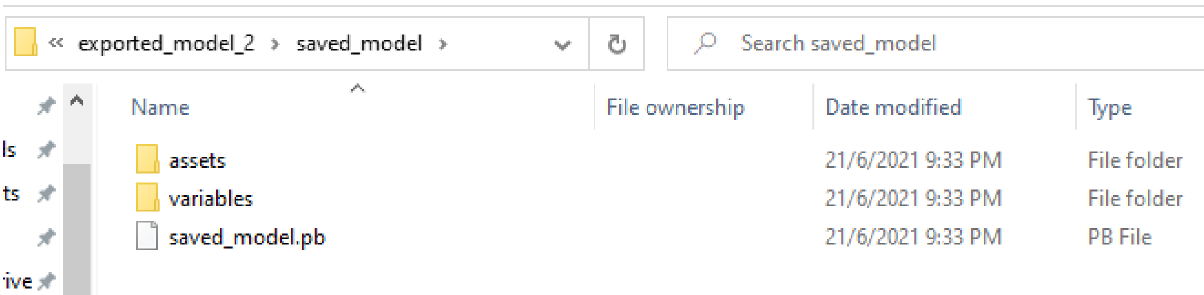


Figure 5.10: Output save_model.pb file.

Next, the save_model file covert to the TensorFlow lite file and add the metadata to the file refer to figure 5.12. The metadata stores the standard of the model description that readable by the Android code generator. Hence it is important to add the metadata into our TensorFlow lite file. Lastly, This TensorFlow lite file will be use to deploy in the mobile application.

```
# Convert the model
converter = tf.lite.TFLiteConverter.from_saved_model('Tensorflow/workspace/models/my_ssd_mobnet_2/exported_model_2/saved_model') #
tflite_model = converter.convert()

# Save the model.
with open('model.tflite', 'wb') as f:
  f.write(tflite_model)
```

```
# Add metadata
ObjectDetectorWriter = object_detector.MetadataWriter
_MODEL_PATH = "Tensorflow/workspace/models/my_ssd_mobnet_2/tflite/model.tflite"
# Task Library expects label files that are in the same format as the one below.
_LABEL_FILE = "Tensorflow/workspace/models/my_ssd_mobnet_2/tflite/labelmap.txt"
_SAVE_TO_PATH = "Tensorflow/workspace/models/my_ssd_mobnet_2/tflite/model_metadata.tflite"
# Normalization parameters is required when reprocessing the image. It is
# optional if the image pixel values are in range of [0, 255] and the input
# tensor is quantized to uint8. See the introduction for normalization and
# quantization parameters below for more details.
# https://www.tensorflow.org/lite/convert/metadata#normalization_and_quantization_parameters)
_INPUT_NORM_MEAN = 127.5
_INPUT_NORM_STD = 127.5

# Create the metadata writer.
writer = ObjectDetectorWriter.create_for_inference(
    writer_utils.load_file(_MODEL_PATH), [_INPUT_NORM_MEAN], [_INPUT_NORM_STD],
    [_LABEL_FILE])

# Verify the metadata generated by metadata writer.
print(writer.get_metadata_json())

# Populate the metadata into the model.
writer_utils.save_file(writer.populate(), _SAVE_TO_PATH)
```

Figure 5.11: TFlite file conversion.

5.2 Banknotes Recognition Mobile Application System Implementation

5.2.1 Pre-Configuration for Application Development

This application developed with android studio version 4.2.2 in Windows operating system. In the base design, we used the TensorFlow Lite object detection android demo, a stable pipeline in-build with the connection between the TensorFlow Lite file and the camera stream (The TensorFlow Authors, 2021). We utilize the environment and deploy modification to meet our project criteria.

TensorFlow lite file generated at system implementation of recognition model and label map text file generated, including the object name as shown in Figure 5.12. This two-file deployed at the asset folder in an Android folder structure.

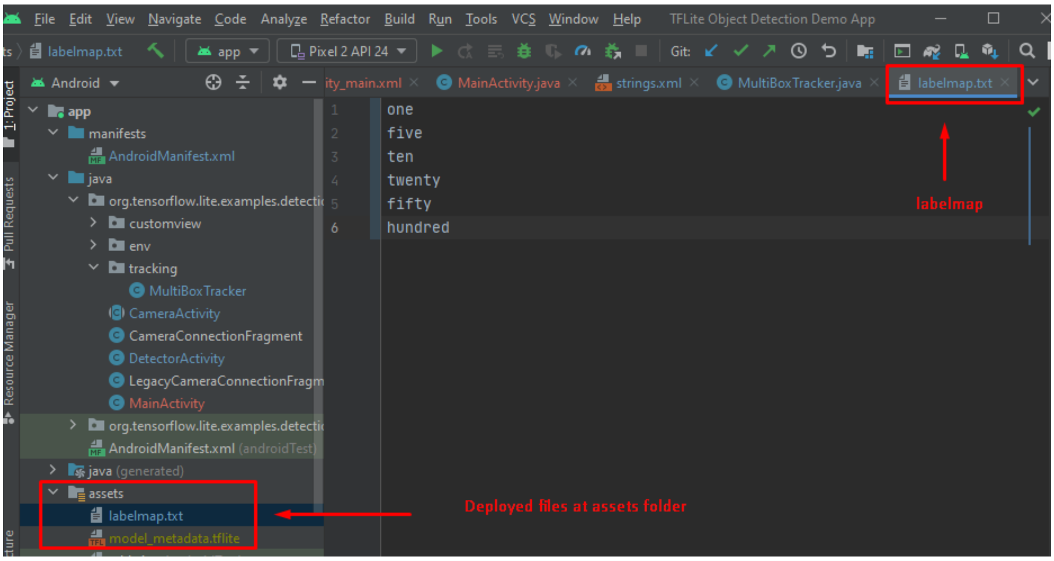


Figure 5.12: TensorFlow lite and label map file deployment.

In the DetectorActivity.java we need to manual configure the input size which compatible to our recognition model, TensorFlow Lite file and label map file as refer to figure 5.13.

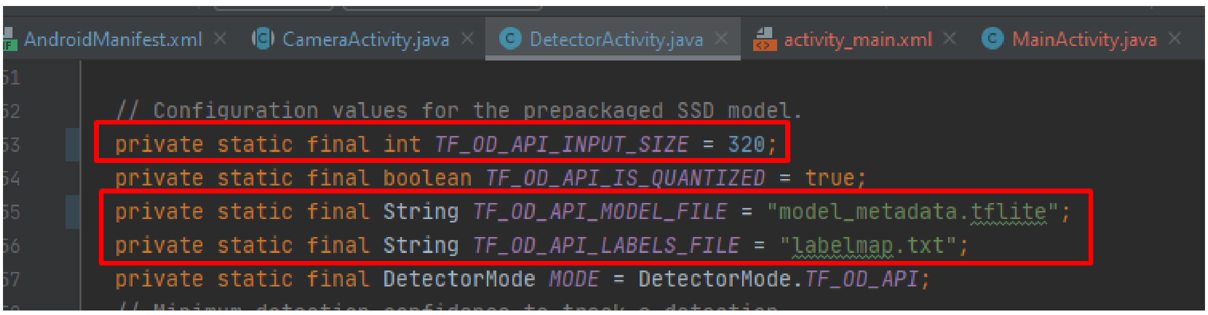


Figure 5.13: Configuration on input pipeline.

5.2.2 Function Implementation for Mobile Application Development

Recognition Audio Feedback Function

Audio mp3 files prepared with corresponding audio of each value of the banknotes. These files stored in the raw folder and Media player API from android libraries used to create the attribute of the six of the audio inputs, as shown in figure 5.14.

```
mp_1 = MediaPlayer.create(context.getApplicationContext(), R.raw.one);
mp_5 = MediaPlayer.create(context.getApplicationContext(), R.raw.five);
mp_10 = MediaPlayer.create(context.getApplicationContext(), R.raw.ten);
mp_20 = MediaPlayer.create(context.getApplicationContext(), R.raw.twenty);
mp_50 = MediaPlayer.create(context.getApplicationContext(), R.raw.fifty);
mp_100 = MediaPlayer.create(context.getApplicationContext(), R.raw.hundred);
```

Figure 5.14: Create media player.

