SAFECROSSING:

PEDESTRIAN CROSSING ASSISTANT APP FOR VISUALLY IMPAIRED

BY

Lok Jun Leong

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Safe Crossing: Pedestrian crossing assistant app for visually impaired

Academic Session: JAN 2020

I LOK JUN LEONG

(CAPITAL LETTER)

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

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

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
10, Jalan Mutiara 3/8
Taman Mutiara Mas
81300, Skudai, Johor

Ng Hui Fuang
Supervisor's name

Date: 18 APRIL 2022

Date: 18/04/2022

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 18 APRIL 2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***LOK JUN LEONG*** (ID No: ***18ACB04199***) has completed this final year project/ dissertation/ thesis* entitled "Safe Crossing: Pedestrian crossing assistant app for visually impaired" under the supervision of ***Dr NG HUI FUANG*** (Supervisor) from the Department of Computer Science, **Faculty**/Institute* of FICT, and _____ (Co-Supervisor)* from the Department of _____, Faculty/Institute* of _____.

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

Yours truly,

(*Lok Jun Leong*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Lok Jun Leong

Date : 18 APRIL 2022

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and appreciation to my supervisor, Dr. Ng Hui Fuang, for providing me with the opportunity to study in the field of object recognition. Other than that, thanks to my supervisor, Dr. Ng Hui Fuang, for keeping me on track in the field of object recognition, and for following up on my project process and giving me advice every week. Again, a million thanks to my supervisor and moderator.

ABSTRACT

This proposal is a project of a mobile assistant app for visually impaired people. Visually impaired peoples have risk when crossing road and they are no pedestrian around to assist them. In addition, most of the traffic light in Malaysia is didn't install pedestrian signal system and for those installed is lack of maintenance. Hence this project is aims to develop an application to assists visually impaired to cross the road with safe condition. Since currently most of the people own their smartphone included visually impaired people, hence this project will fully implement in smartphone. Object detection is one of the functions will included in this project, it uses for recognize the surrounding object to assists the user cross the road. Besides, implement media as the output of application. By the experiment finding, TensorFlow is the suitable open source for create machine learning on object detection. In conclusion, this project will be using TensorFlow to train detection custom model and implement in mobile application to assist visually impaired crossing road.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

Until 2017, 253 million people were visually impaired [1]."Visually impaired" refers to a person who is partially or completely blind. Some activities may be difficult for people who are visually impaired. One of the most difficult or dangerous activities they will face is crossing the road. From the word "safe crossing" in the title, the main purpose of this project is to make sure that the visual impairer's users are in a safe condition when crossing the road. Therefore, this project is used to design and develop an application for the visually impaired and assist them so that they can cross the road safely.

Referring to one of Watanabe's surveys comparing the usability of smartphones and feature phones, the results are divided into two categories: one for people who are blind, with 43 respondents, and the other for people with low vision, with 38 respondents. In the resulting blind category result, 8 respondents believe the smartphone is more advantageous (18.6%), while 29 believe it is both advantageous and inconvenient (64.45%). On the other hand, the result of low vision categories has 17 respondents answering that smartphones were better than feature phones (44.7%), which means it can be shown that smartphones can be acceptable to most of the people who are visually impaired [2]. Hence, in this project, a good interface will be developed to help users easily use the application on a smartphone.

Figure 1.1 *Pedestrian Signal Push Button*

Figure 1.2 *Blind Crossing Track*

1.1 Problem Statement and Motivation

Most countries now provide technology on the pedestrian crossing traffic light, as in **Figure 1.1,** to assist the visually impaired. The technology is referred to as pedestrian signal. One of the most common pedestrian signals is turning the information into sound. It provides a high-pitched sound to notify the visually impaired that they can cross the road safely. Besides that, the blind crossing track in **Figure 1.2** is very useful for conveying the message to visually impaired people with the cane. However, most of the pedestrian crossings in Malaysia are not installed with those pedestrian signals, and unfortunately, most of the installed pedestrian signals also lack maintenance. Therefore, it will be more difficult for visually impaired people to recognize the road situation or surroundings. Moreover, visually impaired people need assistance when crossing a road. This is because they are unable to recognize the surroundings and road situation. Without assistance from others, it may put them in a dangerous situation.

This project aims to greatly reduce the danger of visually impaired people when they are crossing the road and to overcome the lack of electronic travel aids at pedestrian crossings. As stated currently, visually impaired people prefer to use a smartphone to assist them. Therefore, the project is being developed so that it could be an application that could assist the function of their eyes and help them recognize the road and cars so that it could greatly reduce the danger of visually impaired people when they are crossing the road and overcome the lack of electronic travel aids at pedestrian crossings.

1.2 Objectives

- **Aims to develop an application that assists the visually impaired to cross the road safely as the title "Safe Crossing"**

In order to help the visually impaired, the application recognizes the surrounding traffic situation when they cross the road. Machine learning is the system that will be used to develop this project. Furthermore, machine learning will be able to recognize pedestrian bridges, pedestrian crossings, and so on.

- **Decrease the interaction time with the application when the application is assisting visually impaired users**.

For the purpose of helping visually impaired people understand the application output, media speech will be used to interact with visually impaired people. The required input from the user will be designed to be simple, such as double tapping and long tapping.

1.3 Project Scope and Direction

The scope of the project is to develop an application that can recognize pedestrian crossings, crossing directions, and coming traffic with a smartphone camera. Therefore, visually impaired people only need to open the application on a smartphone, and the smartphone will recognize the environment automatically. Users could cross the road by following the instructions given by the application. To make a system able to recognize these things, more training data is needed in this application to analyze it. Consequently, the application can be processed and a more accurate route for the user could be generated. "We train the machine learning system before we launch it." It means the system will be able to recognize the surroundings correctly, so the user can cross the road safely even if there is no internet access.

Besides the output of the application. The smartphone flashlight would also be used to notify incoming traffic. Drivers normally face difficulty and have low vision when driving at night. With the help of the smartphone flashlight, the driver will be able to see the pedestrian. A smartphone flashlight plays an important role, especially during the night time, so that the driver is able to notice it earlier. Moreover, a sound effect will be played when the user is crossing the road, and thus it could play a role in notifying other pedestrians who are around the user. Therefore, those pedestrians could assist or make way for the user.

Every touchscreen on the smartphone is an input and output, but the touchscreen is probably just an input for the person who is visually impaired. Additionally, the application display on a smartphone screen is dynamic, and hence, the users need to rely on text-to-speech voice to recognize the applications that are currently on the screen. As a result, the application includes media voice generation to assist users in crossing the street and provides a simpler input method, such as a single tap or double tap, to perform on the application.

1.4 Contributions

By using this application, our visually impaired users are able to cross the road in safe conditions. Besides, the flashlight function plays an important role in the application. It warns oncoming traffic, especially in low-light areas or at night, to avoid an unwanted incident.

1.5 Report Organization

The following chapters build on this project: Chapter 1 is an introduction, Chapter 2 is a literature review, Chapter 3 is a system design, Chapter 4 is a system implementation, Chapter 5 is a system testing and evaluation, and Chapter 6 is a conclusion. The problem statement and motivation, project background, objectives, project scope and direction, project contribution, and report organization all included in the first chapter of this project. The second chapter is a literature review of numerous existing visually impaired assistant applications in the segment in order to assess each product's strengths and drawbacks. The third chapter is about the design of the whole project and discusses the way to achieve the objective. The fourth chapter is the implementation of the application by following the system design. The fifth chapter is testing the application with a demo and evaluation of the application after testing. Last chapter is the conclusion, which has the project review and future work.

Chapter 2

Literature Review

2.1 Review of Existing Application and System

People who are visually impaired did not give up their lives due to their shortcomings, conversely, they lived harder and cherish their life. Hence, there are a few of application and research was developed to assist them in their daily life. Three of the similar applications are taken for review.

2.1.1 SeeLight

Figure 2.1 *SeeLight Application*

The idea of this application is to assist visually impaired people by informing them about the length between the crosswalk destination and the time in seconds it will take for them to cross the road [3]. On the other hand, it also provided the correct direction for users to cross the pedestrian crossing. One of the strengths of this application is that data for streets can be retrieved from the database and calculated by using GPS. However, it is very dependent on internet access and database. For example, it will not function when used in areas or streets that don't have data in its database. Furthermore, network latency will cause issues for the application when retrieving data from its database. Last but not least, the application could not function on an unsignalized pedestrian crossing as it was not equipped with the traffic light.

The big difference between signalized and unsignalized pedestrian crossings is that the signalized pedestrian crossing is equipped with a traffic light to stop the vehicle and it is also

equipped with sound on the traffic light. The use of sound is to notify the pedestrian to cross the road and the sound frequency will change when the crossing period almost ends. This sound function brings more benefits, especially to visually impaired people. They can differentiate between the surrounding situation and the traffic light by sound. Arman claimed that a statistic shows that Malaysia's accidents in road traffic among pedestrians is rising from 24.4 percent in 2006 to 44.2 percent in 2013. He also stated that the Malaysian Institute of Road Safety Research (MIROS) recommended equipping the traffic signal on the pedestrian crossing due to the fact that most drivers in Malaysia will not slow down or stop at a non-signalized pedestrian crossing [4]. In this situation, visually impaired people will be exposed to a dangerous situation when they are crossing the unsignalized pedestrian crossings. Another problem is that the number of hybrid and electric vehicles on road is increasing. As a result, because those vehicles produce less noise or are quieter than traditional fuel-engine vehicles [5], it is a limitation of this application as well as another shortcoming that is brought to the visually impaired people.

