Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples

BY

TEH KAI WEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _Mobile Tour Guide Application with On-site Attraction Recognition__
_for Kampar Temples_____
_____

Academic Session: _202301_____

I __TEH KAI WEN_____
(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,

_kelvin_____ _____

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

Address:
_200, Lorong Seri Koh 8,_____
_Taman Seri Koh 1, 32000____
_Sitiawan, Perak_____

Liew Soung Yue
_____
Supervisor's name

Date: _25/4/2023_____ **Date**: 25/4/2023

FACULTY OF _ Information and Communication Technology **_**

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _25/4/2023_____

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that __Teh Kai Wen_____ (ID No: _19ACB01435_) has completed this final year project entitled "_Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples_" under the supervision of _ Dr Liew Soung Yue_ (Supervisor) from the Department of _Computer and Communication Technology_, Faculty of _ Information and Communication Technology __.

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

Yours truly,

(TEH KAI WEN)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples**" 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 : _TEH KAI WEN_____

Date : _25/4/2023_____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr Liew Soung Yue who has given me this bright opportunity to engage in a mobile application development project. It is my first step to establish a career in the mobile application development field. Dr Liew has been patiently supervising and teaching me throughout the project. A million thanks to you. Next, I would like to thank my friends who have been helping me when I faced some difficulties during the project. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Kampar has many interesting tourism resources. However, tourism in Kampar is not ideal. Many tourists do not actually know Kampar can be a good place for travel. This is a big waste and regret for Kampar. Thus, how to improve tourism in Kampar is always a challenge for the related industries. Besides, most of the tourists that travel to Kampar do not know the history and relevant information of the attractions in Kampar. If they are interested to know, they need to dig around the internet to find out. The process is quite boring and long. It will cause a lot of inconveniences to the tourists. In order to solve these issues, this project aims to develop a mobile tour guide application with attractions recognition for Kampar. This application enables the users to take/upload an image of the attractions or their cultural relics and buildings. Afterward, the application will identify the image and display the history and relevant information based on the image. The attractions recognition can be achieved by machine learning. Furthermore, the application would also provide a real-time attractions detection feature to the users. Apart from using images for attractions recognition, the users shall open the camera and scan through an attraction. Then, the application might display the relevant information to the users based on the attraction scanned by them. It will enhance the on-site experience of the tourists. In addition, the application provides a list that contains the interesting attractions in Kampar. The users can scroll the list to search for the information and history of an attraction based on their demands. In short, this application can improve tourism and deliver the history and relevant information of the attractions in Kampar. As the main purpose of this project is to prove the concept of on-site and off-site attraction recognition can work and contribute to the tourism of Kampar, the main focus of the project will be narrowed down to recognise 2 temples in Kampar. They are Kampar Lu Ban Temple and Kampar Chinese Temple.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GDP	Gross domestic product
Covid-19	Coronavirus disease
UTAR	Universiti Tunku Abdul Rahman
CNN	Convolutional neural network
2D	Two-dimensional
1D	One-dimensional
SVM	Support vector machine
Open-CV	Open source computer vision library
ORB	Oriented FAST and rotated BRIEF
API	Application programming interface
RGB	Red Green Blue
ResNet	Residual neural network
QR	Quick response
Inc.	Incorporated
etc	Et cetera
SDLC	Systems development life cycle
RAD	Rapid application development
AR	Augmented reality
FYP	Final year project
SDK	Software development kit
OS	Operating system
SMART	Specific, Measurable, Achievable, Relevant, and Time-bound

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Kampar is a small town in Perak state, Malaysia. There are many interesting attractions in Kampar. Each of the attractions has its culture, heritage, and historical value. However, tourism in Kampar is not so prosperous. Most of the tourists that come to Kampar are students and parents. Consequently, their main purpose that coming to Kampar is not really want to travel around Kampar. This is such a great regret and waste for Kampar because it has many interesting tourism resources. The international tourists and the domestic tourists should really travel to Kampar. When they travel to Kampar, they will realize that Kampar is a good place for traveling. Therefore, how to promote Kampar and letting more tourists get to know about Kampar are always the problems and challenges faced by the relevant industries. In addition, the Covid-19 pandemic also has a significant impact on Kampar's tourism industry. During the Covid-19 pandemic, tourism in Kampar is badly affected. There are no tourists on the streets at all and most of the shops are facing closure issues. Nowadays, the Covid-19 pandemic has improved, but tourism in Kampar is still not ideal. Hence, the priority now is how to improve the tourism industry in Kampar.

At the same time, these 2 issues mentioned above also cause a lack of employment opportunities in Kampar. Most of the younger generations are forced to left the town to find job opportunities in big cities. Although the Universiti Tunku Abdul Rahman (UTAR) brings in a lot of businesses, it is still not enough. The job opportunities are too less when compared to the number of graduates every trimester. Moreover, this has not included the students who come out to work after high school. This situation will slow down the economy in Kampar. Thus, tourism is an important role in reviving Kampar as tourism can bring in many job opportunities. Aside from the problems mentioned above, there is also a searching information problem among the tourists. Sometimes, the tourists could see an image of an attraction online but they do not know where is it. They need to dig around the internet when they want to figure out the

answer. The process of searching will be very challenging as they do not know anything when they look at the image.

Furthermore, there is an issue that the tourists do not actually know the histories and relevant information about some of the attractions in Kampar. The tourists just come to Kampar and play around without really getting to know about Kampar. History is very important as it has recorded how past societies, systems, ideologies, governments, cultures, and technologies were built, how they operated, and how they have changed [1]. It helps to position people to see patterns that might otherwise be invisible in the present. In short, it has provided a critical perspective for understanding and solving current and future problems. However, this is not entirely the tourists' fault as there are not many efforts have been put to promote the history of Kampar. Previously, a senior developed a tourism mobile application with attractions recognition for the UTAR campus. But the scope of the attractions is only in the UTAR Kampar campus. As the scope of the attractions is limited, the publicity effect is still can be improved. The histories and relevant information of other attractions in Kampar also need to be promoted, so the tourism in Kampar is able to be improved.

The motivation of this project is to develop a mobile tour guide application with attractions recognition for Kampar. This application is aiming to improve tourism in Kampar. As mentioned above, the tourism downturn in Kampar causes a few problems. Hence, it is essential to emphasize how to improve tourism in Kampar. Tourism in Kampar and the development of Kampar are closely related. Tourism plays an important role in building a prosperous community in Kampar. In order to make Kampar better and better, tourism should be taken seriously.

1.2 Objectives

The objectives of this project will be written by following SMART criteria. SMART is an acronym that stands for Specific, Measurable, Achievable, Relevant, and Time-bound [21]. It helps to ensure that goals are well-defined, realistic, and have a clear path to success.

- **Specific:** Develop a mobile tour guide application with attractions recognition for Kampar that has three main modules

The mobile tour guide application developed in this project should have three main modules. They are the attractions catalogue module, attractions recognition module, and real-time attractions detection module. First, the attractions catalogue module enables the users to perform searching for an attraction on the attractions list provided. Next, the attractions recognition module allows the users to upload an image, and then the application shall recognise the attractions in the image. Then, the real-time attractions detection module allows the users to scan through the attractions to let the application detects the attractions on the screen.

- **Measurable:** Achieve at least 80% accuracy in the attractions recognition module and the real-time attractions detection module

The accuracy of these modules plays an important role in ensuring whether the application is feasible to be used. The application shall be ensured that it can recognise and detect most of the attractions correctly. By achieving a high level of accuracy, the mobile tour guide application will offer an effective and reliable tool for users to explore the attractions in Kampar.

- **Achievable:** The models used in the application will be developed using object detection and image classification techniques, with a focus on accuracy and speed

The models will be developed using object detection and image classification techniques, with a focus on achieving high accuracy and speed. This will require the use of appropriate tools and techniques to ensure that the models can be deployed efficiently on mobile devices. The use of TensorFlow and TensorFlow

Model Maker, which are popular deep learning frameworks, will enable the development of highly accurate and efficient models for object detection and image classification. By leveraging these tools and techniques, the desired outcomes for the application can be achieved.

- **Relevant:** The mobile tour guide application will help enhance the on-site experience of tourists, promote information and history of the attractions, and improve tourism in Kampar

The application focuses on enhancing the tourism experience in Kampar. The application aims to provide a more informative and engaging experience for tourists, promoting the history and information of the attractions. With the accurate, fast, and reliable attractions recognition features, the application will lead to a better overall experience for tourists. This objective is aligned with the goal of promoting tourism in Kampar and improving the economy of the region.

- **Time-bound:** The mobile tour guide application will be developed within the time frame of the Final Year Project period

The development of the mobile tour guide application with attractions recognition in Kampar will be completed within the time frame of the Final Year Project (FYP) period. This will ensure that the project is completed in a timely manner and can be presented as a complete solution at the end of the project period. This will also help to ensure that the project stays focused and that progress is made within the given time frame, allowing for any necessary adjustments and improvements to be made before submission.

1.3 Project Scope

This project is proposed to develop a mobile tour guide application with attractions recognition. Nowadays, there are a lot of tourism mobile applications in the market. Therefore, a novel tourism mobile application can attract more users to use it. In order to solve the problems stated in the problem statement, this application should be developed. The application developed in this project is more paying attention to attractions recognition. Attractions recognition allows the users to take/upload an image of the attraction, then the application will display the history and relevant information after identifying the image. Apart from using images for recognition, the application also allows users to use the phone camera to scan through an attraction. There are 3 main modules that need to be included in this application. They are the attractions catalogue module, the attractions recognition module, and the real-time attractions detection module.

- **Attractions catalogue module**

  In this application, there is a list provided which will contain the interesting attractions in Kampar. The attractions catalogue module allows the users to scroll through the list to search for an attraction. When the users click on an attraction, the application would navigate the users to the page of the attraction selected. On the page of the attraction, it contains the history and relevant information about the attraction. At the same time, there will also be a list on the page of the attraction that contains the meaningful corners of the attraction. By clicking a meaningful corner on the list, the application will further navigate the users to the page that contains the history and relevant information about the meaningful corner. All of the histories and relevant information provided are accurate and updated. This attraction catalogue module is helpful for users in the planning process of a trip. As in the planning process, they might not know which attractions match their preferences and demands. Thus, the list provided can enable them to explore the interesting attractions and plan their trip nicely.

- **Attractions recognition module**

  The attractions recognition module enables the users to take/upload an image of an attraction. Then, the application will identify the image and navigate the

users to the page of the attraction. It also can be known as off-site attractions recognition. The attractions recognition module has 2 functions. They are the single attraction recognition function and the multiple attractions recognition function. The main difference between them is the single attraction recognition function only can recognise 1 attraction at a time but the multiple attractions recognition function can recognise more than 1 attraction at a time. The single attraction recognition function can be achieved by image classification technique with machine learning, meanwhile, the multiple attractions recognition function can be achieved by object detection technique with machine learning.

Therefore, there are 2 types of models to be produced. To start with, a lot of the images of the attractions in Kampar need to be collected from different angles, colours, and brightness. This step is to ensure the accuracy of the models produced. Next, these images will be used to pass into the models to learn. As a result, the models are coupled with this application to perform attractions recognition. Furthermore, the attractions recognition module shall also recognize the meaningful corners of the attractions.

- **Real-time attractions detection module**

  To make the attractions recognition process more convenient and faster, the real-time attractions detection module would be included in the application. The working principle of the real-time attractions detection module is similar to the concept of the augmented reality (AR) technique. The users do not need to upload/take an image for recognition. They can just easily use the phone camera to scan through an attraction. Then, the application might detect the attraction and display the result on the screen. The attraction detected will be labelled with the name and the accuracy of detection. It can be known as on-site attractions recognition. Similar to the attractions recognition module, the real-time attractions detection module also has 2 functions. The first one is the single attraction detection function, and the second one is the multiple attractions detection function. Both of these functions are also achieved by image classification and object detection techniques with machine learning.

1.4 Impact, Significance and Contribution

This project helps to improve tourism in Kampar. To achieve this goal, the application proposed in this project should be novel and attractive to tourists. The application provides tourists with interesting on-site experiences. The tourists can have a more interactive way to get to know about Kampar. Throughout the exploration of Kampar using this application, the history and relevant information also would be propagated to the tourists. It can enrich the knowledge of the tourists. Hence, the history of the older generation will not be forgotten by people. People will always be grateful for the contributions of the older generation. This application shall enhance the experience of travelling in Kampar and indirectly improve tourism in Kampar. Besides, this project also aims to make the process of searching for an attraction in Kampar become easier. With this application, the tourists do not need to dig around the internet to research the attractions in Kampar. The process of searching will be faster and easier. In addition, this project shall enhance the project done by the senior. This project will expand the attractions scope of the senior's project and emphasize more on the on-site experience of the tourists. Therefore, the problems mentioned above can be solved. A more comfortable and better community can be attained in Kampar.

1.5 Background Information

Malaysia is a beautiful country in Southeast Asia. Malaysia is composed of 13 states and 3 federal territories. They are separated by the South China Sea into 2 regions which are West Malaysia and East Malaysia. Malaysia has various tourism resources. In Malaysia, there are many different and awesome historical places. Each of the historical places has its unique culture, heritages, foods, and historical values. Meanwhile, not only those big cities in Malaysia have historical places. For example, Melaka, Penang, Ipoh, Kuala Lumpur, etc. On the contrary, many small towns in Malaysia also have historical places. These historical places have contributed to the development of tourism in Malaysia. This is because the main purpose of most international tourists coming to Malaysia is to explore these historical places. At the same time, most domestic tourists are also interested in exploring these historical places in Malaysia. Tourism plays an important role to boost the economy in Malaysia. After manufacturing

and commodities, tourism is the third biggest contributor to the GDP of Malaysia [2]. Consequently, how to promote tourism has always been a difficult problem discussed by the Malaysian government.

In recent years, the rapid growth in the use of tourism mobile applications establishes a new way of travel in the world. Within the various categories of mobile applications, tourism application is the seventh most popular category of applications being downloaded [3]. This situation reflects the significant impact of tourism mobile applications in the travel industry. With tourism mobile applications being so universal, it is important for the tourism industry as well as mobile application developers to have a strong understanding of the mobile applications landscapes that are on the market and to find out what may be lacking [3]. As the types of tourism mobile applications are too many, tourists nowadays become very picky. Therefore, a tourism mobile application with some useful and powerful features is able to attract tourists to use it. For instance, a tourism mobile application with attraction recognition.

Attraction recognition is a feature that recognizes an attraction from the images taken/uploaded by the users and display the relevant information about it [4]. This technique can be achieved by using machine learning to perform image recognition. Image recognition with machine learning means that uses algorithms to learn the hidden knowledge from a dataset of good and bad samples [5]. This is also known as supervised learning. Instead of the coding, it is more reliant on the dataset provided. The data is first passed into an algorithm. The algorithm will try to learn from the data. Then, a model is trained with the data given. The model is the output of the algorithm that runs on data. As a result, the model will be used in a program.

1.6 Report Organization

There will be 7 chapters in the report. In Chapter 1, it will have an introduction to the project such as the problem statement, project objective, project scope, contributions, and background information. In Chapter 2, some related previous papers done by others and some existing similar applications would be reviewed. Then, the proposed system methodology shall be discussed in Chapter 3. For example, the methodologies used, tools and technologies involved, timeline, and the project overview. Furthermore, Chapter 4 will describe the system design. It should include the user requirements, application flowchart, use case diagram, use case description, and activity diagram. Besides, Chapter 5 is about the illustration of the system implementation. It mainly illustrates how to set up the environments and the processes of training models and developing the mobile application. In Chapter 6, the system evaluation shall be discussed. It includes a discussion about the testing results and performance of the models and the application. Lastly, Chapter 7 would have to make a conclusion about the project and state if there are any recommendations for the project.

Chapter 2

Literature Review

2.1 A Tourist Place Recommendation and Recognition System [6]

The objective of this research paper is to develop an application with a place recommendation system and a place recognition system. The place recommendation system will not be reviewed here. Contrarily, the place recognition system is the focusing point to be reviewed. The technology used by the authors to develop the place recognition system is Convolutional Neural Networks (CNN). CNN is a class of artificial neural networks. It is widely used for computer vision such as image and video recognition. There are 3 layers in CNN architecture which are the convolution layer, pooling layer, and output layer. Each convolution layer will be followed by a computing layer that is used to calculate the local average and the second extract. First, the authors have created a large training dataset by using the image scraping technique. Next, they initialized the weights used in the Convolutional Neural Networks. Then, they used a 2D convolution layer to create a convolution kernel that is convolved with the input layer to conduct a tensor of outputs. A pooling layer is inserted periodically in between 2 convolution layers to downsample the volume spatially, independently in each depth slice. Afterwards, the authors converted the output of the convolutional part into a 1D array by using a flattening layer. Lastly, all the nodes were connected in the dense layer to change the dimensions of the vector.

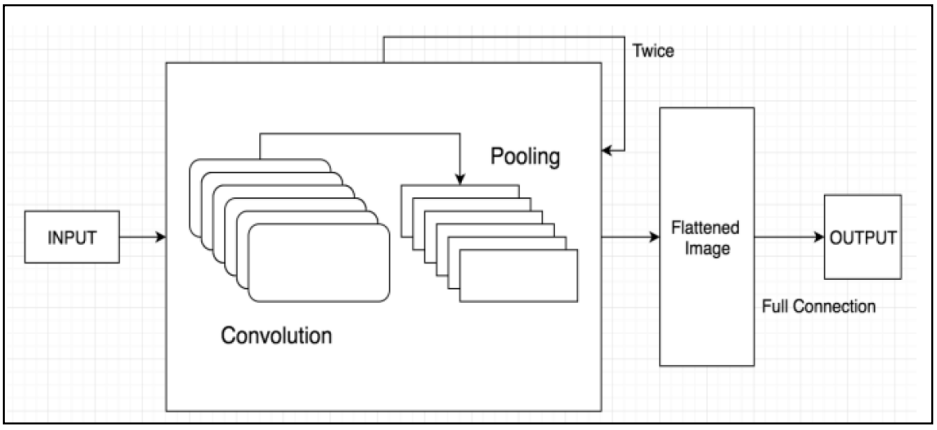


Figure 2.1 Architecture of CNN

- **Strengths of this system**

  First, convolutional neural networks model can perform efficient computation. Special convolution and pooling operations will be used in this model. Afterward, the model will perform parameter sharing. Next, this model can automatically extract the important features of the images without any human supervision. It helps to decrease the need for human effort in developing its functionalities. Thus, this model produces a good accuracy for the recognition of places. The more the images used to train this model, the higher the accuracy of places recognition will be produced.

- **Weakness and limitation of this system**

  The convolutional neural networks model requires a lot of time to be trained. It is not sufficient for normal use. Next, many training data are required to ensure the accuracy of the model. This will cause a time-consuming issue in the process of training this model. With this model, the overfitting problem might occur as it learns highly abstract features and works with fewer parameters. This situation will make the performance of the model drop off.

2.2 A scene recognition algorithm based on deep residual network [7]

The objective of this research paper is to propose a deep residual network-based scene recognition model. This model is known as Scene-RecNet. The model has consisted of 3 parts. They are the classifier, feature extractor, and feature adjustment layer. For the image feature extraction, CNN is used to find out the features of the images. The process will be repeated until the model has learned most of the features and finished the parameter adjustment. The adjustment layer shall be inserted after the original classifier is removed. The feature adjustment layer is used to further integrate and condense the extracted features. As a result, 2 kinds of outputs will be produced. They are the feature vectors and the image classification labels. The setup of the training model can be used to decide the number of feature vectors in an image. When the number of images used to train the model increase, the number of feature vectors also increase. Thus, the model can perform with higher recognition accuracy. First, the image to be recognized is passed through the model. The model will follow the feature vectors to identify the image. Next, it is classified based on the labels obtained. The final image label will be the label with the highest probability.

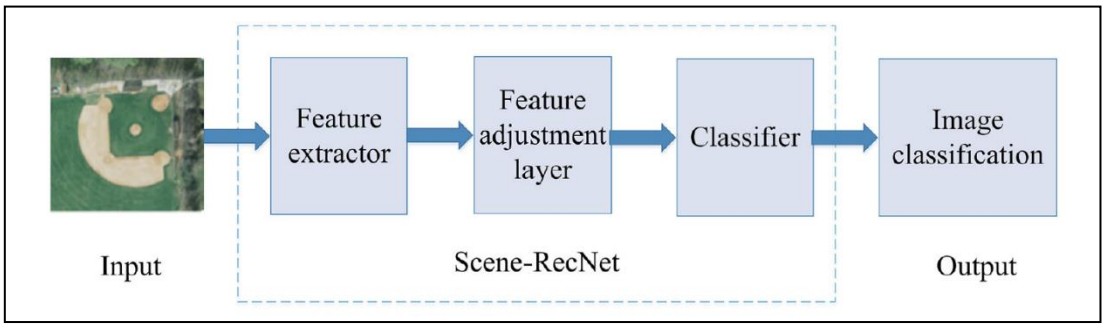


Figure 2.2 Structure of Scene-RecNet

- **Strengths of this system**

  When compared to other models, the model produced in this research paper is more efficient because it can identify the image accurately, faster and have lesser memory overhead. It is focusing on how to extract the features of images with strong representation. At the same time, it also guarantees the recognition rate although an image is condensed or distorted. This deep residual network-based model (Scene-RecNet) produced a higher accuracy of scene image recognition than the convolutional neural networks model. For feature learning, it is better than other existing convolutional neural networks models. In summary, the Scene-RecNet model can calculate faster, need small memory space, and have high recognition accuracy.

- **Weakness and limitation of this system**

  Although the deep residual network is very powerful, it also has its weakness and limitation. First, it is not suitable for normal use as it requires a lot of time to be trained. To illustrate, a deep residual network is also using a convolution neural network method that needs large memory and computation requirements, especially while training. Next, the detection of errors becomes difficult as it will increase the complexity of architecture. However, the learning might be very inefficient if the network is too shallow.

2.3 Recognition of Tourist Attractions [8]

The objective of this research paper is to recognize the specific location using machine learning methods. The CNN is used to find out the features of images and the support vector machine (SVM) is used to output the predicted attraction. The dataset used by the authors contains 10000 images and 10 attractions in Beijing, 1000 images per attraction. A Places-CNN model is first trained as a fixed feature extractor. The model is pre-trained on the 2.5 million images from 205 scene classes. The model parameters will be acquired and used as the input features for the classifiers. When the authors are examining the features extracted, they found the differences between day and night images and features of the same attraction. Therefore, they decided to train the classifiers separately for the day and night images. An additional binary classifier will be used to identify which model to use at test time. To do this, they need to compute the pixel ratio of the image by counting the total number of pixels of each value and then dividing by the total number of pixels. Then, 2 classifiers will be trained. They are the softmax classifier and SVM classifier. Softmax is a function that outputs a value between 0 and 1 that can be interpreted as the probability of a sample z from class j. Besides, the SVM classifier is used to classify the images as either day or night.

$$\sigma(z)_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$$

Figure 2.3 Softmax equation

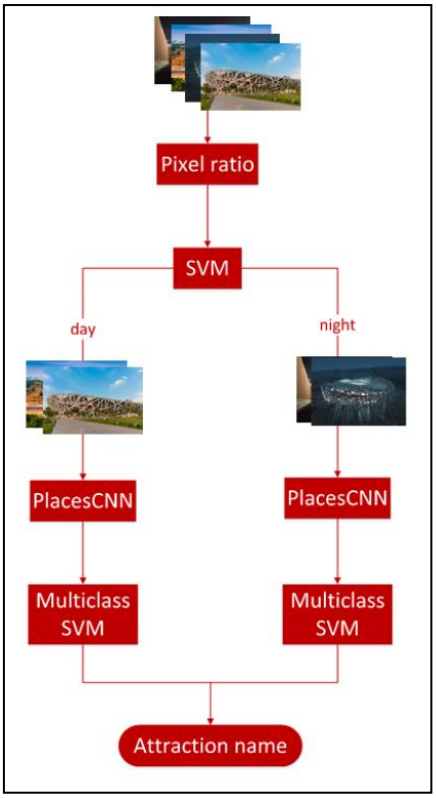


Figure 2.4 Classification pipeline

- **Strengths of this system**

  The SVM classifier produces a good accuracy of places recognition. The SVM classifier performed significantly better than the softmax classifier. As there are some overlaps in the pixel ratio features between day and night images, which would make it difficult to linearly separate them. However, the SVM classifier will seek a hyperplane that will separate 2 classes with as large a distance/margin as possible.