The recognition process returns a list of results with object names and detection confidence levels. We retrieved the results and make validation. When the result obtains an object name and confidence level, over ninety-five per cent will trigger the media player to play the corresponding audio files. The details refer to Appendix B-1 – recognition audio feedback function source code.

On-boarding Guideline Function

This function will only trigger once when the user enters the application for the first time. We used the shared preference function to store a string with a default of false when the application starts the on-resume function trigger and will validate the string value. The validation checks the string value, if it is false hence will proceed to the guideline else will direct the activity to the recognition audio feedback function (moveTosecondary() function) as shown in figure 5.15. This function will only trigger once when the user enters the application for the first time. We used the shared preference function to store a string with the default of false value when the application starts the on-resume function trigger and will validate the string value. The validation checks the string value. If it is a false value, the guideline will proceed to the guideline; otherwise, it will direct the activity to the recognition audio feedback function (moveTosecondary() function), as shown in figure 5.15.

```
protected void onResume() {
    super.onResume();
    SharedPreferences sharedpreferences = getSharedPreferences(getString(R.string.app_name), this.MODE_PRIVATE);
    if (!sharedpreferences.getBoolean(prevStarted,  defValue: false)) {
        SharedPreferences.Editor editor = sharedpreferences.edit();
        editor.putBoolean(prevStarted, Boolean.TRUE);
        editor.apply();
    } else {
        moveToSecondary();
    }
}
```

Figure 5.15: Guideline validation function in MainActivity().

Enter the guideline function, and then the system will trigger the audio feedback with the guideline content. After completing the guideline, it allows the user to swipe left or right to enter the recognition mode. We used touch events to capture users' touch events and trigger the intent function to direct the screen to the recognition screen. The on-touch event function refers to figure 5.16.

```
public boolean onTouchEvent(MotionEvent touchEvent){
    switch(touchEvent.getAction()){
        case MotionEvent.ACTION_DOWN:
            x1 = touchEvent.getX();
            y1 = touchEvent.getY();
            break;
        case MotionEvent.ACTION_UP:
            x2 = touchEvent.getX();
            y2 = touchEvent.getY();
            if(x1 < x2){
            Intent i = new Intent( packageContext: MainActivity.this, DetectorActivity.class);
            startActivity(i);
        }else if(x1 > x2){
            Intent i = new Intent( packageContext: MainActivity.this, DetectorActivity.class);
            startActivity(i);
        }
        break;
    }
    return false;
}
```

Figure 5.16: On touch event function in MainActivity().

Vibrate Feedback Function

The vibration function aims to help the users realize the recognition activity is ongoing so that the user has an awareness of the application's running. The vibration function needs to set the user permission in the android manifest file. The recognition activity on a resume will trigger the vibration service with a looping pattern configuration of vibrating one hundred milliseconds and sleep for one thousand milliseconds. The vibration settings refer to figure 5.17.

```
public synchronized void onResume() {
  LOGGER.d("onResume " + this);
  super.onResume();

  handlerThread = new HandlerThread( name: "inference");
  handlerThread.start();
  handler = new Handler(handlerThread.getLooper());

  // Get instance of Vibrator from current Context
  Vibrator v = (Vibrator) getSystemService(Context.VIBRATOR_SERVICE);

  //...
  long[] pattern = {2000, 100, 1000};

  //...
  v.vibrate(pattern, repeat: 0);
}
```

Figure 5.17: Vibration function in CameraActivity().

5.2.3 User Interface for Mobile Application Development

On-boarding Screen

The application logo set on the top middle of the screen, followed by the guideline title. The centre of the screen includes the guideline content with the corresponding icons. Lastly, at the below, we set a swipe right or left icons. The text style uses bold with a text size of 25 scalable pixels for the main title and 20 scalable pixels for the text content. Figure 5.18 show the user interface of the onboarding screen.

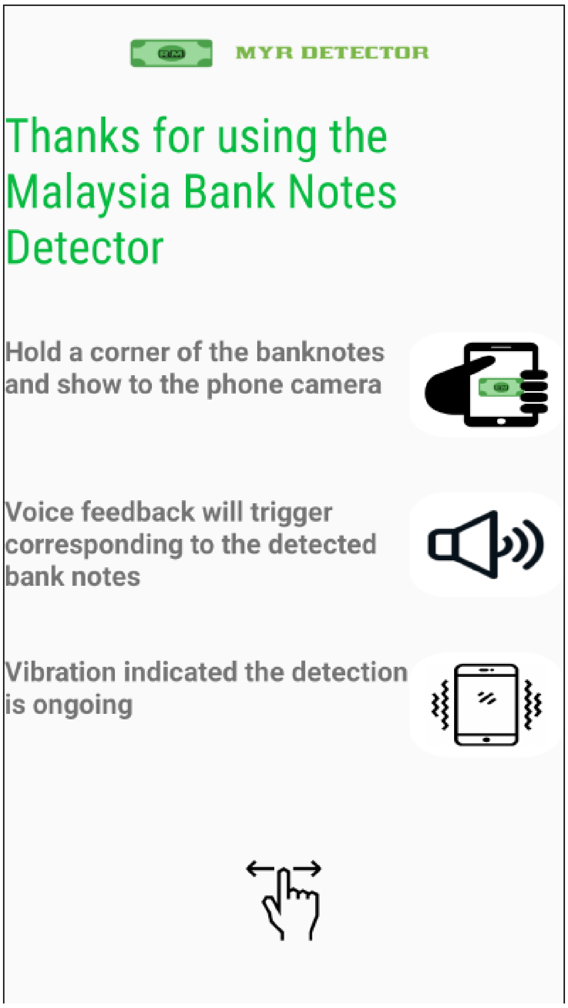


Figure 5.18: On-boarding screen user interface.

Recognition Screen

The recognition screen includes the application name at the top left corner. The middle part is the frame that shows the camera input stream. The bounding boxes with object names and confidence levels will appear once the banknotes detected. Figure 5.19 show the recognition screen user interface.

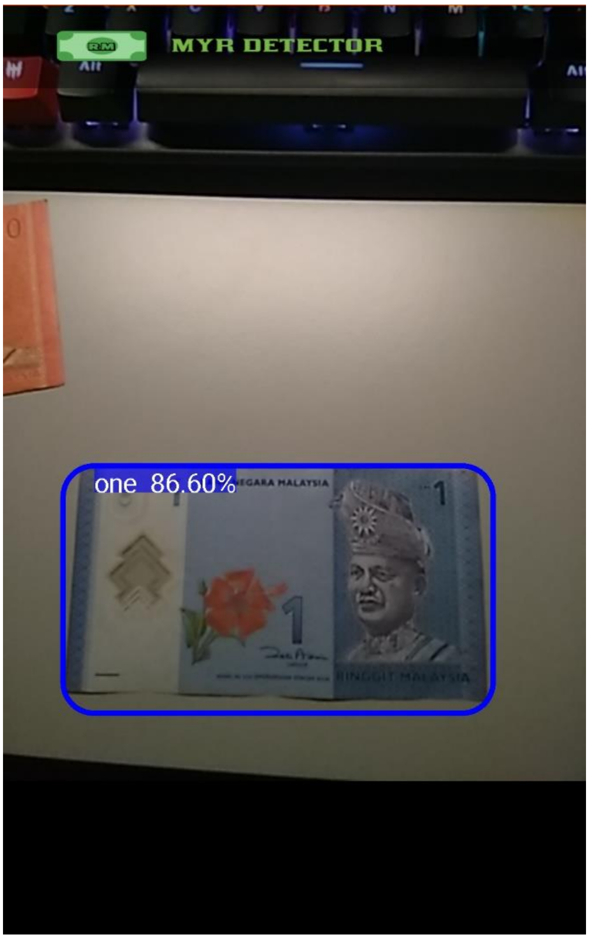


Figure 5.19: Recognition screen user interface.