Therefore, the project should utilize the current technologies to overcome the limitations. The applications need to recognize the pedestrian bridge, incoming traffic, and pedestrian crossing. Moreover, the application could function in both online and offline situations to avoid the latency. Furthermore, the application is able to convey the message to the driver when visually impaired people are crossing, so the driver is able to slow down earlier and so that the visually impaired user can cross the road safely.

2.1.2 Previous Application – Identify pedestrian crossing

According to the article, they are developing an application that could identify and recognize a pedestrian crossing using computer vision [6]. It directly solves the problem of visually impaired people crossing the road by directing them to the pedestrian crossing. However, the weaknesses of the application are that it cannot recognize incoming traffic. Moreover, it will face a problem when the user's living area does not provide any pedestrian crossings. This is because the application provides good assistance to visually impaired people in some areas where it provides the complete pedestrian crossing system. So, it may be a limitation in other countries or areas that do not have a good pedestrian crossing system. This system should recognize incoming traffic and recognize the crossing destination so that it is able to overcome those problems with this application.

2.1.3 Pedestrian Crossing Aid Device

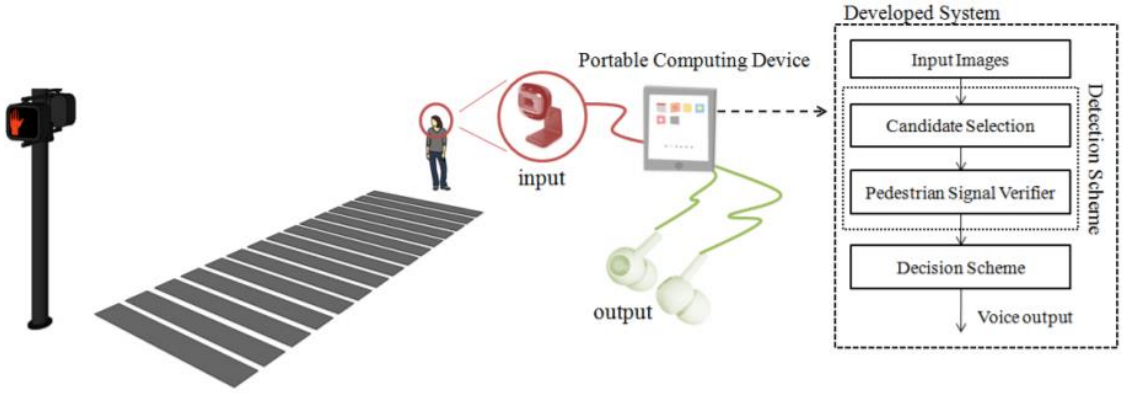


Figure 2.2 *Pedestrian crossing aid device illustration and flowchart*

According to the system, crossing aid devices use object detection to recognize the pedestrian signal as the flowchart of **Figure 2.2** [7]. The system flow is used to detect all possible pedestrian signals and send them to the pedestrian signal verifier. The pedestrian signal verifier validates existent pedestrian signals. Next, the system will process the speech output to the visually impaired pedestrian. This device can detect pedestrian signals and output speech, allowing it to guide a visually impaired pedestrian in front of a pedestrian crossing. Unfortunately, the shortcoming of the device is that it does not have a decision schema on a pedestrian crossing that does not have a pedestrian signal. The object detection method that will be used in the project that will be carried out and will be the same as the proposed idea mentioned above. Additionally, the object detection in this project will increase, which means it could detect more objects. As a result, it could overcome those problems caused by the pedestrian crossing with no pedestrian signal.

2.1.4 Supersence

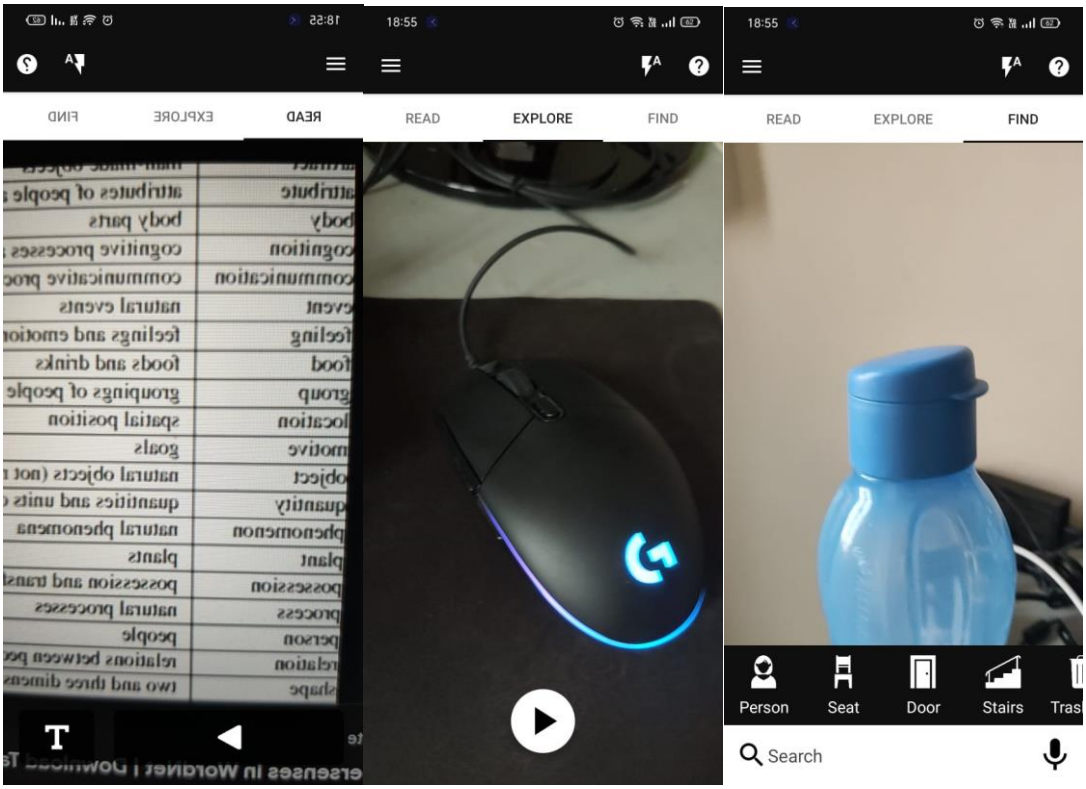


Figure 2.3 *Supersence*

This application was developed by a group of teams, and the team members are based in multiple countries. The application can be downloaded from the Google Play Store [8]. This application's vision is to allow visually impaired people who use technology to experience this new generation of experience-based applications.

Supersence is an application that assists blind and visually impaired users to read, detect the surrounding objects, and find a specific object (see **Figure 2.3)**. Supersense is using AI to train their applications to recognize objects that we interact with every day, such as seats, doors, bathrooms, and so on. Besides, this application could also scan plain text and convert it to speech output. Moreover, this application has a feature for finding objects. In this feature, users have to select the object that wanted to find, and then the application will start scanning. This feature will notify the user by vibrating when a selected item is detected.

Overall, the application is good at solving the common problem that the visually impaired face in daily life. However, the find feature in this application has an option which is find the traffic object. The application does not have any other actions after the traffic object is detected. The only actions from this application are to vibrate and notify the user by output speech. Hence, if

the visually impaired know that the traffic is around but there is no direction and no other assistance to cross the road, they will be in a dangerous situation. Nevertheless, the application is good enough for simple usage for the visually impaired.

2.2 Comparison Analysis on usage of object detection

Computer vision and image processing technology are becoming mature as time flies. Most of the review applications and systems in 2.1 have implemented computer vision. Compared to the previous system and application, gaining a thorough understanding of each system and application's flaws.

2.2.1 Usage of Object Detection

Features / Application And System	Usage of Object detection	Action after get object detection
Application - Identify pedestrian crossing	Detect the nearest pedestrian crossing	Lead visually impaired to the detected pedestrian crossing by speech after detected a pedestrian crossing.
Pedestrian crossing aid device	Detect the pedestrian signal	Validate pedestrian signal light color then lead user to cross the road when the pedestrian light is green.
Supersence	Detect the surrounding object	Convert output speech to user about the surrounding object name when detected specific object.

Table 2.1 *Comparison on usage of object detection*

2.2.2 Analysis on Table 2.1

By analysis of **Table 2.1**, those applications using computer vision technology produce different usages to assist the visually impaired. Besides, those reviewed applications are good at solving their problem statements through object detection. However, there is some shortage of action after getting output from object detection. For, Application - Identify pedestrian crossing is using object detection to find the pedestrian crossing. Hence, the shortage of this application is that no assistance is offered to the visually impaired after leading the visually impaired to a pedestrian crossing. This shortage also applied to the Supersence app. Supersence provided a feature to find a specific object like traffic, but no other action after notifying the visually impaired that traffic is in the area. In conclusion, this project will overcome those shortages. Computer vision technology will also be used. Furthermore, the action when visually impaired users cross roads will be emphasised in this project.

