INTELLIGENT MALAYSIAN BANKNOTE RECOGNITION SYSTEM FOR BLIND PEOPLE

TAN GUAN SENG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

INTELLIGENT MALAYSIAN BANKNOTE RECOGNITION SYSTEM FOR BLIND PEOPLE

TAN GUAN SENG

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2020

DECLARATION

I declare that this report entitled “Intelligent Malaysia Banknote Recognition System for Blind People” is the result of my own work except for quotes as cited in the references.

Signature : …………………………………

Author : ………TAN GUAN SENG…

Date : ………24 JUNE 2020……………

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature : ………………………………………

Supervisor Name : Dr Khairuddin Bin Osman

Date : …………26 JUNE 2010…………

DEDICATION

For my parents, my supervisor, Dr Khairuddin Bin Osman and all my friends.

ABSTRACT

Blind people faced a lot of difficulties in order to interact with the environment because most of the information encoded is visual. One of the serious problems they faced is identifying and recognizing the different currency amount of banknote. This paper presents the development of a system with the aim to classify the Malaysian banknotes for blind people so that allows blind people to handle the banknotes in their daily life independently. The objectives of this project are to study Tensorflow deep learning platform for Malaysian Banknote classification, to develop a Malaysian Banknote classification mobile application using image classification with Tensorflow deep learning platform as well as to analysis and identify the classification performance (accuracy and inferencing time) of developed application against different brightness (dark and bright). The development of this system involves the use of Tensorflow to train a newly classification model for Malaysian Banknote and android studio software for application. For the hardware part, a smartphone as an application will be used for blind people in classifying Malaysian banknote. A result of 80% and 83.33% accuracies with an average time of 486.10ms and 488.69ms for dark area and bright area respectively is achieved. In the end of the project, an android

application which can be used for banknote classification is successfully developed with high accuracy and less inference time.

ABSTRAK

Orang buta menghadapi banyak masalah untuk berinteraksi dengan alam sekitar kerana kebanyakan maklumat yang dikodkan adalah visual. Salah satu masalah serius yang mereka hadapi ialah mengenal pasti dan mengenali jumlah mata wang yang berbeza dari wang kertas. Makalah ini membentangkan perkembangan sistem dengan matlamat untuk mengkelaskan wang kertas Malaysia untuk orang buta supaya membolehkan orang buta mengendalikan wang kertas dalam kehidupan seharian mereka secara bebas. Objektif projek ini adalah untuk mempelajari model pengelas Tensorflow yang boleh digunakan dalam mengklasifikasikan wang kertas Malaysia, untuk membangunkan aplikasi telefon yang boleh mengklasifikasikan kelas-kelas wang kertas Malaysia yang berbeza untuk orang buta serta analisis dan mengenal pasti prestasi aplikasi(ketepatan dan masa inferens) yang dihasilkan terhadap kecerahan yang berbeza(cerah dan gelap). Perkembangan sistem ini melibatkan penggunaan Tensorflow untuk melatih model klasifikasi baru untuk wang kertas Malaysia dan perisian studio android untuk permohonan. Untuk bahagian perkakasan, telefon pintar sebagai aplikasi akan digunakan untuk orang buta dalam mengklasifikasikan wang kertas Malaysia. Hasil daripada ketepatan 80% dan 83.33% dengan purata masa 486.10ms dan 488.69ms untuk kawasan gelap dan kawasan *terang masing-masing dicapai. Pada akhir projek, aplikasi android yang boleh digunakan untuk klasifikasi wang kertas berjaya dibangunkan dengan ketepatan yang tinggi dalam masa yang singkat.*

ACKNOWLEDGEMENTS

It is a matter of great pleasure and benefit for me to introduce this report of Project Sarjana Muda (PSM). I would like to thank to those individuals for their consistent support and advice on completing my project.

As a matter of first importance, I would like to thank to Dr. Khairuddin Bin Osman who is my supervisor for giving me the chance to start this project under his guidance. The door to Dr. Khairuddin's office was always open whenever we ran into a trouble spot or faced the problems during the completion of this project. Besides, thanks to Profesor Madya Dr Kok Swee Leong and Ir Dr Ridza Azri Bin Ramlee, the panels for PSM 1 and PSM 2. Their suggestions and comments help me have a clearer idea and better understanding on the problems and weakness that need to overcome to complete my project. The most importantly, I would like to express my profound gratitude to my parents and friends for presenting us with unfailing support and non-stop encouragement through the process of researching and writing on this paper. The successful completion of this project would now not have been possible without their help. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

SURF	:	Speed Up Robust Features
JPG	:	Joint Photographic Experts Group
SIFT	:	Scale-invariant Feature Transform
FREAK	:	Fast Retina Keypoint
BRISK	:	Binary Robust Invariant Scalable Keypoints
OpenCV	:	Open Source Computer Vision
GPU	:	Graphic Processing Unit
AI	:	Artificial Intelligent
iOS	:	iPhone Operating System
SDK	:	Software Development Kit
DTC	:	Decision Tree Classifier
ROI	:	Region of interest
RGB	:	Red, Green, and Blue
RAM	:	Random Access Memory
CPU	:	Central Processing Unit
API	:	Application Programming Interface
WIFI	:	Wireless fidelity
UI	:	User Interface

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Background

Sight is a thing that one sees or that can be seen and an significant skill for the existence of human beings. Vision is not only a skill that helps us in task management, but also brings an effect to human behaviors. Furthermore, blindness can be defined as the inability to see anything, including light. If you're partially blind, you have limited vision. For instance, you may have blurry vision or the inability to distinguish the shapes of objects. Complete blindness means you can't see at all. Legal blindness refers to vision that's highly compromised. There are several eye diseases and conditions can cause blindness, Glaucoma is a disease which can damage human optic nerve and affect the transmission of visual information from our eyes to brain. Macular degeneration usually affects older adults will destroys the part of human eye that enables to see details. Other than that, a lazy eye also affects human eyes to see details and it may cause to vision loss. Blindness affects several factors including the psychological behavior of people. For example, a blind people will more likely to feel stress and depress compare to a person with normal vision. There are also more likely to suffer from worry, and short of social relationships.

Blind people experienced problems in their daily activities. The lack of accessibility for the visually impaired is central to a number of the issues the blind or low visual individuals face. Leisure is another one on the list. There is a limited number of inclusive/accessible activities for the visually impaired, which are as simple as a museum visit. Moreover, accessible books are not abundant either. According to the World Blind Union, “more than 90% of all published material is not accessible to the blind or partially sighted.” Furthermore, they will not be able to recognize banknote without a training program on how to recognize different currency amount of banknote. This will definitely affect their daily life which allows them to handle the banknotes independently in their daily life.

There are currently various currency recognizing devices available internationally such as Euro Banknote Recognition System for Blind People which used of modified Viola and Jones algorithms and Speed Up Robust Features (SURF) technique for banknote detection. Wearable device for Malaysian Banknotes recognition based on embedded decision tree classifier. However, these devices are not really suitable in classifying Malaysian banknote as it is still inconvenient to bring it and the processing time is also slow. The purpose of this project is to develop a banknote recognition mobile application using deep learning technique to identify different amount of Malaysian currency including RM1, RM5, RM10, RM20, RM50, and RM100.

Besides that, Artificial intelligence (AI) is possibly for machines to learn from experience, adjust to new inputs and perform human-like tasks. Most AI examples that you hear about today from chess-playing computers to self-driving cars are mostly rely heavily on deep learning and natural language processing. Using these technologies, computers can be trained to accomplish specific tasks by processing large amounts of data and recognizing patterns in the data. The capability of learning unsupervised data which is unstructured or unlabeled is also one of the abilities can be found on deep learning networks. Recently, there is a new machine learning framework – TensorFlow which is an open source library created by Google Brain Team to ease the process of obtaining data, training models, serving predictions, and refining future results.

1.2 Objectives

The objectives of this project are to evaluate the students on:

i. To study Tensorflow deep learning platform for Malaysian Banknote classification.

ii. To develop a Malaysian Banknote classification mobile application using image classification with Tensorflow deep learning platform.

iii. To analysis and identify the classification performance (accuracy and inferencing time) of developed application against different brightness (dark and bright)

1.3 Problem statement

In 2018, a total of 15,000 subjects were examined with a response rate at 95.3%. The population for prevalence of blindness was 1.2%, severe visual impairment was 1.0%, moderate visual impairment was 5.9%. Also, untreated cataract which has the population of 58.6%, diabetic retinopathy 10.4%, and glaucoma 6.6% were the commonest causes of blindness. A total of 75.6% of diseases are present in this population which will cause blindness. The results show there is an increase patient in Malaysia 2018[1]. Nowadays in Malaysia, the way to help blind people to differentiate banknote is touching on the small embossed lines and dots on the banknote in order to distinguish different banknote by their sense of touch. Although Braille markings have been incorporated in Malaysian Ringgit but the limitation is that some of them might be unable to understand the Braille Feature on the banknote. Although Braille plays an important role in helping visual impairment to access information and education independently, but the challenges and problem to learn Braille like emotional downsides of individuals, lack of resources, and lack of perceived purpose are still existing for blind people[2]. Besides that, nowadays in Malaysia, the way to help people to differentiate banknote also including the Ultraviolet Light counterfeit banknote detectors to detect the counterfeit banknote . However, the limitation is that this detector is expensive and not affordable causing it's usage is lower.

A variety of treatment is available like corneal transplantation and osteo-odonto-keratoprosthesis. However, this kind of surgical has the limitation which is very pricey which they unable to effort. Currently, for blind community to be able to identify and handle the banknotes, many blind users distribute them in advance, by value, into different pockets, this allows them to know the amount they are carrying. However, this classification has the problem that requires qualifying time or a third people to help them. Besides that, a wallet with many dividers also become confusing when blind people will sort through many denominations. There are many devices available in the market but not acceptable to detect Malaysian Ringgit banknote and some of them are very pricey as hospital equipment is increasingly expensive and fraudulent efforts are also increasing. Therefore, a solution is proposed to offer an artificial vision for recognizing the value of currency in Malaysia.

1.4 Scope of Work

This project is to develop a mobile application which enables blind community to classify banknotes through it so that they can handle money by their own with ease. This project will cover only the development of a portable system which helps in classification of Malaysian Banknote (RM1, RM5, RM10, RM20, RM50, RM100) However, it will not cover the currency of other countries and Malaysian coins. It will not cover the classification of counterfeit as well. The software, method, type of images, performance measurement, and data size are shown in Table 1.1.

Table 1.1: Scope of work

Software	• Python 2.7 & Python 3.5 • Android Studio
Method	• Deep Learning • TensorFlow
Type of Images	• JPG image file
Performance Measurement	• Accuracy • Inference time • Lighting situation
Data Size	• 1000 images for each different classes of Malaysian banknote (RM1, RM5, RM10, RM20, RM 50, RM100)

1.5 Brief Description of the Methodology

This project is will be divided into three stages which are planning, implementing and analysis. For planning stage, researches are collected for this project through literature review from journal, book and research paper from library and internet. All the related studies and knowledges about this project will be stated in planning phase. With regard the implementing phase, sample data required will be collected via images taken by phone, online sources like video and images and also by the data augmentation. A TensorFlow model will be created which involve the training dataset of total 6 classes of Malaysian Banknote. After training part, the model is then optimized to make it utilize which can compatible and process in mobile application. There are total 6 different classes of collected data which are RM1, RM5, RM10, RM20, RM50, and RM100. The classifier will learn to classify different amount of Malaysian banknote. After obtaining the results, the accuracies and average inferencing time will be calculated to measure the performance of the system. Lastly, the overall project will be discussed and identified whether it is successful or no achieved with the objectives.

1.6 Thesis Organization

This thesis contains five chapters which are shown as below.

Chapter 1: Introduction

This chapter introduces the background of this project. This chapter also clearly states the objectives and problem statement for this project. Furthermore, scope of work, description of methodology and thesis outline will be listed out in this chapter as well.

Chapter 2: Literature review

The theory of deep learning and TensorFlow are described, summarized and identified in this chapter. Besides, the development of mobile application and the review of past research papers about the development of portable device which can classify different Malaysian Banknote will be introduced in this chapter as well.

Chapter 3: Methodology

This chapter describes the techniques being used to obtain the results and the method to analysis of the results for this project. First, sample images will be collected. After that, the existing model is then retrained to classify new categories which consists of Malaysian Banknote. Lastly, collect the results and analysis through the acquired results.

Chapter 4: Result and Discussion

This chapter presents the result of this project and discussion of the result obtained. The result shown are corresponding to the methodology. Analysis of the result for this project and any future work can be done to enhance and improve the result of this project will be discussed in discussion part.

Chapter 5: Conclusion

The chapter will conclude everything about this project which including all the 5 chapters. The strength and weakness of the project will be discussed in this section. The future work for this project will also be discussed in this chapter.

CHAPTER 2

BACKGROUND STUDY

This chapter will illustrate about the perspectives and techniques used in past research and investigates the scope to which the project is associated and existing studies and theories. Besides that, the theories and concepts that have been used in solving problem stated in this project will be depicted in this chapter. Next, this chapter will also present the objectives and problems of the project in the field of research. Finally, any research hypothesis related to the research methodology will be clearly stated.

2.1 Banknote Detection for blind people

In this modern time, Malaysian banknote has changed over the year and it came to Fourth series until today. The fourth banknotes series of Malaysian banknotes came out with new design in terms of designation as well as the materials. Instead of changing the banknote to polymer materials for RM1 and RM5, Malaysian banknote also retain the existing sizes and colours to assist the public and visually impaired person. In order to make the banknotes more user friendly to the blind community, Malaysian also assimilate Braille features and come out with Malaysian banknotes which have small embossed lines and dots on them for blind people to feel with their fingers and differentiate different amount of currency. Currently, a lot of researchers have been performed in the domain of banknote detection and recognition device. In order to recognize the banknote, various image processing technique and artificial intelligence methods have been employed for handle this challenge. Most recent technique is computer vision like SURF, SIFT, FREAK, BRISK are being used in the development of a devices which assist blind people in their daily life to handle money independently[3], [4]. Standard computer vision techniques and Deep learning approach for automatic metal corrosion (rust) detection is being compared. The test has been performed by classifying images and calculating the total accuracy for two different approaches. The results show that OpenCV based model showed a total accuracy of 69% while deep learning algorithm achieved an accuracy up to 78% in both "rust" and "non-rust" cases. Therefore, Deep Learning algorithm performs better in real case scenario as it is more accurate compare to Standard computer vision (OpenCV)[5].

2.2 Deep learning

Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning has been widely used until today because of three significant reasons: the dramatically increased chip processing abilities (GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms[6], [7]. Besides that, Deep learning is a method that uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation.

Each successive layer uses the output from the previous layer as input. It is then applied some form of gradient descent for training via back propagation. In common words, deep learning is an artificial intelligent function that works like human brain through data processing and creating patterns for decision making. Furthermore, it is also a subset of machine learning in Artificial Intelligent (AI) which capable to learn from unstructured or unlabeled data. Therefore, it is a useful tool which can be used in developing an object recognition system in real-time.