Chapter 6: System Evaluation

6.1 Evaluation Tools

We used TensorBoard, a visualization toolkit, to visualize a deep learning model's training process and performance. The checkpoint file generated from the training process is needed as the input file to the TensorBoard. Figure 6.1 shows the localhost address, which contains the visual data generated from the TensorBoard. The interface of the TensorBoard refers to figure 6.2.

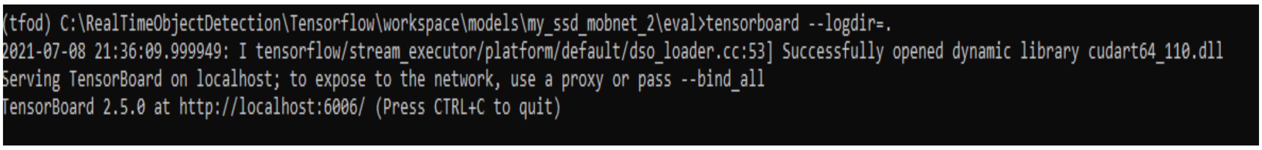

```
(tfod) C:\RealTimeObjectDetection\Tensorflow\workspace\models\my_ssd_mobnet_2\eval>tensorboard --logdir=.
2021-07-08 21:36:09.999949: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudart64_110.dll
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.5.0 at http://localhost:6006/ (Press CTRL+C to quit)
```

Figure 6.1: TensorBoard evaluation command.

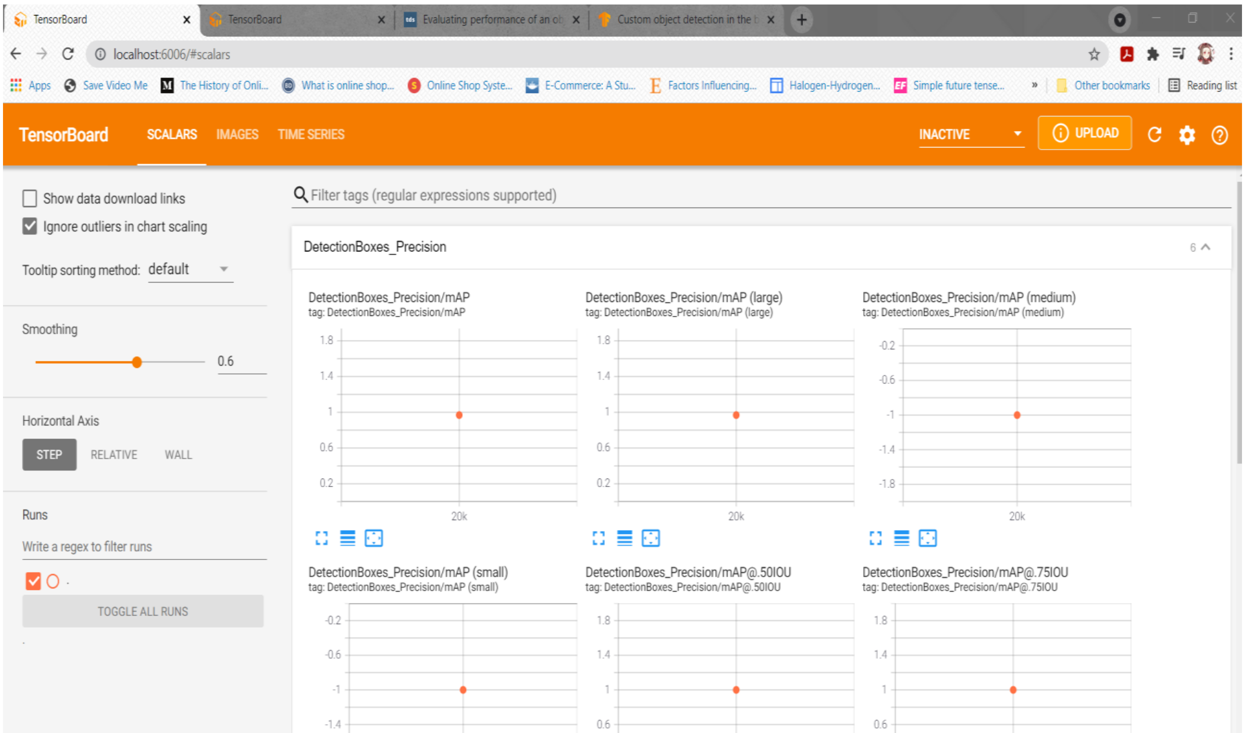


Figure 6.2: TensorBoard evaluation interface.

6.2 Model Performance Evaluation and Discussion

Object detection models include the classification process and localization process. Intersection over union will be the concept for computes the bounding boxes between the prediction and ground truth. The system performance will have relied on the deep learning model, resulting in an accuracy value, which depends on precision and recall. Precision refers to the accuracy of the model, and recall refers to the rate of positive results.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

TP = True positive
TN = True negative
FP = False positive
FN = False negative

Figure 6.3: Precision and recall formula for deep learning model.

Our chosen model is a single shot detection mobile net v2 structure, and it results in a lower time to perform and computational power to perform detection. Hence it suitable to deploy on the mobile application. In addition, the mean average precision reaches ninety-six percent and the recall score of ninety percent, which is a good score for the training results. Figure 6.4 shows the mean average precision result, and figure 6.5 shows the recall score result.

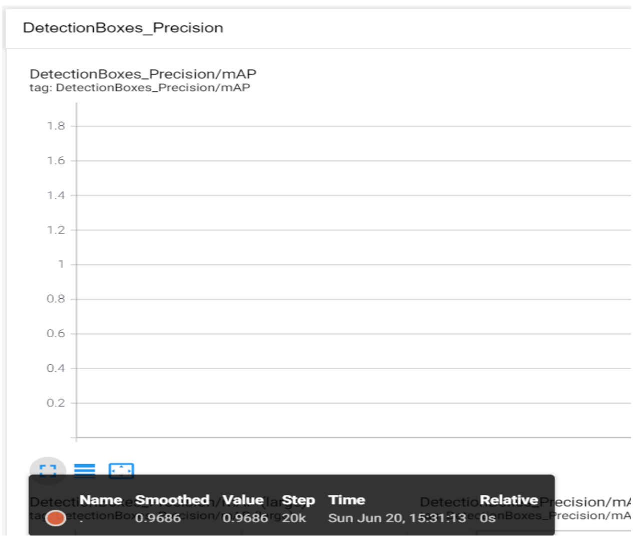


Figure 6.4: Mean average precision score.

Figure 6.5: Recall score.

6.3 Model Training Loses Result

The result from the TensorBoard, the losses include the classification, localization, regularization, and total losses have a significantly decreased slope, indicating the learning process is doing well.

<u>Classification loses at twenty thousand steps: 0.03178</u>

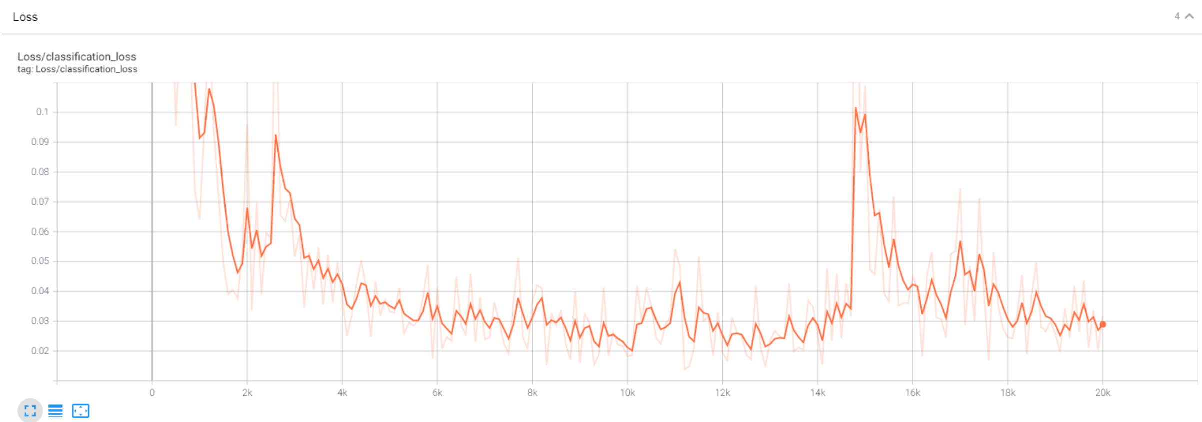


Figure 6.6: Classification loses graph.