Chapter 3

System Design

3.1 Overall System Flow

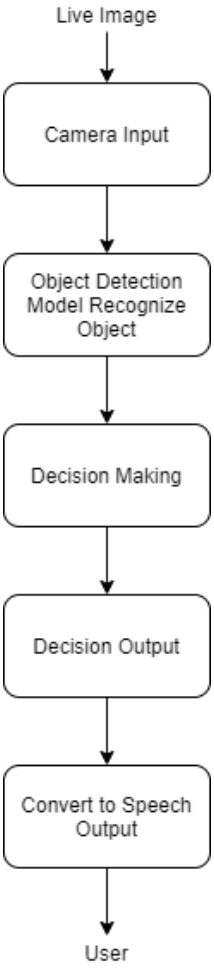


Figure 3.1 *System Flow Chart*

The overall application system design is represented by **Figure 3.1** flow chart. The application gets a live image from the device's camera. The live image will then act as the input to the object detection model to recognize the object. Next, the object detection model will generate the output by returning the result percentage of the detected object. After that, the system will make the decision by using the output from the object-detection model. The decision-making system is discussed in 3.3.2 as a decision schema. Later, the application will convert the output to speech-based. Lastly, the speech output will convey the message to the users, and this system flow will be repeated until the users finish crossing the road.

3.2 Training image detection method

Image classification and object detection are the methods under consideration in this project. Object detection has been chosen as the project detection method. The reason for choosing object detection in this project is that it requires real-time detection and it could detect more than one object in an image.

3.3 Decision Schema (Flow Diagram)

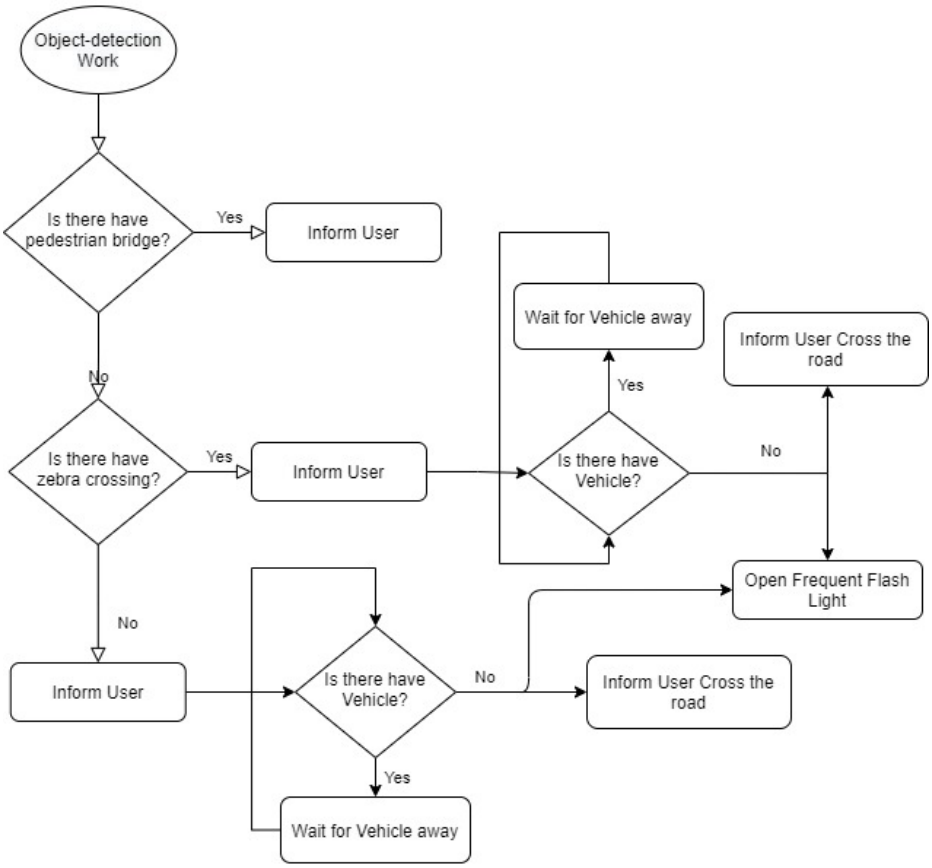


Figure 3.2 *Decision Schema flow diagram*

Figure 3.2 is a decision flowchart diagram of the application. The trained object detection model starts working when the users are ready to cross the road. In the first step, the object detection model will detect the environment around the user. If there are pedestrian bridges, the application will inform the user to cross the pedestrian bridge. Next, guide the direction of the pedestrian bridge to the user. On the other hand, if there is a pedestrian crossing, the application will also inform the user to cross at the pedestrian crossing and guide the direction of the pedestrian crossing. When the user is heading toward the direction of the pedestrian crossing, the application will start to detect the availability of the vehicles before the user

crosses the road. The application will inform the user to cross the road once the vehicle has gone away. In the meantime, the application will display a frequent flashing light to notify the incoming traffic. The conditions will be divided into two categories: those with pedestrian crossings and those without.

3.4 System Activity Diagram

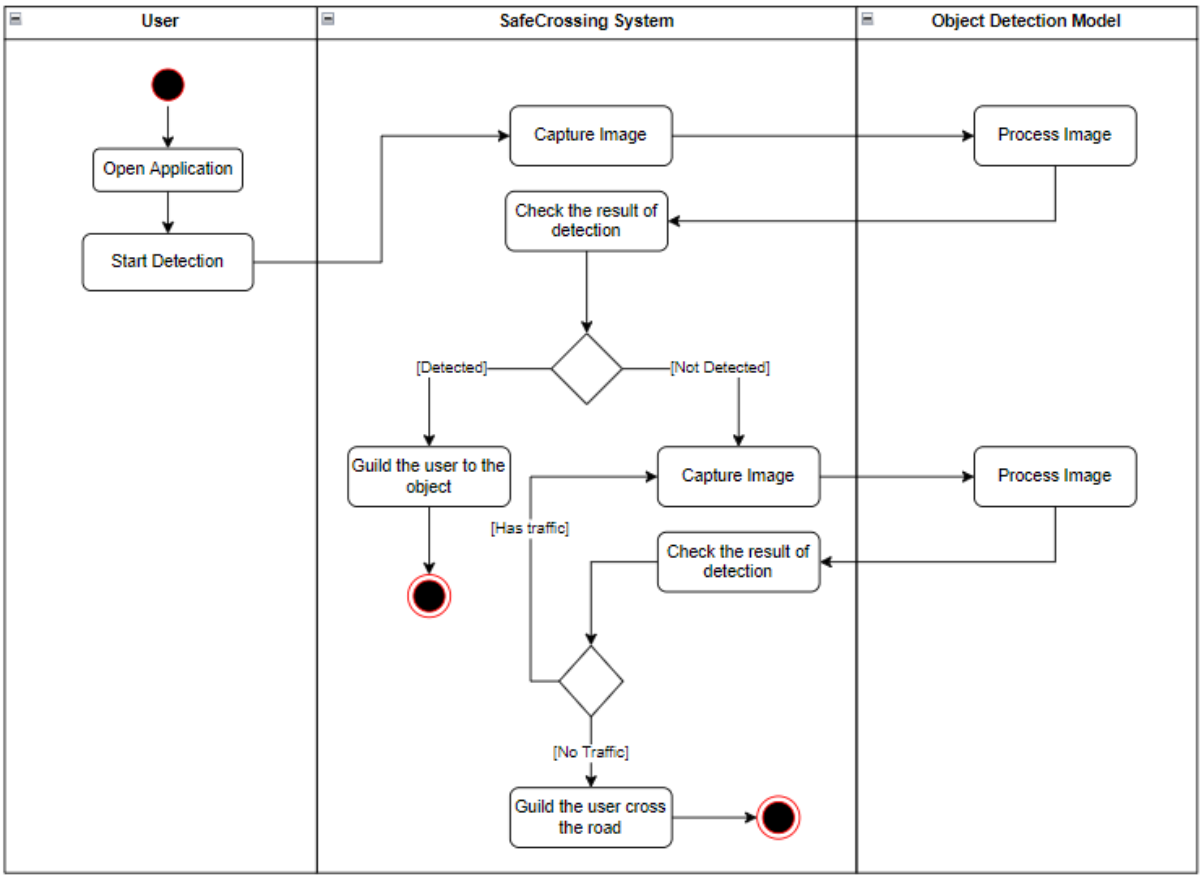


Figure 3.3 *Activity diagram*

In Figure 3.3, an activity diagram of the system is shown. There is a user, a SafeCrossing application system, and an object detection model. The activity starts when the user opens the application. The system captures the image from the device when the user starts the detection. The captured image will be processed in an object detection model. Thus, the model returns the detection result to the system. The system processes the returned result of the image and makes the decision. If detected, a pedestrian bridge or pedestrian crossing will guide the user to that object's direction. Continue capturing the image and letting the model process it if no pedestrian crossing or pedestrian bridge is detected. Moving on, checking the traffic. If the traffic is detected, it will repeat the capture image until no traffic is detected. In the following,

the system will guide the user to cross the road when there is no traffic detected, and then the activity will end.

Chapter 4

System Implementation

4.1 Methodologies and General Work Procedures

The development of this proposal will be divided into two categories. One of the categories is the training of the model for the detection. Besides that, the other category is mobile application program development. The object detection model would be used when the application is developing. Hence, this project will train the object detection model first.

4.1.1 Tools to use

Software:

1. Anaconda

   Anaconda is software that creates a virtual environment for Python and is compatible with the Windows operating system. Besides, the Python libraries that only work on Linux systems can also be run in Anaconda.

2. TensorFlow

   TensorFlow is an open-source software to create machine learning models for multiple devices and platforms such as mobile, desktop, cloud, and so on.

3. Android Studio

   Android Studio IDE is used for developing mobile applications. Object detection will be implemented in the mobile application. Android Studio IDE will be used to write a program to get the output of object detection by following the system flow. Besides, program for open torch light and text-to-speech function will be written and implemented in this tool.