- **Weakness and limitation of this system**

  The features of images extracted by the Places-CNN are not the exact features required by the authors. Then, it causes difficulty for the softmax classifier to linearly separate. Places-CNN also requires a lot of time to be trained. In this research paper, the authors would like to further extract the images' features from the output of the first convolution layer if they have been given more time and then train the remaining layers.

2.4 Mobile Travel Guide using Image Recognition and GPS/Geo Tagging [4]

The objective of this research paper is to develop a travel guideline application for tourists. The main function of this application is to let tourists click/upload a picture of an attraction. Next, the application will identify the picture and provide its detailed information. This technique can be achieved by using Open-CV (Open Source Computer Vision Library) for image recognition. By using open-cv, the pictures that are in the database will be compared with the picture selected by users. Then, the histogram comparison is used along with feature detection using the ORB (Oriented FAST and Rotated BRIEF) algorithm to compare the 2 pictures. Besides, the application also provides the function of searching by name. The tourists can enter the name in the text box provided. After searching the name in the database, the result will be displayed to the users. In addition, this application has the functionality to point an attraction on the map. This function is achieved by adding the Google Maps API into the application. Similarly, a list that contains all the nearby attractions will be provided to users as the authors added the Google Places API to this application.

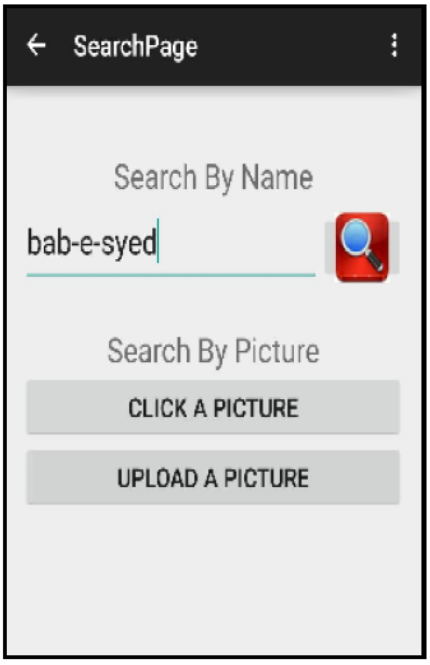


Figure 2.5 Search page

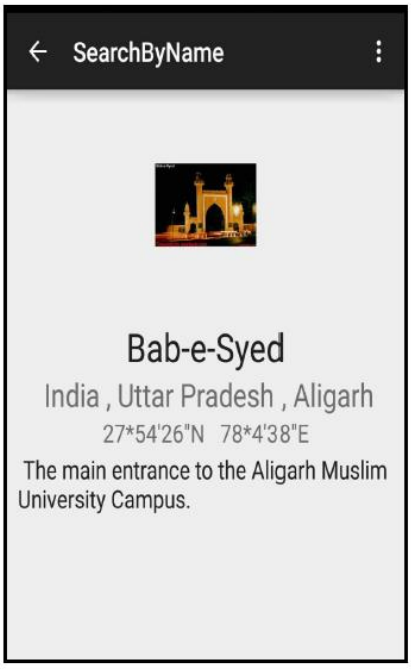


Figure 2.6 Search by name page

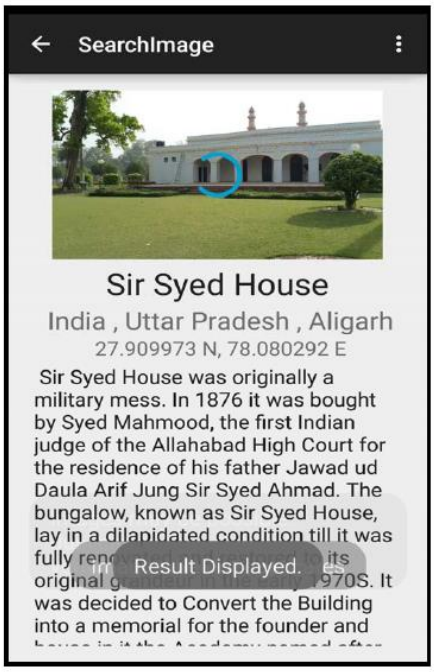


Figure 2.7 Search by image page

Figure 2.8 Map page

- **Strengths of this system**

  This application provides a few ways to deliver information related to a monument or famous spot. It aims to reduce the demand of a hired guide. Next, it also provides some useful functions that might help the tourists. For the tourists who are visiting a new attraction, this application shall be useful for them. It helps the tourists get to know well about a place and get information about nearby attractions.

- **Weakness and limitation of this system**

  The Google Maps API added to this application is meaningless. Instead of creating new functionality, the authors are just putting the Google Maps API into this application. The Google Maps API can just let users search the places. However, the regular Google map can do better than it.

2.5 Tourist Attractions Classification using ResNet [9]

The objective of this research paper is to develop a smart tourism application that can classify tourist attractions automatically by a picture. The datasets are formed by the pictures of attractions in Jakarta and Depok. They are taken from Google's search engine with different angles, lighting, and size. Next, the datasets will be pre-processed by converting them into a RGB (Red Green Blue) model. Afterward, the images are resized to 150x200 for speeding up the running time during the classification process. The k-fold cross validation method is used to train the model. The k-fold cross validation will divide the data into k sections. One of the parts will be used to do the testing and the rest are used to train the model. The process is repeated many times, with each subsample being used exactly once. As a result, the outcomes of each k will be summed and averaged. In this application, the k value used by the authors is 5. Subsequently, the ResNet50 is used by the authors as a method of classification. The ResNet50 is a residual network with 50 layers.

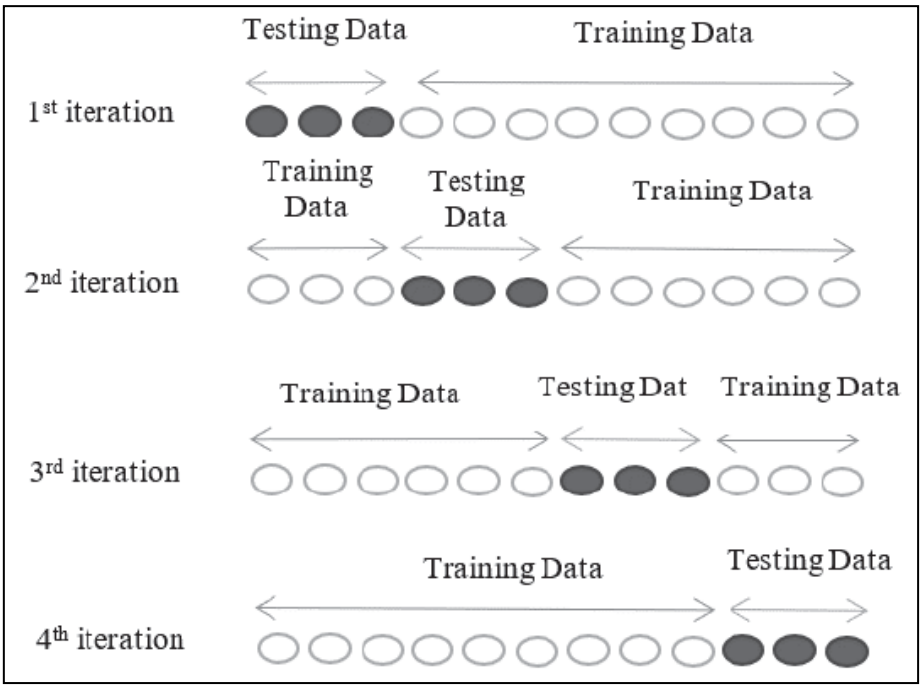


Figure 2.9 K-fold cross validation

- **Strengths of this system**

  ResNet50 runs faster at inference time as it has fewer layers when compared to other types of ResNet. ResNet50 is able to handle the research problem quite well. With the ResNet50, the accuracy of recognition is very high.

- **Weakness and limitation of this system**

  Although the accuracy is already very high, it is still can be further improved by using different types of ResNet and the k value in k-fold cross validation.

2.6 Review of Existing System

2.6.1 Google Lens

Google Lens is a recognition application developed by Google using its image recognition technology. It aims to extract relevant information related to an object. It uses visual analysis based on a neural network to identify the object. Google Lens enables users to search the object by text, barcodes, QR codes, and pictures uploaded. The users can also use the camera to search for an object. However, the users must take a photo of the object first. Then, it will display the relevant search results, web pages, and information.

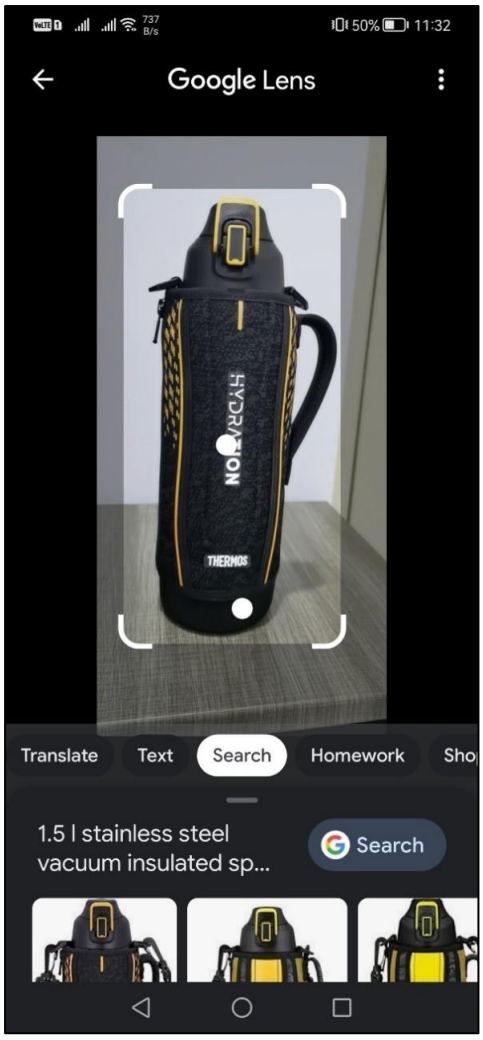


Figure 2.10 Search by camera (Google Lens)

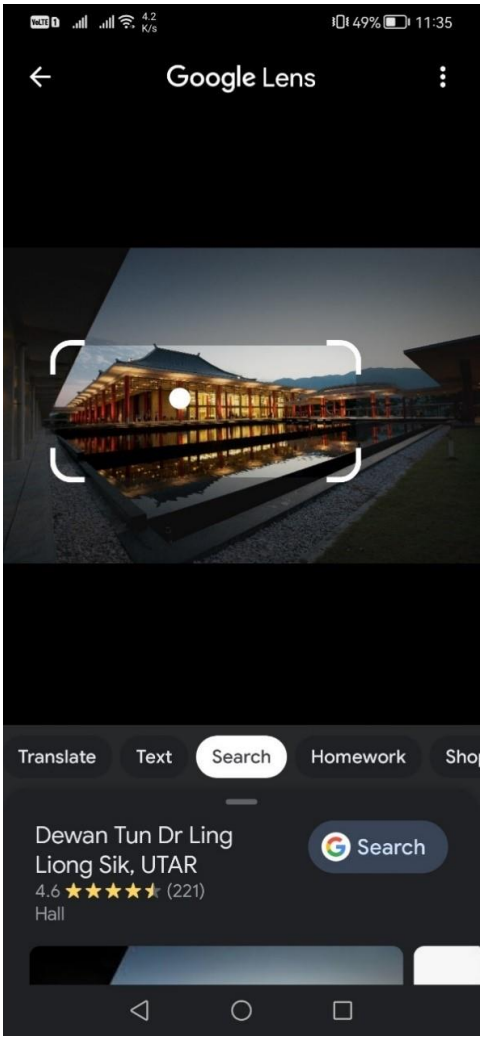


Figure 2.11 Search by picture uploaded (Google Lens)

- **Strengths of this system**

  Google Lens can identify an object accurately. The performance of the image recognition technology is very high. Next, it provides the users with multiple ways to identify an object. It is a user-friendly application.

- **Weakness and limitation of this system**

  Although Google Lens can identify an object quite well, it does not directly show the relevant information of the object. After the application identifies an object, the application only displays the object's name, a search button, and some web pages links. The users need to further click on the search button, then the application will navigate the users to the Google page with the object searched. Thus, this process will become meaningless as the users can search for the object on Google. The users still need to dig around the internet to find the information related. When the users do not know an object, the application will become useful in this situation.

2.6.2 CamFind

CamFind is a mobile application that enables users to search something just by taking/uploading an image. CamFind is an image recognition application developed by Image Search, Inc. First, the users can either take a photo using the camera or upload an image from the gallery. Then, the application will identify the image and display the result. In addition, CamFind also contains a community function that allows users to share anything in this community. The users can follow other users to uncover more on the social feeds.

Figure 2.12 Search by camera (CamFind)

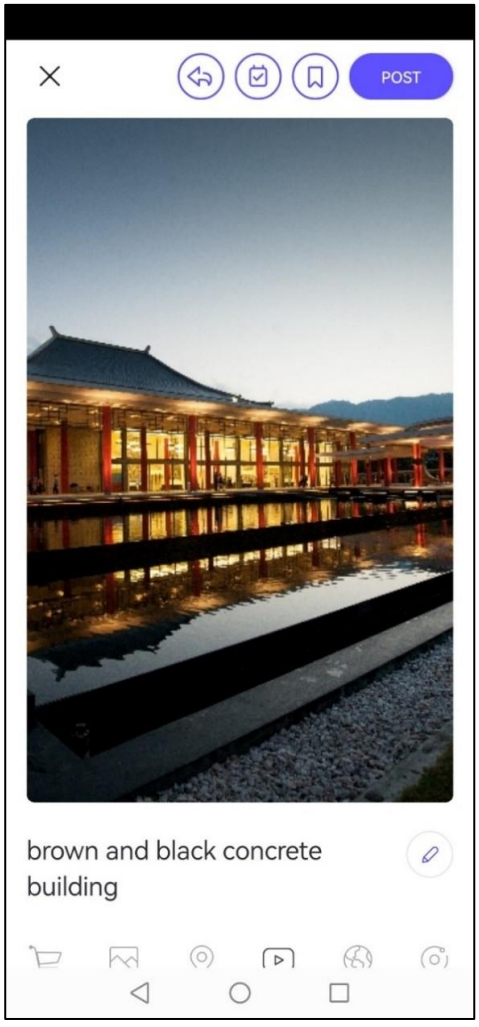


Figure 2.13 Search by picture uploaded (CamFind)

- **Strengths of this system**

  The community function in this application provides a platform to let users share their findings and communicate with other people. It helps in improving social interaction.

- **Weakness and limitation of this system**

  The accuracy of the recognition is low. The application does not identify the object correctly. Next, the application will just display the name of the object identified. It will not display any information related to the object. The performance of this application is not good.

Chapter 3

System Methodology/Approach

3.1 Methodologies and General Work Procedures

The methodology used in this project is prototyping. It is a software development methodology which is focusing on the use of working models that are constantly refined based on the feedback from the users [10]. It enables the developers to create only the prototype of the solution to demonstrate its functionality to the users [11]. By using this methodology, all of the modifications are allowed before developing the actual application. In fact, the upfront analysis is not so important in this project. As this project is non-critical, so not many careful upfront analyses are needed. When compared to other methodologies, the prototyping methodology can embrace the changes well. The developers are able to make the changes easily when they found that there is something wrong or lacking something in the application. At the same time, the developers can also discover the new requirements and try them out quickly.

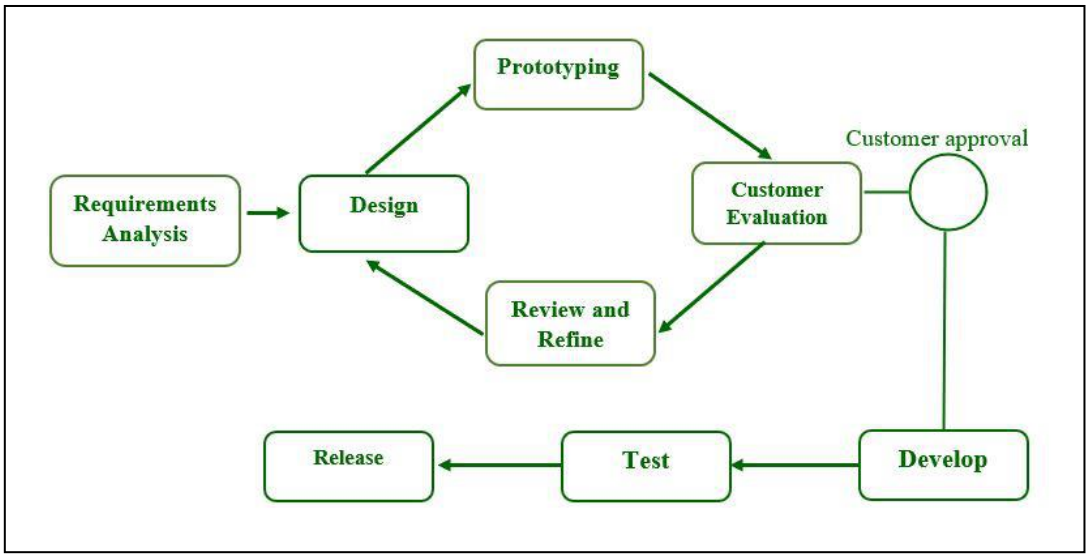


Figure 3.1 Prototyping methodology phases

The prototyping methodology has 6 Software Development Life Cycle (SDLC) phases.

i. Requirements analysis phase

Requirements analysis is the first phase in the prototyping methodology. In this phase, all of the requirements of the project need to be precisely defined. For example, the main modules and functions needed to be developed in the application.

ii. Quick design phase

Next, quick design is the second phase of this methodology. A simple design of the application will be formed in this phase. However, it is not an absolute design. It aims to provide a brief idea of the application to the users. The user interface design will be created. Hence, it might help in the development of the prototype.

iii. Prototyping phase

The third phase in this methodology is prototyping. During this phase, an actual prototype is designed based on the knowledge gathered from the quick design phase. It is a small working model of the desired application. The first version of the prototype is developed.

iv. Customer evaluation phase

Then, the fourth phase is customer evaluation. The proposed application is presented to the users (supervisor and friends) for an initial evaluation in this phase. It is beneficial to find out the performance model's strengths and weaknesses. The feedback and suggestion are gathered from the users.

v. Refining prototype phase

Afterward, the fifth phase in this methodology is refining the prototype. The refinement of the prototype might be conducted based on the feedback and suggestion gathered from users. This process will be repeated until the users are satisfied with the upgraded prototype. Consequently, a final version of the prototype is produced.

vi. Implementation and maintenance phase

The last phase in this methodology is implementation and maintenance. In this phase, the complete application is developed based on the final prototype. The application will undergo routine maintenance to reduce downtime and prevent major failures.

3.2 Tools and Technologies Involved

3.2.1 Hardware requirements

- **Laptop**

  The main device used to complete this project. Most of the processes in the project will be done on this device.

Description	**Specifications**
Model	HP Pavilion 14-ce1064TX
Processor	Intel(R) Core(TM) i7-8565U
Operating System	Windows 10
Graphic	NVIDIA GeForce MX150 2GB DDR5
Memory	16GB DDR4 RAM
Storage	1TB SSD + 1TB HDD

Table 3.1 Specifications of laptop

- **Smartphones**

  These devices were used to take the images of the attractions for the dataset to be learned and trained. They are also used to run the application prototype.

	Description				
	Model	**Processor**	**Operating System**	**Memory**	**Storage**
Specifications	Huawei Mate 30	Huawei Kirin 990	EMUI 10 (Based on Android 10)	8GB RAM	128GB ROM
	Huawei Mate 50 Pro	Snapdragon 8+ Gen 1	EMUI 13	8GB RAM	256GB ROM

Table 3.2 Specifications of smartphones

3.2.2 Software requirements

Software	**Description**
Visual Studio Code	Visual Studio Code is a well-organized code editor and supports various types of development operations. It also provides some useful extensions to increase the ease of use. When compared to Android Studio, Visual Studio Code is a lightweight IDE [12]. When running a big project, it does not cause much burden on the computer. It can perform operations faster and smoother. In this project, Visual Studio Code will be the main software used to develop the application.
Android Studio	Android Studio is a piece of software that helps users to build an Android-based application. It supports various programming languages such as Java, Kotlin, C++, etc. Android Studio provides the ability to create an Android emulator that enables users to run their applications virtually. The users can create various types of virtual devices to test the

	applications. In this project, Android Studio will be used to create the Android emulator that runs the application.
Jupyter Notebook	Jupyter Notebook is a server-client software that permits the users to edit and run the notebook documents via a web browser. It can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet. This is the main software to perform machine learning for the project.
LabelImg	LabelImg is a software used for image annotation and labelling. Specifically, this is a graphical image annotation tool that is commonly used in computer vision and machine learning [22]. This tool allows users to annotate images by drawing bounding boxes around objects of interest and labelling these objects with tags. These labelled images are then used to train machine learning models for tasks such as object detection and image recognition.

Table 3.3 Software requirements for the project

3.2.3 Software development kit used

Programming language	**Description**
Flutter	Flutter is a software development kit (SDK) mainly used for user-interface design. It is open-source and supports application development for cross-platform. Flutter is using the Dart language for its main programming language. Unlike other SDKs, Flutter uses various widgets for application development [13]. Flutter is simple and has high performance when compared to others. It makes the application development process faster, more productive, and more

	flexible. Flutter will be used for developing the application in this project.

Table 3.4 Software development kit used in the project

3.2.4 Programming languages used

Programming language	**Description**
Dart	Dart is an object-oriented programming language that is mainly used for client development, especially mobile applications. It is specialized for the demands of user interface development [14]. Dart enables the users to build the application faster on any platform. It is easier to use and understandable. With the basic concept of the Dart language, the users are able to begin the development. In this project, Dart is the main programming language used in the Visual Studio Code to develop the application.
Python	Python is a high-level programming language with dynamic semantics. It is stable, flexible, and simple. Hence, it is the perfect programming language for various machine learning and artificial intelligence projects. There are many Python machine learning libraries and packages available. It is also very attractive for Rapid Application Development (RAD) because it has high-level built-in data structures, combined with dynamic typing and dynamic binding. Python will be the main programming language used in the Jupyter Notebook. It is used to perform machine learning in this project.

Table 3.5 Programming languages used in the project

3.2.5 Libraries used

Library	Description
TensorFlow	TensorFlow is an open-source software library used to perform machine learning and artificial intelligence projects. It is mainly focusing on the training and inference of machine learning. It aims to simplify the process of developing and executing advanced analytics applications for users. For this reason, TensorFlow will be the Python machine learning library used in this project.
TensorFlow Lite Model Maker	TensorFlow Lite Model Maker is a lightweight version of TensorFlow. It is designed especially for mobile devices and embedded devices. It would simplify the process of training a model. TensorFlow Lite Model Maker applies transfer learning which can help in reducing the amount of dataset needed and faster the training time. TensorFlow Lite Model Maker will also be used in the Python machine learning of the project.

Table 3.6 Libraries used in the project

3.3 Timeline

Gantt chart for FYP1

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Revise IIPSPW Report														
Define scope and requirements														
Researching for relevant information														
Gathering dataset														
Training model with TensorFlow														
Coding the application														
Refinement and testing														
Report writting														
Presentation														

Figure 3.2 Timeline for FYP1

Gantt chart for FYP2

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Revise FYP1 Report														
Identify new requirements														
Researching for relevant information														
Gathering dataset														
Retraining the model with TensorFlow														
Coding the application														
Refinement and testing														
Report writting														
Presentation														

Figure 3.3 Timeline for FYP2

3.4 Project Overview

Figure 3.4 has showed the overview of this project. In this project, there are 2 models needed to be produced. They are the image classification model and the object detection model. The first stage of this project is to gather the dataset. This project will use an image dataset and the images are taken using the phone camera. In order to prepare the dataset of the image classification model, the images taken are classified into different classes and stored in different folders. However, for the dataset of the object detection model, the images are needed to be annotated with the labels by using LabelImg. Afterwards, the dataset is ready to be used. The second stage in this project is to train the models. TensorFlow library will be used to train the models in the Jupyter Notebook. The dataset should be split into a training set and a testing set. The training set is used to train the models while the testing set is used to evaluate the models. As a result, the 2 models will be produced and they are stored separately in 2 .tflite files. Subsequently, the application would be developed in Visual Studio Code using Flutter. As Flutter provides a TensorFlow model package, the models stored in the .tflite files would be imported into the application. By using the predefined function provided, the models can be called and return the recognition results to the application.