Localization loses at twenty thousand steps: 4.8874e-3

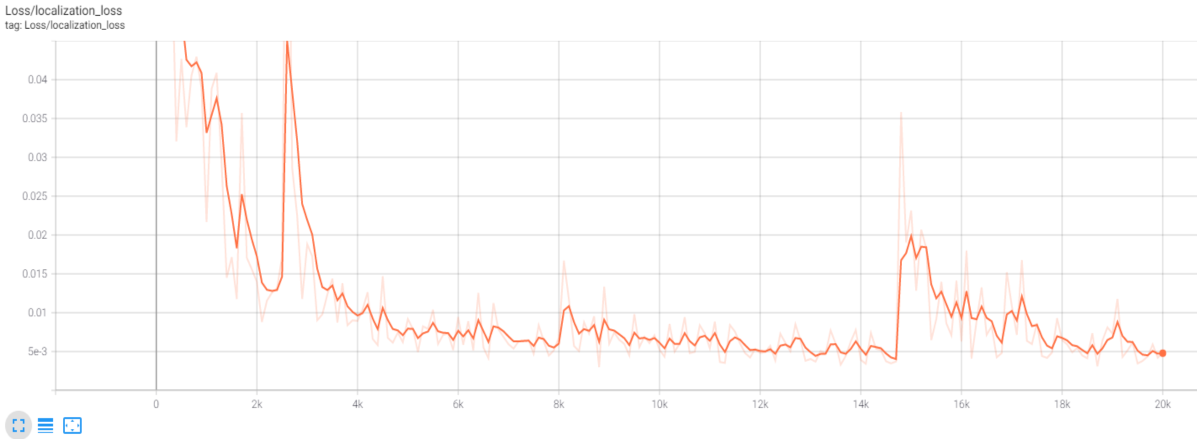


Figure 6.7: Localization loses graph.

Regularization loses at twenty thousand steps: 0.0684

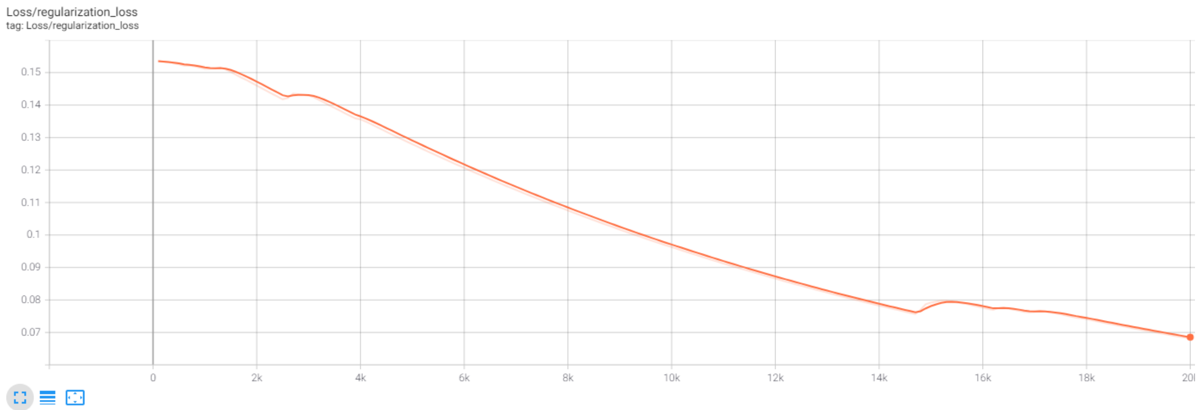


Figure 6.8: Regularization loses graph.

Total loses at twenty thousand steps: 0.1048

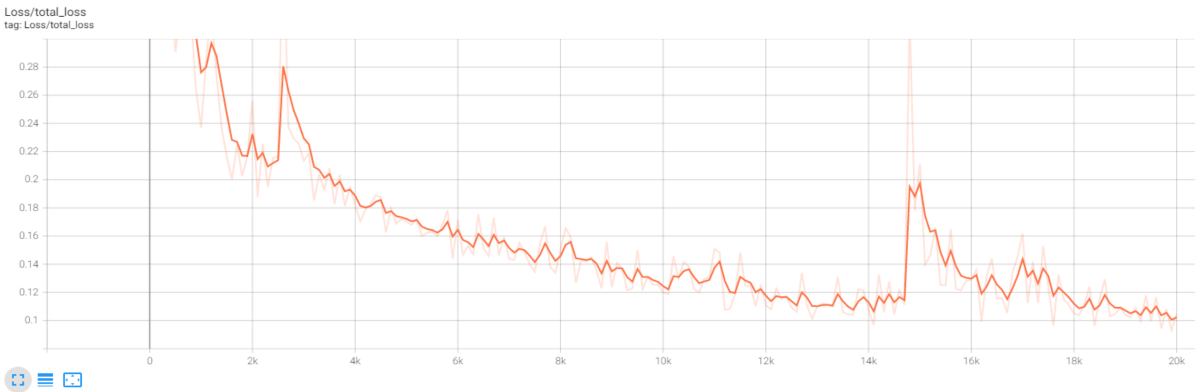


Figure 6.9: Total loses graph.

6.4 Model Performance on Android Devices

We used the benchmark tools provided by TensorFlow, which measures the model's performance in the application, such as initialization time, inference warmup time, inference steady time. The benchmark tool is an APK file installed through Android studio with an ADB command in the terminal. The execution command runs with a configuration of using 4 CPUs.

```
C:\Users\ROG\ApkProjects\android_aarch64_benchmark_model>adb -d shell am start -S -n org.tensorflow.lite.benchmark/
.BenchmarkModelActivity --es args '"--graph=/data/local/tmp/model.tflite --num_threads=4"'
Stopping: org.tensorflow.lite.benchmark
Starting: Intent { cmp=org.tensorflow.lite.benchmark/.BenchmarkModelActivity (has extras) }
```

Figure 6.10: Benchmark execution command.

The output from the benchmark application will prompt at the logcat in the android studio. The performance data from the output refer to table 6.1. The system achieved an average of 55686 microseconds while performing inference or recognizing the banknotes.

```
2021-07-15 00:34:43.646 5228-5228/org.tensorflow.lite.benchmark I/tflite_BenchmarkModelActivity: Running
 TensorFlow Lite benchmark with args: --graph=/data/local/tmp/model.tflite --num_threads=4
2021-07-15 00:34:43.650 5228-5228/org.tensorflow.lite.benchmark I/tflite: Initialized TensorFlow Lite
 runtime.
2021-07-15 00:34:46.952 5228-5228/org.tensorflow.lite.benchmark E/tflite: Average inference timings in us:
 Warmup: 62905, Init: 1483, Inference: 55686.3Overall max resident set size = 34.3555 MB, total malloc-ed
 size = 0 MB, in-use allocated/mmapped size = 21.9456 MB
   Inference time us:count=50 first=55906 curr=54688 min=53924 max=62677 avg=55686.3 std=1479
2021-07-15 00:34:46.962 5228-5228/org.tensorflow.lite.benchmark W/Activity: Slow Operation: Activity
 org.tensorflow.lite.benchmark/.BenchmarkModelActivity onCreate took 3316ms
```

Figure 6.11: Output result of model performances from logcat.

Table 6.1: Performance measurement for model in Android.

Performance Attributes	**Performance result (us)**	**Description**
Warmup	62905	The average time to runs at the start.
Initialization	1483	The average time to load the model and build up the interpreter objects.
Inference	55686	The average inference time while model detecting objects.

Chapter 7: System Testing

7.1 Banknotes Recognition Application Unit Testing

The test case conducted with the use case on mobile application. This testing showed the mobile application intended to do and ensure defects if any before deploy to use. This chapter shows the test result for each test case.

7.1.1 First-time Guideline Test Case

Testing on the guideline function of the application while the users using the application for the first time and vice versa.

Table 7.1: First-time guideline test case.