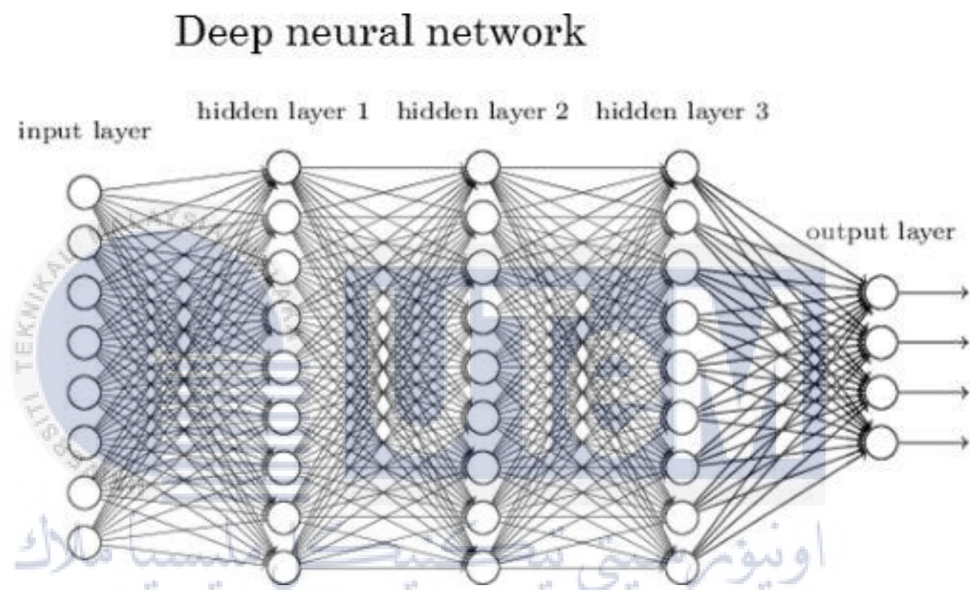

Figure 2.1: Deep learning neural network

2.3 TensorFlow

In machine learning, there are various library like Keras, Theano, and TensorFlow. TensorFlow is now getting more popular and it is open source library which developed by Google under the Apache 2.0 License in November 2015 while Keras was now become a part of TensorFlow as well[8]. TensorFlow can be used and support multiple languages including python. It is a flexible system including training and inference algorithms for deep neural network models as it is capable of detecting objects in images, human speech, analyzing video as well as predicting the properties of potential medicines. The flexibility of TensorFlow ease the engineers to design and deploy sophisticated deep learning architectures. Several Google services use TensorFlow in production and they released it as an open-source project which has been widely used by people for machine learning research[9], [10], [11].

2.4 Development of mobile application

Mobile application development is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Hence, the mobile development process involves creating installable software bundles (code, binaries, assets, etc.), implementing backend services such as data access with an API, and testing the application on target devices. They are also available on personal computers, mainly within web browsers[12]. In development process, user interface design is also an essential element in the building of mobile apps. Mobile UI considers contexts, input, screen, and mobility as outlines for design. The components required both hardware and software for interface. User input allows for users to manipulate a system, and output of the device indicate the effects of the users' manipulation. One of the most common approaches being used is Software Development Kids (SDK) as it was the major development technique which capitalizes on native programming languages such as Java, C*, .NET and their individual SDKs. Mobile apps are fast and relatively easy to develop and low-cost maintenance for public.

2.5 Critical Review

Table 2.1 shows the various research paper on the development of portable device which helps blind people in classifying different type of banknote.

Table 2.1: Critical Review

Reference	Approach	Strength	Weakness
Euro Banknote Recognition System for Blind People (Dunai et al. 2017) [13]	Method - Modified Viola and Jones algorithms - Speed Up Robust Features (SURF) Software - Python 2 - Android Studio Hardware - Raspberry Pi -Android-based Samsung Galaxy -3D printed sunglasses	Cheap electronics –Raspberry Pi, Raspberry Pi NoIR camera, Android based smart phone - Due to infrared lasers of the camera, the system is able to work both in dark environment and daylights. - Synthetic Speech	- No portable for blind people with an extra Raspberry and smartphone. - Extra cost to buy for Raspberry Pi - The mean banknote recognition time is 11 seconds.
Wearable Device for Malaysian Ringgit Banknotes Recognition Based on Embedded Decision Tree Classifier (Ghazali et al. 2018) [14]	Method - Features extraction of RGB values. - Embedded Decision Tree Classifier (DTC) Software - Matlab Hardware - colour sensor TC230 - Lilypad Arduino	- The device is designed in small circle-shaped as a necklace - low cost and low power consumption	- Detection based on the colour feature only while cannot detect through other features like size

	Performance Measurement - 84.7% accuracy is achieved		
Currency Recognition System for Visually Impaired: Egyptian Banknote as a Study Case (Semary et al. 2015) [15]	Method - Image foreground segmentation -Histogram Enhancement - Region of interest (ROI) Software - Matlab R2012a - Android Studio - Opencv library Hardware - Samsung Galaxy GT-N800 tablet Performance Measurement - 89% - Average running time for Matlab system is 10 seconds - Average running time for android system is 12 seconds	- The device is designed in android application with no icons and no manual configuration wanted. -Sound is pronounced by the application.	- The overall required time (12 seconds) to process the system is too long compared to others
Currency Recognition on Mobile Phones (Alexeff 2014)	Method	- High accuracy - Portable device which can run on	- SIFT on Android is patended and supposed to pay them for its use.

[16]	- Scale-Invariant Feature Transform (SIFT) Software - Android Studio - OpenCV library Hardware - An android run on versions 2.3 and above - Android camera with at least 1.3MP. Performance Measurement - 96.7% accuracy	android smartphone - Total recognition pipeline used is 1.09 seconds	
Robust and Effective Component-based Banknote Recognition for the Blind (Hasanuzzaman, Yang, and Tian 2012) [17]	Method -Speeded Up Robust Features (SURF) Software - OpenCV library - Android Studio Hardware - camera - microphone for speech command input - speakers -mini-computer (PDA)	- high accuracy: high true recognition rate and low false recognition rate -robustness: handles different banknotes in various conditions -high efficiency: recognizes banknotes quickly - Portable device which is designed using wearable camera - ease to use	- SURF is patended and required to pay in order to use library on Java - average speed is 2.5 seconds

	Performance Measurement -100% true recognition accuracy		

According to the research paper, "Euro Banknote Recognition System for Blind People" by Larisa Dunai Dunai 1, Mónica Chillarón Pérez, Guillermo Peris-Fajarnés and Ismael Lengua Lengua[13], they propose a Euro banknote detection algorithms based on Haar features proposed by Viola and Jones. This method is based on the classification of the integral image, and not by analyzing each image pixel, which significantly reduces the processing time. The Viola and Jones method detects the edges in order to identify areas of interest by analyzing the two rectangles. It also detects the lines and analyzes the four rectangles as well as detects shapes by analyzing the three rectangles. For the selection of characteristics and classification, a set of 200 images named positives was taken with Raspberry Pi Camera. There are total 452 generic images which is used as negative background images. Afterwards, adding negative background and rotation have modified the positive images creating 2000 training samples. In the end, 2000 positive images and 2000 negative images were tested for banknote detection. For the recognition part, a 97.5% hit rate was obtained with two hundred photographs of the €5, €10, €20, €50, and €100 banknote. However, this kind of system will require both the Raspberry Pi and Pi camera as well as the Android smartphone. Therefore, it is not suitable and inconvenient for a blind people to bring a smartphone along with Raspberry Pi. Furthermore, a Raspberry Pi will require power bank to operate as well and it will cause more trouble for them to know the condition of power bank whether it is out of battery or even broken down.

According to the research paper, “Wearable Device for Malaysian Ringgit Banknotes Recognition Based on Embedded Decision Tree Classifier” by Nurul Fathiah Ghazali, Muhammad Amir As’ari1, Mohd Najeb Jamaludin, Lukman Hakim Ismail, Hadafi Fitri Mohd Latip and Abdul Hafidz Omar[14], they propose a method which used the Colour Sensor with IR filter as the input, Lilypad Arduino as the microcontroller and processing part and Lilypad Buzzer as the cueing module to represent each banknote with different notes in developing this device. There are overall four main processes that needs to be done in producing the Malaysian banknotes device which are data acquisition and collection, data analysis, system development, and system evaluation. For data collection part, Red (R), Green (G), and Blue (B) values for each Malaysian banknote was recorded by colour sensor. Next, for the data analysis, the three features (RB, RG, and GB) were used as the main features in modelling the DTC using MATLAB. For the system development, there are total two things need to be done which is the internal and external part. The internal part consists of the processor and programming part which model the optimum DTC in Arduino Lilypad. Lastly, once the system is developed, 10 subjects were used to test the device and the performance of the device in terms of accuracy is calculated. Overall, the accuracy of 84.7% is achieved and there are classifiers which suffer with sudden performance degradation in RM5 and RM50 and it perhaps due to green as dominant colour component for both banknotes. This is also the weakness of this device as it depends only on colour sensors.

According to research paper, “Currency Recognition System for Visually Impaired: Egyptian Banknote as a Study Case” by Noura A. Semary, Sondos M. Fadl, Magda S. Essa, Ahmed F. Gad[15], a general framework for identifying paper banknotes is proposed. There are 5 main stages in this system which are Image acquisition, Pre-processing, Histogram equalization, ROI extraction, and Template Matching. Firstly, Image acquisition will convert the captured image from RGB image to grayscale version. Then, pre-processing step will remove noises on the image and improve quality before undergo image processing. In order to extract the currency paper from the background, the image is converted to binary version in segmentation process. Binary image is the image that only contains two possible intensity values which usually will be black (0) and white (1). Histogram equalization is used to adjust the contrast based on the image histogram, editing the brightness on the image to make the image clearer. ROI extraction will extract the currency paper and resize it to the dataset height to make comparison. Lastly, template matching preformed to measure the similarity between images in the database and the ROI part. In this paper, simple image processing techniques like thresholding, noise removal, histogram equalization and segmentation are used to extract the ROI and facilitate the template matching procedure. Correlation is used after that to carry out comparison process between the capture image and dataset image. The accuracy of the system reached 89% under the Matlab system. The android system designed is better if compare with[14] which designed in another portable devices and give signal to blind people by using Lilypad Buzzer. However, this system required 12 seconds for android system which is slower than the previous currency recognition system [13].

According to the research paper, “Currency Recognition on Mobile Phones”, by Suriya Singh, Shushman Choudhury, Kumar Vishal, C.V.Jawahar[16], an application which used to recognize banknote by using computer vision techniques is developed. This application runs on the device without the need of server and it is specially designed for blind people to use in their daily life. They used segmentation method which fixed rectangular region of interest (ROI) while everything outside of this ROI can various background. Once the images captured, a GrabCut algorithm is applied to carry out foreground or background segmentation of the images. Then, instance retrieval is being used to classify the currency bill inside the image. SIFT is being used in this stage in order to locate the keypoints in image which obtained from segmentation. The class with highest keypoint matching is then declared as the result in the classification process. Lastly, they adapted the above steps to a mobile environment which required 2.4GB of storage and 1.5GB RAM. It achieved the recognition accuracy of 96.7% while the processing time is 1.09 seconds. Although this system is developed with high accuracy and low processing time but since SIFT and SURF are patented, and we are supposed to pay them for its usage in order to make a commercial application [18], [19]. Therefore, it is no suitable to use in developing recognition devices on Android application.

According to the research paper, "Robust and Effective Component-based Banknote Recognition for the Blind", by Faiz M. Hasanuzzaman, Xiaodong Yang, YingLi Tian[17], a component-based model is developed for banknote recognition algorithm. The proposed component-based model is more robust in handling partial occlusions. Therefore, as long as a part of components are detected, the system still be able to recognize the whole banknote. There are total seven categories involved in this system. By considering front and back sides of each category, so the total ground truth images will be 14 images with the resolution of 835x354 for each banknote image. It is then applied SURF features for extraction and saved for matching with the query image in recognition process. The reasons of SURF are used as detection and representation is because SURF able to overcome the rotation and scaling change. Besides, SURF facilitates fast internet point localization and matching. The local image features are then extracted by SURF and perform matching performance between reference regions and currency image captured under a variety of circumstances. In this research paper, it come out with a spatial clustering of matching features which ease blind users in the aiming problem during the banknote recognition process. In positive image, the points in the same colour tend to group in the same cluster for the positive image whereas the matched points almost randomly distribute on the image for negative image. The accuracy (100%) is achieved as the SURF manage to handle image rotation, scaling change as well as the illumination change. Besides, various category to differentiate a query image from different background. Lastly, effectiveness of clustering approach helps to filter out negative image and improve accuracy of this system. However, since SIFT and SURF are patented and require payment in order to use this library in Java [18], [19]. Therefore, this is not an appropriate approach to use in Android based system.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In order to achieve the objectives of this project, the appropriate methods and suitable techniques are essential. The methods being used for this project are discussed in this section as well. The researches and literature reviews about this project are studied through different mediums such as books, articles, research papers and web resources. Both hardware and software required in the project development are also being considered before the project started. This chapter including the project planning, flow chart, techniques used to collect sample data, project design and the method used for performance analysis.

3.2 Project Development

To ensure that the project is done correctly based on the objective stated, strong understanding of the project need to be considered as the main needs in this study. Strong understanding on the hardware and software used in this project was needed to make sure the correct usage of the hardware and software component. The appropriate selection of the components will give some positive impact on the development of the project as it will help the project to be developed smoothly.

The system can help the blind people to classify the Malaysian banknotes by using Tensorflow deep learning platform for image classification. The process of retrieving data to determine whether the picture snapped by the blind people are correct or not will actually be done in the application itself as trained by TensorFlow software to ensure the safeties and security that need to fulfilled in order to make a safer environment for the blind people user.

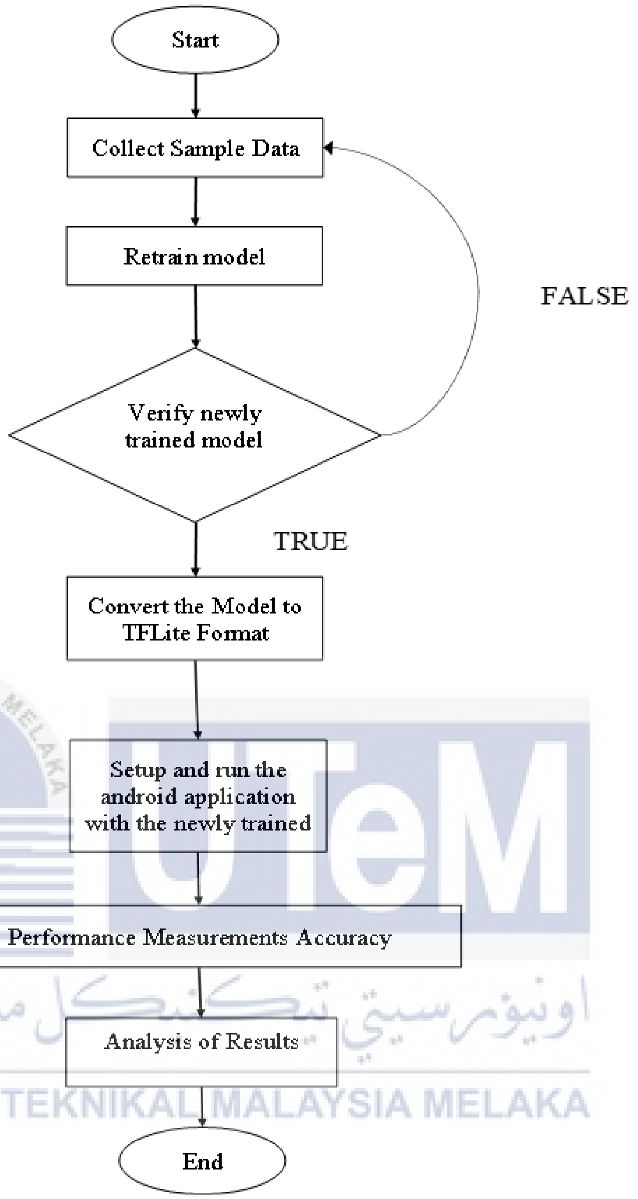

Figure 3.1: Project development flowchart

3.2.1 Split video into frame

Computer Vision (OpenCV) is a powerful tool which can be used to extract every frame from an input video source. This is one of the effective ways in order to collect more image dataset**.** Figure 3.2 shows a simple python code which pass to the location of the video file and split the video with the name “malaysia banknote.mp4” and save all the frames into the same directory. The image saved are in the “jpg” format.