4. TensorFlow Lite

TensorFlow Lite is a software library designed for devices with limited memory and able to fit on mobile GPU

Hardware:

1. Laptop

Table 3.1 Specifications of laptop

Description	Specifications
Model	Asus TUF FA506IU
Processor	AMD Ryzen 7 4800H
Operating System	Windows 10
Graphic Card	GeForce GTX 1660Ti 6GB
Memory	8GB RAM
Storage	512GB SSD

The hardware used to train the object detection model is a local laptop. TensorFlow requires a powerful CPU in order to process the image and train the model. Table 3.1 contains the specifications for training the object-detection model.

2. Smartphone

Table 3.2 Specifications of Smartphone

Description	Specifications
Model	Oppo Reno 10x Zoom
Processor	Snapdragon 855 (7 nm)
Operating System	Android 11
Main Camera	48 MP, f/1.7
Memory	8GB RAM
Storage	256GB

The smartphone used in this project is Android operating system. The smartphone must be having rear camera in order to capture the image in front of user. Besides, the processor of the smartphone is need to be powerful in aiming to process the image in background.

4.1.2 User Requirements

User Requirements:

- The user able to open the application on the smartphone.
- The user shall be at outdoor when using the application.
- The user able to hear clearly the output media from application
- The user able to moving their device toward on different direction

Non-Functional Requirements:

- The system is required to return the detection results within 2 seconds.
- The system is required to run on Android Platform.

4.2 Object Detection Model

A custom object detection model is required to be trained in this project before developing the mobile application. Object detection model is a model that able to identify the object trained in this model. This stage can be done several times in the future by increasing the training image dataset and training time. Thus, the result of the custom object detection model will be more accurate on object detection. To train a custom model, this project will follow the tutorial provided by Evan, 2018 on GitHub [9]. The tutorial is about guiding the methods for training the custom object-detection model, and the software used is TensorFlow version 1.13. An object detection API repository will be downloaded from the TensorFlow official GitHub. However, the TensorFlow object detection API is only compatible with the Linux operating system. He is guiding by using Anaconda in his tutorial. Lastly, TensorFlow will be set up and installed in the Anaconda environment.

4.2.1 Training and Test image Dataset

Figure 4.1 *Image Dataset*

The related images of traffic roads, as in **Figure 4.1,** will be collected. It is one of the junctions in Malaysia with pedestrian crossings. A total of 360 images have been collected to train the model. 80 percent of the collected images will be used for training. The other 20 percent of the collected images will be used for testing after the model is trained. Thus, 300 images will be allocated for training while 60 images are for testing. This project is planned to have 3 object classes, which are pedestrian crossing, pedestrian bridge, and a class of vehicles, which includes cars and motors.

4.2.2 Label Image Object on training dataset

Figure 4.2 *Labellmg*

```
<folder>train</folder>
<filename>TP192.PNG</filename>
<path>C:\tensorflow1\models\research\object_detection\images\train\TP192.PNG</path>
<source>
    <database>Unknown</database>
</source>
<size>
    <width>1157</width>
    <height>618</height>
    <depth>3</depth>
</size>
<segmented>0</segmented>
<object>
    <name>pedestrian crossing</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>455</xmin>
        <ymin>350</ymin>
        <xmax>931</xmax>
        <ymax>516</ymax>
    </bndbox>
</object>
```

Figure 4.3 *Output xml file*

The training and testing images have to be labelled in order to let TensorFlow understand where the object is when training the model. As shown in **Figure 4.2**, the open source software Labellmg is used to manually plot the object in the training and testing image dataset. The image can be plotted easily with this software. However, TensorFlow is still unable to understand the plotting in this step. So, this software will generate an XML file for each training and testing image that is plotted with object label information as in **Figure 4.3**. Those XML

files will be generated as a TFRecords file to serve as the input when training a model in TensorFlow.

4.2.3 Run Training on Object Detection Model with Anaconda

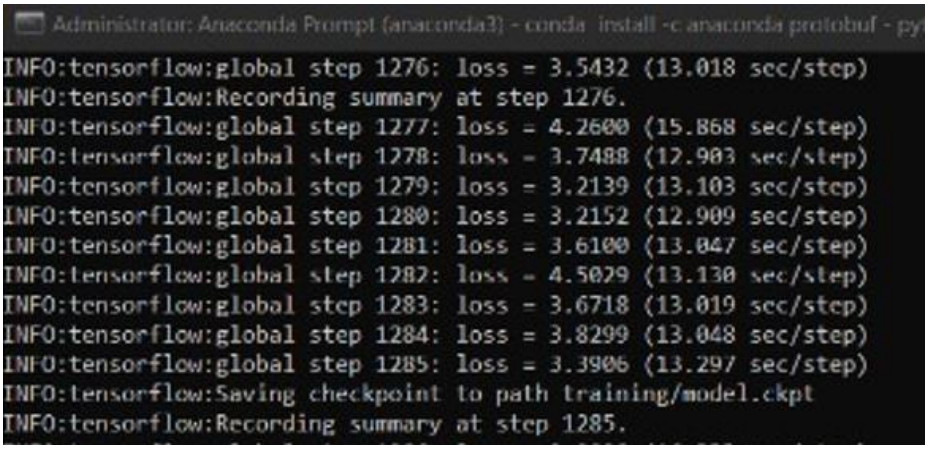


Figure 4.4 *Training Model*

The training started after all the TensorFlow environment was setup and installed in Anaconda. **Figure 4.4** is the output of Anaconda when training the custom object detection with TensorFlow. The training loss will start higher and become lower and lower as training progresses. In this project, it takes eight hours to get the lower loss. Manually stop the training when the loss becomes lower and will not drop anymore. TensorFlow will generate the last step of the train as the model.

4.2.4 Convert to TensorFlow Lite Model

The custom model will be trained successfully with the correct method and be ready to be used to detect the object. However, the model could not be implemented and run on mobile platforms. This is because the model training is only compatible with devices with large memory. As a result, the model must be converted to TFLite file format and can only be implemented in TensorFlow Lite. In order to achieve that, the model will be converted into the TensorFlow Lite method with the API provided by TensorFlow. Next, the mobile application will be developed, and the converted model will be implemented into the mobile application.

Once the object detection custom model in TFLite file format is ready, the application will load the model by coding in the Android Studio IDE. Moving on to the next, the application will then transform the input data. This is because the input data from the android camera generally does not match the raw input data for the model. Thus, resizing the input image in the function is necessary to progress. After that, the application will run the custom object

detection model. Lastly, the application will interpret the output of the model which is the object that trained in that model.

4.3 Mobile Development

4.3.1 Rebuild on TensorFlow Lite Application

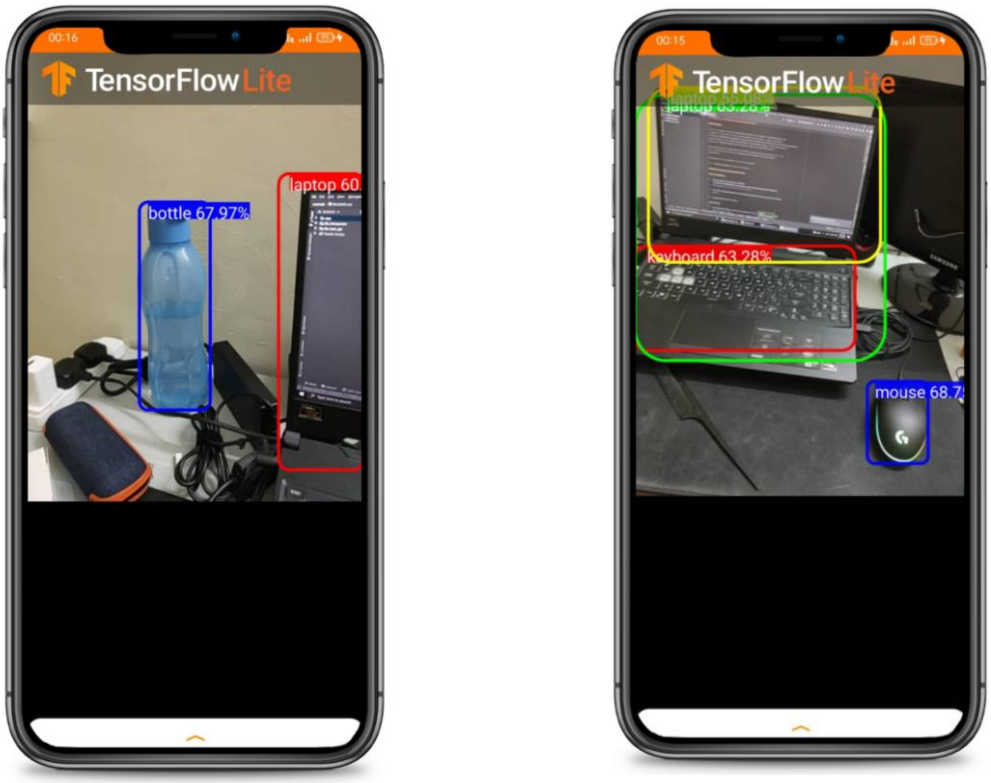


Figure 4.5 *TensorFlow Lite example application*

TensorFlow provided an example application for running the TensorFlow Lite model on Android, which is a TensorFlow Lite application. The application is able to keep detecting the object and the model keeps returning the detected result in front of the device's camera once it opens. It provides an example of a pre-trained model that can detect the things that we commonly see in our daily life, such as laptops, keyboards, tables, and so on. However, trained custom object detection model will implemented in this application. The example of the application is shown in **Figure 4.5.** The example application has the function for preview camera, process the detected image from the preview camera and a camera connection fragment to display the live camera on the application activity. Besides, the application also included request user permission for camera. This project will rebuild this application by

modifying the detection mode and changing the example model to the custom model that plan to train in this project.