Test Case ID	Feature Name	Summary	Precondition	Execution Steps	Actual Result
CASE_001	Open application	Verify that application install successfully.	-	1. Click on the application icons 2. Application start up	Pass
CASE_002	Enter guideline screen	Verify the guideline screen appear.	1.Executed CASE_001 2.User open application for the first time.	1. View the guideline screen appear.	Pass
CASE_003	Not enter guideline screen	Verify the guideline screen not appear.	1.Executed CASE_001 2.User open application not for the first time.	1. Navigation to recognition activity.	Pass

CASE_004	Voice Feedback - guideline	Verify the voice feedback guideline triggered.	1.Executed CASE_001 2.User open application for the first time.	1. View the guideline context. 2.Listen to the voice feedback guideline. 3.Ensure voice feedback align with the context.	Pass
CASE_005	Navigate to recognition activities.	Verify the navigation to recognition activity successfully.	1.Executed CASE_001 2.User open application for the first time.	1. Swipe left or right on the screen. 2. Navigate to recognition screen.	Pass

7.1.2 Banknotes Recognition Test Case

Testing on the banknote's recognition activity and the responding screen while users enter the recognition mode and ensure the detection activity are normal.

Table 7.2: Banknote's recognition test case.

Test Case ID	**Feature Name**	**Summary**	**Precondition**	**Execution Steps**	**Actual Result**
CASE_006	Enter banknotes recognition screen	Verify that recognition screen appears successfully.	1.Executed CASE_001 / CASE_005	1. Click on the application icons 2. Application start up	Pass
CASE_007	Recognition Boxes with object name	Verify the boxes with object name	1.Executed CASE_006	1. User move the phone camera to	Pass

	and confident level	and confident level appear		Malaysia banknotes	
CASE_008	Recognition Boxes with object name and confident level	Verify the boxes with object name and confident level not appear	1.Executed CASE_006	1. User move the phone camera to other object except from Malaysia banknotes	Pass

7.1.3 Banknotes Recognition Voice Feedback Test Case

Testing on the voice feedback function, while the feedback is precise responding to each type of the banknotes.

Table 7.3: Banknote's recognition voice test case.

Test Case ID	**Feature Name**	**Summary**	**Precondition**	**Execution Steps**	**Actual Result**
CASE_009	Voice Feedback (RM1)	Verify that voice feedback for RM1 successfully.	1.Executed CASE_007 2.Confident level>=95 percent	1. Move phone camera to Malaysia banknotes RM1 2. Hear voice feedback of RM1	Pass
CASE_010	No Voice Feedback (RM1)	Verify that no voice feedback for RM1 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM1 2. Not hear voice feedback of RM1	Pass

CASE_011	Voice Feedback (RM5)	Verify that voice feedback for RM5 successfully.	1.Executed CASE_007 2.Confident level>=95 percent	1. Move phone camera to Malaysia banknotes RM5 2. Hear voice feedback of RM5	Pass
CASE_012	No Voice Feedback (RM5)	Verify that no voice feedback for RM5 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM5 2. No hear voice feedback of RM5	Pass
CASE_013	Voice Feedback (RM10)	Verify that voice feedback for RM10 successfully.	1.Executed CASE_007 2.Confident level>=95 percent	1. Move phone camera to Malaysia banknotes RM10 2. Hear voice feedback of RM10	Pass
CASE_014	No Voice Feedback (RM10)	Verify that no voice feedback for RM10 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM10 2. No hear voice feedback of RM10	Pass
CASE_015	Voice Feedback (RM20)	Verify that voice feedback for	1.Executed CASE_007	1. Move phone camera to Malaysia	Pass

		RM20 successfully.	2.Confident level>=95 percent	banknotes RM20 2. Hear voice feedback of RM20	
CASE_016	No Voice Feedback (RM20)	Verify that no voice feedback for RM20 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM20 2. No hear voice feedback of RM20	Pass
CASE_017	Voice Feedback (RM50)	Verify that voice feedback for RM50 successfully.	1.Executed CASE_007 2.Confident level>=95 percent	1. Move phone camera to Malaysia banknotes RM50 2. Hear voice feedback of RM50	Pass
CASE_018	No Voice Feedback (RM50)	Verify that no voice feedback for RM50 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM50 2. No hear voice feedback of RM50	Pass
CASE_019	Voice Feedback (RM100)	Verify that voice feedback for RM100 successfully.	1.Executed CASE_007	1. Move phone camera to Malaysia banknotes RM100	Pass

			2.Confident level>=95 percent	2. Hear voice feedback of RM100	
CASE_020	No Voice Feedback (RM100)	Verify that no voice feedback for RM100 successfully.	1.Executed CASE_007 2.Confident level<95 percent	1. Move phone camera to Malaysia banknotes RM100 2. No hear voice feedback of RM100	Pass

7.1.4 Vibration Test Case

Testing on the vibration function works normally while the application start.

Table 7.4: Vibration test case.

Test Case ID	**Feature Name**	**Summary**	**Precondition**	**Execution Steps**	**Actual Result**
CASE_021	Vibration enables	Verify that vibration occurs successfully.	1.Executed CASE_001 / CASE_005	1. Feel vibration feedback	Pass

7.2 User Verification on Banknotes Recognition Application

We invited five testers from different age group to conduct with the application in blindfolded condition. Tester 1 is a student at childhood age group (0-14), tester 2 and 3 are student at youth age group (15-24), tester 4 is an accountant at adulthood age group (25-59) and tester 5 is manager at seniority age group (60 and above). The experiences and details feedback responded to the banknote's recognition application recorded as a preference for future development and research. From the survey, we observed that the application design is satisfying on navigation, simpleness, using experience and recognition speed of the Banknotes recognition application. However, some dissatisfaction was due to the guideline speed so fast that users might miss out on the details, should provide Multilanguage's and enable to recognize while holding the banknotes not limited to horizontal position.

Table 7.5: Application verification feedback and results.

No.	**Application verification survey**	**Rating Results (1-5)/ Feedback**				
		Tester 1	**Tester 2**	**Tester 3**	**Tester 4**	**Tester 5**
1	On a scale of 1 to 5, rate the clear delivery on the guidelines of the mobile application.	3	4	3	4	3
2	On a scale of 1 to 5, rate the navigation experience of the mobile application.	3	3	3	4	3
3	On a scale of 1 to 5, rate the easy to use of	4	3	4	5	4

	the mobile application.					
4	On a scale of 1 to 5, rate your experience using the mobile application.	4	4	5	4	4
5	On a scale of 1 to 5, rate the recognition speed of the mobile application.	4	4	4	5	4
6	On a scale of 1 to 5, rate how often you will use the mobile application.	4	4	5	4	5
7	What do you like most about the mobile application?	Guideline provided	Easy to scan	The application able to recognize with fast and accurate.	Feel safe to having known the value of the money	Feel safe to having known the value of the money
8	what do you like the least about the	The way to scan is restricted	Cannot repeat detect	The guideline speed is too fast.	Cannot repeat detect	The guideline speed is too fast.

	mobile application?					
9	What do you think the mobile application should improve on?	detect not limited to horizontal position	provide multi-language on guide	The guideline speed can be slower.	The repeat detection	provide multi-language on guide

Chapter 8: Conclusion and Future Works

8.1 Project Review

In this project, we successfully build a prototype mobile application that will bring difference for people who suffer from vision impairments. We bring a Malaysia banknotes recognizer mobile application to their daily lives and help them recognize banknotes in more accessible and more accurate ways. People with vision impairments have difficulty recognizing banknotes, and it is even challenging for people who are not born blind. Current banknotes recognition-based application unable to guide blind users in identifying the notes; hence, it lacks effectiveness.

We had studied the problem while visually impaired people recognize Malaysia banknotes in real life. The bezel on the banknotes is not presentable to let the visual impairments feel and recognize the value of the banknotes. We utilize the deep learning technique to differentiate the banknotes.

Most importantly, the extraction method of the convolutional neural network we use will enhance accuracy and be lightweight when identifying the banknotes. As a result, the system reached a mean average accuracy of 97% and an average inference time of 0.06 seconds on detecting the currency. Nevertheless, the features such as voice guidelines, recognition feedback, and vibration indicator successfully deployed and brought more usable on the mobile application for the visual impairment's users. Lastly, we validated the application with users to obtain feedback to improve in future development.