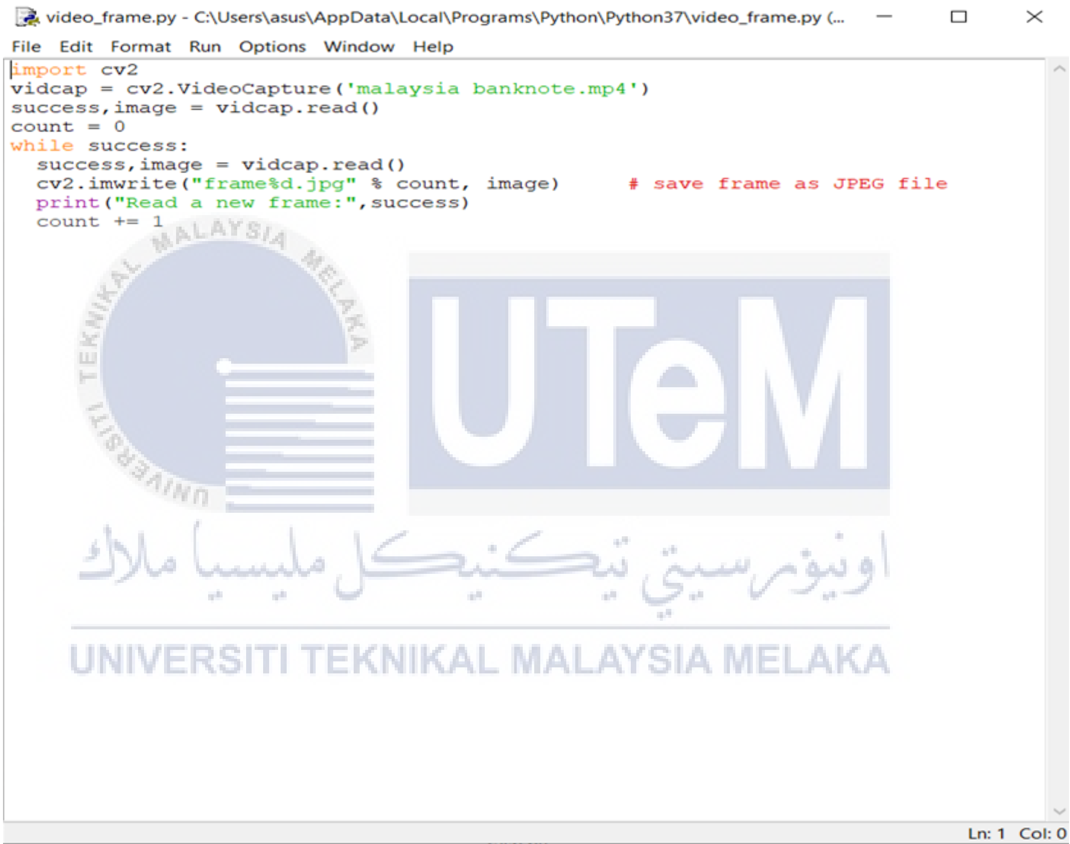

Figure 3.2: Split video to frame

3.2.2 Image capture through mobile phone

Besides, there are some of the images that collected through mobile phone. This is the most important part as it represents the real-life situation image capture through the phone and this approach allows developers to decide the way on how to handle the banknotes before proceeding to the training process. Figure 3.3 shows the image that capture through the android phone with camera specifications of 13-megapixel.

Figure 3.3: Camera images

3.2.3 Data Augmentation

Data augmentation is the process of increasing the amount and diversity of data. We do not collect new data, rather we transform the already present data. By applying the technique above, a large number of datasets like one hundred or more are managed to be collected. However, in order to get good performances by deep learning algorithms, the more data the merrier. Therefore, data augmentation which come out with artificially augmenting our dataset and help us to solve the problem of data shortage. Figure 3.4 shows the simple python code which carried out the data augmentation process to generate 1000 image dataset from our original dataset.

```
import Augmentor
path_to_data = "rm1"
p = Augmentor.Pipeline("rm1")
p.rotate90(probability = 0.8)
p.rotate180(probability = 0.9)
p.rotate270(probability = 0.5)
p.random_brightness(probability = 0.7, min_factor = 0.4, max_factor = 1.0)
p.zoom(probability = 0.2, min_factor = 1.1, max_factor = 1.5)
num_of_samples = int(1e3)
p.sample(num_of_samples)
```

Figure 3.4: Data Augmentation

3.3 Retrain the model

Since TensorFlow is an open source and well developed with mobilenet-v1 model. Therefore, the model which is already trained by google on 1000 classes is used and apply transfer learning which will retrain the final layer of already mobilenet-v1 with new categories that we want to use for classification. Since TensorFlow architecture supports both large-scale training and inference which including GPU and CPU. Therefore, retraining process on CPU will be used for training part. However, it is desirable to use GPU for retraining process to accelerate the training time in the case that include a large amount of dataset like 2000 images for each class. After retraining process, verification of the newly trained model is a must in order to make sure the newly trained model work perfectly.

3.4 Setup and run the application with newly trained model

The retrained model will then be converted to lite model is also necessary to optimize the retrained model and use it on mobile application. This step is required as the size of the model is still big and definitely not suitable for mobiles. Then, develop and run the application with the newly trained model which works faster and better without expensive equipment in rural parts of the world. An android application will then be developed which will use the photo taken by the user for backend predictions and return the results in terms of sound. The android app will be designed and developed as the steps in Figure 3.5.

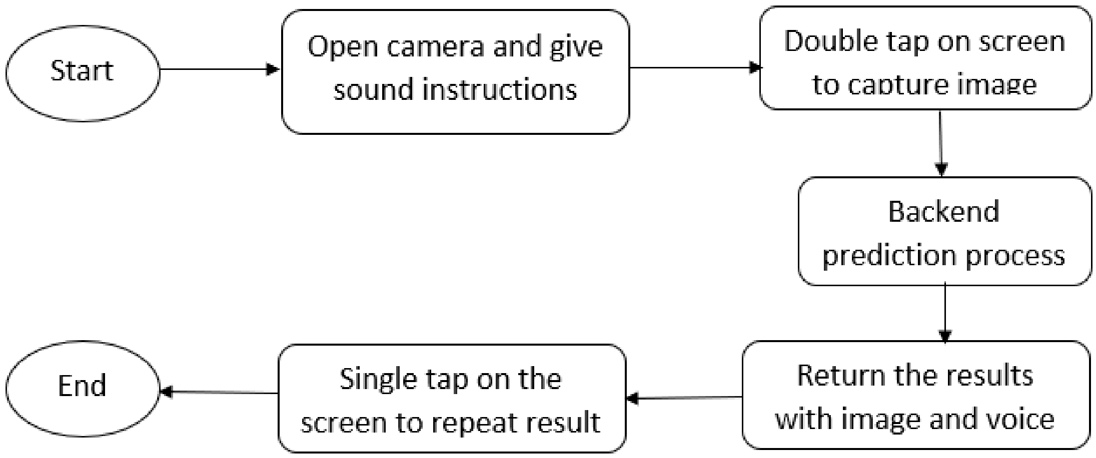

Figure 3.5: Android application design steps

3.5 Performance Analysis

The algorithm will then analyze based on it's accuracy and inference time against different brightness. About 30 samples for each kind of classes will be collected and tested via this system in order to calculate the mean accuracy and average time required for this model run on the mobile application. Performance Analysis will be collected in table form by considering the image prediction accuracy, the inferencing time, and the background of the image like brightness in order to perform analysis measurement. The application is also designed by adding a Java Excel API which enable developers to read, write and modify the Excel spreadsheets dynamically in order to collect results more efficiently. Furthermore, the Android apps are designed with the use of light sensor on the phone in order to collect the reading based on different lux values. The results collected are based on lux which is a concept to measure the intensity of light at a specific point. It signifies the brightness obtained at a point in room by illuminating a light source. Table 3.1 shows the lux values for various tasks [20].

Table 3.1: Lux values for various tasks

Activity	**Illumination (lux, lumen/m^2)**
Public areas with dark surroundings	20-50
Simple orientation for short visits	50-100
Working areas where visual tasks are only occasionally performed	100-150
Warehouses, Homes, Theaters, Archives	150
Easy Office Work, Classes	250
Normal Office Work, PC Work, Study Library	500

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Training results

After collecting all the data, the training process of the final layer of the networks begins. Then, the 4000 training steps are run by the scripts and 10 images are randomly chosen by each of the steps from the training dataset for prediction process . Those predictions are then compared with the actual results, the results are updated to the final layer's weight as comparison so that our computer can learn from it through back propagation process. The further study on the training steps will be carried out to investigate the best training step to train a model. Figure 4.1 shows the process during training with total accuracy, cross entropy as well as the validation accuracy after each training step. The percentage of the images in each class were labeled with the correct class is represented by the training accuracy. Validation accuracy is the precision of correctly labelled image from the images that selected from random classes. Cross entropy is the measure from the field of information theory, building upon entropy and generally calculating the difference between two probability distributions.

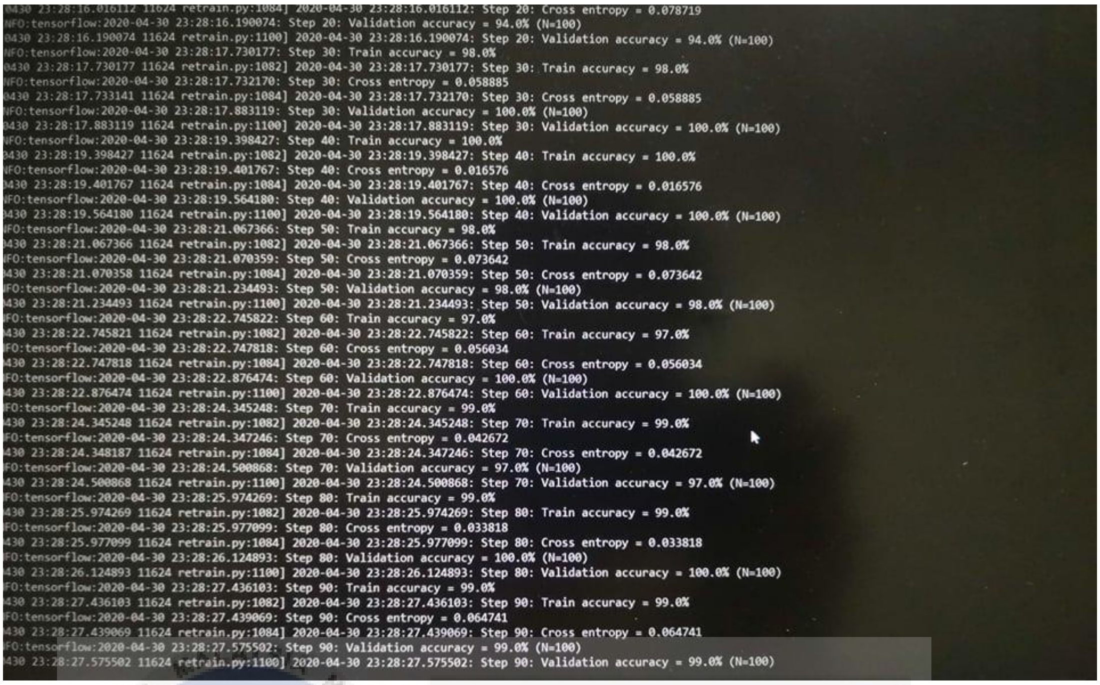

Figure 4.1: Training process

In Deep Learning, to improve something you often need to be able to measure it. TensorBoard is a tool for providing the measurements and visualizations needed during the machine learning workflow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more. TensorBoard will update automatically every 30 seconds during the process of training a new model. If the training graph was increasing while the validation starts to decrease, this can be said that the training process is overfitting, which means that the newly trained model will only classify based on the training sample and it will not recognize when a new photo is feed in. Creating more training data by resizing, scaling, and rotating of the training data set can improve the overfitting. Figure 4.2 and Figure 4.3 showed the accuracy as well as the cross entropy for training and validation graph and the throughout the 4000 training steps. The overall accuracy for newly trained model was 99.83 % while the overall loss was below 4%.

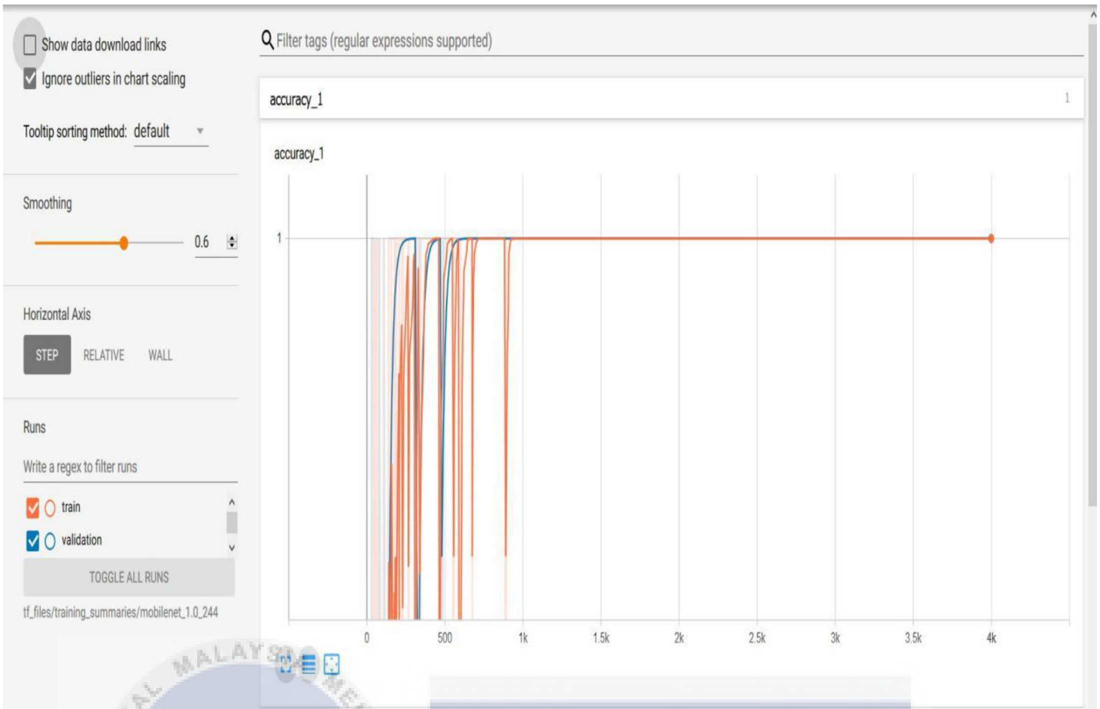

Figure 4.2: Graph of accuracy for training and validation

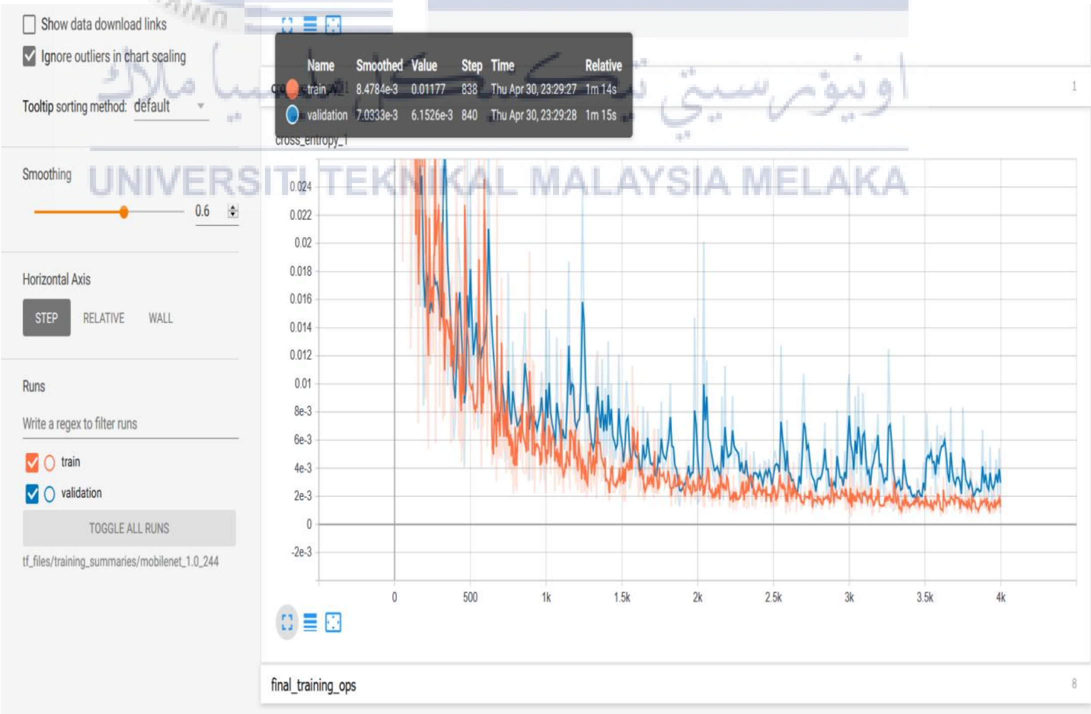

Figure 4.3: Graph of cross entropy for training and validation

4.2 Testing with the newly trained model

The newly trained model was tested with each classes of Malaysian Banknote both front side and back side. After testing process, the retrained model is applied to application on android phone where the application is developed by using android studio software tool.