4.3.2 Modify on the Detect Image Function

```
List<Detector.Recognition> return_results = detector.recognizeImage(croppedBitmap);

runOnUiThread(
        new Runnable() {...});

return return_results;
```

Figure 4.6 *Process Image*

The existing application provided a function for detection. In this project, the detection works only when the user wants to cross the road. Hence, a function was created for detection and returns a list of the results of the detection. This function is called detect_image(). This application will call this function if needed to get the result of detection from a live image. The list of results includes the label of the object and the score percentage of the detected object. Once this function is called, it will process the current image that is captured from the live image and return the object detection result to the function. **Figure 4.6** is one of the codes for the detection function. The 'List<detector.Recognition>' is the array list from the TensorFlow API and it is also the output of 'recognizeImage' function from the TensorFlow API. The list includes ID, title, confidence (a score for how good the recognition is) and location (a rectangle box results on where the object is in the image bitmap). 'CroppedBitmap' is the resized capture image from the live image in order to be read by the custom object detection model.

4.3.3 Application input and output

```
@Override
public boolean onDoubleTap(MotionEvent e) {

    if (!is_detecting) {
        Toast.makeText(context,  text: "Detection Start", Toast.LENGTH_SHORT).show();
        decision_making();
        vibe.vibrate(pattern,  repeat: -1);
        is_detecting = true;
        mp_detect_start.start();
        return true;
    }
    return true;
}

@Override
public void onLongPress(MotionEvent e) {
    if (is_detecting) {
        is_detecting = false;
        Toast.makeText(context,  text: "Detection Stop", Toast.LENGTH_SHORT).show();
        vibe.vibrate( milliseconds: 500);
        mp_detect_stop.start();
    }
}
```

Figure 4.7 *Code of Onclick Function*

The targeted user of this application is visually impaired. Hence, this application provided input and feedback to the user. The input of the application is only double click and hold click. **Figure 4.7** is the code for the click listener. A double click starts the detection. The detection is halted by a long press. Furthermore, the vibrate feedback is for the user. The device will vibrate when the user long presses or double clicks the application using the function 'getSystemService (Context.VIBRATOR_SERVICE). Besides, when the application is detecting around it will also trigger a small vibration. The user could receive their feedback when the detection starts or stops. Besides, a pre-set media sound will play when the detection stops or starts. The media speech when the user double clicks: "Detection Start Working, Please Moving Your Device Around. Long Press To Stop Detection.". In the meantime, two small vibration will be triggered. The reason for letting the user move around the device is to let the camera capture the surroundings when the detection starts. If the detection is running and the user long press the media speech, it will play: "The Detection Was Stopped." and it occur together with a long device vibration.

4.3.4 Decision Making Function

```
Timer timer = new Timer();
timer.schedule(() -> {
        if (!is_detecting) {
            return;
        }
        vibe.vibrate( milliseconds: 100);
        List<Detector.Recognition> decision_result_1 = new ArrayList<>(detect_image());
        for (final Detector.Recognition results : decision_result_1) {
            //if have pedestrian bridge
            if (results.getTitle().contains("pedestrian bridge") && results.getConfidence() >= match_scores) {
                have_pb = true;
                pb_degree_direction = degree_direction;
            }
            //if have pedestrian crossing
            if (results.getTitle().contains("pedestrian crossing") && results.getConfidence() >= match_scores) {
                pc_degree_direction = degree_direction;
                have_pc = true;
            }
        }
}, delay: 0, period: 1000);
```

Figure 4.8 *Code in decision_making()*

The decision was made when the detection started. When the detection starts, it will call the decision_making() function. **Figure 4.8** is the main code that gets the object detection result from the live image. A timer was created for repeating detecting the environment and getting the output of the detection results. The timer will repeat every 1 second. Furthermore, the function inside the timer will check if there is any pedestrian bridge or pedestrian crossing detected by getting the output from the 'detect_image()' function. If detected on a pedestrian bridge or pedestrian crossing and the confidence is more than or equal to 50 percent, the boolean will become true. Besides, the device will vibrate when this application is detected to let the user know the device is working on detecting it. Furthermore, the 'pb_degree_direction' will be assigned to get the detected direction by getting the data from the device's 'SensorManager' system. The sensor system in Android is powered by the accelerometer hardware. It is frequently used as a device compass.

A scheduled function will run after 10 seconds to check if there is any detected pedestrian bridge or pedestrian crossing. If there is a pedestrian bridge or pedestrian crossing, the detected direction will be passed to another function that leads the user to the correct direction for a pedestrian bridge or pedestrian crossing. However, it will detect surrounding incoming traffic if both of them are not detected.

4.3.5 Guiding to Correct Direction

```
do {
    if (!is_detecting) {
        return;
    }
    double direction_result = (item_degree - degree_direction + 180 + 720) % 360 - 180;

    if (direction_result < 0) {
        mp_move_to_left.start();
    } else if (direction_result > 0) {
        mp_move_to_right.start();
    }
}while (item_degree != degree_direction);
```

Figure 4.9 *Code to Guiding User Direction*

This is a function to calculate which direction the user should move when a pedestrian bridge or a pedestrian crossing is detected. The ‘direction_result’ is the value that determines whether the user should turn left or right. ‘item_degree’ is the desirable direction, ‘degree_direction’ is the current direction. Both of these values are between 0 and 360. Hence, subtracting "degree_direction’ from ‘item_degree’ and adding 180. However, it is required to do a modulus and the modulus (%) does not support on negative value. To overcome it, add 720. Next, modulus 360. After modulus, the result will be between 0 and 360. Lastly, minus 180 will get the result. **(desired_direction – current_direction +180 + 720) % 360-180** is the final calculation. If the value is positive, it indicates that it is on the right; if it is negative, it indicates that it is on the left. The do-while loop will keep looping and playing the media about moving right or moving left, and it will break the loop when the current direction matches the desired direction. If the desired direction is for pedestrian crossing, the applications will start detect surrounding traffic.

4.3.6 Detect surrounding traffic

```
Timer timer = new Timer();
timer.schedule(() → {
        vibe.vibrate( milliseconds: 100);
        if (!is_detecting) {
            timer.cancel();
            return;
        }
        List<Detector.Recognition> decision_result_1 = new ArrayList<>(detect_image());
        for (final Detector.Recognition results : decision_result_1) {
            //if have car
            if (results.getTitle().contains("car") && results.getConfidence() >= match_scores) {
                mp_detect_car.start();
                count_second = count_second + 50;
            } //if have motor
            else if (results.getTitle().contains("motor") && results.getConfidence() >= match_scores) {
                mp_detect_motor.start();
                count_second = count_second + 50;
            }else {
                count_second--;
            }
        }
        if(count_second>150){ //avoid the detection too long
            count_second = 150;
        }
        if (count_second < 1) {//when no detected any traffic
            timer.cancel();
            mp_able_crossing.start();
            while (mp_able_crossing.isPlaying()) {
                //wait the mp end
            }
            start_crossing_road();
        }
}, delay: 0, period: 1000);
```

Figure 4.10 *Code to Detect Car*

This function will be called when the user is in the direction of a pedestrian bridge or there is a situation in the user's surroundings that does not have both a pedestrian crossing and a pedestrian bridge. **Figure 4.10** is the main code of this function. This function keeps detecting the incoming traffic. If there is any car or motorcycle, the media will be there to inform the user. Moreover, there is a 'count_second' that keeps counting down, and its default initial value is 10 seconds. The application will inform users there is no traffic and instruct users to cross the road when the 'count_second' value becomes zero. Despite this, when a car or motorcycle is detected, the 'count_second' value increases by 5 seconds. Besides, there is a condition to avoid the 'count_second' more than 15 seconds.

4.3.7 Start Crossing Road

```
//blinking flashlight
timer.schedule(() -> { camera2Fragment.turnFlash( flag: true); }, delay: 0, period: 250);
timer.schedule(() -> { camera2Fragment.turnFlash( flag: false); }, delay: 125, period: 250);

//start a looping alert sound
final MediaPlayer mp_crossing_alert = MediaPlayer.create( context: this, R.raw.crossing_alert_sound);
mp_crossing_alert.setLooping(true);
mp_crossing_alert.start();
```

Figure 4.11 *Code of crossing road function*

```
public void turnFlash(boolean flag) {
    previewRequestBuilder.set(CaptureRequest.FLASH_MODE, flag ? CaptureRequest.FLASH_MODE_TORCH : CaptureRequest.FLASH_MODE_OFF);
    previewRequest = previewRequestBuilder.build();
    try {
        captureSession.setRepeatingRequest(
                previewRequest, captureCallback, backgroundHandler);
    } catch (CameraAccessException e) {
        e.printStackTrace();
    }
}
```

Figure 4.12 *Function of turn flash on and off*

This function will only be triggered when there is no traffic around the user. **Figure 4.11** shows two scheduled timers. The timers are to turn on and off the device's flash light with frequency to notify incoming traffic if there is any. The flash is along with the camera class in Android, and the camera class is in use with a live camera. The device's camera cannot be opened again because the camera is in use. Therefore, the open flash light function should be written in another Java class in order to open the flash light. **Figure 4.12** shows the function of turning the flash on and off by using 'CaptureRequest.FLASH_MODE_TORCH'. It is a function of the open torch light when the camera is previewing (live image). Additionally, a media alert sound will play and loop until the user stops the detection.