8.2 Problems Encountered

The environment settings for the deep learning model need to be precise and up to date. We faced the error to utilize the graphics processing unit on training the deep learning network due to incompatible of the corresponding version of the CUDA software, cuDNN library, TensorFlow version, and the compatible graphic processing unit. Next, the first training result is not performing well due to the input data. We re-collected quality images that mimic the handling position while performing inference and manual augmentation, such as captured input images with rotation angles with front and back. Eventually, the performance increased and performing well while performing inferences.

Nevertheless, the first attempted of the project used the classification method on the deep learning model. While deployed to the mobile application designed to capture the input video stream from the camera, it faces issues while detecting the actual banknote's position and causing the wrong result feedback of the banknotes values. We solved this by changing the method to the object detection API, which provides the detecting object's bounding boxes before inferences. Hence the results are way much improved with a video stream as the input for the detection model.

8.3 Future Work

Some improvements can be implemented from the user's feedback, such as including Multilanguage, add a dataset to enable recognition in more positions, and the repetition on recognizing a single value. Furthermore, we will consider adding a Malaysia coins reorganization feature into this mobile application and export the file to support offline usage in future planning. Furthermore, enhanced features such as voice recognition on controlling the application can be introduced. Nevertheless, a computational calculator can be implemented to accumulate the values detected and feedback to the users.

REFERENCES

Aljutaili, D. S. *et al.* (2018) 'A Speeded up Robust Scale-Invariant Feature Transform Currency Recognition Algorithm', *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, 12(June).

Alon, A. S. *et al.* (2020) 'EyeBill-PH: A Machine Vision of Assistive Philippine Bill Recognition Device for Visually Impaired', *2020 11th IEEE Control and System Graduate Research Colloquium, ICSGRC 2020 - Proceedings*, (August), pp. 312–317. doi: 10.1109/ICSGRC49013.2020.9232557.

Central Connecticut State University (2016) *CS 410/530 - Software Engineering class notes*. Available at: https://cs.ccsu.edu/~stan/classes/CS410/Notes16/02-SoftwareProcesses.html [Accessed: 19 August 2021].

Doudera, M. (2020) 'Cash Reader: Bill Identifier'. Available at: [Mobile app]. [Accessed 15 August 2020].

Fentahun Zeggeye, J. and Assabie, Y. (2016) 'Automatic Recognition and Counterfeit Detection of Ethiopian Paper Currency', *International Journal of Image, Graphics and Signal Processing*, 8(2), pp. 28–36. doi: 10.5815/ijigsp.2016.02.04.

Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang Li, Frederick Liu, Jaeyoun Kim, J. L. (2020) *TensorFlow Model Garden*. Available at: https://github.com/tensorflow/models [Accessed: 4 August 2021].

IBM Cloud Education (2020) *Deep Learning, IBM Cloud Education*. Available at: https://www.ibm.com/cloud/learn/deep-learning [Accessed: 19 August 2021].

IDEAL Group, I. (2014) 'IDEAL U.S. Currency Identifier'. Available at: [Mobile app]. [Accessed 15 August 2020].

J.Thomas, E. (2019) *From: Visually Impaired: Now What?* Available at: https://visionaware.org/blog/visually-impaired-now-what/ada-at-29-websites-and-apps-still-not-fully-accessible/ [Accessed: 1 August 2021].

Jasmin Sufri, N. A. *et al.* (2017) 'Image based ringgit banknote recognition for visually impaired', *Journal of Telecommunication, Electronic and Computer Engineering*, 9(3–9), pp. 103–111.

Linkon, A. H. M. *et al.* (2020) 'Deep learning approach combining lightweight CNN architecture with transfer learning: An automatic approach for the detection and recognition of bangladeshi banknotes', *Proceedings of 2020 11th International Conference on Electrical and Computer Engineering, ICECE 2020*, pp. 214–217. doi: 10.1109/ICECE51571.2020.9393113.

Mahesh Bahrani, S. (2020) 'Deep Learning Approach for Indian Currency Classification', *International Journal of Engineering Applied Sciences and Technology*, 5(6), pp. 335–340. doi: 10.33564/ijeast.2020.v05i06.049.

Malaysia Bank Negara (2020) *CURRENT BANKNOTE SERIES.* Available at: https://www.bnm.gov.my/currency [Accessed: 21 August 2020].

Maurer, M. (2007) *Is the Failure to Produce Tactile Currency Really a Matter of Discrimination?* Available at: https://nfb.org//images/nfb/publications/bm/bm07/bm0702/bm070202.htm [Accessed: 1 August 2021].

MCT Data (2018) 'MCT Money Reader'. Available at: [Mobile app]. [Accessed 15 August 2020].

Mittal, Shubham and Mittal, Shiva (2018) 'Indian Banknote Recognition using Convolutional Neural Network', *Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018*, pp. 1–6. doi: 10.1109/IoT-SIU.2018.8519888.

Myanmar Assistive Technology Team (2020) 'Myanmar Money Reader'. Available at: [Mobile app]. [Accessed 15 August 2020].

Palm, N. (2012) *Born Blind Vs. Becoming Blind.* Available at: http://playingtheblindcard.blogspot.com/2012/01/born-blind-vs-becoming-blind.html [Accessed: 1 August 2021].

Rimal, K. (2019) 'Cash Recognition for Visually Impaired'. Available at: [Mobile app]. [Accessed 15 August 2020].

Sandler, M. *et al.* (2018) 'MobileNetV2: Inverted Residuals and Linear Bottlenecks', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

Semary, N. A. *et al.* (2016) 'Currency recognition system for visually impaired: Egyptian banknote as a study case', *2015 5th International Conference on Information and Communication Technology and Accessibility, ICTA 2015*, (December). doi: 10.1109/ICTA.2015.7426896.

Sufri, N. A. J. *et al.* (2019) 'Vision Based System for Banknote Recognition Using Different Machine Learning and Deep Learning Approach', *ICSGRC 2019 - 2019 IEEE 10th Control and System Graduate Research Colloquium, Proceeding*, (August), pp. 5–8. doi: 10.1109/ICSGRC.2019.8837068.

The TensorFlow Authors (2021) 'TensorFlow Lite Object Detection Android Demo'. Git code. Available at: https://github.com/tensorflow/examples/tree/master/lite/examples/object_detection/android [Accessed: 4 August 2021].

Tom (2019) *How Blind People Identify Paper Money.* Available at: https://blindcoincollector.com/2019/02/18/how-blind-people-identify-paper-money/#:~:text=Identification solutions by governments and central banks&text=Along with this%2C blind people,is used based on length. [Accessed: 21 August 2020].

Tzutalin (2015) 'LabelImg'. Git code. Available at: https://github.com/tzutalin/labelImg [Accessed: 4 August 2021].

World Health Organization (2019) *Blindness and vision impairment*. Available at: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment [Accessed: 21 August 2020].