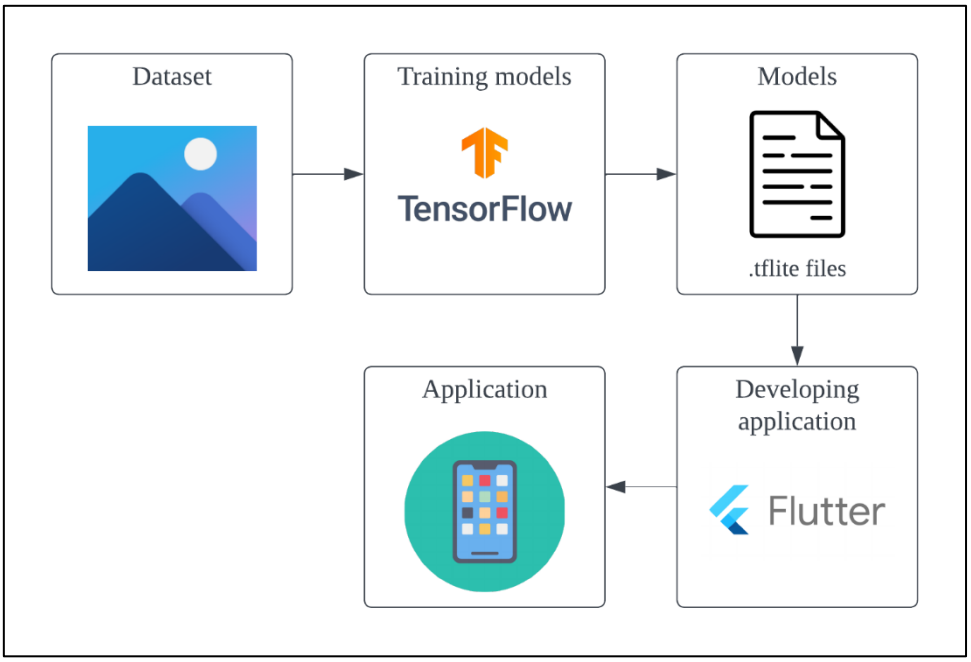


Figure 3.4 Project overview

Chapter 4

System Design

4.1 User Requirements

4.1.1 Functional requirements

- As a user, I can upload an image from the gallery to the application.
- As a user, I can take a photo using the phone camera through the application.
- As a user, I can view the recognition results after uploading the image to the application.
- As a user, I can open the phone camera to scan through the attractions.
- As a user, I can view the detection results when scanning through the attractions.
- As a user, I can view the list which contains various attractions.
- As a user, I can view detailed information about an attraction.
- As a user, I can view detailed information about a meaningful corner in a specific attraction.

4.1.2 Non-functional requirements

- As a user, I can use the application on any Android phone.
- As a user, I can use the application without occurring failure.
- As a user, I can view the recognition and detection results within 3 seconds.
- As a user, I can easily use the application without much learning about it.

4.2 Application Flowchart

Figure 4.1, Figure 4.2 and Figure 4.3 have showed the flowchart of the application. Figure 4.1 mainly shows the flow of the attractions catalogue module. When the users launch the application, the application would first display the attractions list page to the users. If the users click on an attraction on the list, the application would navigate users to the attraction detail page. On the attraction detail page, it should display the detailed information and history about the attraction. There will also have a list that contains the meaningful corners of the attraction. When the users further click on a meaningful corner of the attraction, the application should navigate them to the meaningful corner detail page. On the attraction detail page and the meaningful corner detail page, there is a back button on the app bar that is used to back to the previous page.

Next, Figure 4.2 is mainly showing the flow of the attractions recognition module. This module has 2 functions, the single attraction recognition function and the multiple attractions recognition function. The users can navigate to the recognition page to perform the single attraction recognition function, meanwhile, the users can select the "Image Detection" on the popup menu to perform multiple attractions recognition function. Both of the functions will need the users to select a method to upload an image to perform the recognition. There are 2 methods that enable users to upload an image. The first one is selecting an image from the gallery, and the second one is taking a photo using the phone camera. After the users upload the image, the image should be passed to the models to perform recognition.

Both of the functions will use different models, the image classification model is used for the single attraction recognition function and the object detection model is used for the multiple attractions recognition function. The single attraction recognition function can only recognise an attraction based on the image uploaded, so the application will display only 1 recognition result and then navigate the users to the page of the recognised attraction. However, the multiple attractions recognition function can recognise multiple attractions in the image, so the application will use multiple bounding boxes to point out the recognised attractions. Then, the users should be navigated to the page of the attraction after they click one of the bounding boxes. The application shall check on the result and navigate users to the relevant page. If the

recognition result is an attraction, the application navigates users to the attraction detail page. Otherwise, the recognition result is a meaningful corner of an attraction. The application navigates users to the meaningful corner detail page.

Furthermore, Figure 4.3 is mainly showing the flow of the real-time attractions detection module. The working principle of this module is similar to the attractions recognition module. It also has 2 functions, the real-time single attraction detection function and the real-time multiple attractions detection function. The real-time single attraction detection function will use the image classification model and the real-time multiple attractions detection function will use the object detection model. First, the users should navigate to the live detection page. There will have 2 buttons displayed on the page. The "Single Attraction" button represents the real-time single attraction detection function while the "Multiple Attractions" button represents the real-time multiple attractions detection function. After the users click on the button, the application navigates the users to the relevant page to perform detection.

Both functions allow the users to scan through the attractions using the phone camera. The application would send every frame to the relevant model to perform detection. For the real-time single attraction detection function, the application would display a bounding box to point out the detected attraction and a navigation button on the screen after the image classification model detects an attraction. The users can click on the navigation button to navigate to the relevant page of the attraction. For the real-time multiple attractions detection function, the application would display multiple bounding boxes to point out the detected attractions on the screen after the object detection model detects the attractions. The users shall click one of the bounding boxes to navigate to the relevant page of the attraction.

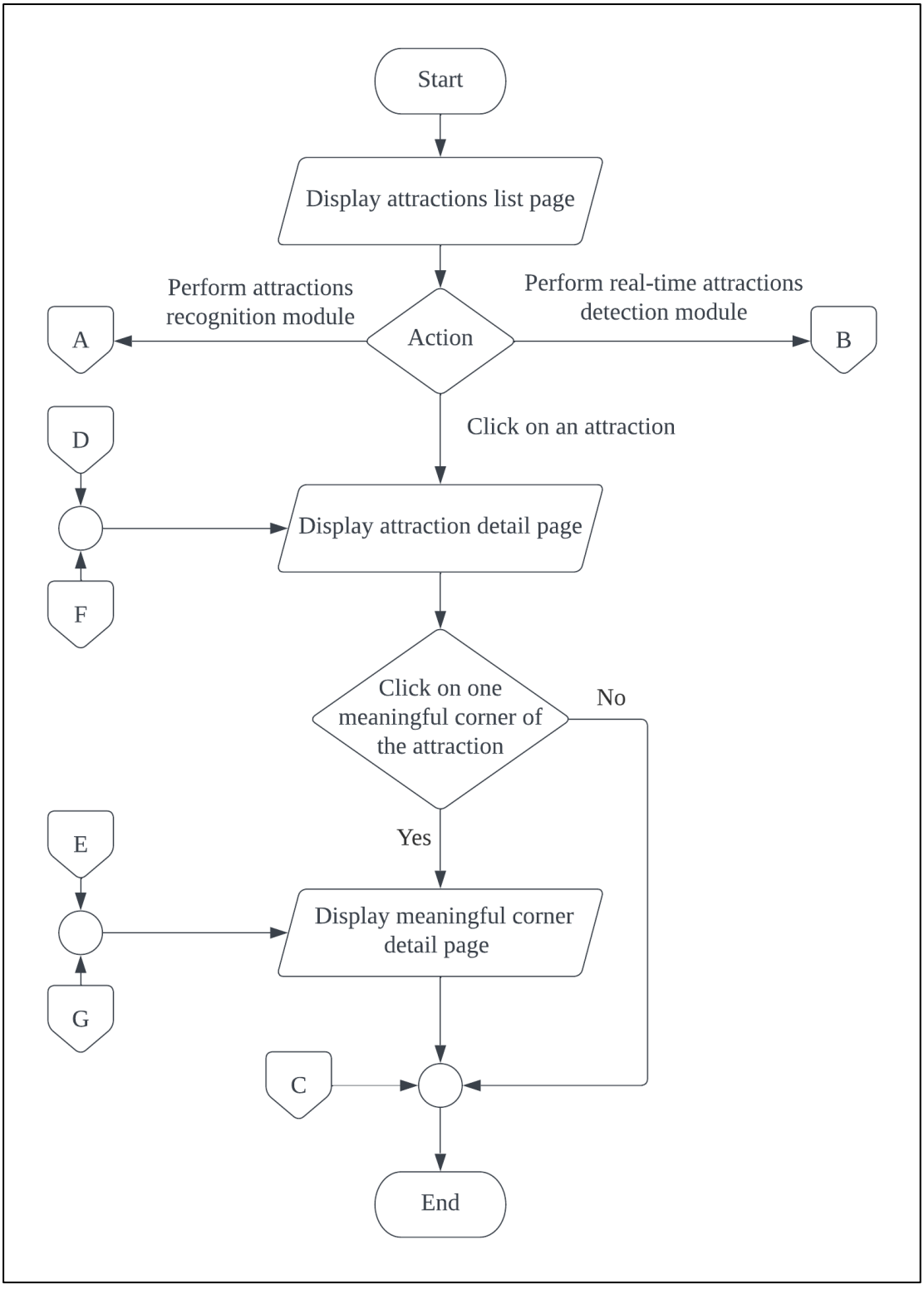


Figure 4.1 Flowchart of Attractions Catalogue Module

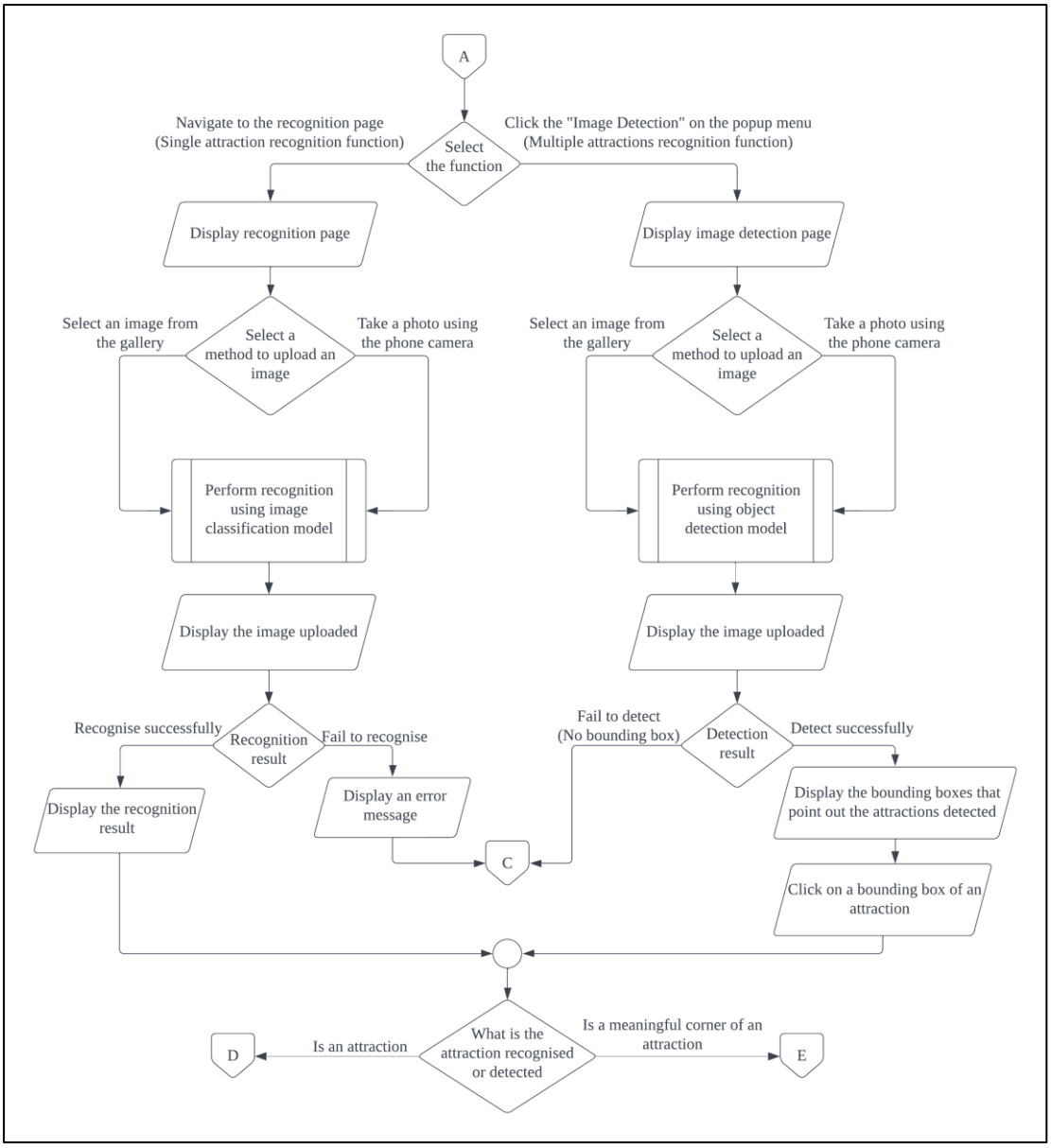


Figure 4.2 Flowchart of Attractions Recognition Module

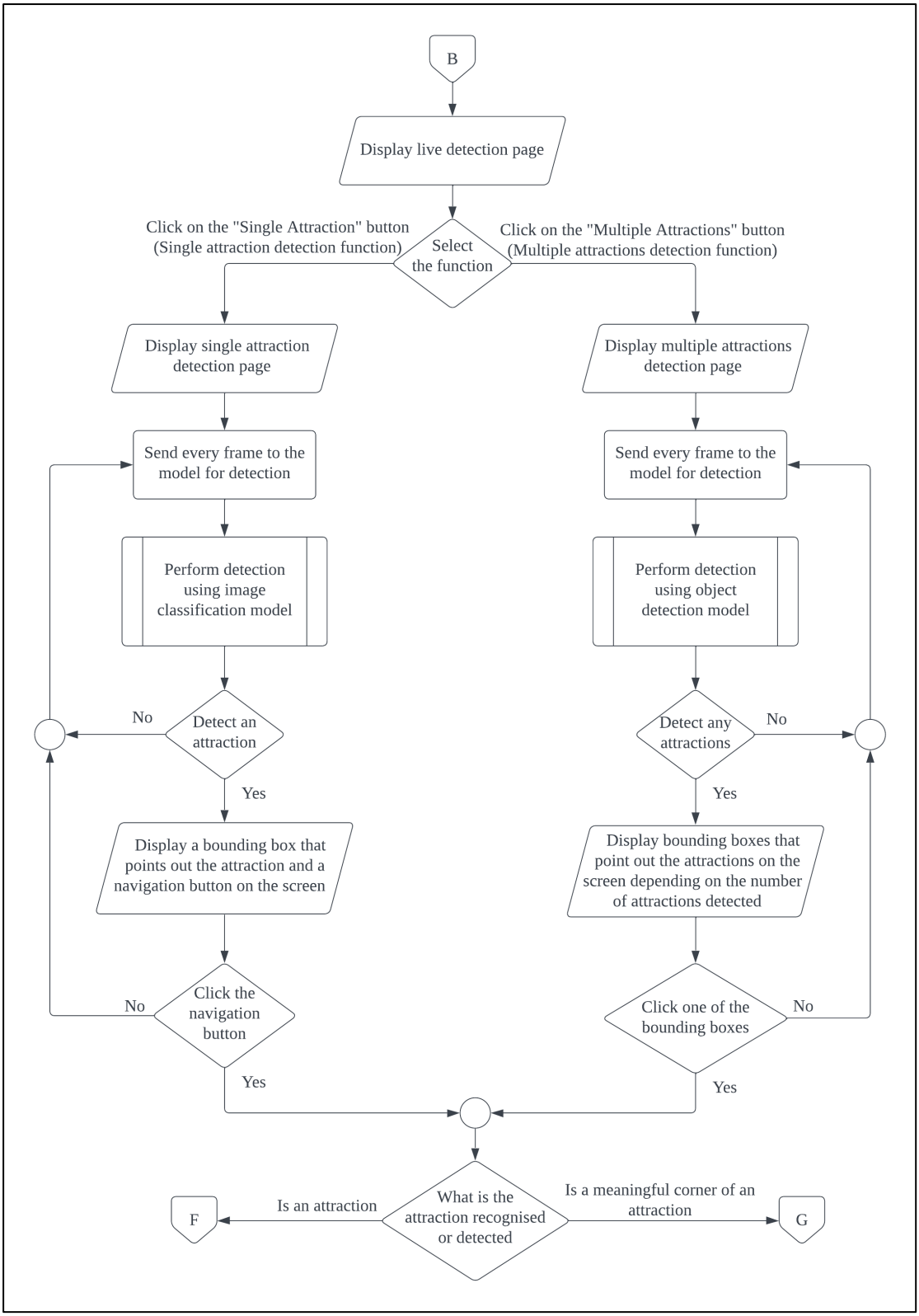


Figure 4.3 Flowchart of Real-time Attractions Detection Module

4.3 Use Case Diagram

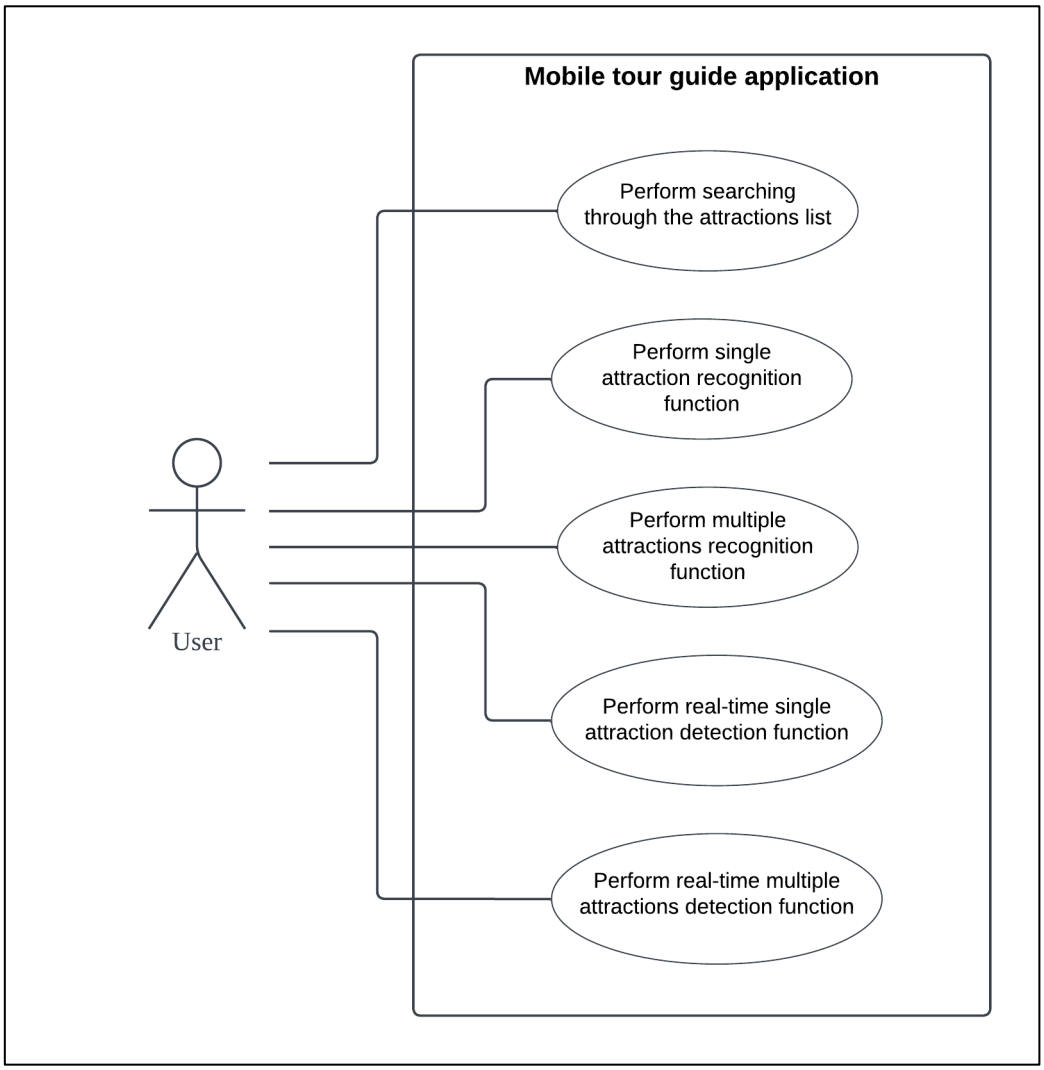


Figure 4.4 Use Case Diagram

4.4 Use Case Description

4.4.1 Use Case Description for Perform searching through the attractions list

Use Case ID	001	**Version**	1.0
Use Case	Perform searching through the attractions list		
Purpose	Allow user to search the information and history of the attractions through the attraction list.		
Actor	User		
Trigger	The user launches the application.		
Precondition	-		
Scenario Name	**Step**	**Action**	
Main Flow	1	The user launches the application.	
	2	The system displays the attractions list page.	
	3	The user scrolls through the list and clicks on an attraction.	
	4	The system displays the attraction detail page.	
	5	The user views the information and history of the attraction selected.	
	6	The user clicks on a meaningful corner of the attraction through the meaningful corners list provided.	
	7	The system displays the meaning corner detail page.	
	8	The user views the information and history of the meaning corner of the attraction selected.	
Rules	-		

Author	Teh Kai Wen

Table 4.1 Use Case Description for Perform searching through the attractions list

4.4.2 Use Case Description for Perform single attraction recognition function

Use Case ID	002	**Version**	1.0
Use Case	Perform single attraction recognition function		
Purpose	Enable the system to recognise the attraction in the image uploaded by the user.		
Actor	User		
Trigger	The user navigates to the recognition page.		
Precondition	-		
Scenario Name	**Step**	**Action**	
Main Flow	1	The user navigates to the recognition page.	
	2	The system displays the recognition page.	
	3	The user clicks on the floating button on the bottom right corner of the page.	
	4	The system displays 2 methods to allow users to upload an image.	
	5	The user selects the method to upload an image.	
	6	The system passes the image to the image classification model to perform recognition.	

	7	The system displays the image uploaded and the recognition result on the page.
	8	The system navigates the user to the page of the recognised attraction.
Sub Flow – Select an image from the gallery	5a.1	The user selects the method of selecting an image from the gallery.
	5a.2	The system opens the gallery of the phone.
	5a.3	The user selects an image from the gallery.
	5a.4	The system continues Main Flow Step 6.
Sub Flow – Take a photo using the phone camera	5b.1	The user selects the method of taking a photo using the phone camera.
	5b.2	The system opens the phone camera.
	5b.3	The user takes a photo using the phone camera.
	5b.4	The system continues Main Flow Step 6.
Alternate Flow – Fail to recognise	7.1	The system displays the image uploaded on the page.
	7.2	The system displays an error message.
	7.3	End of the flow.
Rules	-	
Author	Teh Kai Wen	

Table 4.2 Use Case Description for Perform single attraction recognition function

4.4.3 Use Case Description for Perform multiple attractions recognition function

Use Case ID	003	**Version**	1.0
Use Case	Perform multiple attractions recognition function		
Purpose	Enable the system to recognise multiple attractions in the image uploaded by the user.		
Actor	User		
Trigger	The user clicks "Image Detection" on the popup menu.		
Precondition	-		
Scenario Name	**Step**	**Action**	
Main Flow	1	The user clicks "Image Detection" on the popup menu.	
	2	The system displays the image detection page.	
	3	The user selects the method to upload an image by clicking the floating button on the bottom right of the page.	
	4	The system passes the image to the object detection model to perform recognition.	
	5	The system displays the image uploaded on the page and uses the bounding boxes to point out the recognised attractions.	
	6	The user clicks one of the bounding boxes.	
	7	The system navigates the user to the page of the attraction.	
Sub Flow – Select an image from the gallery	3a.1	The user selects the method of selecting an image from the gallery.	
	3a.2	The system opens the gallery of the phone.	