4.3.8 Auto Increase Volume

```
AudioManager audioManager = (AudioManager) getApplicationContext().getSystemService(Context.AUDIO_SERVICE);
int volume_level = audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);
audioManager.setSpeakerphoneOn(true);

//the volume raise to 10
while (volume_level <= 12) {
    audioManager.adjustStreamVolume(
            AudioManager.STREAM_MUSIC,
            AudioManager.ADJUST_RAISE,
             flags: AudioManager.FLAG_PLAY_SOUND | AudioManager.FLAG_SHOW_UI);
    volume_level = audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);
}
```

Figure 4.13 *Auto Increase Device's Volume*

The media play is the important output of this application. When the user's device volume is low, they might difficult to hear the device media output especially the application is using at outdoor. Therefore, when this application is opened, the volume will be increase to 12. The function is implemented by 'AudioManager' as **Figure 4.13.**

4.4 Application Design

4.4.1 User Interface

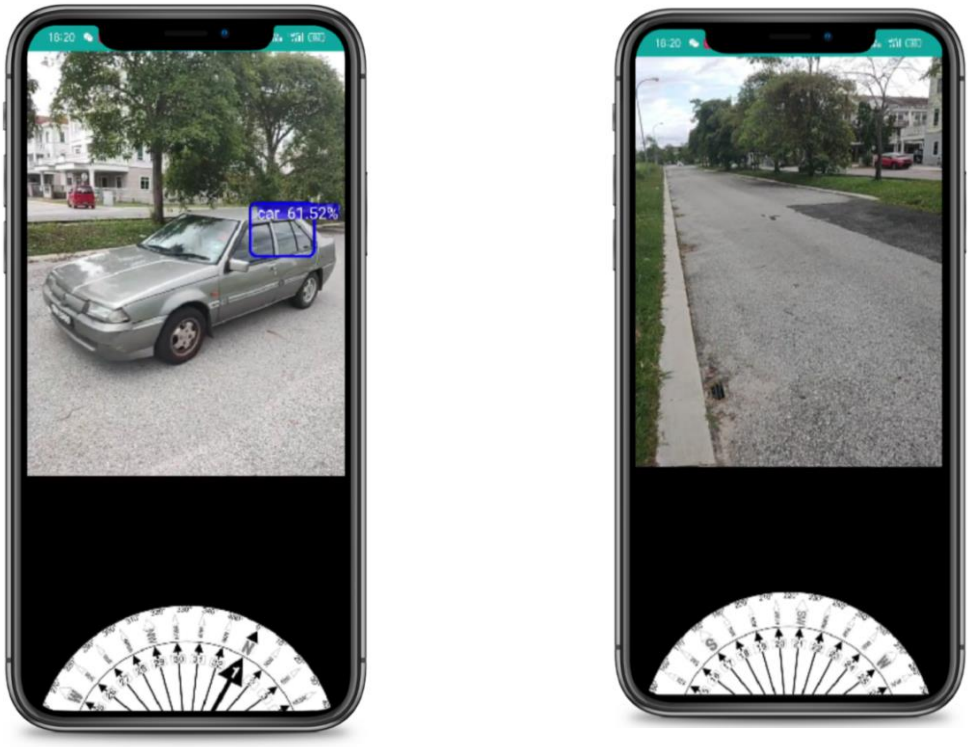


Figure 4.14 *Application Interface*

The user interface in this project was not important. The reason is that the target user of this application is visually impaired. However, only the compass and the camera preview (live image) are shown in this application. Camera preview and compass can let the other user, or a user that is only partially blind, look at the screen for a guidance.

4.4.2 Application Logo Design

Figure 4.15 *Splash Interface*

Figure 4.16 *Application Icon*

The design of the application logo is wanted to represent the application is for visually impaired using at outdoor. The logo has two people which both are visually impaired. They right hands are holding the cane and left hands is holding the smartphones. For the splash screen is using the logo and added some city background.

4.5 Implementation Issues and Challenges

The development of object detection models has faced a challenge in collecting the training images. In order to train a good model, the training image is required to get the image that is in the point of view of the user toward the pedestrian crossing or pedestrian bridge. However, it is difficult to collect a huge number of training images in a period of development time. Therefore, most of the training images are collected from Google Maps Street View. Hence, some of the collected images are not from the point of view of the user when carrying a smartphone. As a result, the object detection model for this project is not the perfect one.

Chapter 5

System Testing and Evaluation

5.1 Testing Case Without Pedestrian Crossing and Pedestrian Bridge

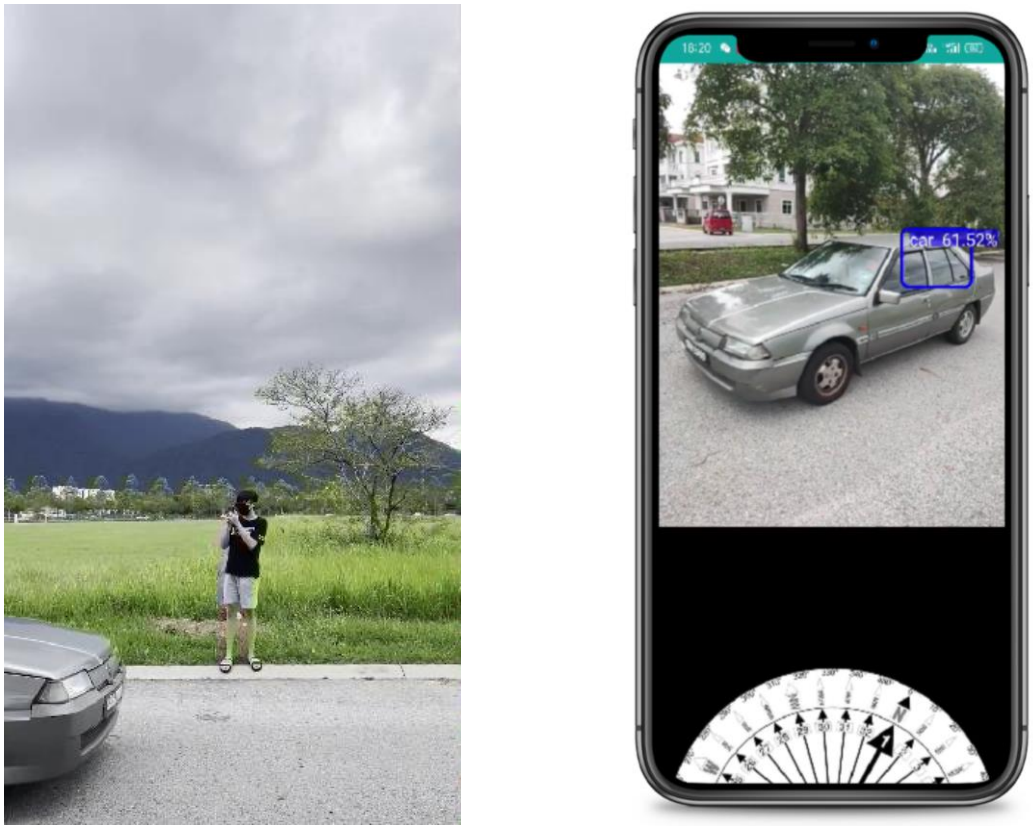


Figure 5.1 *Test Case 1*

Figure 5.1 depicts the first case of the testing, which takes place on an empty road on a day without pedestrian crossings and pedestrian bridges. There is one driver and one user involved in this case. Pedestrian crossings and pedestrian bridges were not detected by the application. Thus, the application will start detecting surrounding vehicles. When the application detects the car, it slightly delays due to the image processing time. However, the detection is still acceptable. In result, the application is able to detect the incoming traffic and avoid visually impaired explore in a danger road. It is one of the objectives of this project, assists visually impaired cross the road in safe condition.