4.2.1 Front side and back side of the currency

The images are randomly chosen from the class or new ones to test the retrain network. The python script "label_image.py" is to classify the images Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, Figure 4.8, Figure 4.9, and Figure 4.10 show the testing process from random image which taken by android phone. The results have shown all the banknotes with front side and back side are predicted correctly. The newly trained model successfully achieved high precision and low evaluation time from the classification process.

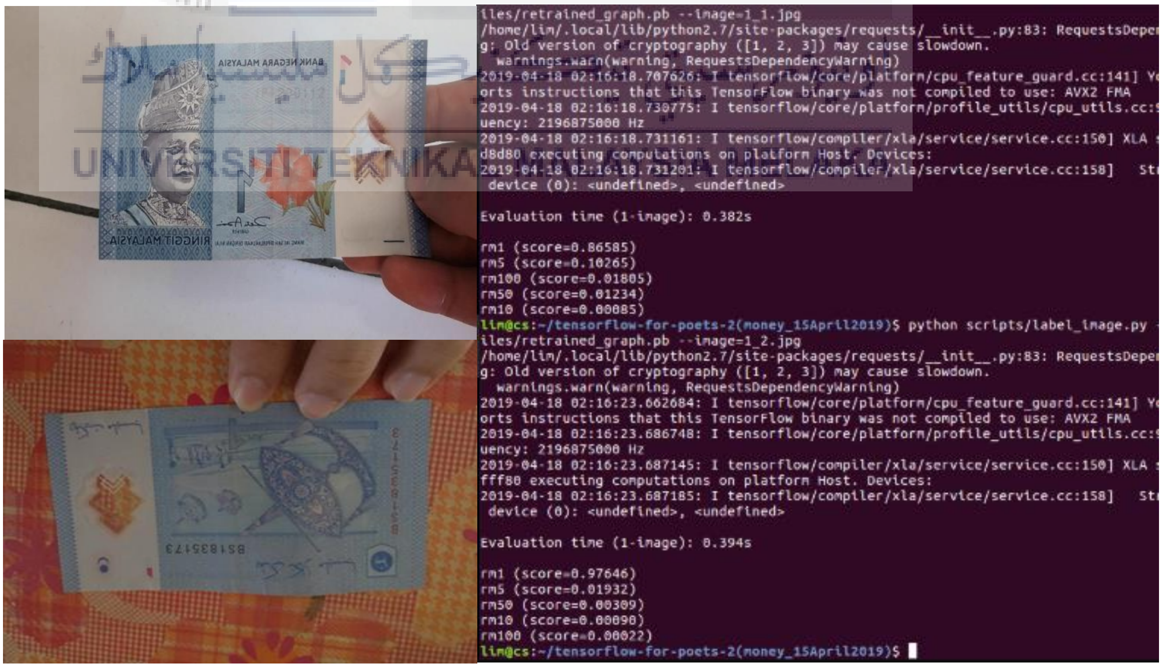

Figure 4.4: Front side and back side of RM1

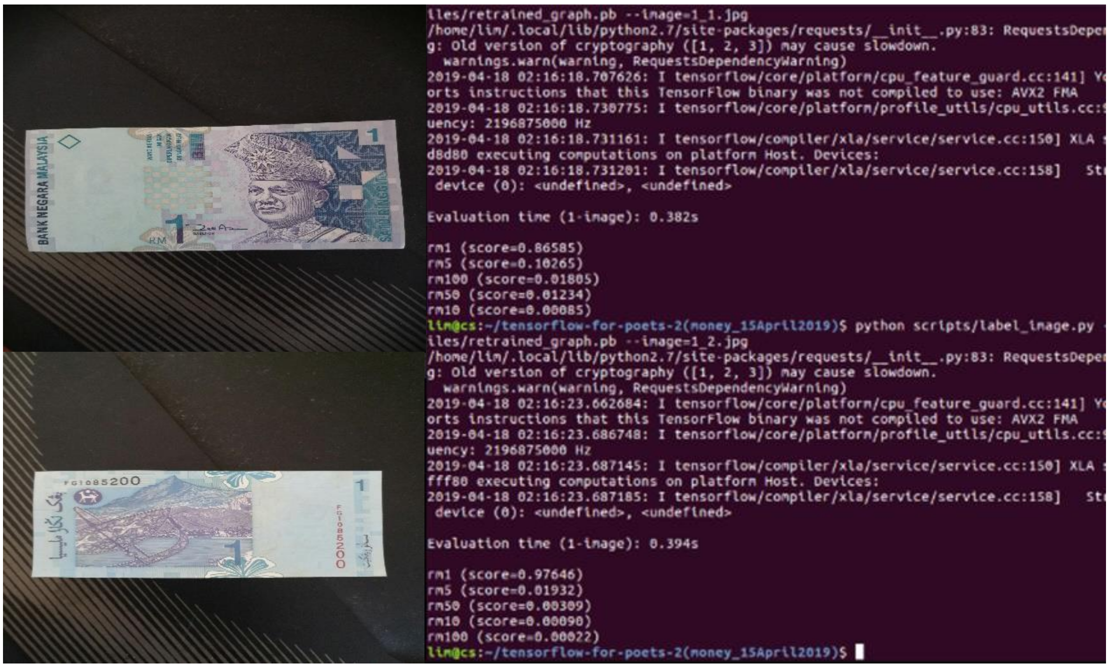

Figure 4.5: Front side and back side of RM1

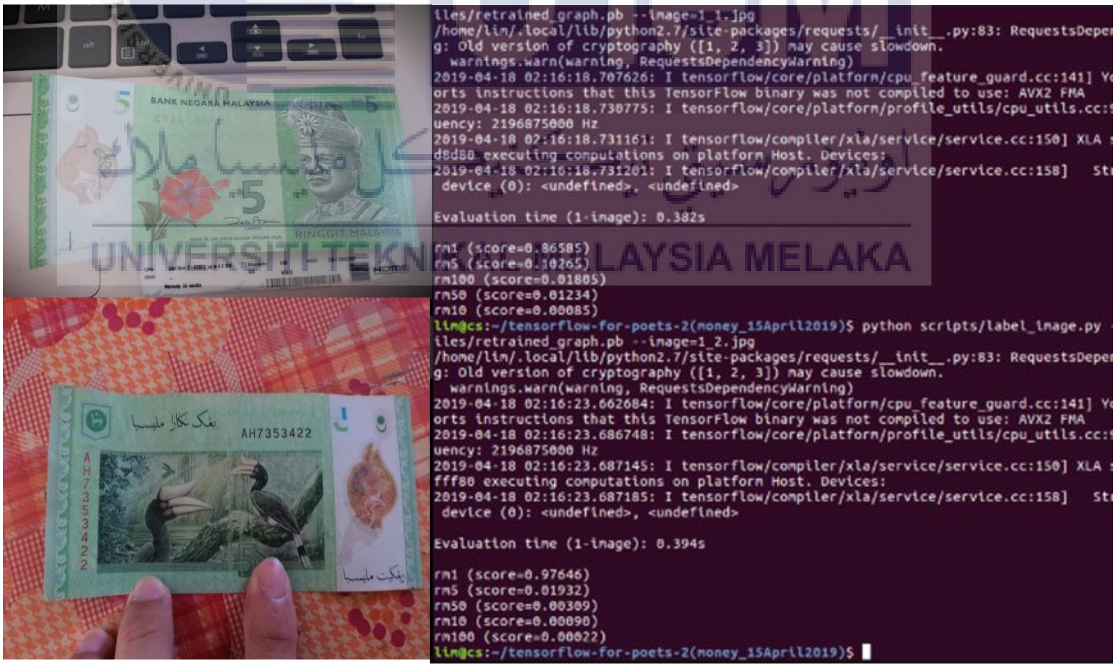

Figure 4.6: Front side and back side of RM5

Figure 4.7: Front side and back side of RM10

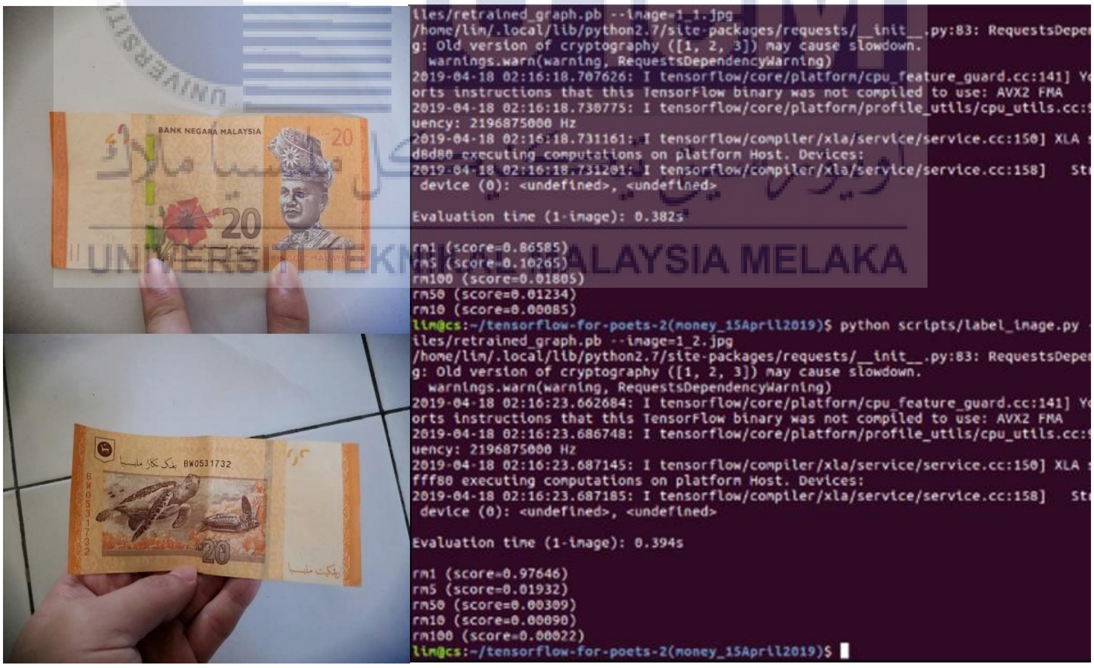

Figure 4.8: Front side and back side of RM20

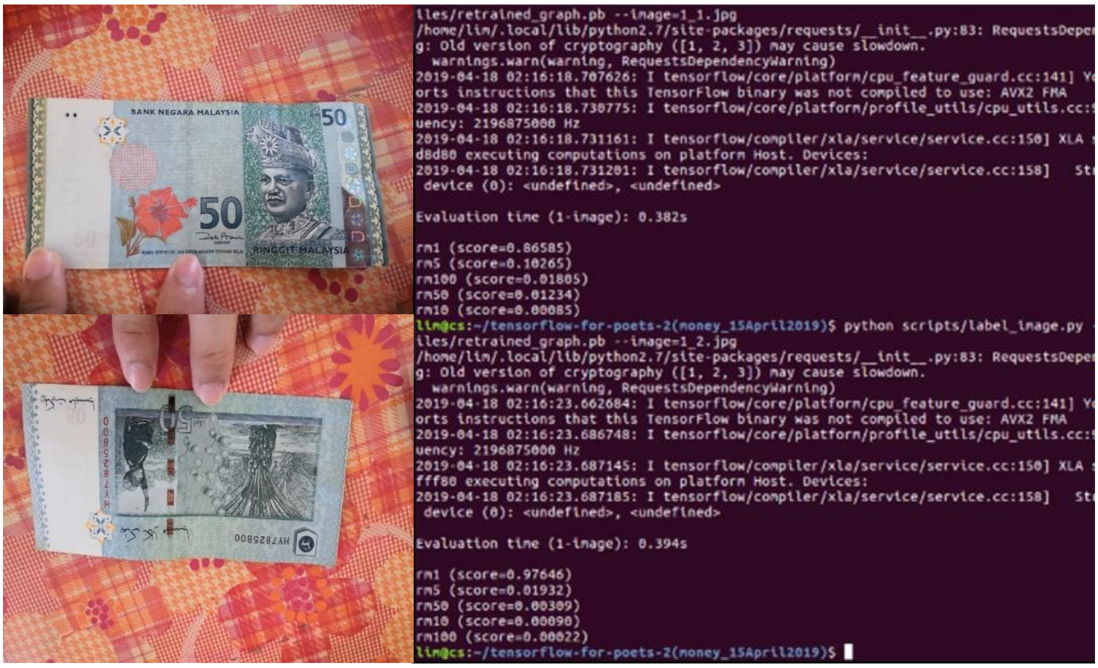

Figure 4.9: Front side and back side of RM50

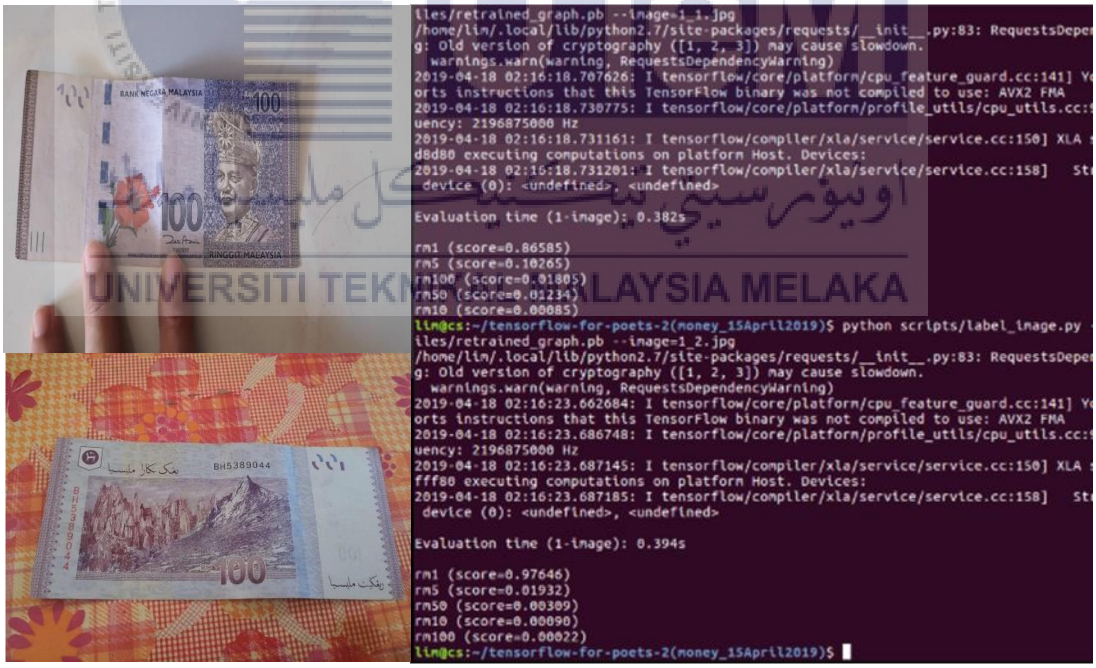

Figure 4.10: Front side and back side of RM100

4.3 Performance Analysis

In the world that full of technology, speed and accuracy are very significant to make the product with outstanding value in this market. Besides that, the precision and swiftness are played an important role in order to maintain or increase the level of customer satisfaction. Therefore, a performance analysis of the product is needed to determine the market value of it. Once the retrained model converts to tensorflow lite, it can be applied to the phone and do classification process totally inside a phone without necessity of WIFI or online data. For this project, an android application is developed for the usage of blind people. The data is collected and is listed in excel file as shown in Figure 4.11.