	3a.3	The user selects an image from the gallery.
	3a.4	The system continues Main Flow Step 4.
Sub Flow – Take a photo using the phone camera	3b.1	The user selects the method of taking a photo using the phone camera.
	3b.2	The system opens the phone camera.
	3b.3	The user takes a photo using the phone camera.
	3b.4	The system continues Main Flow Step 4.
Alternate Flow – Fail to detect any attractions in the image	5.1	The system displays the image uploaded on the page.
	5.2	End of the flow.
Rules	-	
Author	Teh Kai Wen	

Table 4.3 Use Case Description for Perform multiple attractions recognition function

4.4.4 Use Case Description for Perform real-time single attraction detection function

Use Case ID	004	**Version**	1.0
Use Case	Perform real-time single attraction detection function		
Purpose	Enable the user to scan through an attraction using the phone camera, and then the system will detect the attraction on the screen.		
Actor	User		
Trigger	The user clicks the "Single Attraction" button on the live detection page.		
Precondition	-		
Scenario Name	**Step**	**Action**	
Main Flow	1	The user clicks the "Single Attraction" button on the live detection page.	
	2	The system displays the real-time single attraction detection page.	
	3	The user uses the phone camera to scan through an attraction.	
	4	The system sends every frame to the image classification model to perform detection.	
	5	The model detects an attraction in the frame.	
	6	The system displays a bounding box to point out the attraction and a navigation button on the screen.	
	7	The user clicks on the navigation button.	
	8	The system navigates the user to the page of the attraction.	

Alternate Flow – Fail to detect any attraction in the frame	5.1	The model fails to detect any attraction in the frame
	5.2	The system reconducts Main Flow Step 4.
Rules	-	
Author	Teh Kai Wen	

Table 4.4 Use Case Description for Perform real-time single attraction detection function

4.4.5 Use Case Description for Perform real-time multiple attractions detection function

Use Case ID	005	**Version**	1.0
Use Case	Perform real-time multiple attractions detection function		
Purpose	Enable the user to scan through multiple attractions using the phone camera, and then the system will detect the attractions on the screen.		
Actor	User		
Trigger	The user clicks the "Multiple Attractions" button on the live detection page.		
Precondition	-		
Scenario Name	**Step**	**Action**	
Main Flow	1	The user clicks the "Multiple Attractions" button on the live detection page.	
	2	The system displays the real-time multiple attractions detection page.	

	3	The user uses the phone camera to scan through multiple attractions.
	4	The system sends every frame to the object detection model to perform detection.
	5	The model detects attractions in the frame.
	6	The system displays multiple bounding boxes to point out the attractions on the screen depending on the number of detected attractions.
	7	The user clicks one of the bounding boxes.
	8	The system navigates the user to the page of the attraction.
Alternate Flow – Fail to detect any attraction in the frame	5.1	The model fails to detect any attraction in the frame
	5.2	The system reconducts Main Flow Step 4.
Rules	-	
Author	Teh Kai Wen	

Table 4.5 Use Case Description for Perform real-time multiple attractions detection function

4.5 Activity Diagram

4.5.1 Activity Diagram for Perform searching through the attractions list

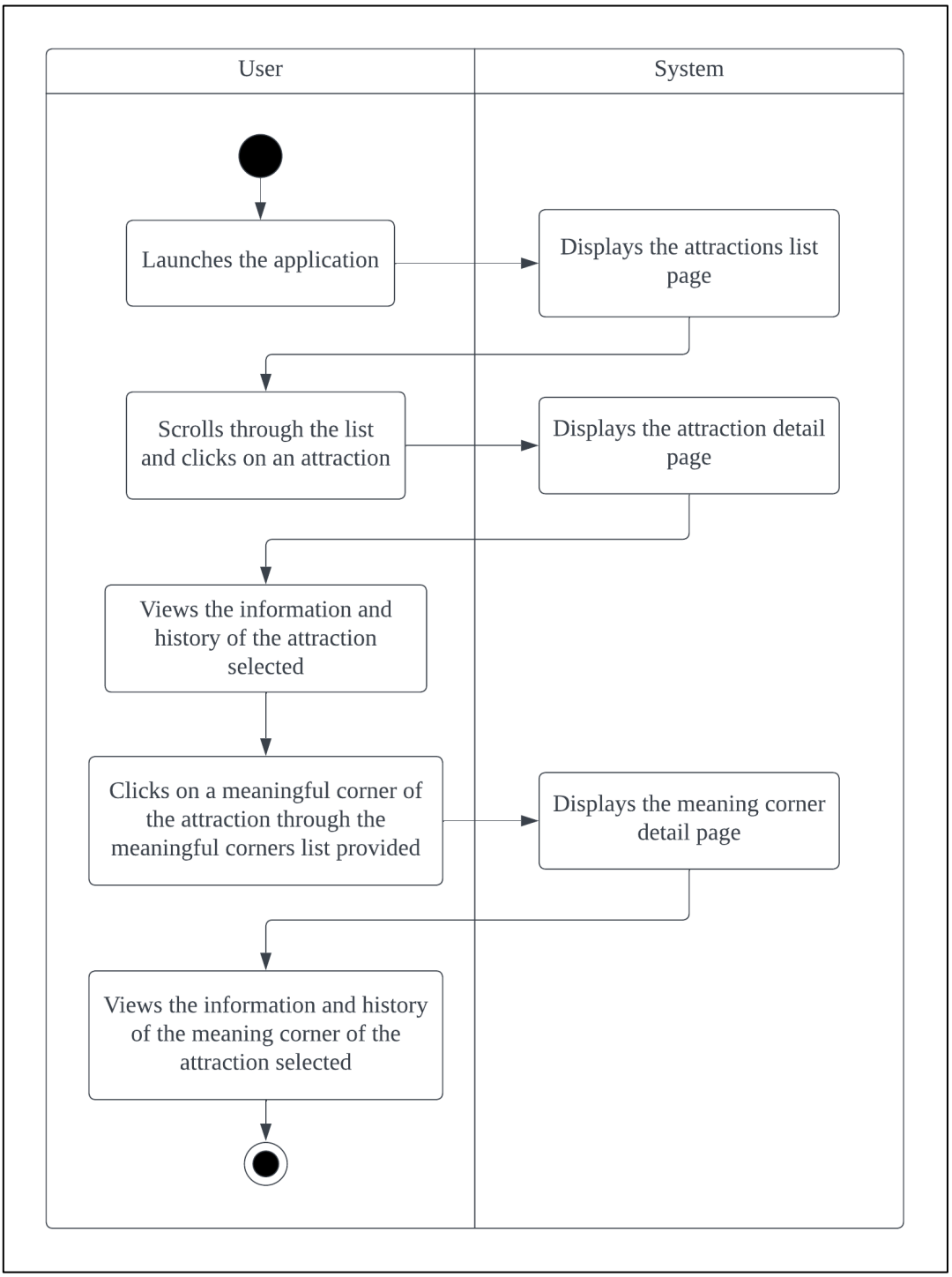


Figure 4.5 Activity Diagram for Perform searching through the attractions list

4.5.2 Activity Diagram for Perform single attraction recognition function

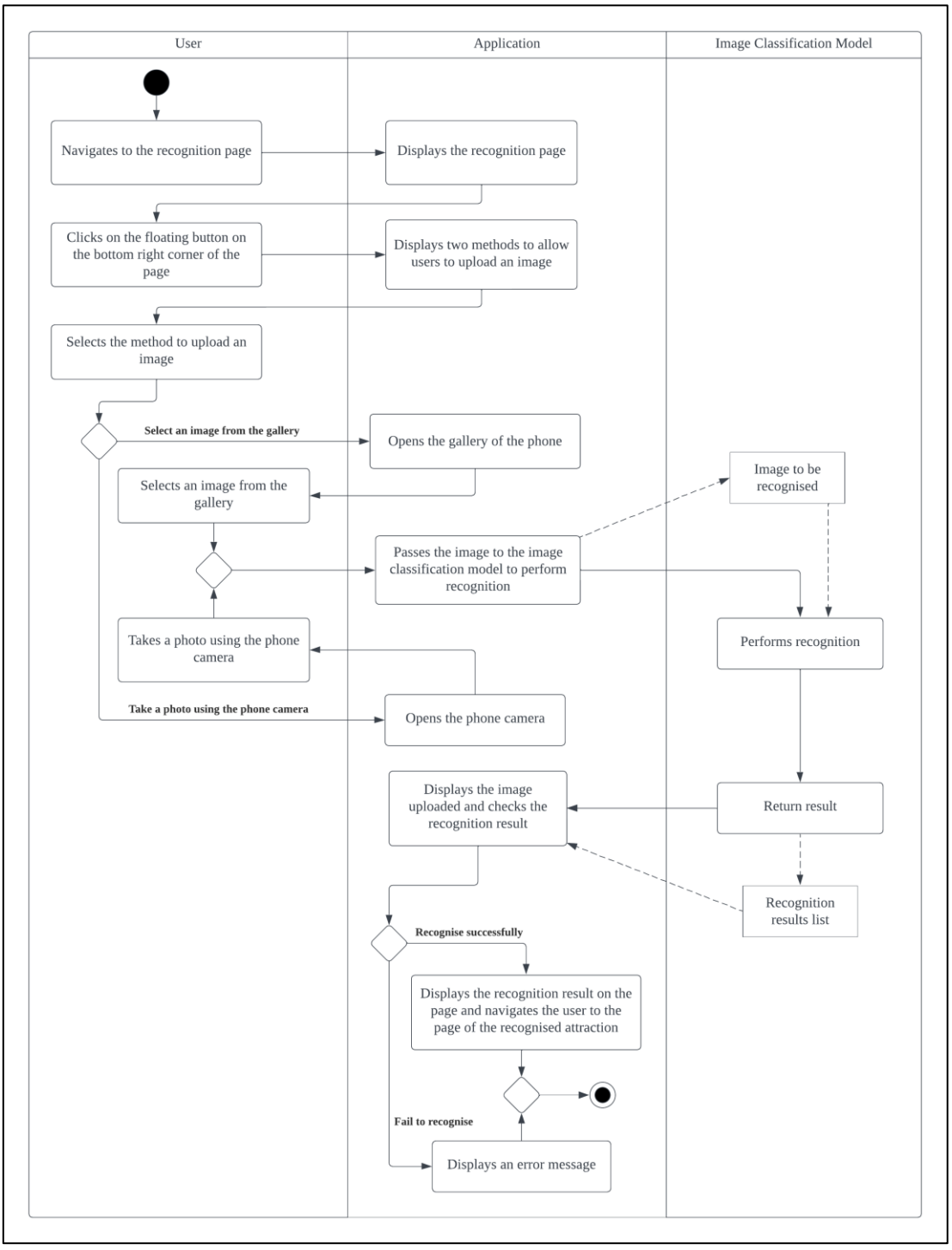


Figure 4.6 Activity Diagram for Perform single attraction recognition function

4.5.3 Activity Diagram for Perform multiple attractions recognition function

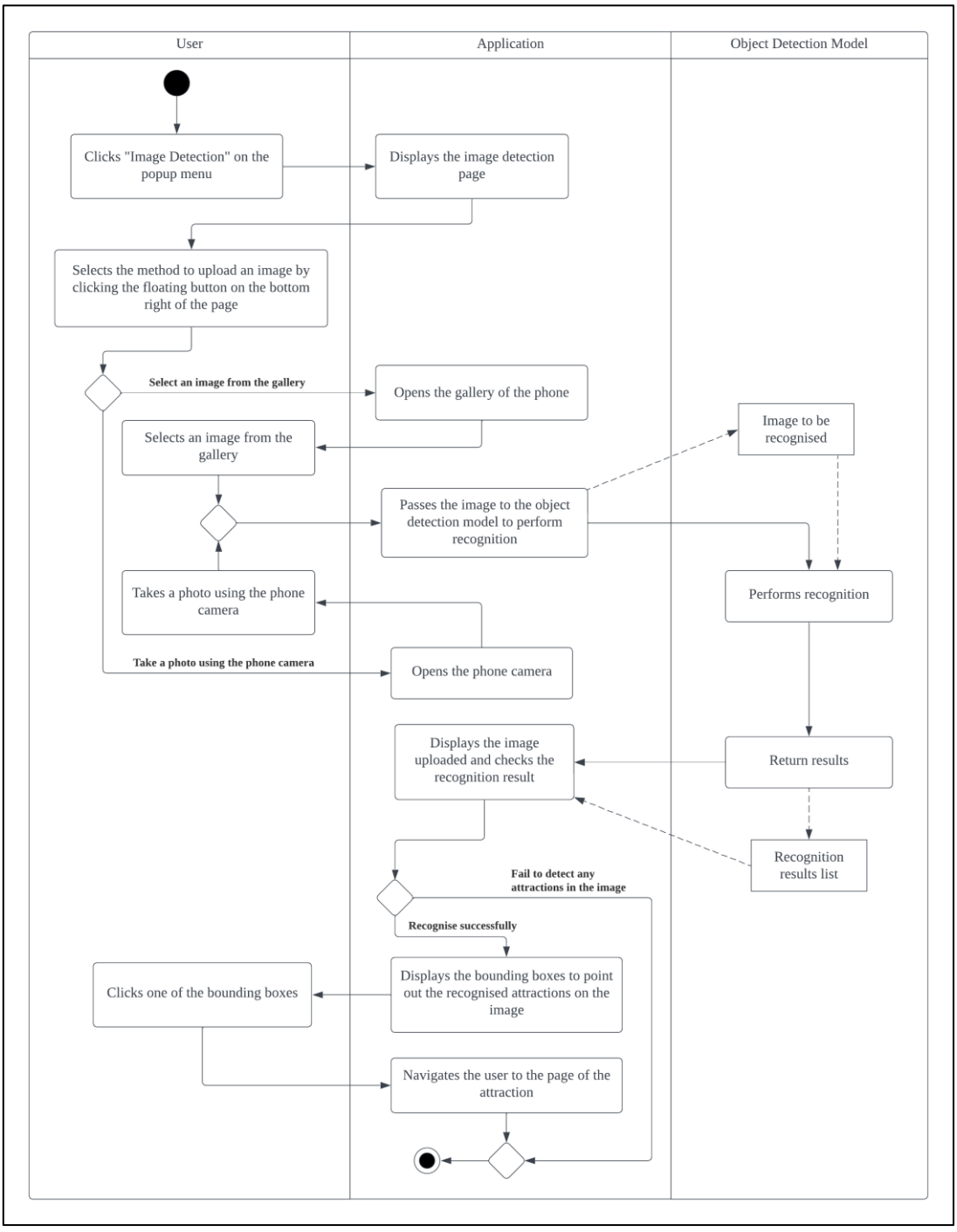


Figure 4.7 Activity Diagram for Perform multiple attractions recognition function

4.5.4 Activity Diagram for Perform real-time single attraction detection function

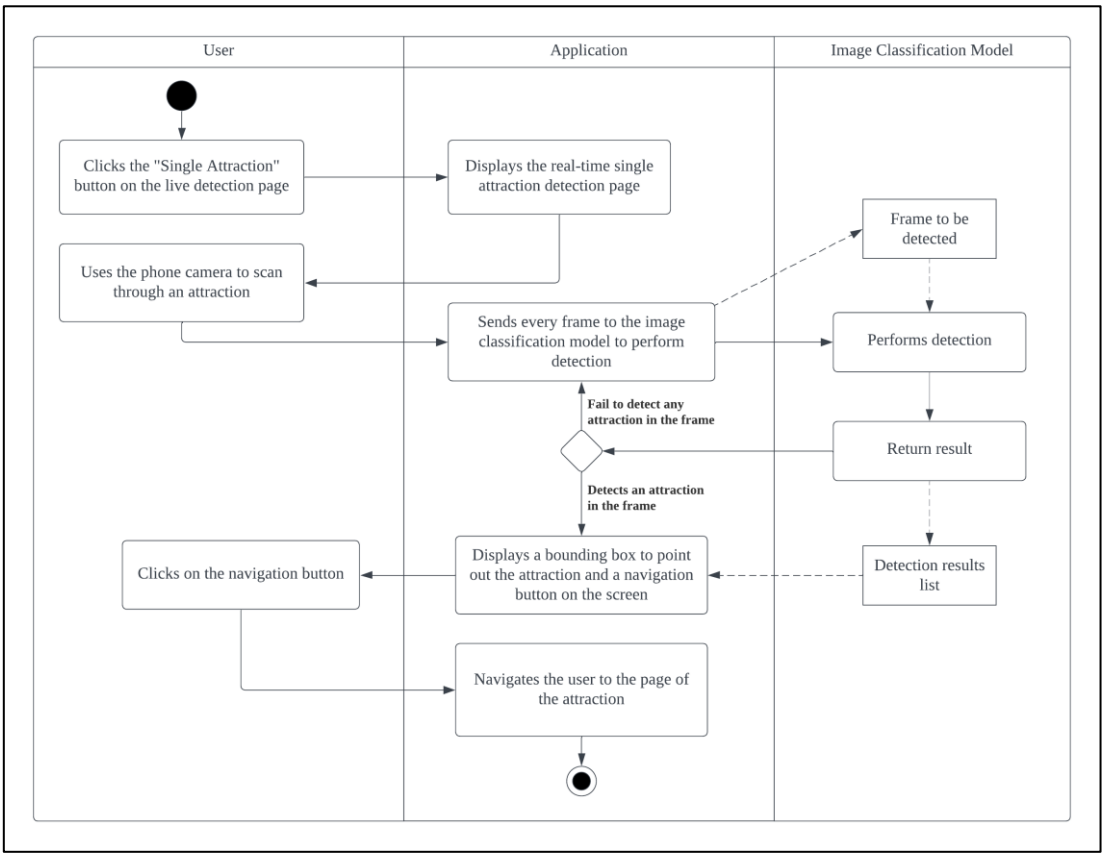


Figure 4.8 Activity Diagram for Perform real-time single attraction detection function

4.5.5 Activity Diagram for Perform real-time multiple attractions detection function

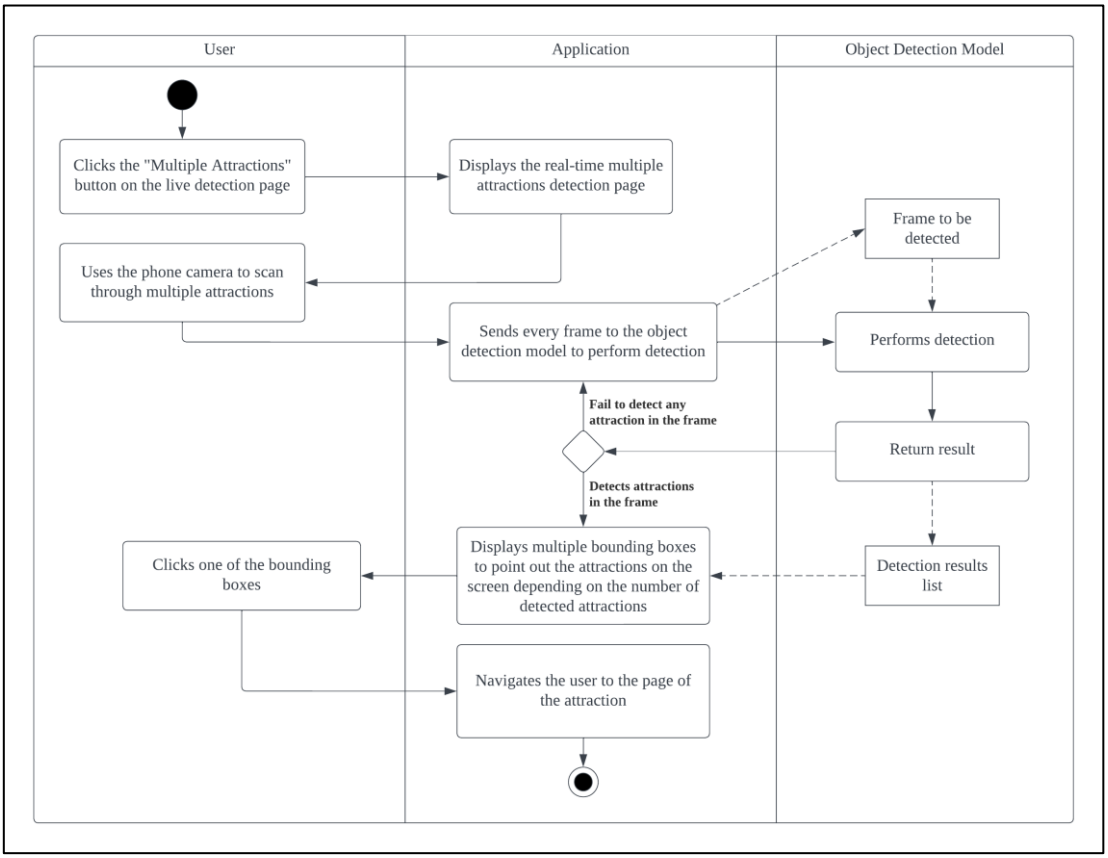


Figure 4.9 Activity Diagram for Perform real-time multiple attractions detection function

Chapter 5

System Implementation

In this project, the main focus will be narrowed down to recognise 2 temples in Kampar. They are Kampar Lu Ban Temple and Kampar Chinese Temple. Although these 2 temples are different attractions, they have some similarities. Therefore, if the recognition of these 2 temples is successful, the concept of this project can be proved to be feasible.

5.1 Setting up

5.1.1 Software

Before starting to develop the application, there are 4 software needed to be installed and downloaded on the laptop:

1. Visual Studio Code (version 1.77)
2. Android Studio (version Chipmunk 2021.2.1)
3. Jupyter Notebook (version 6.3.0)
4. LabelImg (version 1.5.0)

5.1.2 Software development kit

There is also a software development kit (SDK) needed to be installed and downloaded:

1. Flutter SDK (version 3.7.5)

5.1.3 Libraries

In Jupyter Notebook, there are 2 libraries needed to be installed:

1. TensorFlow (version 2.9.1 & 2.5.0)
2. TensorFlow Lite Model Maker (version 0.3.4)

5.2 Process of Training Image Classification Model

In this project, an image classification model will be used to perform the single attraction recognition function and the real-time single attraction detection function.

5.2.1 Gathering Dataset for Image Classification

For the dataset, 2,327 images are taken physically using the phone camera at the 2 temples. These images taken contain images of the temples and images of some meaningful corners at the temples. They are taken from different angles, conditions, and camera setups. Next, the images taken will be classified into 15 classes. They are stored accordingly in 15 sub-folders which are labelled with the name. The dataset is ready to be used.

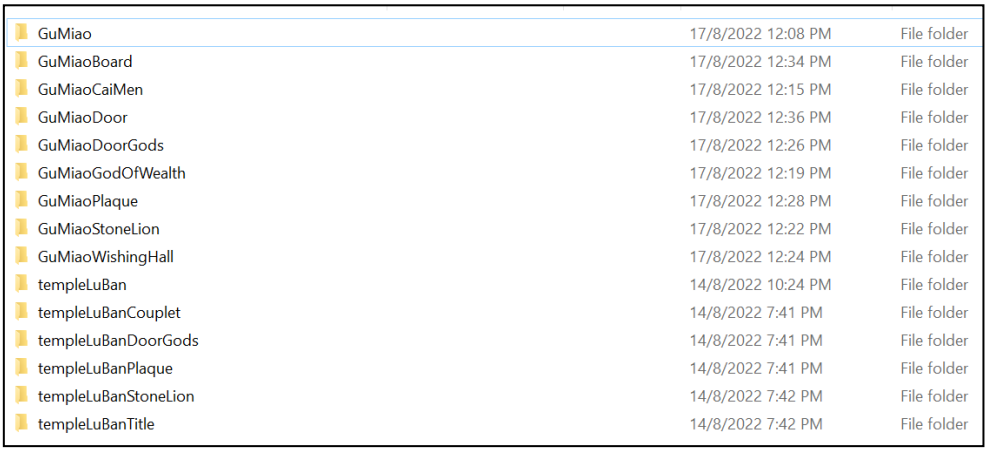


Figure 5.1 Dataset for Image Classification

5.2.2 Importing Libraries Needed for Image Classification

The next step is to import all of the libraries needed into the Jupyter Notebook. For example, os, numpy, tensorflow, tflite_model_maker, and matplotlib.pyplot. OS library provides a convenient way to use the dependent functions of the operating system. It is mainly used to interact with the file system. Furthermore, the NumPy library is a common library for Python to work with arrays. When compared to the lists in Python, NumPy arrays are operated faster than lists. Then, the Matplotlib library is used to create visualizations in Python. It will be used to visualize the recognition results.