5.2 Testing Case for Pedestrian Crossing Available

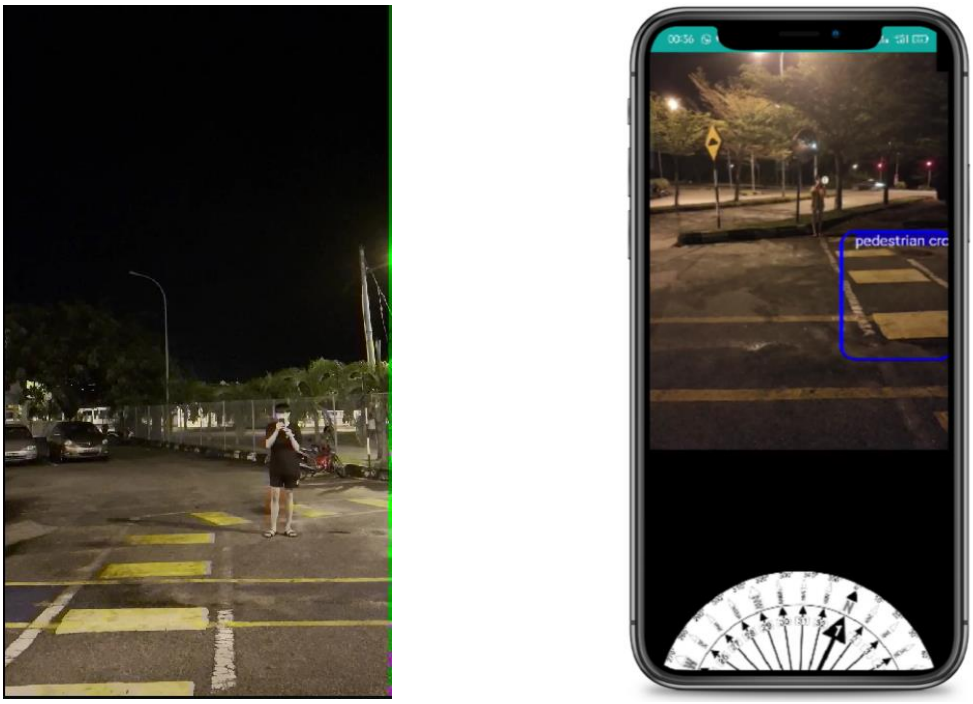


Figure 5.2 *Test Case 2 Detection*

Figure 5.2 is the test case 2. This test case 2 is a demo at night and involves a small road with a pedestrian crossing. One user will be involved in this demo. The demo is to test the situation where there is a pedestrian crossing but there is no traffic light and to test the functions of input and output of the application. The case is assuming the user wants to cross the road and doesn't know whether there is a pedestrian crossing or pedestrian bridge around him. In this test case, the user opens the application and double-clicks for start detection. Next, the application detects pedestrian crossing and leads the user to the direction of the pedestrian crossing. When the user is facing the right direction of the pedestrian crossing, the system will vibrate. In following, it detects the surrounding traffic and informs the user to cross the road when there is no traffic detected. As a result, from the demo, the detection and the compass direction were working well. Furthermore, the input can work well by double-clicking and long-pressing the device's screen. Besides, the output media of the application is audible because it automatically raises the system sound and is quiet around the user. However, users could wear earphones when in a noisy street. The only thing that is not perfect is the detection result, which might be delayed as mentioned in test case 1. Lastly, the application accomplishes the project's objective by

shortening interaction time and providing convenient interactions between the application and the user.

5.3 Testing Case for Frequent Flashlight and Alarm Media

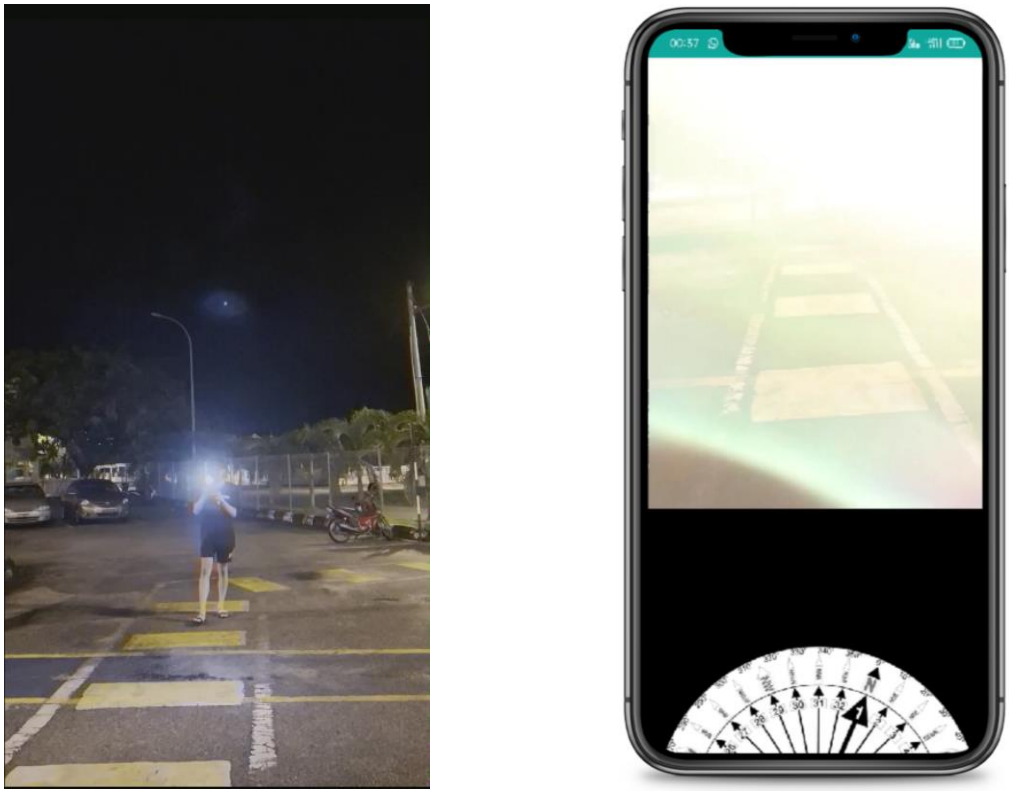


Figure 5.3 *Test Case 2 flashlight*

On the other hand, this test also tests the application's frequent flashlight and the alarm sound when the user crosses the road. From **Figure 5.3** it is shown that the flash light is able to notify others, such as incoming traffic. Besides, the alarm sound has the same function as the flashlight, which notifies others that the user is crossing a road. The objective of these functions is to decrease the danger of a user when he or she is crossing the road.

Chapter 6
Conclusion

6.1 Project Review and Discussion

The aim of the project is to let the smartphone be the eyes of the visually impaired. Hence, the major aim of the project will be object detection and the function of the application. The application is able to recognize a pedestrian crossing and detect traffic. Besides, the application implemented the sensor to improve the accuracy of the direction. Furthermore, the application is equipped with pre-trained media speech to guide the user who is visually impaired. Even though, this project is not able perfectly avoid unwanted accidents in some cases. When crossing the street, the visually impaired must still use a cane.

6.2 Contributions

This project brings out the safety assistant for the visually impaired. Besides, it increases the confidence of the visually impaired when they are outdoors. This project tries to fully use the function of the device to increase the safety of the user. The smartphone can also be useful to them.

6.3 Future Work

In the future, the accuracy of detection can be further improved by increasing the number of training images. However, it is difficult to collect the huge training images. The reason is that the point of view on the traffic road in the image might affect the final object detection model. For a good object detection model, the training image dataset has to be at the same angle as the users when they hold the smartphone. Moreover in future work, the distance of travel of users with the pedestrian crossing can be calculated with a better algorithm in order to make the application perfect for bringing a safe environment for the visually impaired when they are outdoors and crossing the road.

References

[1] P. Ackland, S. Resnikoff and R. Bourne, "World blindness and visual impairment: despite many successes, the problem is growing," *Community Eye Health*, 2017.

[2] T. Watanabe, T. Yamaguchi and K. Minatani, "Advantages and Drawbacks of Smartphones and Tablets for Visually Impaired People —— Analysis of ICT User Survey Results," *IEICE Transactions on Information and Systems*, 2015.

[3] A. Buczkowski, "SeeLight app lets you crowdsource traffic lights data to help blind people walk around the city," 28 September 2015. [Online]. Available: https://geoawesomeness.com/seelight-app-lets-you-crowdsource-traffic-lights-data-to-help-blind-people-walk-around-the-city/.

[4] A. Ahmad, "Are you one of those who ignores zebra crossings?," New Straits Times, 4 February 2018. [Online]. Available: https://www.nst.com.my/cbt/2018/02/331845/are-you-one-those-who-ignores-zebra-crossings.

[5] S. Mohd Syazwan, B. Md. Deros, H. A. Aqbal, H. Azhar and M. J. Zulhaidi, "REVISITING PEDESTRIAN CASUALTIES IN MALAYSIA AND THE ESCALATING NEW THREATS," *Malaysian Journal of Public Health Medicine*, pp. 104-110, 2018.

[6] D. Ahmetovic, C. Bernareggi and S. Mascetti, "Zebralocalizer: identification and localization of pedestrian crossings," 2011.

[7] H. Song and P. Wachirawit, "Pedestrian crossing aid device for the visually impaired," 2014.

[8] Mediate, "Supersence," 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.mediate.supersense&hl=en&gl=US.

[9] Evan, "TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10," 22 June 2019. [Online].
Available: https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 3**
Student Name & ID: Lok Jun Leong 18ACB04199	
Supervisor: Dr Ng Hui Fuang	
Project Title: SafeCrossing: Pedestrian crossing assistant app for visually impaired	

1. WORK DONE

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

20% of mobile application development and object detection model training

2. WORK TO BE DONE

Continue the mobile application development

3. PROBLEMS ENCOUNTERED

Hard to understand the existing TensorFlow application

4. SELF EVALUATION OF THE PROGRESS

The progress should be complete 40%.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 5**
Student Name & ID: Lok Jun Leong 18ACB04199	
Supervisor: Dr Ng Hui Fuang	
Project Title: SafeCrossing: Pedestrian crossing assistant app for visually impaired	

1. WORK DONE

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

50% progress of the mobile application. Which include the flow of the mobile detection and media.

2. WORK TO BE DONE

Gather more training image data for train the third custom model.

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

Mobile develop progress is going good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 7**
Student Name & ID: Lok Jun Leong 18ACB04199	
Supervisor: Dr Ng Hui Fuang	
Project Title: SafeCrossing: Pedestrian crossing assistant app for visually impaired	

1. WORK DONE

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

Collected more than 300 training image and ready to be train. The application function program on the whole flow has been done.

2. WORK TO BE DONE

Implement on the compass on the application in order to know the direction.

3. PROBLEMS ENCOUNTERED

No problem encounters

4. SELF EVALUATION OF THE PROGRESS

The progress of the application is done before week 8.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 9**
Student Name & ID: Lok Jun Leong 18ACB04199	
Supervisor: Dr Ng Hui Fuang	
Project Title: SafeCrossing: Pedestrian crossing assistant app for visually impaired	

1. WORK DONE

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

The compass is implemented and the new custom object detection model is trained

2. WORK TO BE DONE

Fixed the bug and add more media output for application

3. PROBLEMS ENCOUNTERED

The compass is implemented but the problem is the calculation for calculate the left or right of the degree direction.