rm1	0.94	rm50	2%	rm100	2%	609	100		rm50	0.34	rm5	34%	rm100	3%	457	100
rm1	0.97	rm10	1%	rm100	1%	573	110		rm5	0.5	rm50	44%	rm100	3%	469	110
rm1	0.97	rm10	1%	rm100	1%	679	120		rm5	0.89	rm1	9%	rm50	1%	504	120
rm1	0.89	rm100	10%	rm10	0%	640	130		rm5	0.51	rm1	33%	rm50	12%	465	130
rm50	0.32	rm1	32%	rm100	7%	483	140		rm5	0.9	rm50	10%	rm100	0%	515	140
rm50	0.32	rm1	32%	rm100	7%	534	150		rm5	0.46	rm50	30%	nomoney	20%	545	150
rm1	0.84	rm50	10%	nomoney	3%	470	160		rm5	0.99	rm50	1%	rm1	1%	589	160
rm1	0.68	rm100	22%	rm50	9%	494	170		rm5	0.99	rm50	1%	rm1	1%	536	170
rm1	0.92	rm50	4%	rm5	2%	457	180		rm5	0.99	rm50	1%	rm1	1%	499	180
rm1	0.85	rm100	13%	rm50	1%	446	190		rm50	0.08	rm1	23%	rm5	8%	516	190
rm1	0.94	rm50	4%	rm100	2%	429	200		rm5	1	rm50	0%	rm100	0%	597	200
rm1	0.93	nomoney	3%	rm10	2%	432	210		rm5	0.59	rm50	40%	rm1	1%	654	210
rm1	0.93	nomoney	4%	rm10	1%	430	220		rm50	0.02	rm5	2%	rm1	1%	587	220
rm1	0.93	rm100	4%	rm50	1%	428	230		rm50	0.02	rm5	2%	rm1	1%	717	230
rm1	0.55	rm100	23%	rm50	11%	471	240		rm5	0.49	rm50	48%	rm100	2%	468	240
rm1	1	rm100	0%	rm50	0%	434	250		rm5	0.81	rm50	14%	rm10	2%	489	250
rm1	0.99	rm5	0%	rm50	0%	430	260		rm5	0.49	rm50	48%	rm100	2%	482	260
rm1	0.44	nomoney	43%	rm10	8%	430	270		rm5	0.66	rm50	32%	rm1	1%	474	270
rm1	0.68	rm50	11%	rm100	10%	426	280		rm50	0.27	rm5	27%	rm1	1%	474	280
rm1	0.46	rm50	26%	rm100	17%	425	290		rm5	0.8	rm1	9%	rm50	7%	475	290
rm1	0.96	rm50	4%	nomoney	0%	429	300		rm50	0	rm1	1%	rm100	1%	492	300
rm1	0.85	rm50	7%	nomoney	6%	431	310		rm50	0.02	rm5	2%	rm1	1%	446	310
rm1	0.85	rm50	7%	nomoney	6%	444	320		rm5	0.61	rm50	24%	rm100	7%	447	320

Figure 4.11: Collected data in excel file

The collected data are then plotted in a graph to ease the process of analyzing the performance on classifying the Malaysian Banknote.

4.3.1 Result from the android phone

The developed application is applied to two different android phones which are Samsung J7 and Huawei Nova 4 respectively and the results are collected and analyzed. Table 4.1 and Table 4.2 show the specification of android phone selected while Figure 4.12 shows the result displayed on the Android phone.

Table 4.1: Specification of android phone Samsung J7

Processor	Exynos 7870. Processor Speed: 1.6 GHz Octa-core
RAM	2GB
Internal storage	16GB
Rear Camera	8-megapixel
Operating System	Android 7.0

Table 4.2: Specification of android phone Huawei Nova 4

Processor	Octa-core (4x2.4 GHz Cortex-A73 & 4x1.8 GHz Cortex-A53)
RAM	6GB
Internal storage	128GB
Rear Camera	13-megapixel
Operating System	Android 9.0

Figure 4.12: Results displayed on the Android phone

4.3.1.1 Accuracy test on Android phone (Samsung J7)

Figure 4.13, Figure 4.14, Figure 4.15, Figure 4.16, Figure 4.17, and Figure 4.18 below showed the graph for percentage of confidence level against the different brightness for android phone with 8-megapixel. Each classes of banknote are tested with 30 times in dark area (20 until 80 lux) and bright area (100 until 160 lux) respectively. Figure 4.13 showed the number of predictions which failed (below 40%) at bright area was higher compared to number of failed predictions in dark area. The number of wrong predictions at dark area for RM 1 was 7 while bright area was 12 wrong predictions. Therefore, the accuracy achieved for RM1 in dark area (76.67%) was higher than the accuracy obtained at bright area which has only 60%. Figure 4.14 presented the confidence level of RM5 against brightness. The graph showed the results that the number of failure or wrong predictions occurred at dark area (15 times) which was higher than the number of wrong predictions in bright area (5 times). For Figure 4.15, the number of images predicted correctly for RM10 was 21 in dark side and 26 for bright side area where each of them achieved above 70% correct prediction for the 30 number of images tested. The overall confidence level for RM20 in Figure 4.16 obtained the least confidence level with 20% only which indicated that RM20 was easier to be detected whenever it is inside the image taken. The number of images predicted correctly for RM20 was 29 and 27 for dark area and bright area respectively. Figure 4.17 showed that RM50 achieved only 15 times predicted correctly when tested in dark area while 23 times for bright area. Figure 4.18 obtained the highest number of correct predictions for RM100 in bright area which is (29 out of 30) while dark area has 24 out of 30 images are correctly predicted.

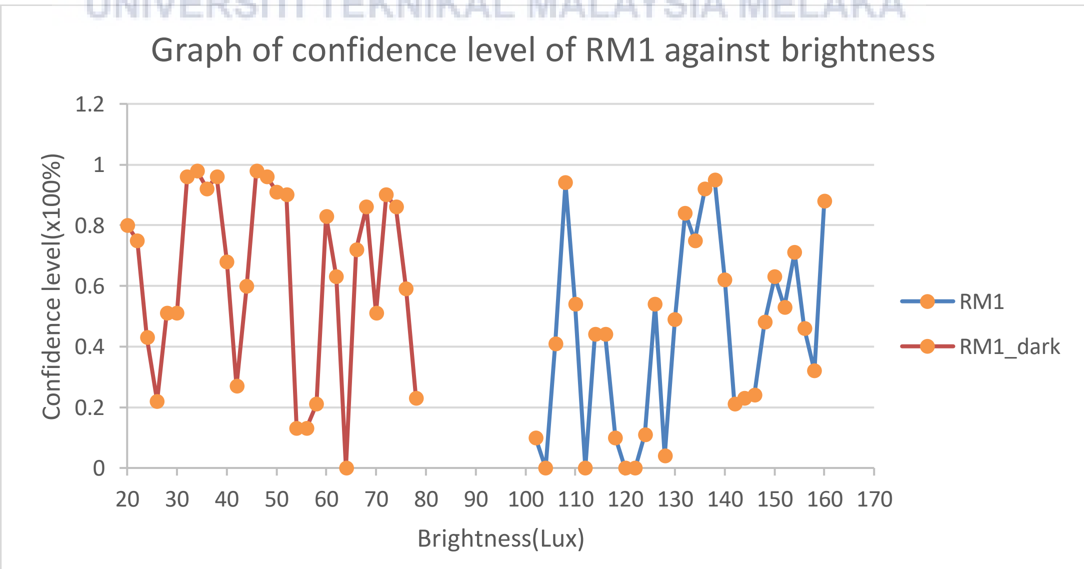

Figure 4.13: Graph of confidence level of RM1 against brightness (8-Megapixel camera)

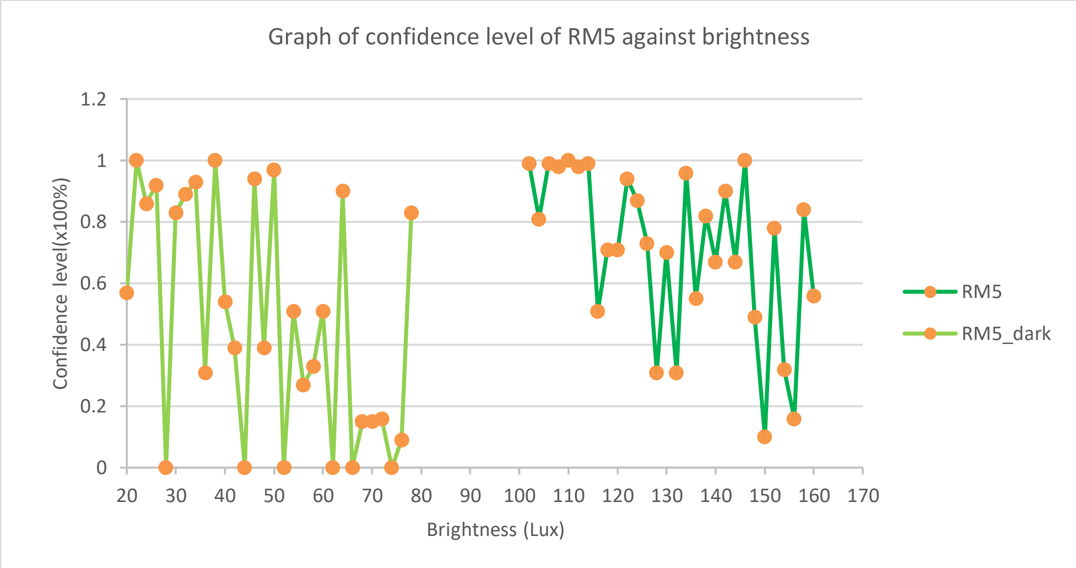

Figure 4.14: Graph of confidence level of RM5 against brightness (8-Megapixel camera)

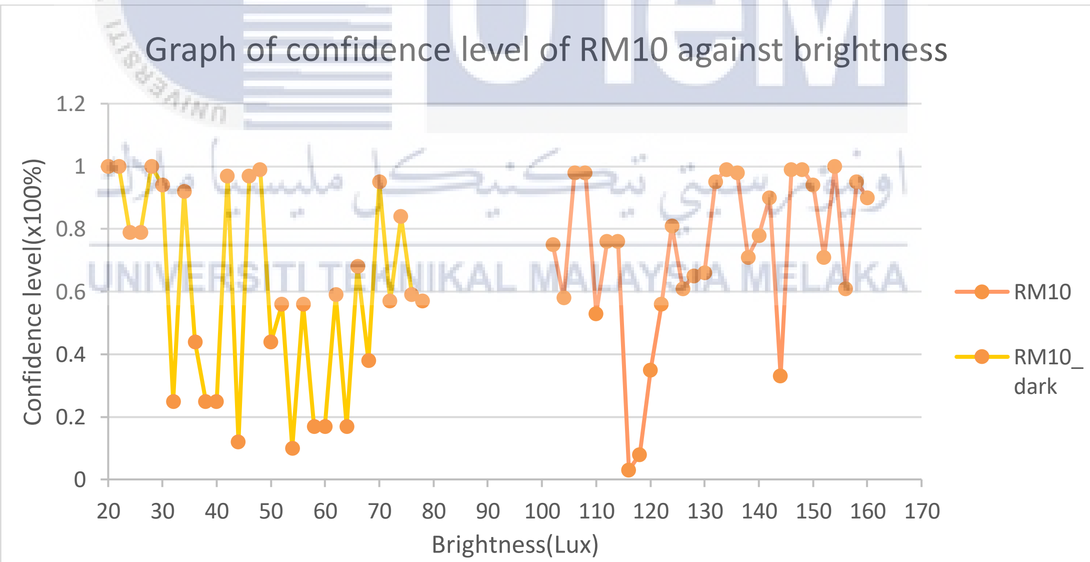

Figure 4.15: Graph of confidence level of RM10 against brightness (8-Megapixel camera)

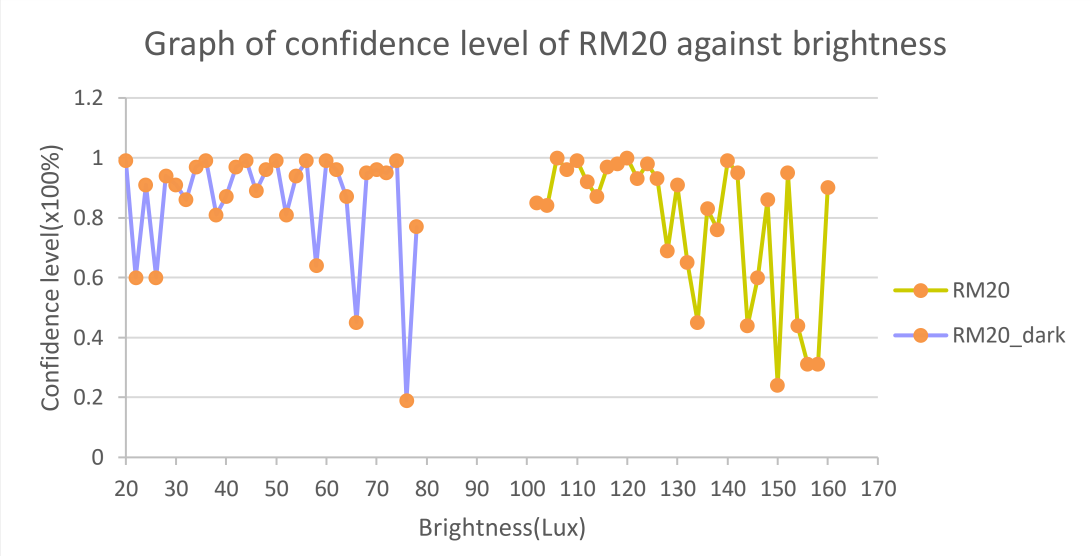

Figure 4.16: Graph of confidence level of RM20 against brightness (8-Megapixel camera)

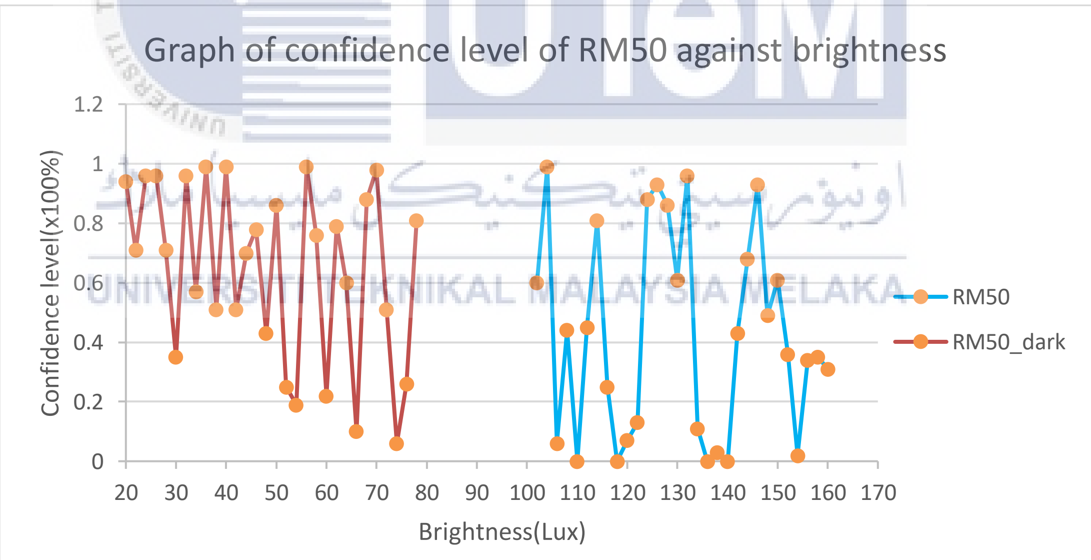

Figure 4.17: Graph of confidence level of RM50 against brightness (8-Megapixel camera)