APPENDICES

Appendix A-1: Deep learning config pipeline.

```
model {
  ssd {
    num_classes: 6
    image_resizer {
      fixed_shape_resizer {
        height: 320
        width: 320
      }
    }
    feature_extractor {
      type: "ssd_mobilenet_v2_fpn_keras"
      depth_multiplier: 1.0
      min_depth: 16
      conv_hyperparams {
        regularizer {
          l2_regularizer {
            weight: 4e-05
          }
        }
        initializer {
          random_normal_initializer {
            mean: 0.0
            stddev: 0.01
          }
        }
        activation: RELU_6
        batch_norm {
          decay: 0.997
          scale: true
          epsilon: 0.001
        }
      }
      use_depthwise: true
      override_base_feature_extractor_hyperparams: true
      fpn {
        min_level: 3
        max_level: 7
        additional_layer_depth: 128
      }
    }
    box_coder {
      faster_rcnn_box_coder {
        y_scale: 10.0
        x_scale: 10.0
        height_scale: 5.0
        width_scale: 5.0
      }
    }
    matcher {
      argmax_matcher {
        matched_threshold: 0.5
        unmatched_threshold: 0.5
        ignore_thresholds: false
        negatives_lower_than_unmatched: true
        force_match_for_each_row: true
        use_matmul_gather: true
      }
    }
    similarity_calculator {
      iou_similarity {
      }
    }
```

```
box_predictor {
  weight_shared_convolutional_box_predictor {
    conv_hyperparams {
      regularizer {
        l2_regularizer {
          weight: 4e-05
        }
      }
      initializer {
        random_normal_initializer {
          mean: 0.0
          stddev: 0.01
        }
      }
      activation: RELU_6
      batch_norm {
        decay: 0.997
        scale: true
        epsilon: 0.001
      }
    }
    depth: 128
    num_layers_before_predictor: 4
    kernel_size: 3
    class_prediction_bias_init: -4.6
    share_prediction_tower: true
    use_depthwise: true
  }
}
anchor_generator {
  multiscale_anchor_generator {
    min_level: 3
    max_level: 7
    anchor_scale: 4.0
    aspect_ratios: 1.0
    aspect_ratios: 2.0
    aspect_ratios: 0.5
    scales_per_octave: 2
  }
}
post_processing {
  batch_non_max_suppression {
    score_threshold: 1e-08
    iou_threshold: 0.6
    max_detections_per_class: 100
    max_total_detections: 100
    use_static_shapes: false
  }
  score_converter: SIGMOID
}
normalize_loss_by_num_matches: true
loss {
  localization_loss {
    weighted_smooth_l1 {
    }
  }
  classification_loss {
    weighted_sigmoid_focal {
      gamma: 2.0
      alpha: 0.25
    }
  }
```

```
      classification_weight: 1.0
      localization_weight: 1.0
    }
    encode_background_as_zeros: true
    normalize_loc_loss_by_codesize: true
    inplace_batchnorm_update: true
    freeze_batchnorm: false
  }
}
train_config {
  batch_size: 10
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    random_crop_image {
      min_object_covered: 0.0
      min_aspect_ratio: 0.75
      max_aspect_ratio: 3.0
      min_area: 0.75
      max_area: 1.0
      overlap_thresh: 0.0
    }
  }
  sync_replicas: true
  optimizer {
    momentum_optimizer {
      learning_rate {
        cosine_decay_learning_rate {
          learning_rate_base: 0.08
          total_steps: 50000
          warmup_learning_rate: 0.026666
          warmup_steps: 1000
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
  fine_tune_checkpoint: "Tensorflow/workspace/pre-trained-models/ssd_mobile
  num_steps: 50000
  startup_delay_steps: 0.0
  replicas_to_aggregate: 8
  max_number_of_boxes: 100
  unpad_groundtruth_tensors: false
  fine_tune_checkpoint_type: "detection"
  fine_tune_checkpoint_version: V2
}
train_input_reader {
  label_map_path: "Tensorflow/workspace/annotations/label_map.pbtxt"
  tf_record_input_reader {
    input_path: "Tensorflow/workspace/annotations/train.record"
  }
}
eval_config {
  metrics_set: "coco_detection_metrics"
  use_moving_averages: false
}
```

```
eval_input_reader {
  label_map_path: "Tensorflow/workspace/annotations/label_map.pbtxt"
  shuffle: false
  num_epochs: 1
  tf_record_input_reader {
    input_path: "Tensorflow/workspace/annotations/test.record"
  }
}
```

Appendix B-1: Banknote's recognition threshold function.

```
boolean one = false;
boolean five = false;
boolean ten = false;
boolean twenty = false;
boolean fifty = false;
boolean hundred = false;

private void processResults(final List<Recognition> results) {
  final List<Pair<Float, Recognition>> rectsToTrack = new LinkedList<~>();

  screenRects.clear();


  if(results != null && results.size() >= 1){
    Recognition recognition = results.get(0);

    if (recognition != null){
      String title = recognition.getTitle();
      Float confi = 100 * recognition.getConfidence();

      try{
        if(!one && title.equalsIgnoreCase( anotherString: "one\r")&& confi>95){
          mp_1.start();
          one = true;
          five = false;
          ten = false;
          twenty = false;
          fifty = false;
          hundred = false;
        }else if(!five && title.equalsIgnoreCase( anotherString: "five\r")&& confi>95){
          mp_5.start();
          one = false;
          five = true;
          ten = false;
          twenty = false;
          fifty = false;
          hundred = false;
```

```
            hundred = false;
        }else if(!ten && title.equalsIgnoreCase( anotherString: "ten\r")&& confi>95){
            mp_10.start();
            one = false;
            five = false;
            ten = true;
            twenty = false;
            fifty = false;
            hundred = false;
        }else if(!twenty && title.equalsIgnoreCase( anotherString: "twenty\r")&& confi>95){
            mp_20.start();
            one = false;
            five = false;
            ten = false;
            twenty = true;
            fifty = false;
            hundred = false;
        }else if(!fifty && title.equalsIgnoreCase( anotherString: "fifty\r")&& confi>95){
            mp_50.start();
            one = false;
            five = false;
            ten = false;
            twenty = false;
            fifty = true;
            hundred = false;
        }else if(!hundred && title.equalsIgnoreCase( anotherString: "hundred")&& confi>95){
            mp_100.start();
            one = false;
            five = false;
            ten = false;
            twenty = false;
            fifty = false;
            hundred = true;
        }
    } catch (IllegalStateException e) {
        e.printStackTrace();
    }
  }
}
```

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 1**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE
Recollecting the images for the training process, mainly adapt the images where holing the currency which applicable to the position while holding the currency for detection. Re-labeling for each image into xml files using the labelimg software.
2. WORK TO BE DONE
Train the model using the ready images and xml files, fine-tune and export model files.
3. PROBLEMS ENCOUNTERED
Time-consuming for ready the images.
4. SELF EVALUATION OF THE PROGRESS
Keep the progress in timeline.

`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

Environment set up for the training process, including the file path and pre-install software needed, such as protobuff and Cuda toolkit, to utilize the laptop GPU power used on Tensorflow.

The pre-trained model was trained and exported using TensorFlow API and Jupiter notebook.

2. WORK TO BE DONE

Convert the model file to the Tensorflow lite file, which applicable on android.

3. PROBLEMS ENCOUNTERED

The environment setup is complicated due to the unstable of the installation. To meet all the criteria, such as the version using and the Tensorflow version must be aligned. Hence time-consuming to set up the environment. Nevertheless, after setup, the Time-consuming for the training process is enormous.

4. SELF EVALUATION OF THE PROGRESS

The model exported can still be improved but needs to be tested on android to stabilize the deployment on the android environment. Hence will focus on developing one android application.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

Tf lite file generated, successfully deployed on android application. Designing the feedback for users while using the application.

2. WORK TO BE DONE

Build out function such like voice feedback, and vibration for the application.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Progress all still on timeline, keep up.

`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

Mobile application function deployed:

- Voice feedback
- Vibration
- Feedback voice once config level over 95 percent.

2. WORK TO BE DONE

a. Build out the guideline for the user
- Vibration enabled
- Voice guideline enabled

b. Test case for the application

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Progress all still on timeline, keep up.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 5**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE Mobile application function deployed: - Guideline for first time user - Vibration enabled - Voice guideline enabled
2. WORK TO BE DONE - Performance evaluation - Test case design for the application - Perform testing on the test case.
3. PROBLEMS ENCOUNTERED -
4. SELF EVALUATION OF THE PROGRESS Progress all still on timeline, keep up.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

- Performance evaluation done on the model through Tensor Board
- Android profiler test evaluation on the application
- Test case deployed and result recorded.
-

2. WORK TO BE DONE

Documentation refinement:

- System design on the application

Documentation:

- System implementation

3. PROBLEMS ENCOUNTERED

- Test case, for guideline not pass. (Further debugging on the error occur)

4. SELF EVALUATION OF THE PROGRESS

Progress all still on timeline, keep up.

`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

- System design refined
- Document on system implementation and supporting material
- System guidelines refined.