TensorFlow library and TensorFlow Lite Model Maker library are used to create and train the model. They have some predefined functions that make the training process easier and faster. The version of the TensorFlow library used is 2.9.1.

```
In [1]: import os

        import numpy as np

        import tensorflow as tf
        assert tf.__version__.startswith('2')

        from tflite_model_maker import model_spec
        from tflite_model_maker import image_classifier
        from tflite_model_maker.config import ExportFormat
        from tflite_model_maker.config import QuantizationConfig
        from tflite_model_maker.image_classifier import DataLoader

        import matplotlib.pyplot as plt
```

Figure 5.2 Libraries Needed for Image Classification

5.2.3 Loading Dataset for Image Classification

Before loading the dataset, the path of the dataset located needs to be specified correctly. Afterwards, DataLoader class is used to load the dataset from the path given. The dataset will be split into a training set and a testing set. The ratio of the training set and the testing set is 9:1. The training set is used to train the model while the testing set is used to evaluate the model.

```
In [2]: image_path = r"C:\Users\user\OneDrive\Desktop\Jupyter\dataAll"
        data = DataLoader.from_folder(image_path)
        train_data, test_data = data.split(0.9)

        INFO:tensorflow:Load image with size: 2327, num_label: 15, labels: GuMiao, GuMiaoBoard, GuMiaoCaiMen, GuMiaoDoor, GuMia
        oDoorGods, GuMiaoGodOfWealth, GuMiaoPlaque, GuMiaoStoneLion, GuMiaoWishingHall, templeLuBan, templeLuBanCouplet, temple
        LuBanDoorGods, templeLuBanPlaque, templeLuBanStoneLion, templeLuBanTitle.
```

Figure 5.3 Load Dataset for Image Classification

5.2.4 Training Image Classification Model

A TensorFlow predefined image classification model will be used to train with the dataset. It can achieve high accuracy with fewer computations and parameters. The predefined model that will be used in this project is the EfficientNet-Lite0 model. This is because its model size and latency are lower than other types of EfficientNet-Lite

models, so it is more suitable for the mobile tour guide application. The model has 3 convolutional layers which are a Keras layer, a dropout layer, and a dense layer. The Keras layer is the functional building block of the model [18]. The dropout layer shall remove the noise and prevent the overfitting problem [19]. The dense layer would perform the specific operations on the input and return the output [20]. By default, the model would run in 5 epochs. The number of epochs can be increased to get higher accuracy, but it will cause an overfitting problem. To avoid the overfitting problem, the model is running in the default epochs.

```
In [3]: model = image_classifier.create(train_data)

        INFO:tensorflow:Retraining the models...
        Model: "sequential"
        _________________________________________________________________
         Layer (type)                Output Shape              Param #
        =================================================================
         hub_keras_layer_v1v2 (HubKe  (None, 1280)             3413024
         rasLayerV1V2)

         dropout (Dropout)           (None, 1280)              0

         dense (Dense)               (None, 15)                19215

        =================================================================
        Total params: 3,432,239
        Trainable params: 19,215
        Non-trainable params: 3,413,024
        _________________________________________________________________
        None
        Epoch 1/5


        C:\ProgramData\Anaconda3\lib\site-packages\keras\optimizers\optimizer_v2\gradient_descent.py:108: UserWarning: The `lr`
        argument is deprecated, use `learning_rate` instead.
          super(SGD, self).__init__(name, **kwargs)

        65/65 [==============================] - 308s 5s/step - loss: 1.0733 - accuracy: 0.8413
        Epoch 2/5
        65/65 [==============================] - 308s 5s/step - loss: 0.6270 - accuracy: 0.9981
        Epoch 3/5
        65/65 [==============================] - 310s 5s/step - loss: 0.6131 - accuracy: 1.0000
        Epoch 4/5
        65/65 [==============================] - 311s 5s/step - loss: 0.6048 - accuracy: 1.0000
        Epoch 5/5
        65/65 [==============================] - 309s 5s/step - loss: 0.5998 - accuracy: 1.0000
```

Figure 5.4 Train Image Classification Model

5.2.5 Evaluating Image Classification Model

After training the image classification model, the model is evaluated using the testing set. The accuracy of the model evaluation is 100%.

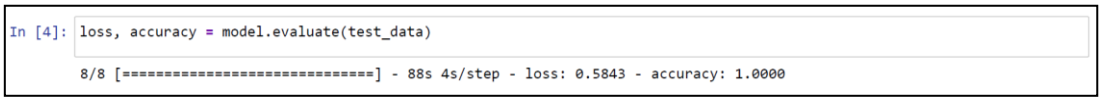


Figure 5.5 Evaluate Image Classification Model

5.2.6 Exporting Image Classification Model

Post-training quantization technique will be performed when exporting the image classification model. It is a conversion technique that helps in reducing model size and latency. The float16 quantization is used in this project. After exporting the model, the model is stored in a .tflite file and the labels are stored in a .txt file.

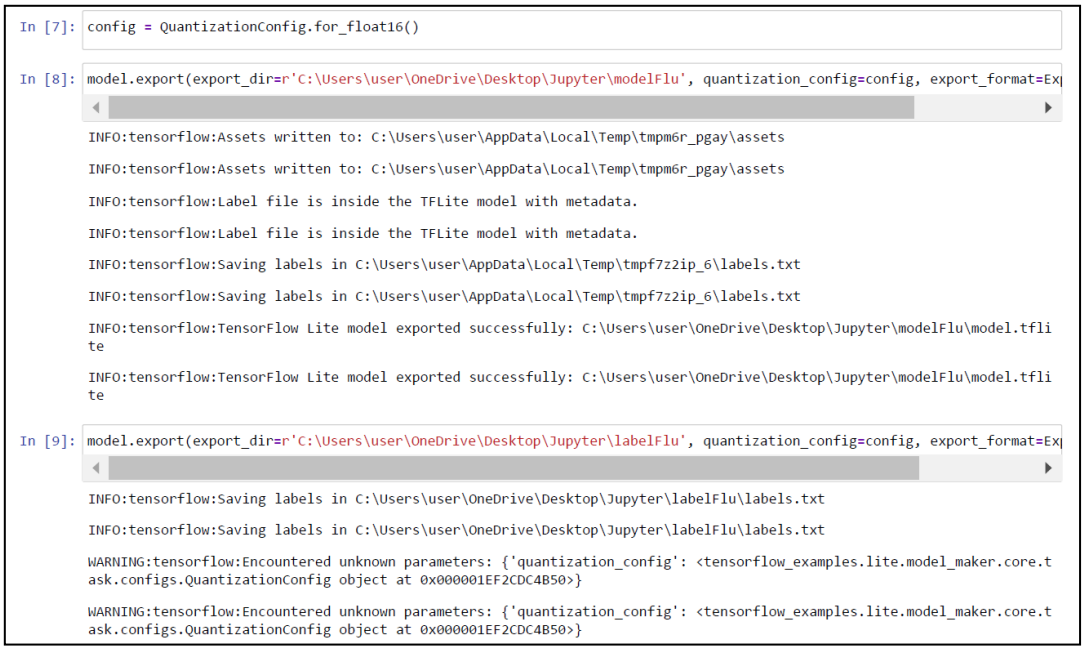


Figure 5.6 Export Image Classification Model and Labels

5.3 Process of Training Object Detection Model

In this project, an object detection model will be used to perform the multiple attractions recognition function and the real-time multiple attractions detection function.

5.3.1 Gathering Dataset for Object Detection

To prepare the dataset for object detection, the image dataset gathered in the image classification will be used back. All images will be randomly split into 2 sets, the training and validation sets, and then stored in 2 folders. The ratio of splitting is 8:2. Next, the images are labelled by using LabelImg. It is used to create the bounding boxes around attractions in the images and assign labels to them. The labels and the bounding box coordinates are saved into the XML files in PASCAL VOC data format. Each image should be associated with an XML file. Then, the dataset for object detection is ready to be used.

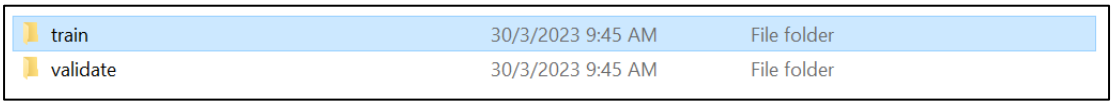


Figure 5.7 Dataset for Object Detection

Figure 5.8 LabelImg

5.3.2 Importing Libraries Needed for Object Detection

In the Jupyter Notebook, the relevant libraries are imported such as the os, numpy, tensorflow, and tflite_model_maker. The version of the TensorFlow library used is 2.5.0.

```
In [1]: import numpy as np
        import os

        from tflite_model_maker.config import ExportFormat
        from tflite_model_maker import model_spec
        from tflite_model_maker import object_detector

        import tensorflow as tf
        assert tf.__version__.startswith('2')

        tf.get_logger().setLevel('ERROR')
        from absl import logging
        logging.set_verbosity(logging.ERROR)
```

Figure 5.9 Libraries Needed for Object Detection

5.3.3 Loading Dataset for Object Detection

Afterwards, the predefined from_pascal_voc function is used to load the dataset with the PASCAL VOC format.

```
In [2]: train_data = object_detector.DataLoader.from_pascal_voc(
            'C:/Users/user/Desktop/Object Detection/images/train',
            'C:/Users/user/Desktop/Object Detection/images/train',
            ['GuMiao', 'GuMiaoBoard', 'GuMiaoCaiMen', 'GuMiaoDoor', 'GuMiaoDoorGods', 'GuMiaoGodOfWealth', 'GuMiaoPlaque', 'GuMiaoStoneLi
        )

        val_data = object_detector.DataLoader.from_pascal_voc(
            'C:/Users/user/Desktop/Object Detection/images/validate',
            'C:/Users/user/Desktop/Object Detection/images/validate',
            ['GuMiao', 'GuMiaoBoard', 'GuMiaoCaiMen', 'GuMiaoDoor', 'GuMiaoDoorGods', 'GuMiaoGodOfWealth', 'GuMiaoPlaque', 'GuMiaoStoneLi
        )
```

Figure 5.10 Load Dataset for Object Detection

5.3.4 Select Suitable Object Detection Model Architecture

In object detection, TensorFlow provides 5 types of EfficientDet-Lite model architecture. EfficientDet-Lite is a family of object detection models, meanwhile, EfficientNet-Lite is a family of image classification models. Both of them are optimized for mobile and embedded devices. EfficientDet-Lite uses a combination of feature pyramids and bi-directional feature networks to detect objects of varying sizes and

aspect ratios. In this project, the EfficientDet-Lite0 model architecture will be used. This is because the model needs to perform real-time object detection in the application, so the model requires to be operated faster and in a smaller size. Although the accuracy of the EfficientDet-Lite0 model might be lower than other EfficientDet-Lite models, there is no harm to the application if the EfficientDet-Lite0 model can have sufficient accuracy to detect the attractions.

Model architecture	Size(MB)*	Latency(ms)**	Average Precision***
EfficientDet-Lite0	4.4	37	25.69%
EfficientDet-Lite1	5.8	49	30.55%
EfficientDet-Lite2	7.2	69	33.97%
EfficientDet-Lite3	11.4	116	37.70%
EfficientDet-Lite4	19.9	260	41.96%

Figure 5.11 Comparison Between EfficientDet-Lite Models

```
In [3]: spec = model_spec.get('efficientdet_lite0')
```

Figure 5.12 Select EfficientDet-Lite0 Model

5.3.5 Training Object Detection Model

Next, the model will be trained with the training dataset in 20 epochs.

```
In [5]: model = object_detector.create(train_data, model_spec=spec, epochs=20, batch_size=8, train_whole_model=True, validation_data=val_

Epoch 1/20
232/232 [==============================] - 877s 4s/step - det_loss: 1.2656 - cls_loss: 0.9298 - box_loss: 0.0067 - reg_l2_loss:
0.0654 - loss: 1.3311 - learning_rate: 0.0090 - gradient_norm: 2.4584 - val_det_loss: 1.3291 - val_cls_loss: 0.9093 - val_box_l
oss: 0.0084 - val_reg_l2_loss: 0.0656 - val_loss: 1.3947
Epoch 2/20
232/232 [==============================] - 837s 4s/step - det_loss: 0.6201 - cls_loss: 0.4681 - box_loss: 0.0030 - reg_l2_loss:
0.0658 - loss: 0.6859 - learning_rate: 0.0098 - gradient_norm: 2.9232 - val_det_loss: 0.6420 - val_cls_loss: 0.4680 - val_box_l
oss: 0.0035 - val_reg_l2_loss: 0.0659 - val_loss: 0.7080
Epoch 3/20
232/232 [==============================] - 835s 4s/step - det_loss: 0.4098 - cls_loss: 0.3010 - box_loss: 0.0022 - reg_l2_loss:
0.0661 - loss: 0.4759 - learning_rate: 0.0096 - gradient_norm: 2.4936 - val_det_loss: 0.4460 - val_cls_loss: 0.3326 - val_box_l
oss: 0.0023 - val_reg_l2_loss: 0.0662 - val_loss: 0.5122
Epoch 4/20
232/232 [==============================] - 839s 4s/step - det_loss: 0.3325 - cls_loss: 0.2419 - box_loss: 0.0018 - reg_l2_loss:
0.0662 - loss: 0.3987 - learning_rate: 0.0092 - gradient_norm: 2.2442 - val_det_loss: 0.4188 - val_cls_loss: 0.3124 - val_box_l
oss: 0.0021 - val_reg_l2_loss: 0.0663 - val_loss: 0.4851
Epoch 5/20
232/232 [==============================] - 866s 4s/step - det_loss: 0.2966 - cls_loss: 0.2151 - box_loss: 0.0016 - reg_l2_loss:
0.0663 - loss: 0.3629 - learning_rate: 0.0087 - gradient_norm: 2.2776 - val_det_loss: 0.3939 - val_cls_loss: 0.2861 - val_box_l
oss: 0.0022 - val_reg_l2_loss: 0.0664 - val_loss: 0.4603
Epoch 6/20
232/232 [==============================] - 845s 4s/step - det_loss: 0.2749 - cls_loss: 0.1990 - box_loss: 0.0015 - reg_l2_loss:
0.0664 - loss: 0.3413 - learning_rate: 0.0081 - gradient_norm: 2.1760 - val_det_loss: 0.3370 - val_cls_loss: 0.2232 - val_box_l
oss: 0.0023 - val_reg_l2_loss: 0.0665 - val_loss: 0.4035
```

Figure 5.13 Train Object Detection Model

5.3.6 Evaluating Object Detection Model

After training the model, the model is evaluated using the validation dataset. The average precision (AP) of the model is about 80.96%. It also represents the overall accuracy of the model. In the context of object detection evaluation, a high AP score is generally considered to be good. The AP score measures the quality of the object detection algorithm, based on the precision and recall of the detected objects. A higher AP score indicates that the algorithm is better at detecting objects accurately. The average precision of this model for large objects is about 80.96% while the average recall for large objects is about 86.74%.

```
In [6]: model.evaluate(val_data)

        8/8 [==============================] - 98s 4s/step


Out[6]: {'AP': 0.8096495,
         'AP50': 0.982082,
         'AP75': 0.9287127,
         'APs': -1.0,
         'APm': -1.0,
         'APl': 0.8096495,
         'ARmax1': 0.80438626,
         'ARmax10': 0.864907,
         'ARmax100': 0.8673968,
         'ARs': -1.0,
         'ARm': -1.0,
         'ARl': 0.8673968,
         'AP_/GuMiao': 0.8073899,
         'AP_/GuMiaoBoard': 0.841762,
         'AP_/GuMiaoCaiMen': 0.91442716,
         'AP_/GuMiaoDoor': 0.9156155,
         'AP_/GuMiaoDoorGods': 0.7591672,
         'AP_/GuMiaoGodOfWealth': 0.7228386,
         'AP_/GuMiaoPlaque': 0.952421,
         'AP_/GuMiaoStoneLion': 0.74152666,
         'AP_/GuMiaoWishingHall': 0.76632494,
         'AP_/templeLuBan': 0.85771453,
         'AP_/templeLuBanCouplet': 0.4705526,
         'AP_/templeLuBanDoorGods': 0.83648044,
         'AP_/templeLuBanPlaque': 0.93034655,
         'AP_/templeLuBanStoneLion': 0.81378376,
         'AP_/templeLuBanTitle': 0.81439245}
```

Figure 5.14 Evaluate Object Detection Model

5.3.7 Exporting Object Detection Model

Lastly, the object detection model is stored in a .tflite file and the labels are stored in a .txt file.

```
In [7]: model.export(export_dir='C:/Users/user/Desktop/Object Detection/model5', tflite_filename='ARmodelv1.tflite')

In [9]: model.export(export_dir='C:/Users/user/Desktop/Object Detection/model5', export_format=ExportFormat.LABEL)
```