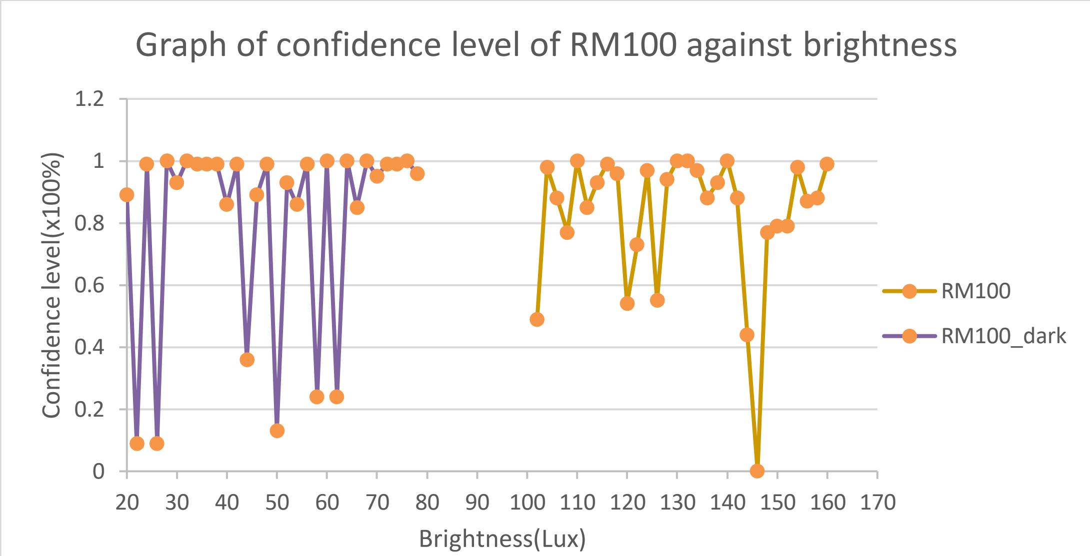

Figure 4.18: Graph of confidence level of RM100 against brightness (8-Megapixel camera)

Table 4.3: Accuracy of Application (Samsung J7) in Dark Area

Banknote	Number of Images predicted correctly	Number of Images test	Average Inferencing time (ms)
RM 1	23	30	443.50
RM 5	15	30	440.60
RM 10	21	30	466.87
RM 20	29	30	481.07
RM 50	23	30	458.07
RM 100	24	30	457.23
TOTAL	135	180	2747.34
Accuracy	75.00%	Average Inference time	457.89

Table 4.4: Accuracy of Application (Samsung J7) in Bright Area

Banknote	Number of Images predicted correctly	Number of Images test	Average Inferencing time (ms)
RM 1	18	30	450.90
RM 5	25	30	462.10
RM 10	26	30	460.93
RM 20	27	30	451.70
RM 50	15	30	459.30
RM 100	29	30	468.23
TOTAL	140	180	2753.16
Accuracy	77.78%	Average Inference time	458.86

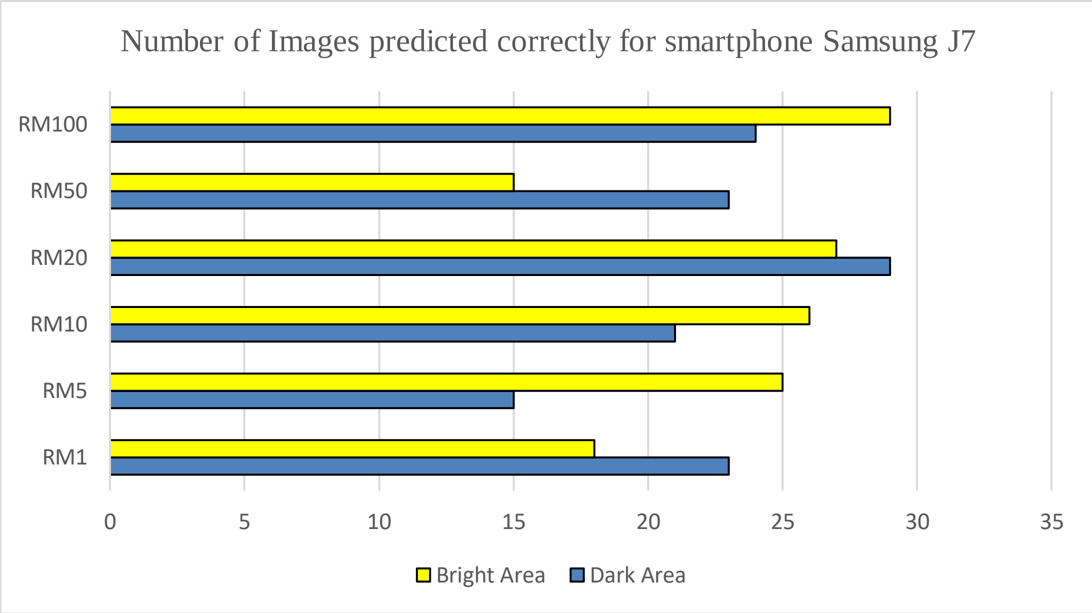

Figure 4.19: Number of images predicted correctly for Samsung J7

Based on Table 4. 3 and Table 4. 4, the number of images predicted correctly for each class was obtained regardless of the brightness are observed. For example, the number of images predicted correctly for RM1, RM20, and RM50 in dark side was higher compared to bright side area while the images predicted correctly for RM5, RM10, and RM100 in bright side was higher compared to those in dark side. This is because the training dataset for each class of banknotes contained different brightness of images. The accuracies obtained were 75% and 77.78% for dark area and bright area respectively. The false predictions of the image can be caused by similar colour of the images and the same shape (rectangular) of the banknote which caused difficulties for the process of classification. Mobilenet is not 100% accurate, therefore varying results are obtained through different kinds of the input images. However, the accuracy of the retrained model can be increased by tightly cropped the images and collect more sample dataset with different background in order to obtain the best performance. From the Table 4.3 and Table 4.4, it is shown that the average inferencing time is almost the same which is 457.89 milliseconds and 458.86 milliseconds for both dark and bright area. It can be concluded that the average inferencing time did not change in large range for both dark and bright condition.

Figure 4.19 showed the bar graph for the number of images predicted correctly for smartphone Samsung J7. From the graph, number of images predicted correctly for categories RM5, RM10, and RM100 in bright area are higher than the number of images predicted correctly in dark area while RM1, RM20, and RM50 showed a higher number of images predicted correctly in dark area. Therefore, the overall of images predicted correctly are almost equal for both bright and dark area.

4.3.1.2 Accuracy test on Android phone (13 Megapixel camera)

Figure 4.20 showed the graph of confidence level of RM1 against brightness for 13-megapixel camera android phone. The results showed that the number of false predictions for dark area has only 4 times while for the bright area has only 2 out of 30 testing images. Figure 4.21 presented that throughout the 30 images tested, there are total 90% accuracy for the predictions in dark area while 76.67% was achieved for the test under the bright condition. Relationship of confidence level against brightness for RM10 shown in Figure 4.22 also showed the significant result that the number of false predictions has only 3 and 6 under dark area and bright area respectively. The false predictions of the image in Figure 4.22 and Figure 4.23 occurred as the similar colour of RM10 and RM20 of Malaysian banknote where RM10 ranked the first in those false predictions of RM20 and RM20 ranked the first for the false predictions when testing with RM10. Figure 4.24 shows that RM50 tested under bright area obtained lower accuracy compared to dark area. For the banknote RM100 as shown in

Figure 4.25 ranked the lowest which has only 16 images are correctly predicted compare to all classes of Malaysian Banknote in dark area.

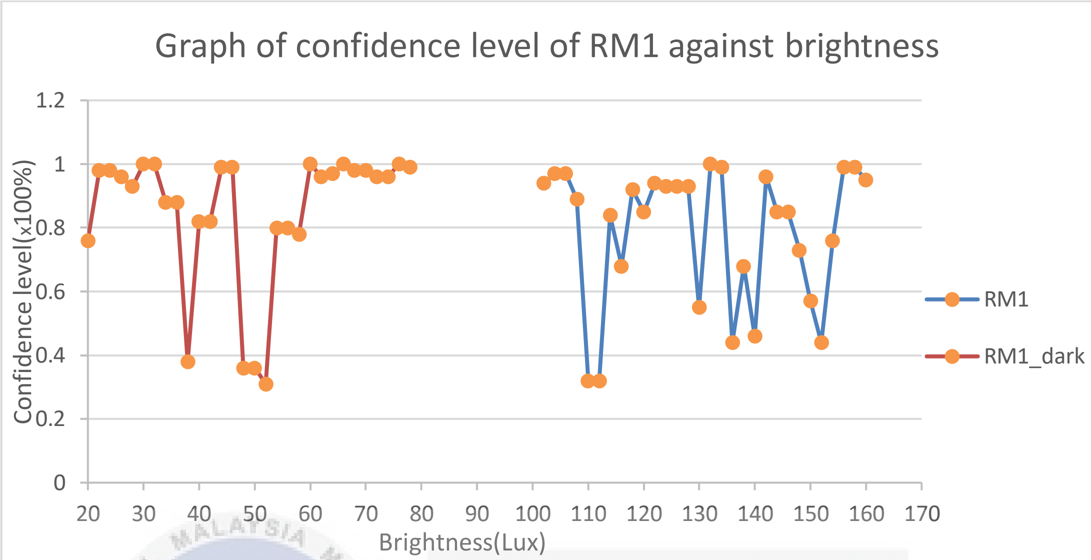

Figure 4.20: Graph of confidence level of RM1 against brightness (13-Megapixel camera)

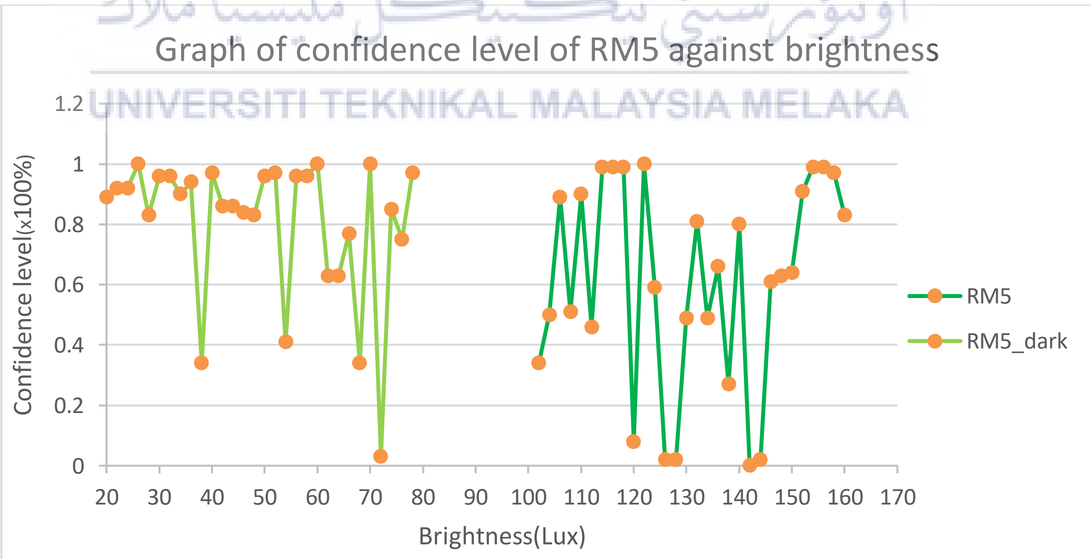

Figure 4.21: Graph of confidence level of RM5 against brightness (13-Megapixel camera)

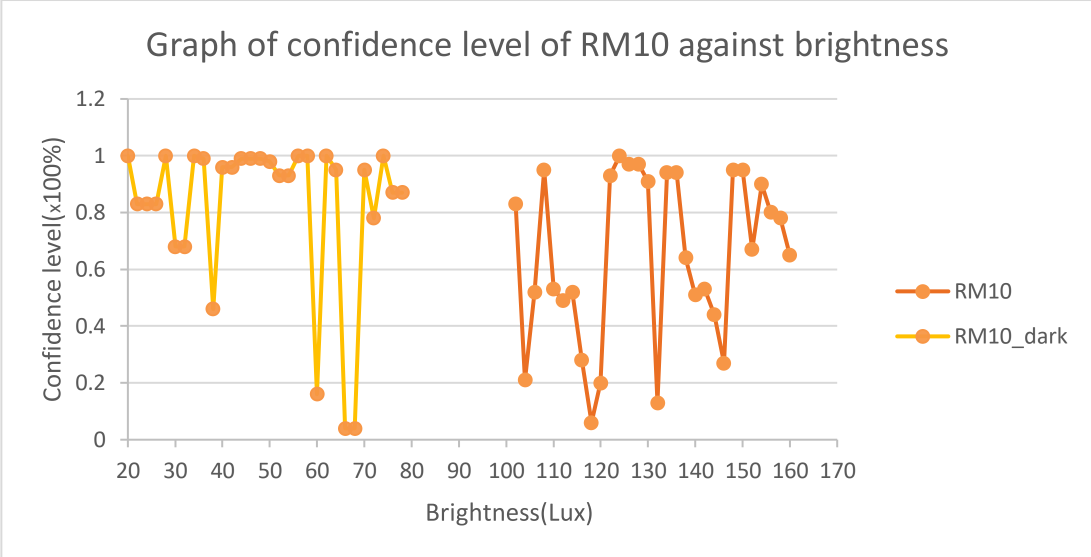

Figure 4.22: Graph of confidence level of RM10 against brightness (13-Megapixel camera)

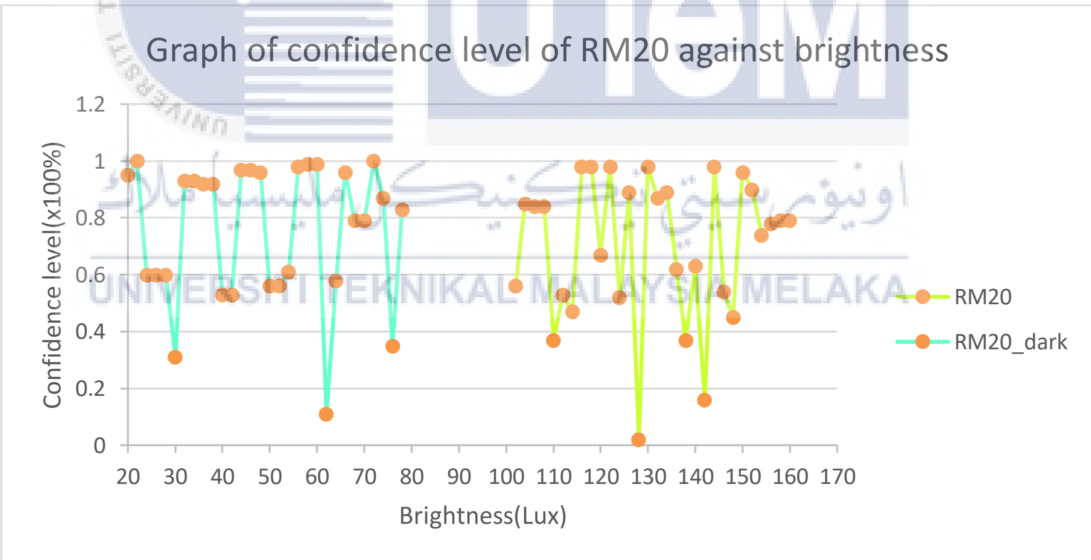

Figure 4.23: Graph of confidence level of RM20 against brightness (13-Megapixel camera)

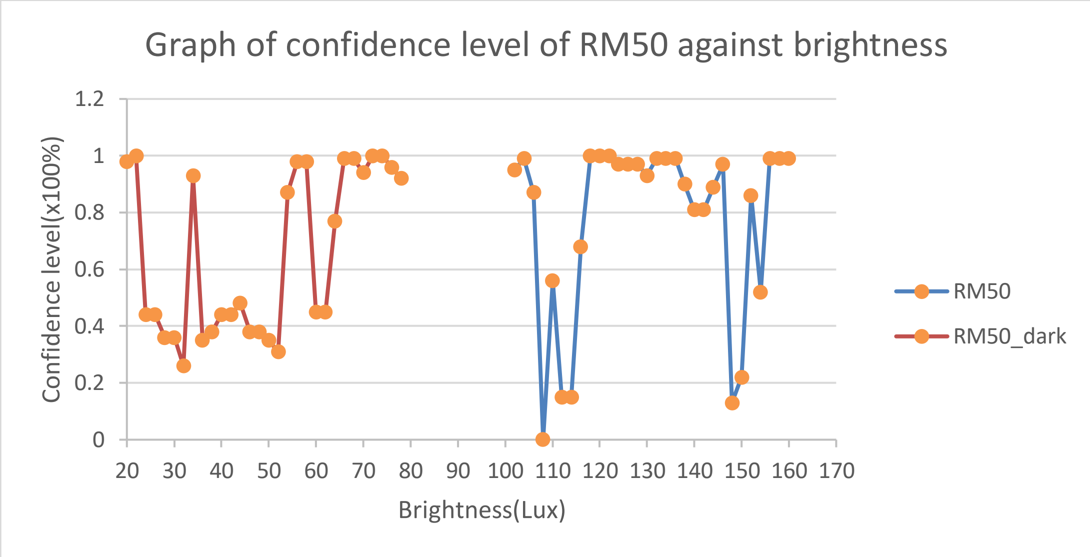

Figure 4.24: Graph of confidence level of RM50 against brightness (13-Megapixel camera)