2. WORK TO BE DONE

- Retest with the tflite performance tester
- Ready to prepared the draft report

3. PROBLEMS ENCOUNTERED

- The test result on the android profiler not visible (discover tflite performance tester)

4. SELF EVALUATION OF THE PROGRESS

- Progress all still on timeline.
- Ready for full documentation

\`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE
- Test result done with tflite performance tester - Draft report prepared for evaluation from supervisor
2. WORK TO BE DONE
- Pending feedback from supervisor - Ready for refinement from the advices.
3. PROBLEMS ENCOUNTERED
- None
4. SELF EVALUATION OF THE PROGRESS
- Progress all still on timeline. - Ready for full documentation

`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

- Feedback received from supervisor
- Refinement according the feedback

2. WORK TO BE DONE

- Refinement on documentation
- Poster re-design

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- Progress 80% ready for final touch up

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE
- Refinement on documentation - Poster redesign
2. WORK TO BE DONE
- Refinement on abstract - Tidy up source code - Ready presentation slide
3. PROBLEMS ENCOUNTERED
- None
4. SELF EVALUATION OF THE PROGRESS
- Progress 95% completed

`

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 11**
Student Name & ID: LEE KYAI LUN 17ACB01383	
Supervisor: Dr. Manoranjitham a/p Muniandy	
Project Title: MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT	

1. WORK DONE

- Increase tester for the validation section
- Refine documentation

2. WORK TO BE DONE

- Sign up documentation
- Ready for submission

3. PROBLEMS ENCOUNTERED

- Tester hard to conduct due to pandemic issues.

4. SELF EVALUATION OF THE PROGRESS

- Progress 100% completed

`

Supervisor's signature

Student's signature

POSTER


CURRENCY RECOGNIZER USING
DEEP LEARNING MODEL
For android
An application to help visual impairment recognize Malaysia Currency with ease
Introduction:
The inability to recognize banknotes during any purchase is one of the biggest challenges first by the VISUALLY IMPAIRED. This application aims to help visual impairment to recognize banknotes by just point the camera to the banknotes.
Methods used:
Deep Learning model
Build a deep learning model through transfer learning to classified Malaysia currency
Android Deployment
Deploy model to android
Model Convert
Convert classification model to tf lite model
Visual Impairment Friendly
Application designed to let Visual Impairments use with ease
Result:
Visually Impaired NOW able to recognize every note
Recognize RM1, RM5, RM20, RM10, RM50 and RM100
Real-Time Capturing
Real-Time capturing with live stream
High Accuracy
Recognition system reached mean average accuracy of 97%
Fast Responds
System average inference time of 0.06 seconds
Voice Feedback
Voice feedback to the user once detect the value of MYR
Voice Navigation
Voice navigation to the first-time users
Proposed by:
Lee Kyai Lun,
Faculty of Information Technology,
Bachelor of Computer Science (Honours)
Project Supervisor:
Dr. Manoranjitham a/p Muniandy

PLAGIARISM CHECK RESULT

Document Viewer

Turnitin Originality Report

Processed on: 27-Aug-2021 10:58 +08
ID: 1636594985
Word Count: 10439
Submitted: 1

1701383_FYP2 By LEE KYAI LUN

Similarity Index	Similarity by Source	
5%	Internet Sources:	3%
	Publications:	1%
	Student Papers:	2%

include quoted   include bibliography   excluding matches < 8 words   mode: quickview (classic) report   Change mode   print   download

1% match (Internet from 23-Apr-2021)
https://play.google.com/store/apps/details?gl=US&hl=en&id=np.com.intelaid.cash

<1% match (Internet from 14-Dec-2020)
https://play.google.com/store/apps/details?hl=en&id=org.ideal.currencyid&showAllReviews=

<1% match (Internet from 14-Sep-2020)
http://eprints.utar.edu.my

<1% match (student papers from 17-Jun-2021)
Submitted to University of Wales Institute, Cardiff on 2021-06-17

<1% match (Internet from 10-Dec-2020)
https://towardsdatascience.com/making-the-printed-links-clickable-using-tensorflow-2-object-detection-api-be42bd65488a?gi=70b7eb966849

<1% match (student papers from 04-May-2021)
Submitted to University of Hertfordshire on 2021-05-04

<1% match (student papers from 16-Aug-2021)
Submitted to Sunway Education Group on 2021-08-16

<1% match (student papers from 01-Jul-2020)
Submitted to University of Mines and Technology on 2020-07-01

<1% match (publications)
N. A. J. Sufri, N. A. Rahmad, N. F. Ghazali, N. Shahar, 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), 2019

<1% match (student papers from 19-Oct-2018)
Submitted to University of New South Wales on 2018-10-19

<1% match (publications)
Jegnaw Fentahun Zeggeye, Yaregal Assabie. "Automatic Recognition and Counterfeit Detection of Ethiopian Paper Currency", International Journal of Image, Graphics and Signal Processing, 2016

<1% match (publications)
Redna A. Almutlaq, Daliyah S. Aljutaili, Suha A. Alharbi, Dina M. Ibrahim. "The impact of using SR-SIFT algorithm on various banknotes", International Journal of Data Science, 2020

<1% match (publications)
Shashank Reddy Danda, Bo Chen. "Toward Mitigating Spreading of Coronavirus via Mobile Devices", IEEE Internet of Things Magazine, 2020

<1% match (Internet from 03-Mar-2019)
https://www.hbc.com.pk/laptops/gaming-laptops/dell-inspiron-15-g3-3579-gaming-laptop

<1% match (Internet from 08-Jun-2021)
https://ichi.pro/tr/tensorflow-nesne-algilama-kilavuzu-tensorflow-2-252181752953859

<1% match (Internet from 08-Apr-2013)
http://www.slideshare.net

<1% match (Internet from 03-Nov-2020)
https://www.tandfonline.com/doi/full/10.1016/j.ejbas.2017.02.004

<1% match (Internet from 01-Apr-2019)
http://trap.ncirl.ie

<1% match (publications)
D. Kavitha, C. H. Renumadhavi. "An efficient multilayer deep detection perceptron (MLDDP) methodology for detecting testicular anomalies with or without congenital heart disease (TACHD)", The Journal of Supercomputing, 2021

<1% match (publications)
Wai Kit Wong, Chi Jie Tan, Thu Soe Min, Eng Kiong Wong. "Fuzzy Logic Based Perceptual Image Hashing Algorithm in Malaysian Banknotes Detection System for the Visually Impaired", Artificial Intelligence Advances, 2021

<1% match ()
Liyana Syazana, Abdul Wahab. "Prototype Of Mobile Learning Application For Physics Entitled Motion", UTeM, 2008

<1% match (Internet from 11-Aug-2020)
https://publications.waset.org/abstracts/search?page=205&q=agribusiness+development

<1% match (Internet from 16-Sep-2020)
http://www.nextplayshop.com

<1% match (publications)
Xinyi Liu, Baoying Liu, Guoqing Liu, Feng Chen, Tianzhang Xing. "MobileAid: A Fast and Effective Cognitive Aid System on Mobile Devices", IEEE Access, 2020

<1% match (publications)
Yan Chai Hum, Hou Ren Tan, Yee Kai Tee, Wun She Yap, Tian Swee Tan, Maheza Irna Mohd Salim, Khin Wee Lai. "The development of skin lesion detection application in smart handheld devices using deep neural networks", Multimedia Tools and Applications, 2021

<1% match (Internet from 28-Aug-2011)
http://www.coursehero.com

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE KYAI LUN
ID Number(s)	17ACB01383
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	MALAYSIA CURRENCY RECOGNIZER MOBILE APPLICATION FOR VISUAL IMPAIRMENT

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 5 **%** **Similarity by source** Internet Sources: 3 % Publications: 1 % Student Papers: 2 %	
Number of individual sources listed of more than 3% similarity: NA	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

____
Signature of Supervisor

Name: Manoranjitham Muniandy

Date: 27th August 2021

____
Signature of Co-Supervisor

Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB01383
Student Name	Lee Kyai Lun
Supervisor Name	Dr Manoranjitham a/p Muniandy

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.
N/A	Front Plastic Cover (for hardcopy)
√	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)
N/A	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)

*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:23th august 2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ________________________ (Signature of Supervisor) Date: 27th august 2021