Figure 5.15 Export Object Detection Model

5.4 Mobile Application Development

5.4.1 Application Logo

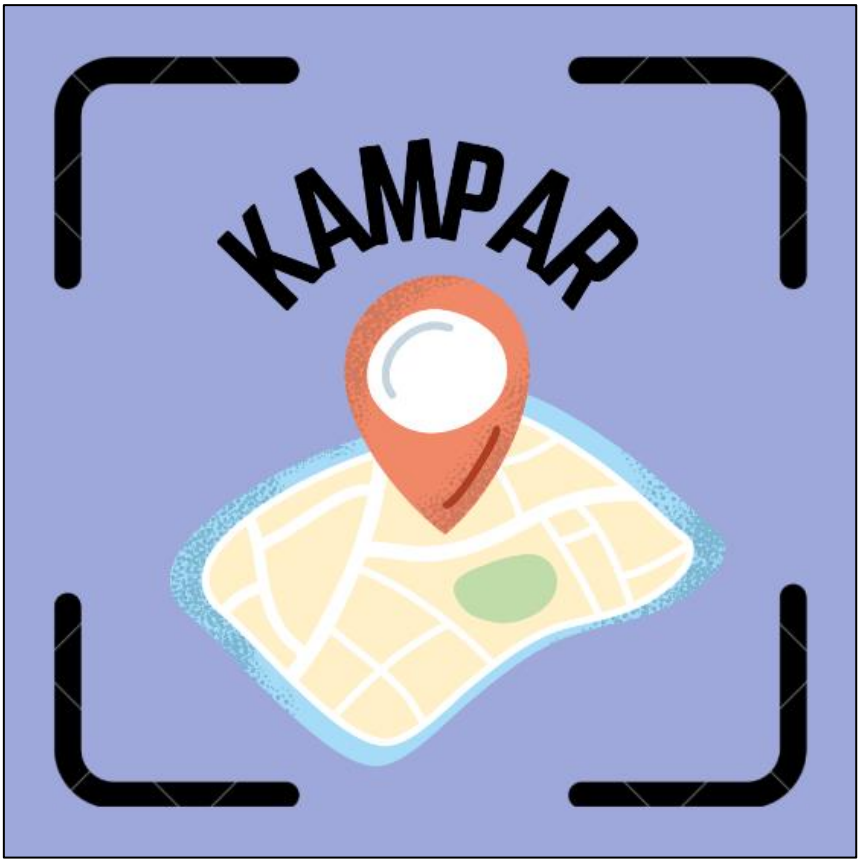


Figure 5.16 Application Logo

5.4.2 Attractions List Page

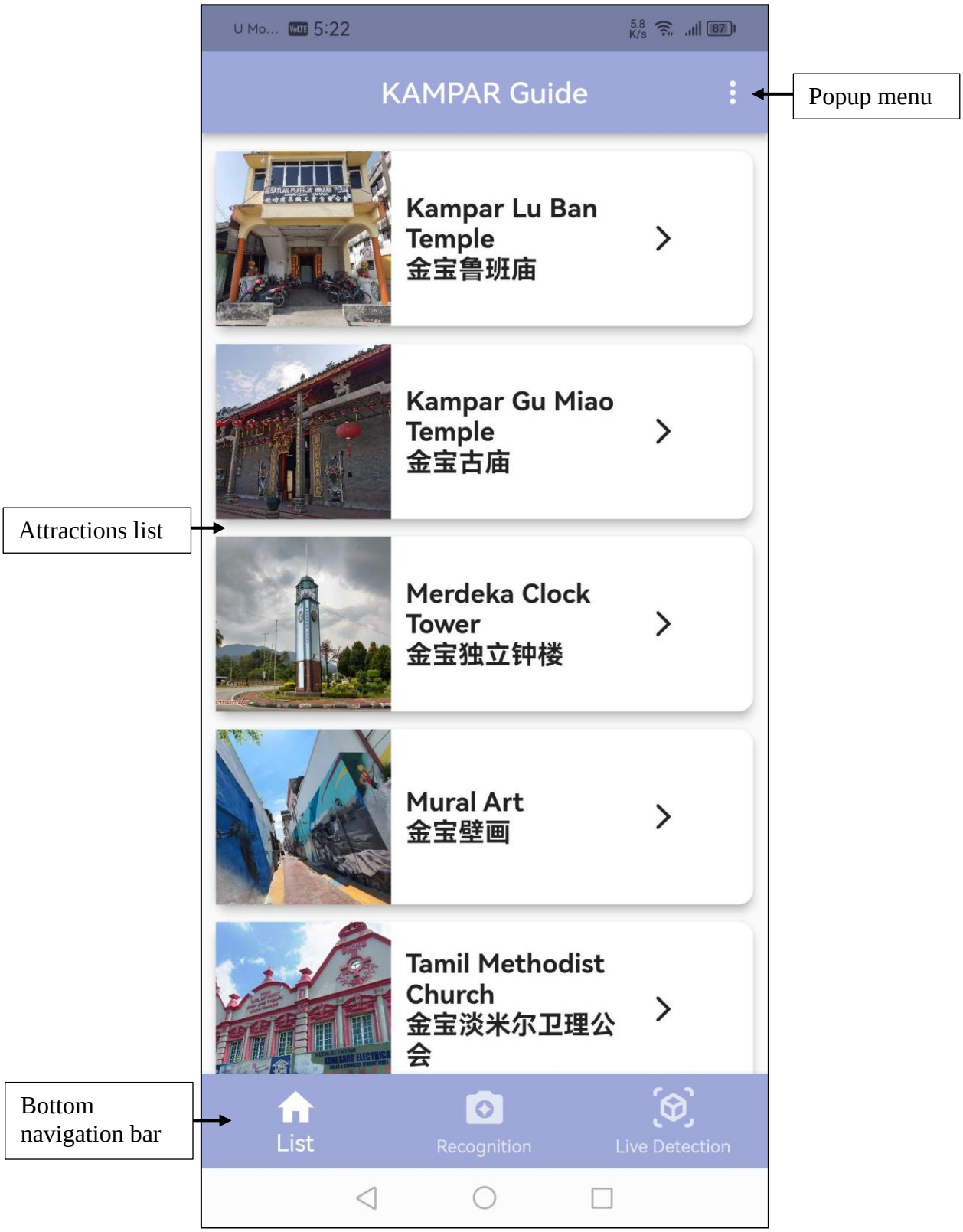


Figure 5.17 Attractions List Page

Open the popup menu

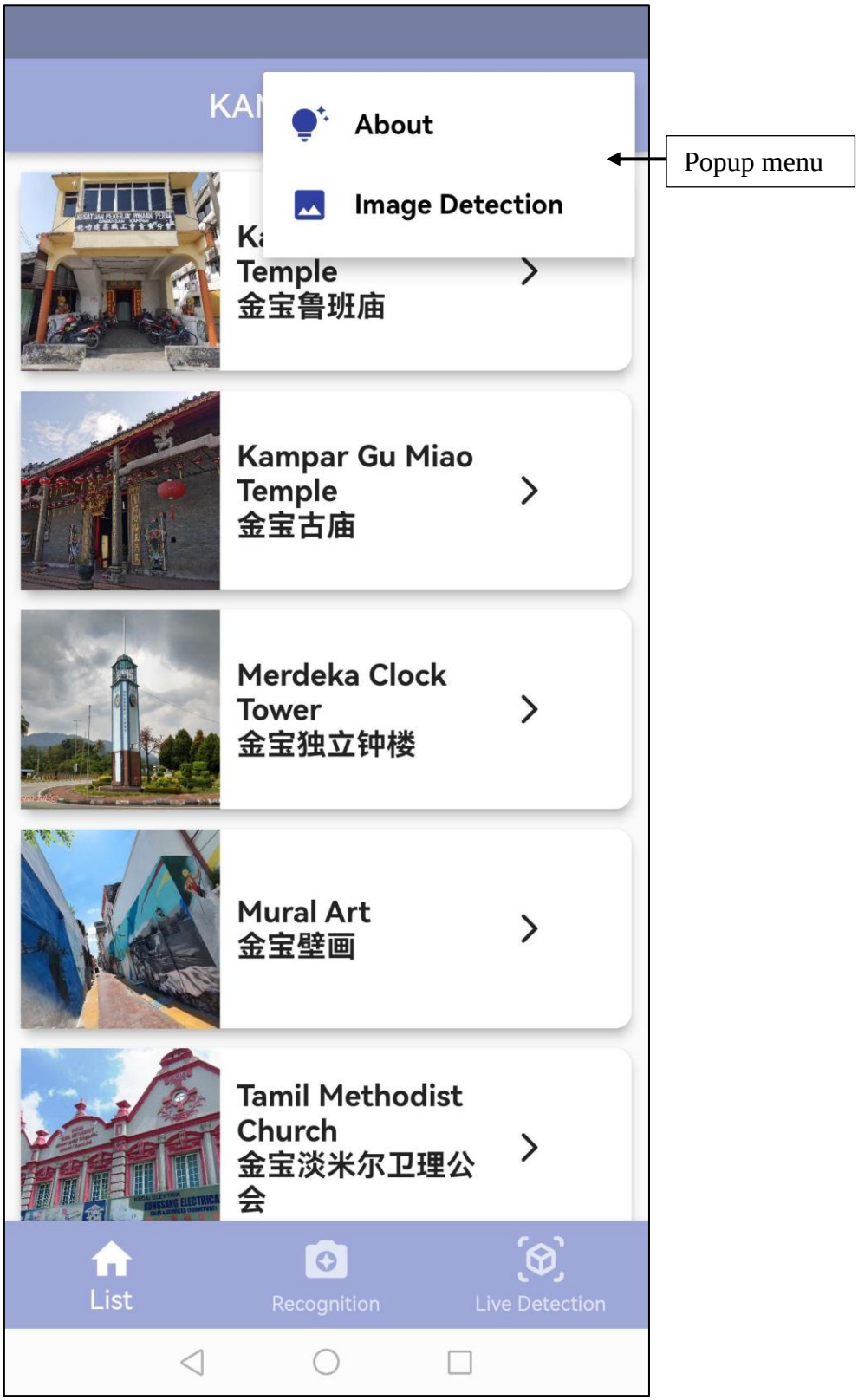


Figure 5.18 Popup Menu

5.4.3 Attraction Detail Page

Figure 5.19 Attraction Detail Page

5.4.4 Meaningful Corner Detail Page

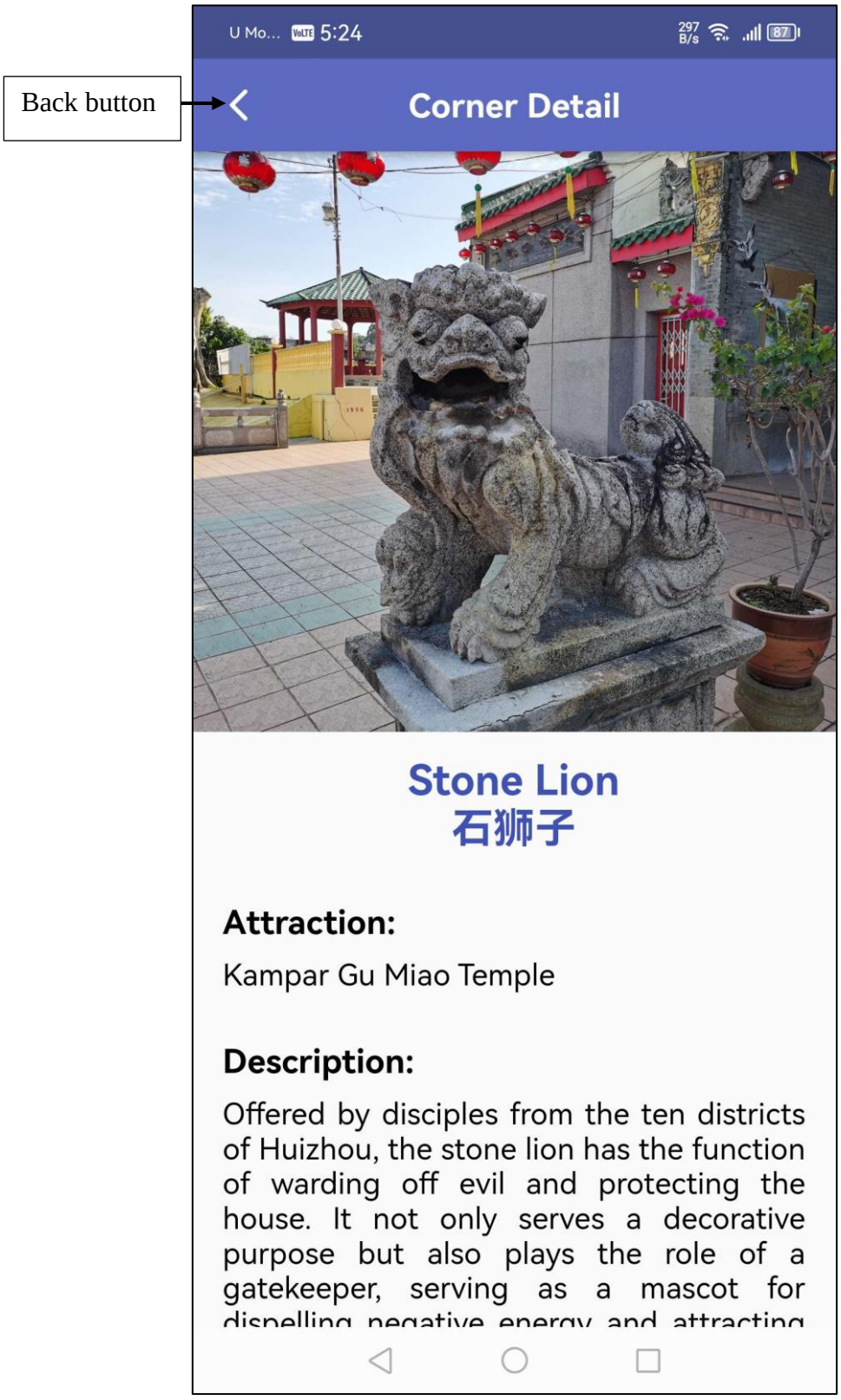


Figure 5.20 Meaningful Corner Detail Page

5.4.5 About Page

Figure 5.21 About Page

5.4.6 Multiple Attractions Recognition Page

Blank page (before uploading an image to perform multiple attractions recognition)

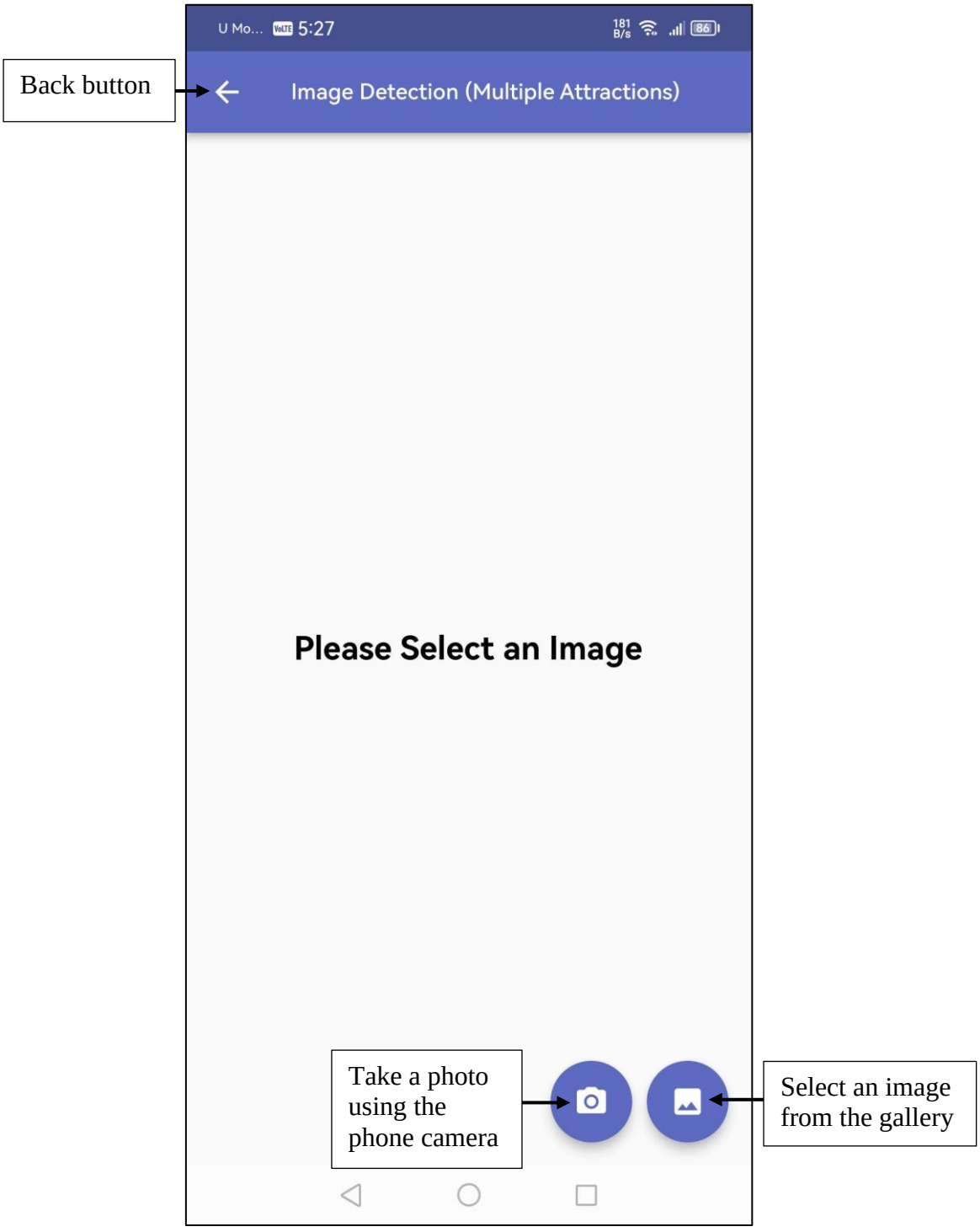


Figure 5.22 Multiple Attractions Recognition Page (Before Upload Image)

Select an image from the gallery

Figure 5.23 Select Image From Gallery

Take a photo using the phone camera

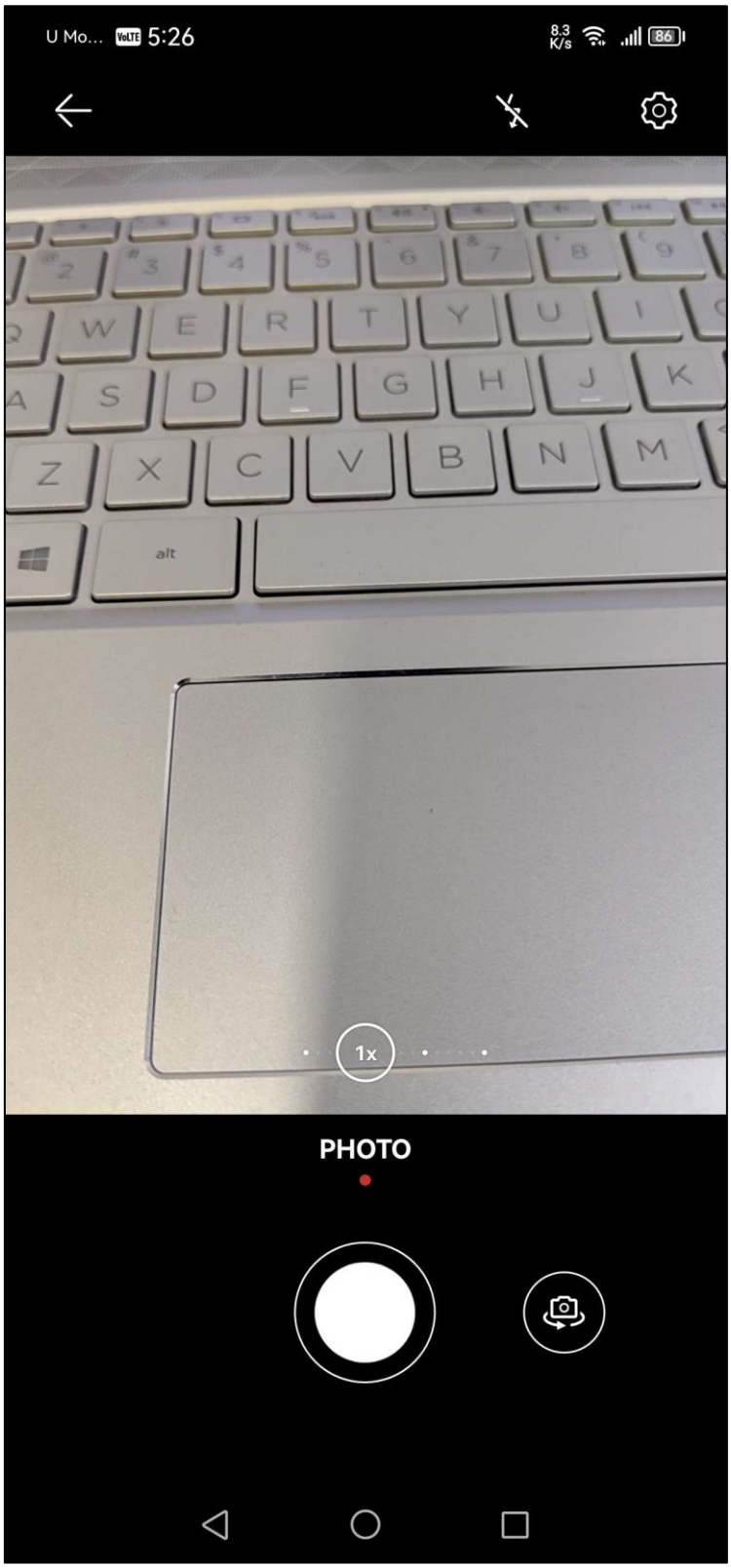


Figure 5.24 Take Photo Using Phone Camera

Display multiple attractions recognition results

After the image is passed to the object detection model, the model will return a list of results. The application uses multiple bounding boxes to point out the recognised attractions in which their confidence scores are > 0.3. The users will be navigated to the relevant page after they click one of the bounding boxes.

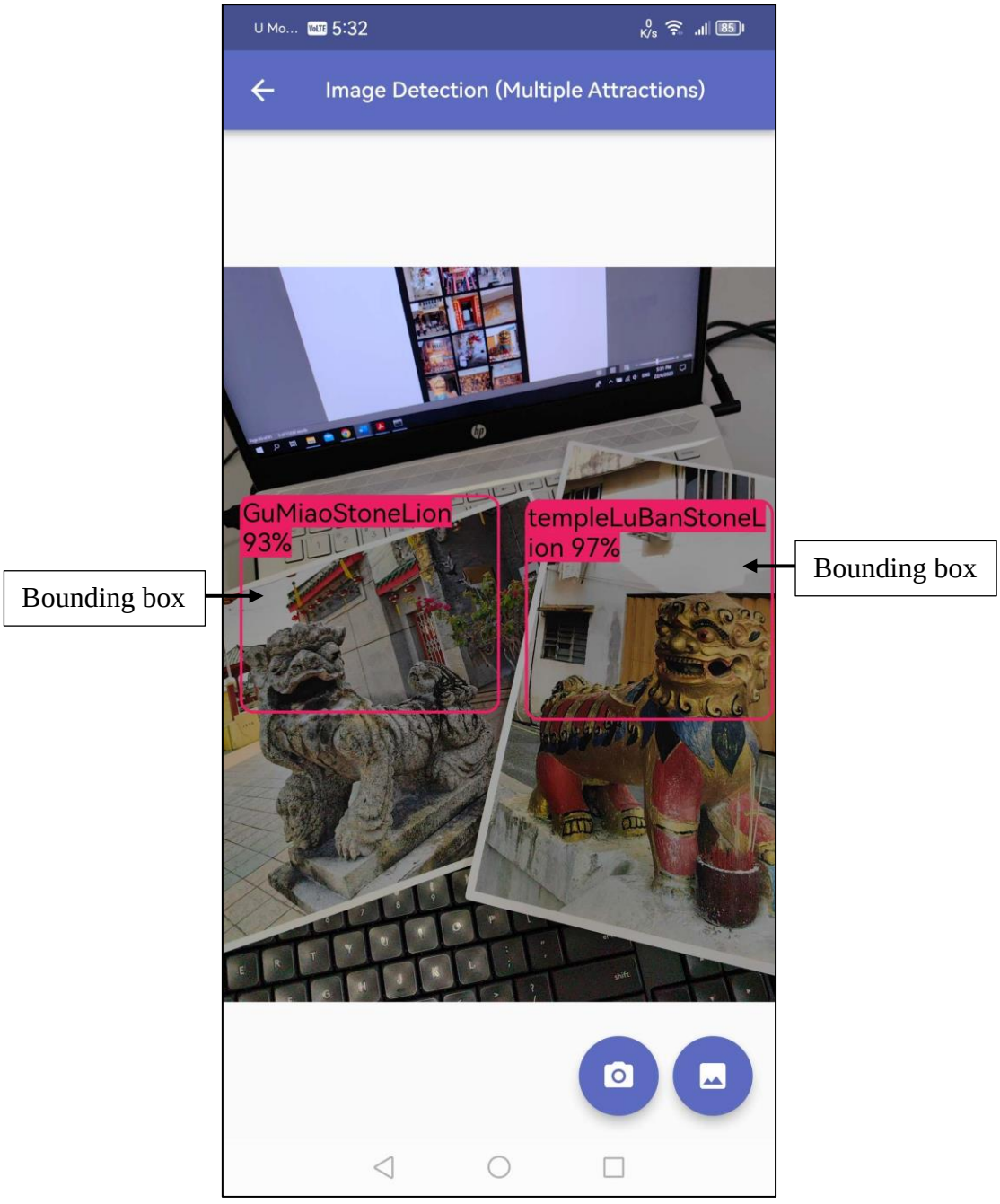


Figure 5.25 Multiple Attractions Recognition Page (After Upload Image)

5.4.7 Single Attraction Recognition Page

Blank page (before uploading an image to perform single attraction recognition)

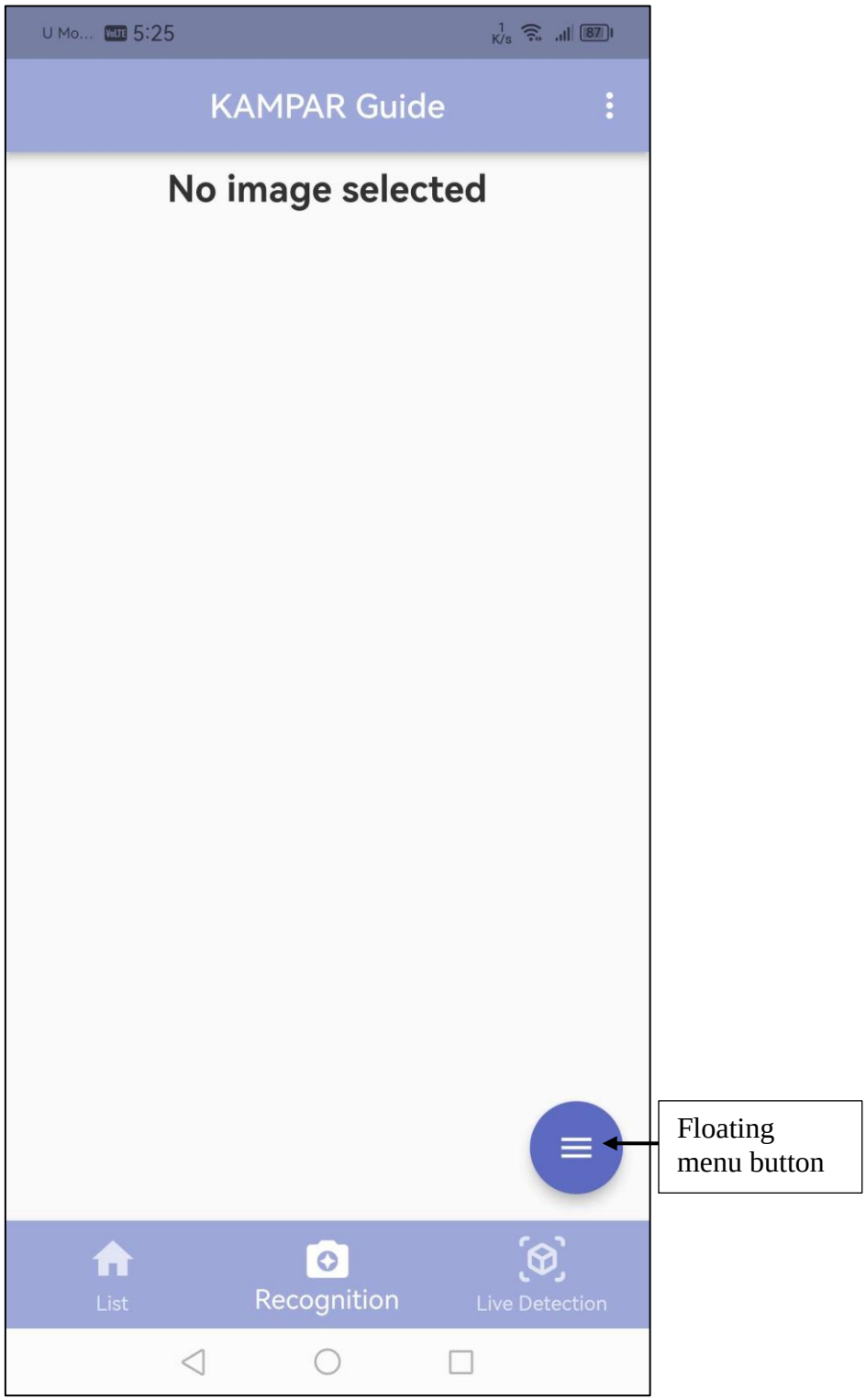


Figure 5.26 Single Attraction Recognition Page (Before Upload Image)

Select the method to upload an image for single attraction recognition

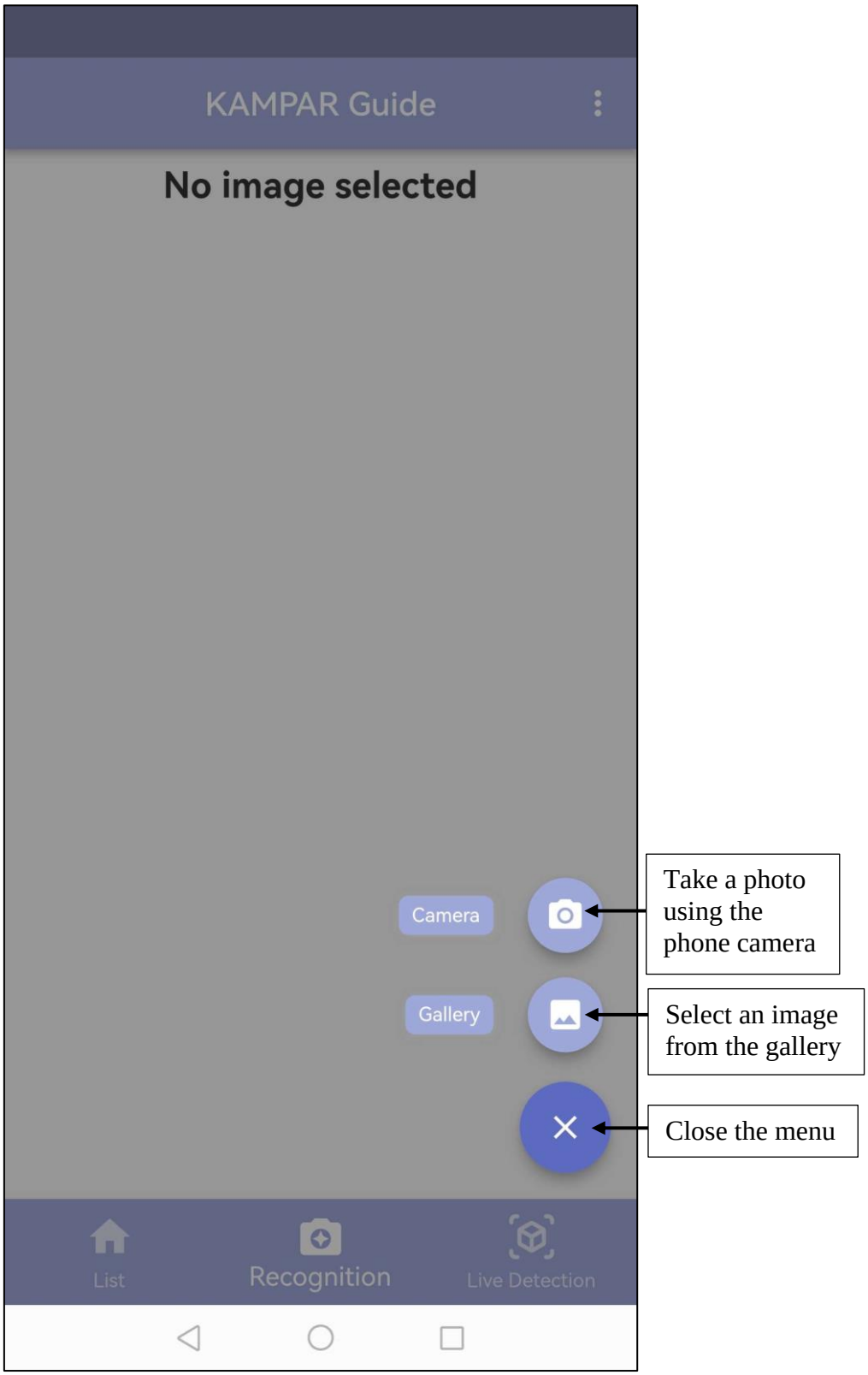


Figure 5.27 Floating Menu

Display single attraction recognition result

After the image is passed to the image classification model, the model will return a list of results. Then, the application selects the result with the highest confidence score and checks if the score is >= 0.25. If yes, the result is considered correct and the application will navigate the users to the relevant page. On the single attraction recognition page, the image uploaded and the results predicted would also be displayed.