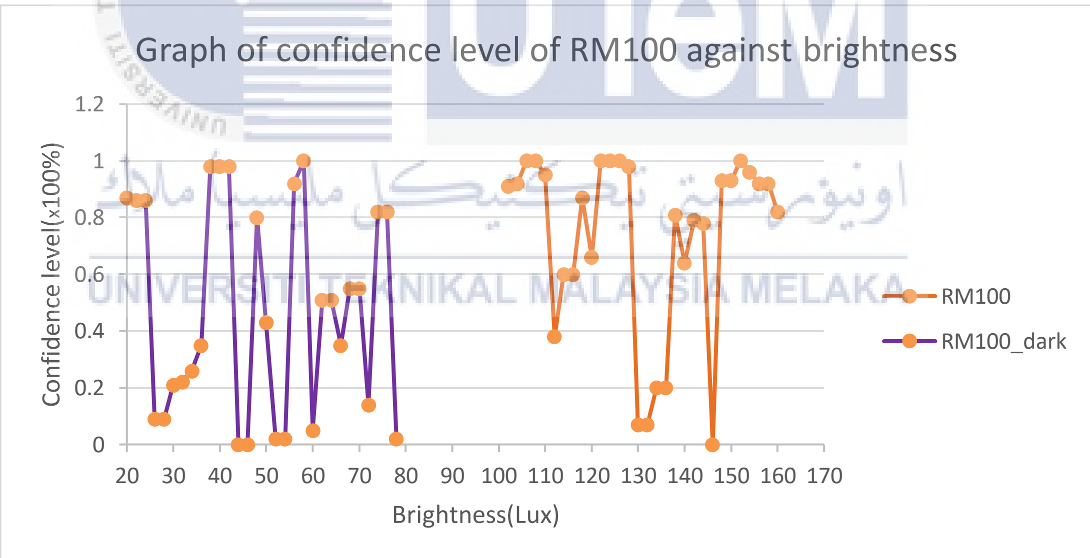

Figure 4.25: Graph of confidence level of RM100 against brightness (13-Megapixel camera)

Table 4.5: Accuracy of Application (Huawei Nova 4) in Dark Area

Banknote	Number of Images predicted correctly	Number of Images test	Average Inferencing time (ms)
RM 1	26	30	507.93
RM 5	27	30	438.50
RM 10	27	30	449.80
RM 20	27	30	488.73
RM 50	21	30	526.73
RM 100	16	30	504.93
TOTAL	144	180	2916.62
Accuracy	80.00%	Average Inference time	486.10

Table 4.6: Accuracy of Application (Huawei Nova 4) in Bright Area

Banknote	Number of Images predicted correctly	Number of Images test	Average Inferencing time (ms)
RM 1	28	30	470.93
RM 5	23	30	501.07
RM 10	24	30	466.10
RM 20	26	30	473.10
RM 50	25	30	517.73
RM 100	24	30	503.23

TOTAL	150	180	2932.16
Accuracy	83.33%	Average Inference time	488.69

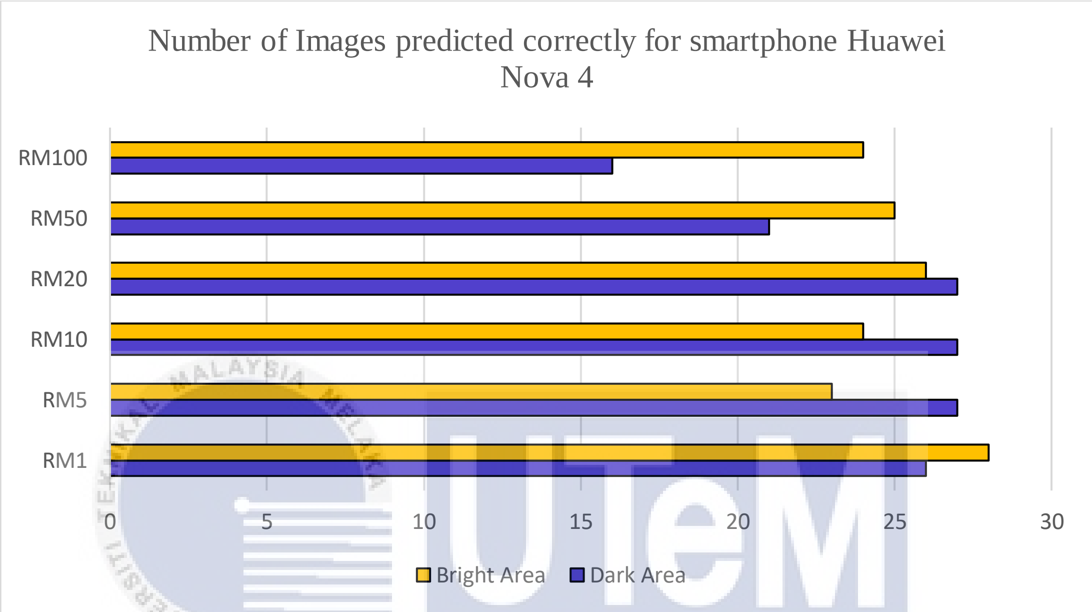

Figure 4.26: Number of images predicted correctly for smartphone Huawei Nova 4

The total accuracies of Malaysian Banknote recognition for dark area and bright area obtained are 80% and 83.33% respectively as shown in Table 4.5 and Table 4.6 .From the Table 4.5 and Table 4.6, it is shown that the average inferencing time is almost the same which is 486.10 milliseconds and 488.69 milliseconds for both dark and bright area. Besides that, the average inferencing time for 13-megapixel android phone is almost the same compare with the 8-megapixel android phone. Therefore, the total accuracies for 13-megapixel android phone showed a better result compared to the accuracies obtained for 8-megapixel android phone. Hence, the specification of the camera used will brings an effect on the results obtained as a clearer image taken helps the classifier to extract features better. Other than that, the performance of the system can be improved through the techniques like increasing the amount of training data set, referring the techniques used based on research paper, and improve the performance with ensembles where the model can be split into different parts. The ensemble prediction will be more robust if each model is skillful but in different ways. [21].

The number of images that predicted correctly are also tested with another smartphone which is Huawei Nova 4 with a higher camera specification to analysis the performance for the developed system. Figure 4.26 showed the number of images predicted correctly for smartphone Huawei Nova 4. RM1, RM50, and RM100 obtained a higher number of images predicted correctly for bright area while RM5, RM10, and RM20 showed the other way around. Based on the graph, the total number of images predicted correctly for both bright and dark area which higher than the total number of images predicted correctly as shown in Figure 4.19 are observed. Therefore, a higher camera specification results a higher accuracy for the prediction process.

4.4 Environment and sustainability

Table 4.7: Environmental and Sustainability

Environmental and Sustainability		
Economic Development	**Social Development**	**Environment Friendly**
➢ Save cost ➢ Save time	➢ Ease blind people handle independently ➢ User-friendly ➢ Provide a comfortable user experience	➢ Virtual and sound pollution ➢ Consume less power

Environmental sustainability can be defined as the responsible which interact with the environment around us to avoid the depletion of natural resources by the developed system. Environment sustainability plays an important role until today for all kinds of project development. It can be divided into three aspects which are economic development, social development and environmental protection as shown in Table 4..

The Malaysian Banknote recognition system that developed in this project has a great effect in the economic development which include cost saving and time saving. Since Tensorflow library allows deep learning process to be used and applied on smartphone devices. Therefore, the developed system will involve the use of application on smartphone only without the need of GPU processer computer as a server for backend processing. Besides, the design of an apps can be just mouse click which is far easier than updating hardware. The use of Tensorflow on smartphone also helps in time saving as the inferencing time will be just a few milliseconds and totally function in without the need of internet.

For social development, the existence of this application helps blind people to handle money independently in their daily life. Also, this application is user-friendly where it is designed such that it can be operated by using touch sensor and the results is returned in terms of sound for them to know the money they hold at that time.

With regard the environment protection, developed software for this project does not really pollute environment except for virtual pollution and sound pollution. Hence, the developed devices will only bring an effect to human eyes and ears as they use smartphone for a longer time. Therefore, suitable time management for the use of smartphone can help to solve this problem. Other than that, since the developed application will be deployed on smartphone without the needs of computer as a server, therefore the total power consume for this device will be same as the power used by smartphone.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter will conclude on overall of the project development that have been done. The suggestions and future work will be stated in this chapter as well so as to help in project development for future.

5.1 Conclusion

In a nutshell, the prevalence of blind people was high, and the process of transaction are still a problem for them as it is difficult for them to handle and classify the money independently. In order to help them differentiate banknote efficiently, a system is designed and developed with the aim to classify different classes of Malaysian Banknote. In order for the system to classify the banknote efficiently and fast, a framework of deep learning (TensorFlow) developed by Google is being used as the fast inference time and high accuracy compared to computer vision and other technique like Matlab which can undergo image processing in order to differentiate different kind of objects. Besides, TensorFlow Lite which is a lightweight solution for mobile and embedded devices is being used as TensorFlow Lite has achieved low latency and optimize for mobile app which helps to deploy the retrained model in mobile phone. An android app has been developed which operates with voice instruction and just a finger tab on the screen in order to snap the image of the banknote for backend classification and then the phone will return the result with voice on the android phone. For the part of analysis and identification of the performance of this system, a total 360 samples are collected for each android phone with different camera specifications. The results collected on excel was collected and analyze through graphs and tables shown in above. The best results achieved were 80% and 83.33% accuracies with an average inferencing time of 486.10 milliseconds and 488.69 milliseconds for dark area and bright area respectively. As a conclusion, Malaysian Banknote Recognition system is developed with the features of low cost, touch sensor, less inference time, voice instructions, and operates in offline state is developed successfully with acceptable accuracies achieved.

5.2 Suggestions for Future Work

Although the system is completed and it can function with desire accuracies in both dark and bright area, but the system is still not a ready product to be pushed to the market or used by this society. This product still has to be done with different performance test by different people and under different conditions to improve this system in order to work in optimum level. Instead of performance test, there are some new implementations or improvements still can be done on this project. In order to increase the accuracy of this system, the more detail feature images and various background images can be included to train for this algorithm. The system can also be further developed with counterfeit banknote by collecting dataset from captured images of genuine and forged banknotes specimen, and wavelet transform tool which was used to extract features from the captured images [22]. Last but not least, more user-friendly user interface (UI) can be designed based on the feedback from blind people to provide an easy, convenient and comfortable way for the users.

REFERENCES

[1] N. F. Ngah *et al.*, "Estimates of visual impairment and its causes from the National Eye Survey in Malaysia (NESII)," *PLoS One*, vol. 13, no. 6, p. e0198799, 2018.

[2] J. Guerreiro, D. Gonçalves, D. Marques, T. Guerreiro, H. Nicolau, and K. Montague, " The today and tomorrow of Braille learning ," no. May 2014, pp. 1–2, 2013.

[3] N. Panah and H. Masoumi, "Banknotes detected using Image Processing Techniques," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 5, pp. 34–44, 2017.

[4] D. Mulmule and A. Dravid, "A Study of Computer Vision Techniques for Currency Recognition on Mobile Phone for the Visually Impaired," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 11, pp. 2277–128, 2014.

[5] L. Petricca, T. Moss, G. Figueroa, and S. Broen, "Corrosion Detection using A.I. : A Comparison of Standard Computer Vision Techniques and Deep Learning Model," *Int. Conf. Comput. Sci. Eng. Inf. Technol.*, pp. 91–99, 2016.

[6] Y. Guo, *Deep Learning for VisuaL Understanding*. 2017.

[7] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no.

7553, pp. 436–444, 2015.

[8] M. Rucci and A. Casile, “Fixational instability and natural image statistics: Implications for early visual representations,” *Netw. Comput. Neural Syst.*, vol. 16, no. 2–3, pp. 121–138, 2015.

[9] J. D. Dignam, P. L. Martin, B. S. Shastry, and R. G. Roeder, “Eukaryotic gene transcription with purified components,” *Methods Enzymol.*, vol. 101, no. C, pp. 582–598, 2016.

[10] C. Huyen, D. Hafner, J. Gautier, M. Luong, and P. Warren, “CS 20SI: Lecture note 1 : Introduction to TensorFlow So what ’ s Tensorflow ? Why TensorFlow,” *Stanford CS 20SI TensorFlow Deep Learn. Res.*

[11] D. Learning *et al.*, “Tensorflow の上に実装した Deep Learning のよる文書分類（パラメータ設定の検討）,” pp. 555–556, 2017.

[12] K. Dobosz, “– DRAFT VERSION,” no. November, 2017.

[13] L. D. Dunai, M. C. Pérez, G. Peris-Fajarnés, and I. L. Lengua, “Euro banknote recognition system for blind people,” *Sensors (Switzerland)*, vol. 17, no. 1, 2017.

[14] N. F. Ghazali, M. Amir, M. N. Jamaludin, L. H. Ismail, F. M. Latip, and A. H. Omar, “Wearable Device for Malaysian Ringgit Banknotes Recognition Based on Embedded Decision Tree Classifier,” vol. 10, no. 1, pp. 129–137, 2018.

[15] N. A. Semary, S. M. Fadl, M. S. Essa, and A. F. Gad, “Currency Recognition System for Visually Impaired : Egyptian Banknote as a Study Case,” no.

December, 2015.

[16] I. Alexeff, “Comment on ‘a critique of the throughtput limit in mass spectrographs,’” *J. Appl. Phys.*, vol. 49, no. 11, p. 5668, 2014.

[17] F. M. Hasanuzzaman, X. Yang, and Y. Tian, “Robust and effective component-based banknote recognition for the blind,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 1021–1030, 2012.

[18] A. Wijesundara, “Object Recognition with OpenCV on Android – Akshika Wijesundara – Medium,” medium.com, 20-Dec-2016. [Online]. Available: https://medium.com/@akshikawijesundara/object-recognition-with-opencv on-android-6435277ab285.

[19] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski, “ORB (Oriented FAST and Rotated BRIEF)¶,” Cascade Classification - OpenCV 2.4.13.7 documentation, 10-Nov-2014. [Online]. Available: https://docs.opencv.org/3.0 beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html#theory.

[20] A. J. (Mod), “Light Meter App on mobile phone: easy way to optimize room lighting and save electricity,” *Bijli Bachao,* 26-Apr-2018. [Online]. Available: https://www.bijlibachao.com/lights/light-meter-app-on-mobile-phone-easy-way-to-optimize-room-lighting-and-save-electricity.html?fbclid=IwAR33P8ADHrRLBlS7wE4l7epwDtxKvo7GsGOK

kgNXo7opSFbDJ_i-FPFHtSc.

[21] Jason Brownlee, “How To Improve Deep Learning Performance,” *Machine Learning Mastery*, 02-May-2019. [Online]. Available: https://machinelearningmastery.com/improve-deep-learning-performance/?fbclid=IwAR0AUtm2peWHwvWvvDi6mzhmj0iclo_3GQacih-Uo4uTS-3ZyQeZzudCdcU.

[22] V. Rajkumar and V. Rajkumar, “Detecting fake banknotes using TensorFlow,” *Medium*, 29-Jul-2017. [Online]. Available: https://medium.com/tensorist/detecting-fake-banknotes-using-tensorflow-be21ffd2c478.