4. SELF EVALUATION OF THE PROGRESS

The application can be fully completed on week 11.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 11**
Student Name & ID: Lok Jun Leong 18ACB04199	
Supervisor: Dr Ng Hui Fuang	
Project Title: SafeCrossing: Pedestrian crossing assistant app for visually impaired	

1. WORK DONE

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

The application is ready and all function is done coding.

2. WORK TO BE DONE

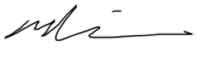
Start to testing application and shooting the application demo video.

3. PROBLEMS ENCOUNTERED

Need to find a road with pedestrian crossing and less traffic in order to take the demo video on the road.

4. SELF EVALUATION OF THE PROGRESS

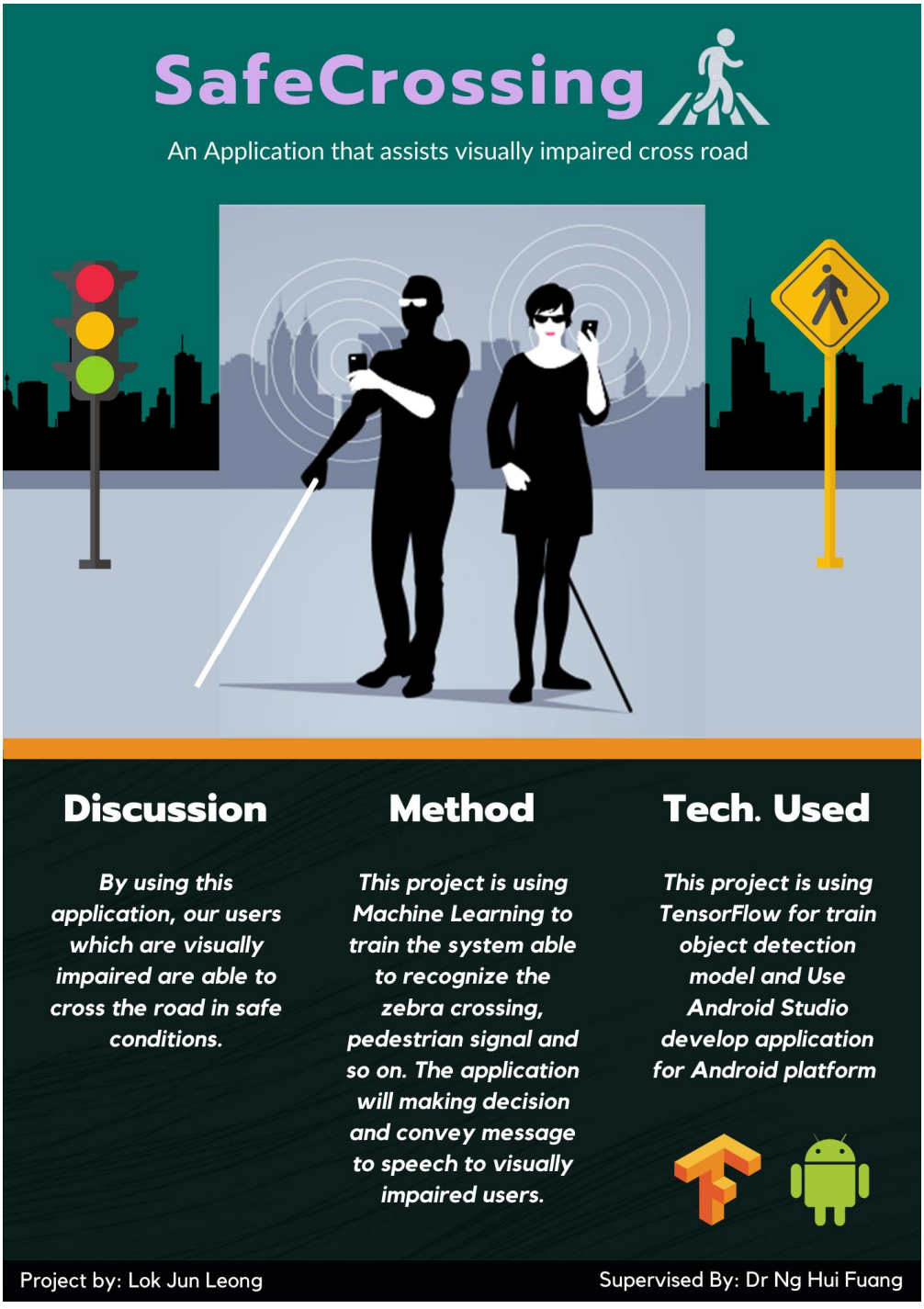
The application is complete and waiting for testing and fix bug.



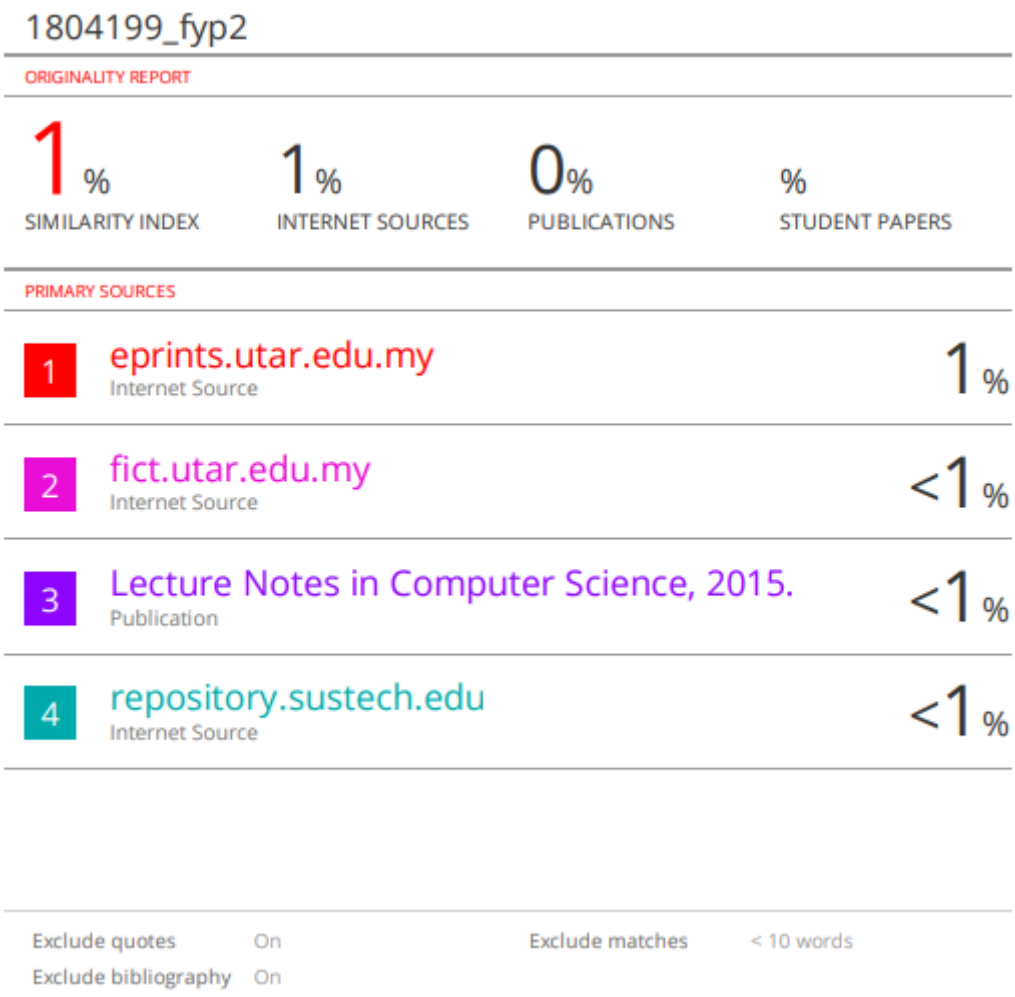
Supervisor's signature

Student's signature

POSTER


SafeCrossing
An Application that assists visually impaired cross road
Discussion
By using this application, our users which are visually impaired are able to cross the road in safe conditions.
Method
This project is using Machine Learning to train the system able to recognize the zebra crossing, pedestrian signal and so on. The application will making decision and convey message to speech to visually impaired users.
Tech. Used
This project is using TensorFlow for train object detection model and Use Android Studio develop application for Android platform
Project by: Lok Jun Leong
Supervised By: Dr Ng Hui Fuang

PLAGIARISM CHECK RESULT



1804199_fyp2

ORIGINALITY REPORT

1%	1%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	1%
2	fict.utar.edu.my Internet Source	<1%
3	Lecture Notes in Computer Science, 2015. Publication	<1%
4	repository.sustech.edu Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

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

`

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lok Jun Leong
ID Number(s)	18ACB04199
Programme / Course	Bachelor of Information Systems (Honours) Information Systems Engineering
Title of Final Year Project	SafeCrossing: Pedestrian crossing assistant app for visually impaired

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 1 % **Similarity by source** Internet Sources: 1 % Publications: 1 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

Signature of Supervisor

Name: Ng Hui Fuang

Date: 18/04/2022

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB04199
Student Name	Lok Jun Leong
Supervisor Name	Dr Ng Hui Fuang

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

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

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

(Signature of Student)
Date: 18 April 2022