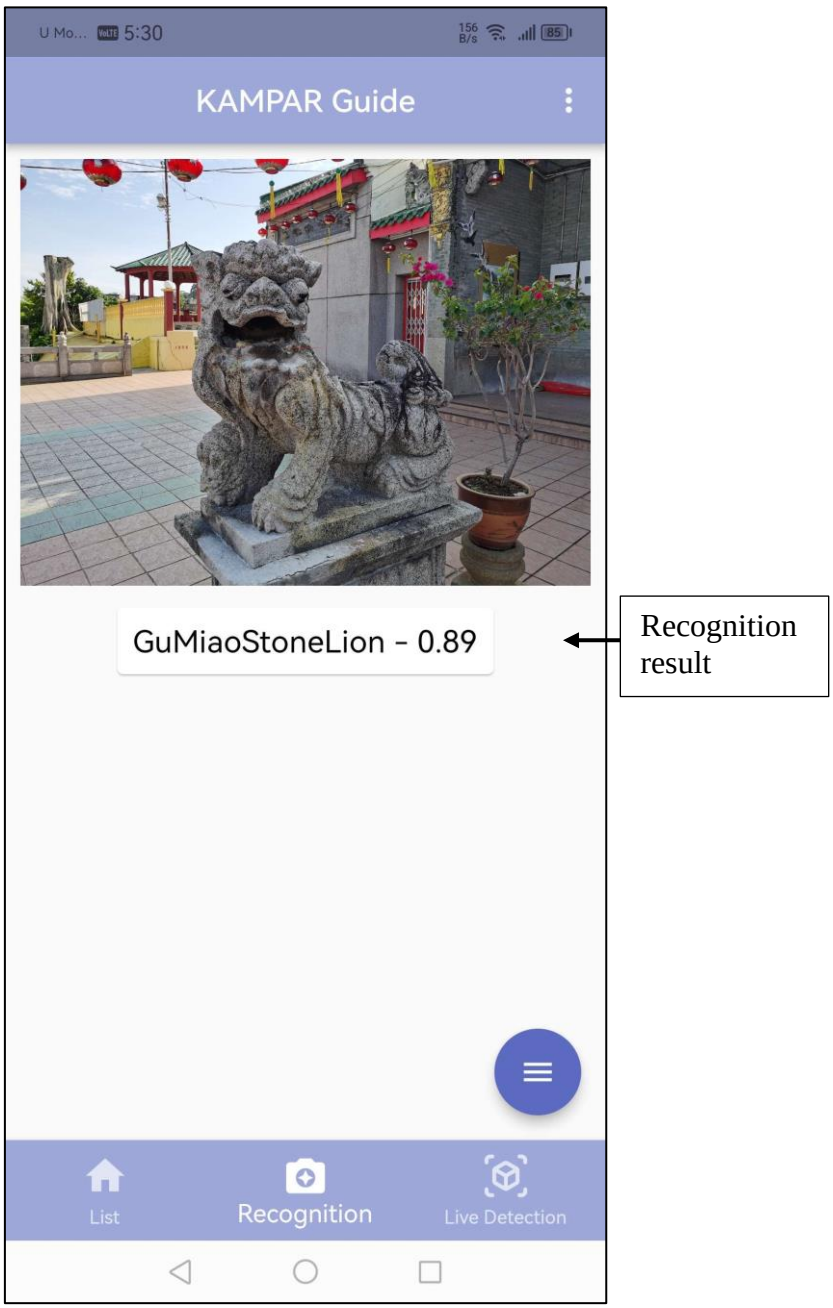


Figure 5.28 Single Attraction Recognition Page (After Upload Image)

Fail to recognise the attraction in the image

If the highest confidence score is < 0.25, the application considers the recognition is failed and displays an error message to the users.

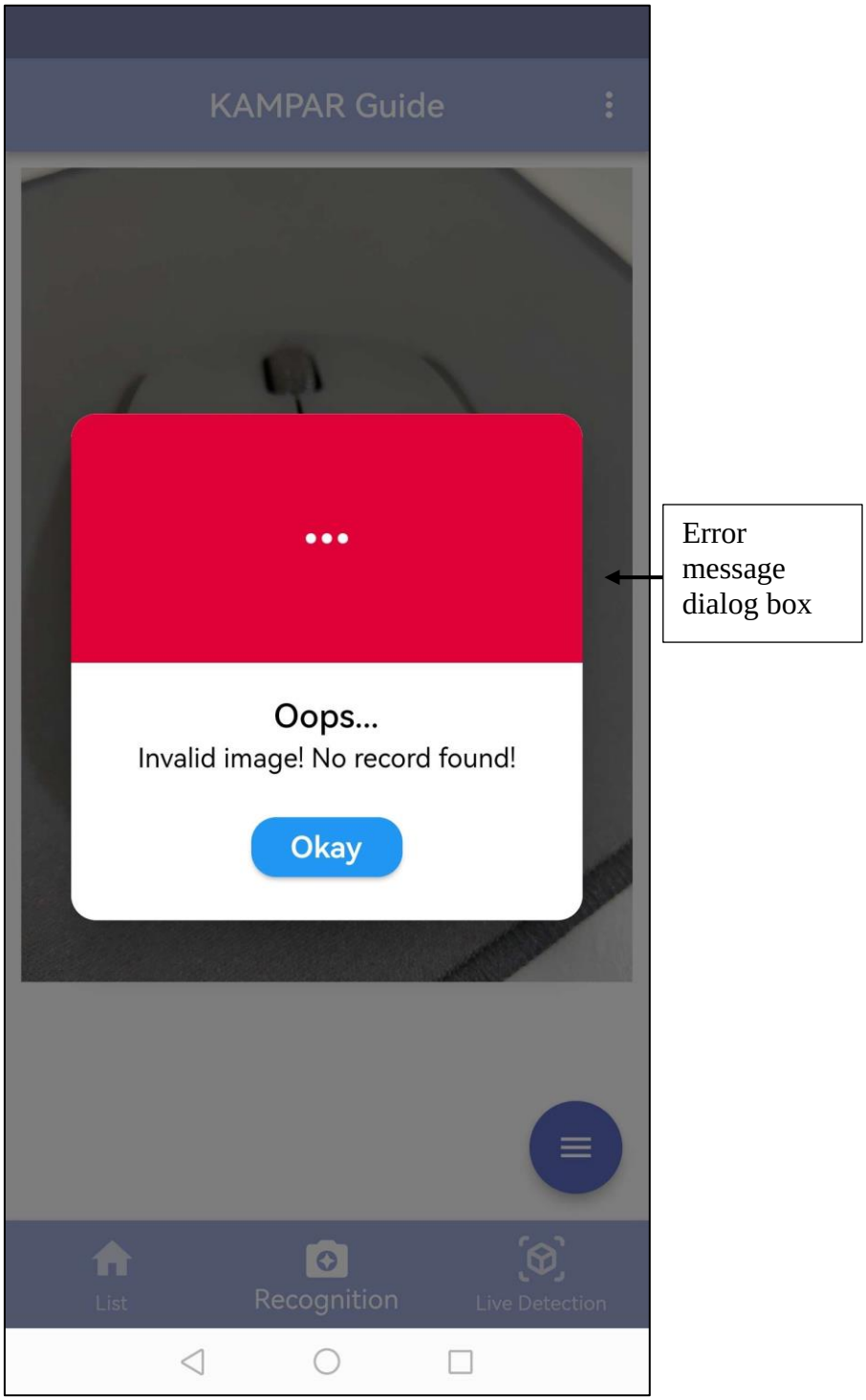


Figure 5.29 Error Message Dialog Box

5.4.8 Live Detection Page

The users can click on the "Single Attraction" button to perform real-time single attraction detection function, meanwhile, the users can click on the "Multiple Attractions" button to perform real-time multiple attractions detection function.

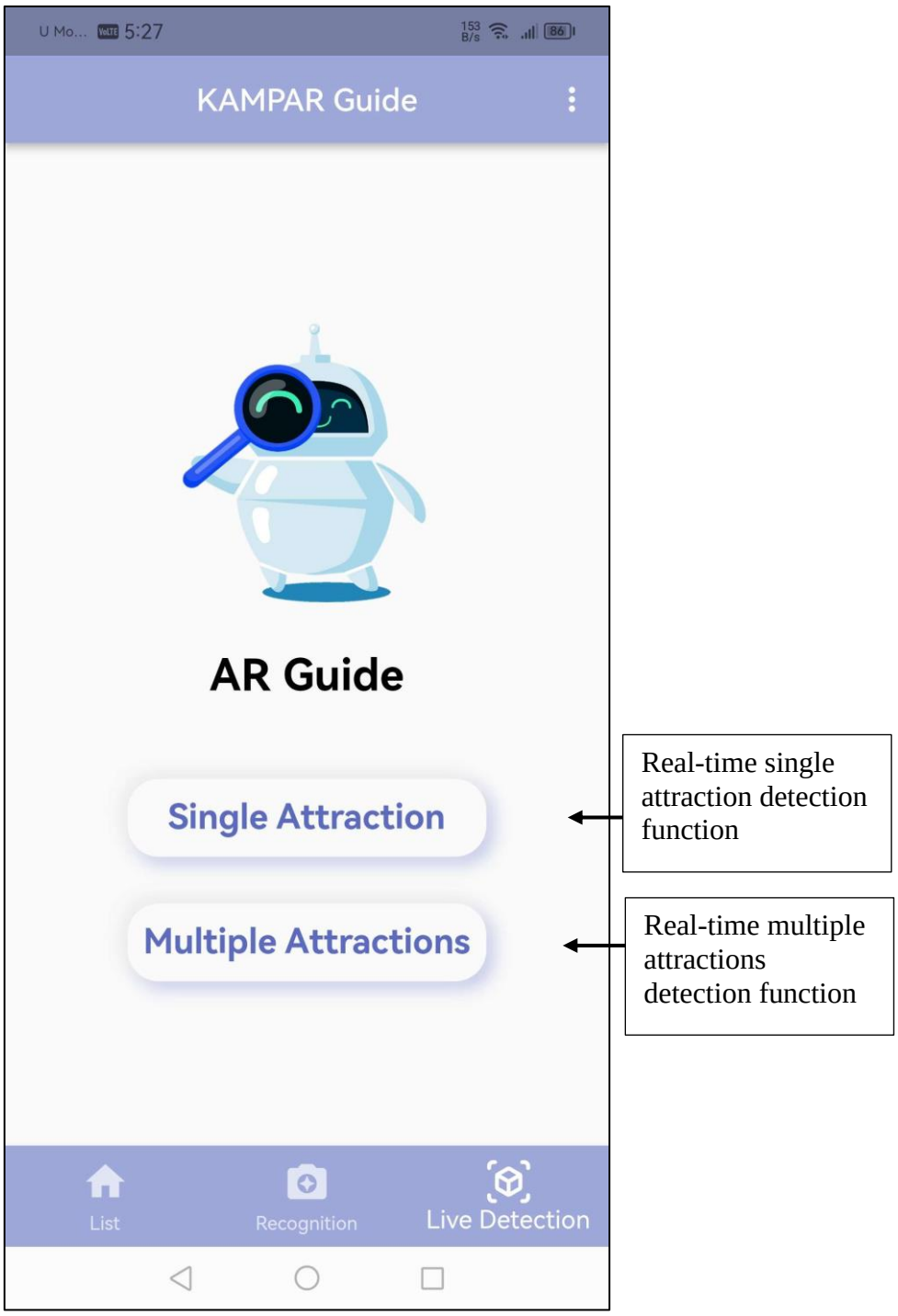


Figure 5.30 Live Detection Page

5.4.9 Single Attraction Detection Page

When the users scan through an attraction, the application sends every frame to the image classification model to perform detection. The model will return a list of results, and the application selects the result with the highest confidence score and checks if the score is >= 0.25. If yes, the application displays a bounding box and a navigation button on the screen. The users will be navigated to the relevant page after they click on the navigation button.

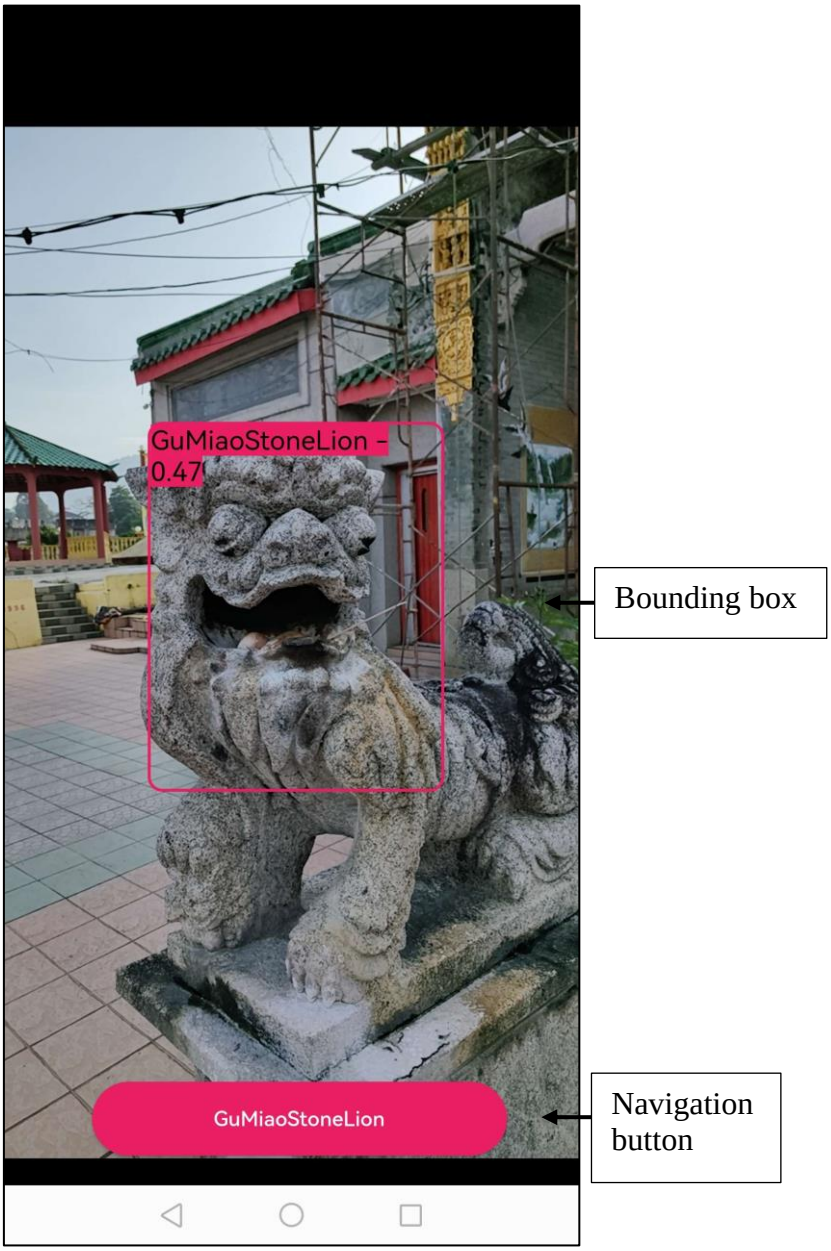


Figure 5.31 Single Attraction Detection Page

5.4.10 Multiple Attractions Detection Page

When the users scan through multiple attractions, the application sends every frame to the object detection model to perform detection. The model will return a list of results, and the application displays multiple bounding boxes on the screen to point out the detected attractions in which their confidence scores are > 0.3. The users will be navigated to the relevant page after they click one of the bounding boxes.

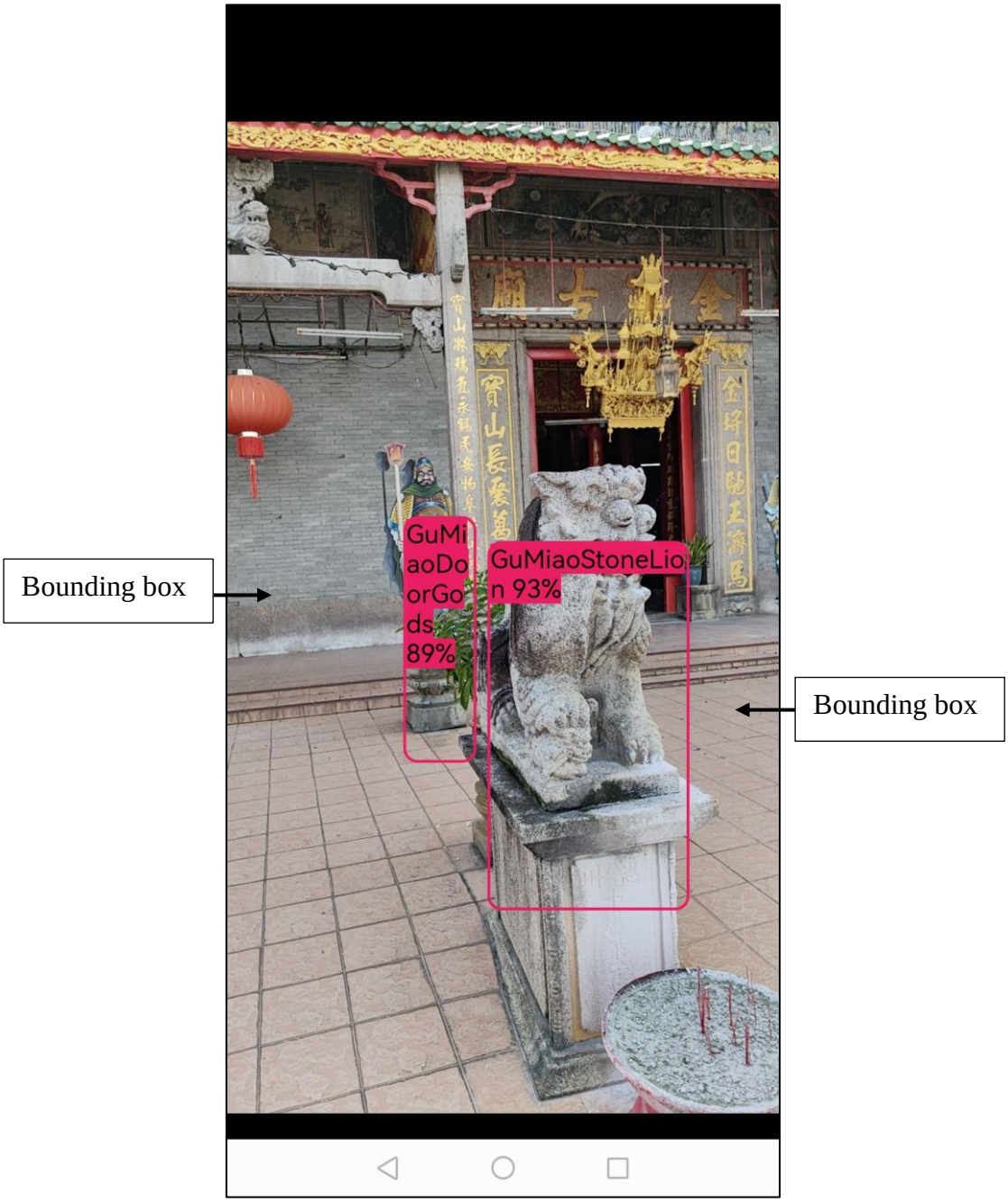


Figure 5.32 Multiple Attractions Detection Page

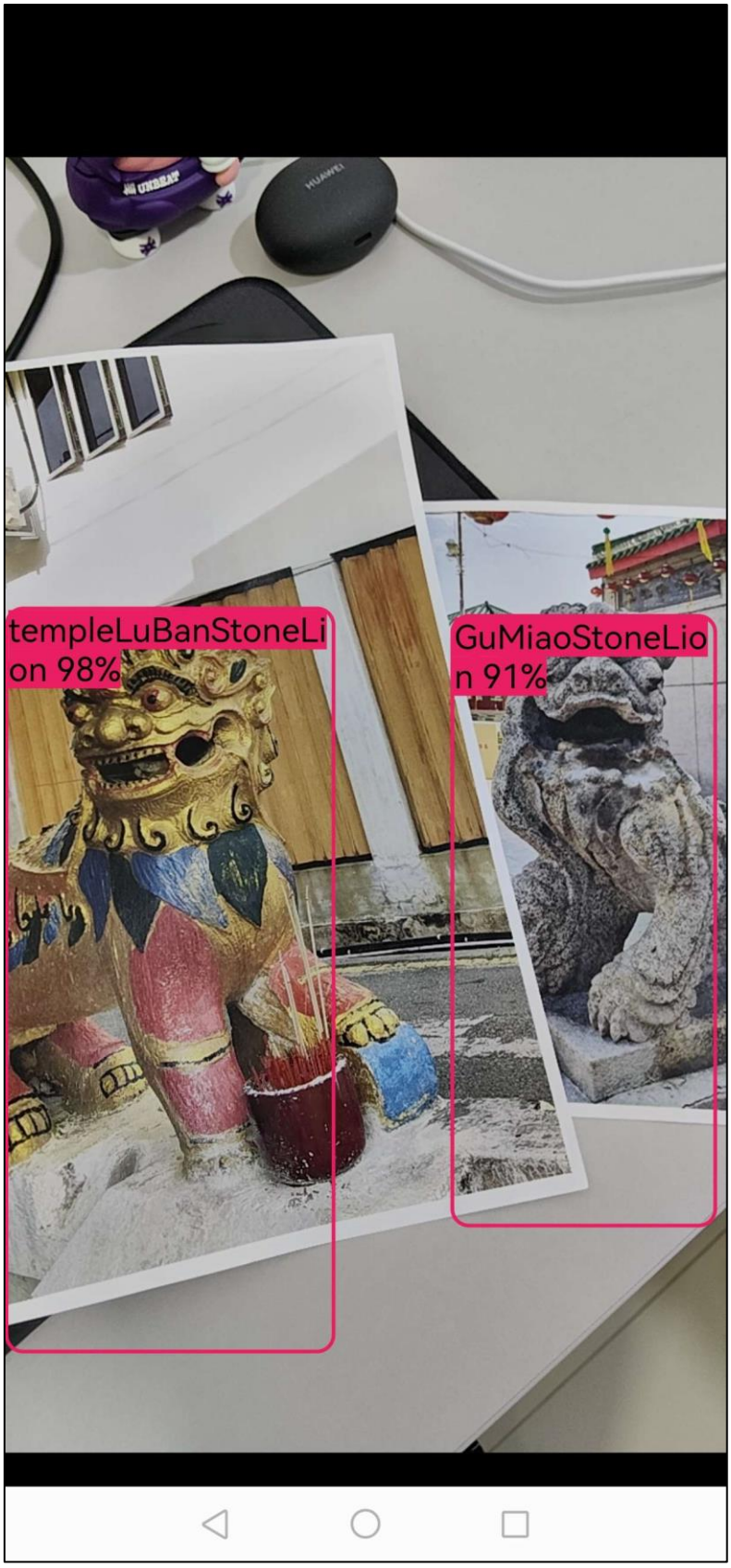


Figure 5.33 Multiple Attractions Detection Page (2)

5.5 Implementation Issues and Challenges

1. Choosing a suitable model architecture

Choosing a suitable model architecture will be a complex process. There are some conditions needed to consider about it. For example, the model size, latency and accuracy. Currently, TensorFlow supports a few types of classification models such as the ResNet50 model, EfficientNet-Lite model, and MobileNetV2 model [15]. The model that will be used in this project is the EfficientNet-Lite model. This is because the EfficientNet-Lite model is more suitable for image recognition. It could have high accuracy and is suitable for mobile devices. After choosing the EfficientNet-Lite model as the main model used in the project, the type of EfficientNet-Lite model is also needed to be specified. The EfficientNet-Lite models have 5 types, EfficientNet-Lite0 to EfficientNet-Lite4 [16]. The higher the model type selected, the higher the accuracy of the model. However, the model will have higher latency and size. It might be a challenge as different conditions when using the application need to be considered. This issue also applies a challenge in the selection of EfficientDet-Lite models for object detection.