APPENDICES

Appendix A: Android Studio Code

(Main Code)

```
package com.example.cslim.samplecamera;

import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.content.res.AssetFileDescriptor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.graphics.drawable.BitmapDrawable;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.net.Uri;
import android.nfc.Tag;
import android.os.Environment;
import android.os.SystemClock;
import android.provider.MediaStore;
import android.support.v4.view.GestureDetectorCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.GestureDetector;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.squareup.picasso.MemoryPolicy;
import com.squareup.picasso.NetworkPolicy;
import com.squareup.picasso.Picasso;

import org.tensorflow.lite.Interpreter;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.MappedByteBuffer;
import java.nio.channels.FileChannel;
import java.util.AbstractMap;
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.Map;
import java.util.PriorityQueue;

import jxl.Workbook;
import jxl.read.biff.BiffException;
import jxl.write.Label;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;
```

```
import jxl.write.biff.RowsExceededException;

import static android.content.ContentValues.TAG;

public class Activity2 extends AppCompatActivity implements GestureDetector.OnGestureListener,
GestureDetector.OnDoubleTapListener,SensorEventListener {

    private long backPressedTime;
    private Toast backToast;
    private GestureDetectorCompat gestureDetector;
    private Button classify_button; //test using button
    // presets for rgb conversion
    private static final int RESULTS_TO_SHOW = 3;
    private static final int IMAGE_MEAN = 128;
    private static final float IMAGE_STD = 128.0f;

    /** Name of the model file stored in Assets. */
    private static final String MODEL_PATH = "optimized_graph.lite";

    /** Name of the label file stored in Assets. */
    private static final String LABEL_PATH = "retrained_labels.txt";

    // tflite graph
    private Interpreter tflite;
    // holds all the possible labels for model
    private List<String> labelList;
    // holds the selected image data as bytes
    private ByteBuffer imgData = null;
    // holds the probabilities of each label for non-quantized graphs
    private float[][] labelProbArray = null;
    // holds the probabilities of each label for quantized graphs
    private byte[][] labelProbArrayB = null;
    // array that holds the labels with the highest probabilities
    private String[] topLables = null;
    // array that holds the highest probabilities
    private String[] topConfidence = null;

    // input image dimensions for the Inception Model
    private int DIM_IMG_SIZE_X = 224; //299
    private int DIM_IMG_SIZE_Y = 224;  //299
    private int DIM_PIXEL_SIZE = 3;

    // int array to hold image data
    private int[] intValues;

    // activity elements
    private ImageView img;
    private TextView label1;
    private TextView label2;
    private TextView label3;
    private TextView Confidence1;
    private TextView Confidence2;
    private TextView Confidence3;

    Context context;
    MyApp mApp;
    SensorManager sensorManager;
    Sensor sensor;

    // priority queue that will hold the top results from the CNN
    private PriorityQueue<Map.Entry<String, Float>> sortedLabels =
```

```
        new PriorityQueue<>(
                RESULTS_TO_SHOW,
                new Comparator<Map.Entry<String, Float>>() {
                    @Override
                    public int compare(Map.Entry<String, Float> o1, Map.Entry<String, Float> o2) {
                        return (o1.getValue()).compareTo(o2.getValue());
                    }
                });

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // initialize array that holds image data
        intValues = new int[DIM_IMG_SIZE_X * DIM_IMG_SIZE_Y];
        super.onCreate(savedInstanceState);
        this.gestureDetector = new GestureDetectorCompat(this, this);
        gestureDetector.setOnDoubleTapListener(this);
        //initilize graph and labels
        try{
            tflite = new Interpreter(loadModelFile());
            labelList = loadLabelList();
        } catch (Exception ex){
            ex.printStackTrace();
        }
        imgData =
                ByteBuffer.allocateDirect(
                        4 * DIM_IMG_SIZE_X * DIM_IMG_SIZE_Y * DIM_PIXEL_SIZE);
        imgData.order(ByteOrder.nativeOrder());
        labelProbArray = new float[1][labelList.size()];

        setContentView(R.layout.activity_2);

        //sound region
        context = getApplicationContext();
        mApp = ((MyApp)getApplicationContext());
//        mApp.play(context,4);
        //

        // labels that hold top three results of CNN
        label1 = (TextView) findViewById(R.id.label1);
        label2 = (TextView) findViewById(R.id.label2);
        label3 = (TextView) findViewById(R.id.label3);
        // displays the probabilities of top labels
        Confidence1 = (TextView) findViewById(R.id.Confidence1);
        Confidence2 = (TextView) findViewById(R.id.Confidence2);
        Confidence3 = (TextView) findViewById(R.id.Confidence3);
        // initialize imageView that displays selected image to the user
        img= (ImageView) findViewById(R.id.img);

        // initialize array to hold top labels
        topLables = new String[RESULTS_TO_SHOW];
        // initialize array to hold top probabilities
        topConfidence = new String[RESULTS_TO_SHOW];
//        loadResult();
        // get current bitmap from imageView
        File sdCard = Environment.getExternalStorageDirectory();

        File directory = new File (sdCard.getAbsolutePath() + "/GUI");

        File file = new File(directory, "temp.jpg"); //or any other format supported

        FileInputStream streamIn = null;
```

```
        try {
            streamIn = new FileInputStream(file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        Bitmap bitmap_orig = BitmapFactory.decodeStream(streamIn); //This gets the image
        try {
            streamIn.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
//        Bitmap bitmap_orig = ((BitmapDrawable)img.getDrawable()).getBitmap();
        // resize the bitmap to the required input size to the CNN
        Bitmap bitmap = getResizedBitmap(bitmap_orig, DIM_IMG_SIZE_X, DIM_IMG_SIZE_Y);
        // convert bitmap to byte array
        convertBitmapToByteBuffer(bitmap);
        // pass byte data to the graph
        long startTime = SystemClock.uptimeMillis();
        tflite.run(imgData, labelProbArray);
        long endTime = SystemClock.uptimeMillis();
        Log.d(TAG, "Timecost to run model inference: " + Long.toString(endTime - startTime));
        //LOAD RESULT
        Picasso.with(Activity2.this).load(file).memoryPolicy(MemoryPolicy.NO_CACHE
).networkPolicy(NetworkPolicy.NO_CACHE).into(img);
        // display the results
        String Timecost =  Long.toString(endTime - startTime);
        printTopKLabels(Timecost);
        sensorManager = (SensorManager) getSystemService(Service.SENSOR_SERVICE);
        sensor = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);

    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this,sensor,SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent intent = new Intent(this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        intent.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
        intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
        startActivity(intent);
    }

    // loads tflite graph from file
    private MappedByteBuffer loadModelFile() throws IOException {
        AssetFileDescriptor fileDescriptor = this.getAssets().openFd(MODEL_PATH);
        FileInputStream inputStream = new FileInputStream(fileDescriptor.getFileDescriptor());
        FileChannel fileChannel = inputStream.getChannel();
        long startOffset = fileDescriptor.getStartOffset();
```

```
    long declaredLength = fileDescriptor.getDeclaredLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY, startOffset, declaredLength);
}

// converts bitmap to byte array which is passed in the tflite graph
private void convertBitmapToByteBuffer(Bitmap bitmap) {
    if (imgData == null) {
        return;
    }
    imgData.rewind();
    bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0, bitmap.getWidth(), bitmap.getHeight());
    // loop through all pixels
    int pixel = 0;
    for (int i = 0; i < DIM_IMG_SIZE_X; ++i) {
        for (int j = 0; j < DIM_IMG_SIZE_Y; ++j) {
            final int val = intValues[pixel++];
            // get rgb values from intValues where each int holds the rgb values for a pixel.
            // if quantized, convert each rgb value to a byte, otherwise to a float
                imgData.putFloat((((val >> 16) & 0xFF)-IMAGE_MEAN)/IMAGE_STD);
                imgData.putFloat((((val >> 8) & 0xFF)-IMAGE_MEAN)/IMAGE_STD);
                imgData.putFloat((((val) & 0xFF)-IMAGE_MEAN)/IMAGE_STD);
        }
    }
}

// loads the labels from the label txt file in assets into a string array
private List<String> loadLabelList() throws IOException {
    List<String> labelList = new ArrayList<String>();
    BufferedReader reader =
            new BufferedReader(new InputStreamReader(this.getAssets().open(LABEL_PATH)));
    String line;
    while ((line = reader.readLine()) != null) {
        labelList.add(line);
    }
    reader.close();
    return labelList;
}

// print the top labels and respective confidences
private void printTopKLabels(String Timecost) {
    // add all results to priority queue
    for (int i = 0; i < labelList.size(); ++i) {
        sortedLabels.add(
                new AbstractMap.SimpleEntry<>(labelList.get(i), labelProbArray[0][i]));
        if (sortedLabels.size() > RESULTS_TO_SHOW) {
            sortedLabels.poll();
        }
    }

    // get top results from priority queue
    final int size = sortedLabels.size();
    for (int i = 0; i < size; ++i) {
        Map.Entry<String, Float> label = sortedLabels.poll();
        topLables[i] = label.getKey();
        topConfidence[i] = String.format("%.0f%%",label.getValue()*100);
    }

    // set the corresponding textviews with the results
    label1.setText("1. "+ topLables[2].toUpperCase());
    Log.i(TAG, topLables[2]);
    label2.setText("2. "+ topLables[1].toUpperCase());
```

```
        Log.i(TAG, topLables[1]);
        label3.setText("3. "+ topLables[0].toUpperCase());
        Log.i(TAG, topLables[0]);
        Confidence1.setText(topConfidence[2]);
        Confidence2.setText(topConfidence[1]);
        Confidence3.setText(topConfidence[0]);

        writeExternalStorage(topLables[2],topLables[1],topLables[0],topConfidence[2], topConfidence[1],
topConfidence[0],Timecost);

        if (topLables[2] .equals("nomoney")){
            Intent intent = new Intent(this, Activity3.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
            startActivity(intent);
        }
        if (topLables[2] .equals("rm1")){
            mApp.play(context,2);
        }
        if (topLables[2] .equals("rm5")){
            mApp.play(context,3);
        }
        if (topLables[2] .equals("rm10")){
            mApp.play(context,4);
        }
        if (topLables[2] .equals("rm20")){
            mApp.play(context,5);
        }
        if (topLables[2] .equals("rm50")){
            mApp.play(context,6);
        }
        if (topLables[2] .equals("rm100")){
            mApp.play(context,7);
        }
    }

    // resizes bitmap to given dimensions
    public Bitmap getResizedBitmap(Bitmap bm, int newWidth, int newHeight) {
        int width = bm.getWidth();
        int height = bm.getHeight();
        float scaleWidth = ((float) newWidth) / width;
        float scaleHeight = ((float) newHeight) / height;
        Matrix matrix = new Matrix();
        matrix.postScale(scaleWidth, scaleHeight);
        Bitmap resizedBitmap = Bitmap.createBitmap(
                bm, 0, 0, width, height, matrix, false);
        return resizedBitmap;
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        this.gestureDetector.onTouchEvent(event);
        return super.onTouchEvent(event);
    }

    @Override
    public boolean onSingleTapConfirmed(MotionEvent e) {
        Log.i(TAG, "onSingleTapConfirmed");
        mApp.stop();
        if (topLables[2] .equals("rm1")){
```

```
            mApp.play(context,2);
        }
        if (topLables[2] .equals("rm5")){
            mApp.play(context,3);
        }
        if (topLables[2] .equals("rm10")){
            mApp.play(context,4);
        }
        if (topLables[2] .equals("rm20")){
            mApp.play(context,5);
        }
        if (topLables[2] .equals("rm50")){
            mApp.play(context,6);
        }
        if (topLables[2] .equals("rm100")){
            mApp.play(context,7);
        }
        return true;
    }

    @Override
    public boolean onDoubleTap(MotionEvent e) {
        Log.i(TAG,"onDoubleTap");
        return true;
    }

    @Override
    public boolean onDoubleTapEvent(MotionEvent e) {
        Log.i(TAG,"onDoubleEvent");
        mApp.stop();
        openMainActivity();
//      camera.startPreview(); //back to activity 1
        return true;
    }

    private void openMainActivity() {
            Intent intent = new Intent(this, MainActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
            startActivity(intent);
    }

    @Override
    public boolean onDown(MotionEvent e) {
        Log.i(TAG,"onDown");
        return true;
    }

    @Override
    public void onShowPress(MotionEvent e) {
        Log.i(TAG,"onShowPress");
    }

    @Override
    public boolean onSingleTapUp(MotionEvent e) {
        Log.i(TAG,"onSingleTapUp");
        return true;
    }

    @Override
```

```
    public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
        Log.i(TAG,"onScroll");
        return true;
    }

    @Override
    public void onLongPress(MotionEvent e) {
        Log.i(TAG,"onLongPress");
        mApp.play(context,0);
    }

    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
        Log.i(TAG,"onFling");
        return true;
    }

    public void writeExternalStorage(String top1, String top2, String top3, String percent1, String
percent2, String percent3, String Timecost){
        String state;
        state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state))
        {
            File Root = Environment.getExternalStorageDirectory();
            File Dir = new File(Environment.getExternalStorageDirectory() + File.separator + "GUI" );
            if (!Dir.exists()){
                Dir.mkdir();
            }
            File file2 = new File(Dir,"firstexcel.xls");
            if(!file2.exists()){
                createExcel(file2);
            }
            else {
                writingExcel(file2, top1, top2, top3, percent1,percent2, percent3,Timecost);
            }
        }
        else
        {
            Log.i(TAG, "SD card not found");
        }
    }
    private void createExcel(File file2) {
        try{
            WritableWorkbook workbook = Workbook.createWorkbook(file2);
            workbook.createSheet("first sheet", 0);
            workbook.write();
            workbook.close();
        }
        catch (Exception e){
            e.printStackTrace();
        }

    }
    //
    public static int column = 0;
    public static int row = 0;
    private static int x = 0;
    private void writingExcel(File file2,String top1, String top2, String top3, String percent1, String
percent2, String percent3, String Timecost) {
        try{
            Log.i(TAG, top1 + "\n" + top2 + "\n" +  top3 +  "\n" + percent1 +  "\n" + percent2 + "\n" +
```

```java
percent3);
//          String [][] result1 = {{top1,percent1},{top2,percent2},{top3,percent3}};
            Workbook wb = Workbook.getWorkbook(file2);
            WritableWorkbook copy = Workbook.createWorkbook((file2),wb);
            WritableSheet copySheet = copy.getSheet(0);
            Label label1 = new Label(column,row,top1);
            Label label2 = new Label(column+1,row,percent1);
            Label label3 = new Label(column+2,row,top2);
            Label label4 = new Label(column+3,row,percent2);
            Label label5 = new Label(column+4,row,top3);
            Label label6 = new Label(column+5,row,percent3);
            Label label7 = new Label(column+6,row,Timecost + "ms");
            //column,row
            copySheet.addCell(label1);
            copySheet.addCell(label2);
            copySheet.addCell(label3);
            copySheet.addCell(label4);
            copySheet.addCell(label5);
            copySheet.addCell(label6);
            copySheet.addCell(label7);
            copy.write();
            copy.close();
            column = 0;
            x = 1;
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        if(event.sensor.getType() == Sensor.TYPE_LIGHT){
            float lux = event.values[0];
            if(x == 1){
                String.valueOf(lux);
                writeExternalStorage2(lux);
            }
            else{
                //do nothing
            }

        }
    }

    public void writeExternalStorage2(float lux){
        String state;
        state = Environment.getExternalStorageState();
        if (Environment.MEDIA_MOUNTED.equals(state))
        {
            File Root = Environment.getExternalStorageDirectory();
            File Dir = new File(Environment.getExternalStorageDirectory() + File.separator + "GUI" );
            File file2 = new File(Dir,"firstexcel.xls");
            writingExcel2(file2,lux);
        }
        else
        {
            Log.i(TAG, "SD card not found");
        }
    }
```

```
    private void writingExcel2(File file2,float lux) {
        try{
            Log.i(TAG, String.valueOf(lux));
            Workbook wb = Workbook.getWorkbook(file2);
            WritableWorkbook copy = Workbook.createWorkbook((file2),wb);
            WritableSheet copySheet = copy.getSheet(0);
            Label label8 = new Label(column+7,row,String.valueOf(lux));
            //column,row
            copySheet.addCell(label8);
            copy.write();
            copy.close();
            row ++;
            column = 0;
            x = 0;
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }

}
```