2. Accuracy issue

There will have an accuracy issue when the image classification model is quantized. Quantization of the model will make the model operates faster and smoother [17]. The size of the model also would be smaller. However, the accuracy of the model is degraded. It affects the performance of the application. For instance, by using the same image, the model that is not quantized can predict the image correctly, but the model that is quantized will predict the image wrongly.

3. TensorFlow library version issue

Very beginning, the object detection model is trained using TensorFlow library version 2.9.1. However, this model will cause an error when the application wants to use it. After some investigation and research, this error is caused by the incorrect order of the model's output. The order of output is changed after TensorFlow library version 2.6.0. However, the TensorFlow package provided by Flutter has not adapted to the change. The Netron APP is used to visualize the infrastructure of the model. Figure 5.33 has showed the output order of the model using version 2.9.1. This is the changed output order, the order is the scores of the bounding boxes, locations of the bounding boxes, number of the bounding boxes, and the categories of the bounding boxes. Figure 5.34 has showed the output order of the model using version 2.5.0, this is the output order supported by the TensorFlow package in Flutter. The supported output order is the locations of the bounding boxes, categories of the bounding boxes, scores of the bounding boxes, and the number of the bounding boxes. Therefore, the TensorFlow library version 2.5.0 is used to train the object detection model in this project.

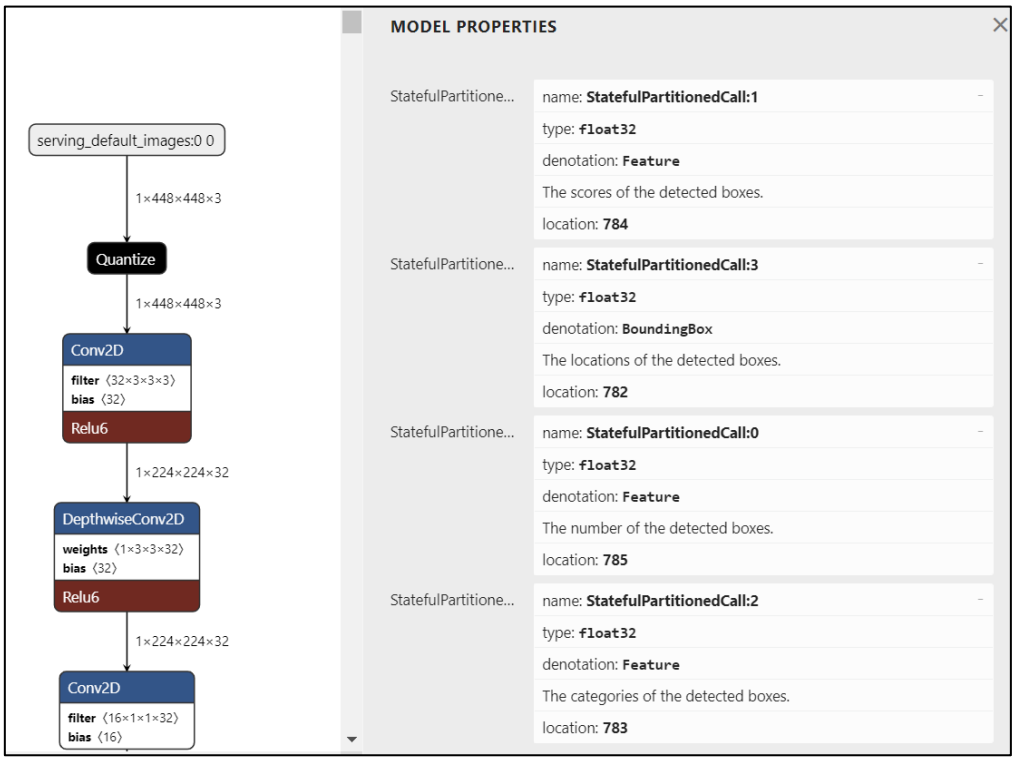


Figure 5.34 Output Order After Change

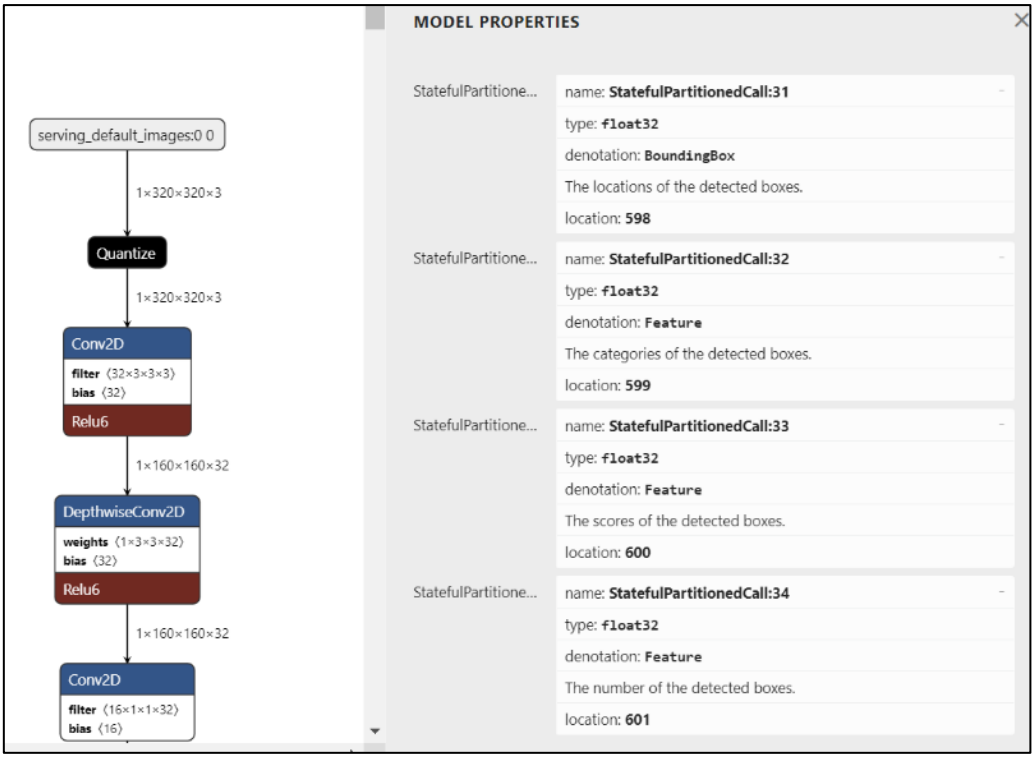


Figure 5.35 Supported Output Order

Chapter 6

System Evaluation and Discussion

6.1 Testing for Image Detection Model

77 images downloaded from the Internet are used to perform testing for the image classification model. After testing, the accuracy of the model is 93.51%. From here, the accuracy of the model is quite high. Figure 6.2 has showed the visualization of the testing results. The labels in red colour indicate that the model has predicted that image wrongly.

```
In [13]: loss, accuracy = model.evaluate(test_data)

         3/3 [==============================] - 6s 1s/step - loss: 0.7893 - accuracy: 0.9351
```

Figure 6.1 Testing for Image Detection Model

Figure 6.2 Visualize Testing Results of Image Classification Model

6.2 Testing for Object Detection Model

455 images are used to perform testing for the object detection model. Figure 6.3 has showed the testing results of the model. The overall accuracy of the model is 79.47%. The average precision of the model for large objects is 79.47%, and the average recall for large objects is 83.08%. High precision indicates that the model is correctly predicting a high proportion of true positive instances while minimizing false positive instances [23]. High recall indicates that a high proportion of actual positives (instances that belong to the positive class) have been correctly identified by the model as positive [23]. The accuracy of the model can be considered good as it can predict most of the attractions correctly.

```
In [10]: model.evaluate_tflite('C:/Users/user/Desktop/Object Detection/model5/ARmodelv1.tflite', val_data)

         455/455 [==============================] - 2340s 5s/step

Out[10]: {'AP': 0.7946509,
          'AP50': 0.986597,
          'AP75': 0.9170687,
          'APs': -1.0,
          'APm': -1.0,
          'APl': 0.7946509,
          'ARmax1': 0.7977913,
          'ARmax10': 0.8307749,
          'ARmax100': 0.8307749,
          'ARs': -1.0,
          'ARm': -1.0,
          'ARl': 0.8307749,
          'AP_/GuMiao': 0.80400676,
          'AP_/GuMiaoBoard': 0.8065805,
          'AP_/GuMiaoCaiMen': 0.9086634,
          'AP_/GuMiaoDoor': 0.89724797,
          'AP_/GuMiaoDoorGods': 0.75081664,
          'AP_/GuMiaoGodOfWealth': 0.72393376,
          'AP_/GuMiaoPlaque': 0.92321837,
          'AP_/GuMiaoStoneLion': 0.7256429,
          'AP_/GuMiaoWishingHall': 0.7643709,
          'AP_/templeLuBan': 0.8196966,
          'AP_/templeLuBanCouplet': 0.47451413,
          'AP_/templeLuBanDoorGods': 0.81175745,
          'AP_/templeLuBanPlaque': 0.9015552,
          'AP_/templeLuBanStoneLion': 0.79696226,
          'AP_/templeLuBanTitle': 0.8107968}
```

Figure 6.3 Testing for Object Detection Model

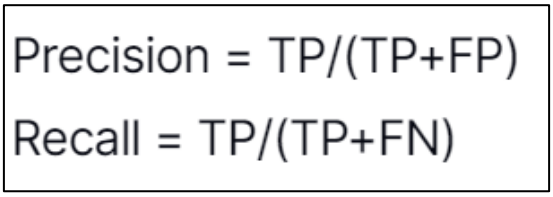


Figure 6.4 Formula for Precision and Recall

Figure 6.5 Visualize Testing Result of Object Detection Model

6.3 Use Case Testing for Application

Use Case	Test Case	Expected Result	Actual Result	Pass/Fail
Perform searching through the attractions list	Click on an attraction on the attractions list	Navigate to the attraction detail page	Navigate to the attraction detail page	Pass
	Click on a meaningful corner of an attraction on the meaningful corners list	Navigate to the meaningful corner detail page	Navigate to the meaningful corner detail page	Pass
Perform single attraction recognition function	Click on the "Camera" floating button	Open phone camera	Open phone camera	Pass
	Take a photo using the phone camera	Upload the image taken to the application for single attraction recognition	Upload the image taken to the application for single attraction recognition	Pass
	Click on the "Gallery" floating button	Open phone gallery	Open phone gallery	Pass
	Select an image from the gallery	Upload the image selected to the application for single attraction recognition	Upload the image selected to the application for single attraction recognition	Pass
	Successfully recognize the attraction in the image	Display the recognition result and navigate to the relevant page of the	Display the recognition result and navigate to the relevant page of the	Pass

		recognised attraction	recognised attraction	
	Fail to recognise the attraction in the image	Display an error message dialog box	Display an error message dialog box	Pass
Perform multiple attractions recognition function	Click on the "Camera" floating button	Open phone camera	Open phone camera	Pass
	Take a photo using the phone camera	Upload the image taken to the application for multiple attractions recognition	Upload the image taken to the application for multiple attractions recognition	Pass
	Click on the "Gallery" floating button	Open phone gallery	Open phone gallery	Pass
	Select an image from the gallery	Upload the image selected to the application for multiple attractions recognition	Upload the image selected to the application for multiple attractions recognition	Pass
	Successfully recognize the attractions in the image	Display multiple bounding boxes to point out the recognised attractions in the image	Display multiple bounding boxes to point out the recognised attractions in the image	Pass
	Click one of the bounding boxes	Navigate to the relevant page of the attraction	Navigate to the relevant page of the attraction	Pass

Perform real-time single attraction detection function	Scan through the attraction to be detected	Send every frame to the image classification model to perform detection	Send every frame to the image classification model to perform detection	Pass
	Detect an attraction in the frame	Display a bounding box to point out the detected attraction and a navigation button on the screen	Display a bounding box to point out the detected attraction and a navigation button on the screen	Pass
	Click on the navigation button	Navigate to the relevant page of the attraction	Navigate to the relevant page of the attraction	Pass
Perform real-time multiple attractions detection function	Scan through multiple attractions to be detected	Send every frame to the object detection model to perform detection	Send every frame to the object detection model to perform detection	Pass
	Detect any attraction in the frame	Display multiple bounding boxes to point out the detected attractions on the screen	Display multiple bounding boxes to point out the detected attractions on the screen	Pass
	Click one of the bounding boxes	Navigate to the relevant page of the attraction	Navigate to the relevant page of the attraction	Pass

Table 6.1 Use Case Testing for Application

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

The development of the mobile tour guide application with on-site attractions recognition for Kampar temples is completed. The project can be considered as achieved its SMART objectives. With the attractions catalogue module, users can easily find an attraction from the list. The users are able to view the history or relevant information of an attraction. A list that contains meaningful corners of the attraction is also provided to users. For the attractions recognition module, the application provides users with the ability to upload an image that lets the models recognise the attractions in the image. The module contains 2 functions, the single attraction recognition function and the multiple attractions recognition function. It enables the flexibility of users in choosing the suitable function to be performed based on their demands. Besides, the real-time attractions detection module provides users with a faster and more interactive way to get to know about an attraction. The users can just use the phone camera to scan through the attractions, and then the models will detect the attractions on the screen. This module also has 2 functions, the real-time single attraction detection function and the real-time multiple attractions detection function.

The models used in the application are developed using image classification and object detection techniques. Thus, there are 2 models incorporated into the application. These models cannot compare which is better, as both of them are developed based on different needs. The image classification model is developed to recognise only 1 attraction in the image or frame. It can fulfil the need of users if the users just want to know a particular attraction. Most of the time, they will just upload an image that contains only that particular attraction if they want to know the information about the attraction. Therefore, the image classification model can provide users with a more convenient way to achieve this. Furthermore, the object detection model is developed to recognise multiple attractions in the image or frame. This model aims to fulfil the need of users who want to know where are the meaningful corners in an attraction. The

accuracies of both models are considered good, the image classification model has a 93.51% accuracy and the object detection model has a 79.47% accuracy.

In conclusion, the concept of the project is proved to be feasible. The application developed might help to improve tourism in Kampar. It enhances the on-site experience of the users when they travel around Kampar. Meanwhile, the application propagates the history and relevant information of the attractions to users. The users shall have a more interesting and interactive way to get to know about Kampar. Thus, the issue of the younger generation gradually forgetting about the history of the older generation might be solved. This project is hoped to have good contributions to the tourism and development of Kampar.

7.2 Recommendation

Although the application is developed successfully, some of the features are still can be improved. First, due to the time constraint, the project is narrowed down to recognise 2 temples in Kampar. The scope of the attractions recognition can be expanded by adding more interesting attractions in order to make the application becomes more powerful and useful for tourists. The accuracy of the models is also can be improved to perform the recognition more accurately. Next, some additional interesting features may be developed and implemented to improve the experience of tourists. For example, besides displaying the information and history, some interactive mini-games can be added after the application successfully recognises the attraction. It can help to attract the kids in using the application. While they are playing the mini-games, they might also learn some information and the history of the attractions throughout the games. In addition, the project may partner with local businesses and attractions to promote the attractions and attract more users. While promoting the attractions, the business will also gain benefits from it. It can also help provide more accurate and up-to-date information about the attractions and improve the overall user experience. Improving and promoting tourism in Kampar is not easy, it might need the efforts of different parties in order to achieve this.

REFERENCES

[1] N. Anglia, "Why is it important to study history?," *Why is It Important to Study History? | Nord Anglia Education*, 29-Apr-2020. [Online]. Available: https://www.nordangliaeducation.com/news/2020/04/29/why-is-it-important-to-study-history#:~:text=Through%20history%2C%20we%20can%20learn. [Accessed: 28-Jun-2022].

[2] T. Promsaka and N. Sakolnakorn, "Important Factors and Policies That Contributed to Tourism in Malaysia Between 1991 and 2018," International Journal of Innovation, Creativity and Change. www.ijicc.net, vol. 14, p. 2020, [Online]. Available: https://www.ijicc.net/images/Vol_14/Iss_12/15210_Sakolnakorn_2020_E1_R.pdf. [Accessed: 30-Jun-2022].

[3] H. Kennedy-Eden, and U. Gretzel, "A taxonomy of mobile applications in tourism," 2012. [Online]. Available: https://ro.uow.edu.au/cgi/viewcontent.cgi?article=3559&context=commpapers. [Accessed: 30-Jun-2022].

[4] R. Fatima, I. Zarrin, M. A. Qadeer and M. S. Umar, "Mobile travel guide using image recognition and GPS/Geo tagging: A smart way to travel," 2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN), 2016, pp. 1-5.

[5] V. Meel, "Image Recognition: The Basics and Use Cases (Guide in 2021)," viso.ai, Sep. 24, 2021. [Online]. Available: https://viso.ai/computer-vision/image-recognition/. [Accessed: 05-Jul-2022].

[6] V. Parikh, M. Keskar, D. Dharia and P. Gotmare, "A Tourist Place Recommendation and Recognition System," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 218-222.

[7] M. Jiafa, W. Weifeng, H. Yahong, and S. Weiguo, “A scene recognition algorithm based on deep residual network,” Systems Science & Control Engineering, vol. 7, no. 1, pp. 243–251, Jan. 2019.

[8] Y. Li, X. Li, and C. Yue, “Recognition of Tourist Attractions,” Nov. 24, 2021.

[9] N. M. Firdaus, D. Chahyati and M. I. Fanany, “Tourist Attractions Classification using ResNet,” 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2018, pp. 429-433.

[10] “Software Prototyping Model and Phases,” GeeksforGeeks, May 01, 2021. [Online]. Available: https://www.geeksforgeeks.org/software-prototyping-model-and-phases/. [Accessed: 08-Aug-2022].

[11] K. Sharma, “Top 12 Software Development Methodologies & its Advantages / Disadvantages | TatvaSoft,” Tatvasoft.com, Dec. 25, 2020. [Online]. Available: https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/. [Accessed: 08-Aug-2022].

[12] L. Cianci, “Best ides for flutter in 2022,” LogRocket Blog, 18-Feb-2022. [Online]. Available: https://blog.logrocket.com/best-ides-flutter-2022/#:~:text=Android%20Studio%20is%20a%20much%20heavier%20IDE%20than%20VS%20Code,when%20you%20load%20your%20project. [Accessed: 08-Aug-2022].

[13] “Flutter tutorial,” Tutorials Point. [Online]. Available: https://www.tutorialspoint.com/flutter/index.htm. [Accessed: 09-Aug-2022].

[14] “Dart programming tutorial - javatpoint,” www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/dart-programming. [Accessed: 10-Aug-2022].

[15] “Image classification with tensorflow lite model maker,” TensorFlow. [Online]. Available: https://www.tensorflow.org/lite/models/modify/model_maker/image_classification. [Accessed: 16-Aug-2022].

[16] "Higher accuracy on vision models with EfficientNet-Lite," The TensorFlow Blog. [Online]. Available: https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html. [Accessed: 22-Aug-2022].

[17] "Post-training quantization :   Tensorflow Lite," TensorFlow. [Online]. Available: https://www.tensorflow.org/lite/performance/post_training_quantization. [Accessed: 20-Aug-2022].

[18] T. V. Team, "Keras layers - everything you need to know," TechVidvan, 05-Jul-2021. [Online]. Available: https://techvidvan.com/tutorials/keras-layers/. [Accessed: 28-Jul-2022].

[19] "Keras - dropout layers," Tutorials Point. [Online]. Available: https://www.tutorialspoint.com/keras/keras_dropout_layers.htm. [Accessed: 28-Jul-2022].

[20] "Keras - dense layer," Tutorials Point. [Online]. Available: https://www.tutorialspoint.com/keras/keras_dense_layer.htm. [Accessed: 28-Jul-2022].

[21] "Objectives and goals: Writing meaningful goals and smart objectives," *Objectives and goals: Writing meaningful goals and SMART objectives - MN Dept. of Health*. [Online]. Available: https://www.health.state.mn.us/communities/practice/resources/phqitoolbox/objectives.html#:~:text=A%20SMART%20objective%20is%20one,organization%20will%20achieve%20a%20goal. [Accessed: 17-Apr-2023].

[22] G. Boesch, "LabelImg for image annotation," *viso.ai*, 09-Jul-2022. [Online]. Available: https://viso.ai/computer-vision/labelimg-for-image-annotation/#:~:text=LabelImg%20is%20a%20straightforward%20and,files%20in%20PASCAL%20VOC%20format. [Accessed: 20-Apr-2023].

[23] “Precision vs. recall: Differences, use cases & evaluation,” *V7*. [Online]. Available: https://www.v7labs.com/blog/precision-vs-recall-guide. [Accessed: 23-Apr-2023].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 1**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Review the FYP1 report.
- Review the model and application developed in FYP1.

2. WORK TO BE DONE

- Convert the FYP1 report to FYP2 report.
- Check and note down what should be done in FYP2.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 2**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

Redefine the objectives of the project by following SMART criteria.

2. WORK TO BE DONE

Study and explore how to implement the real-time attractions detection module.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 3**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Study and explore the object detection technique.
- Review some tutorials on how to develop an object detection model.
- Review some tools that can be used to label the image dataset.

2. WORK TO BE DONE

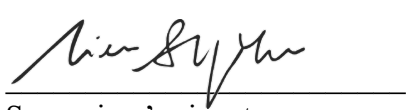
Tidy up the existing image dataset and prepare a dataset for object detection.

3. PROBLEMS ENCOUNTERED

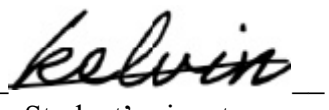
No.

4. SELF EVALUATION OF THE PROGRESS

In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 4**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

Use LabelImg software to label the image dataset.

2. WORK TO BE DONE

Start to train an object detection model.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In Progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 5**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

Train an object detection model.

2. WORK TO BE DONE

- Start to develop the functions of the real-time attractions detection module.
- Incorporate the object detection model into the application.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 6**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Develop the real-time single attraction detection function that uses the image classification model.
- Develop the real-time multiple attractions detection function that uses the object detection model.

2. WORK TO BE DONE

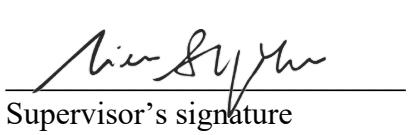
Researching for the reason that caused the problem.

3. PROBLEMS ENCOUNTERED

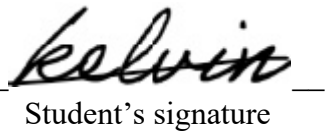
The application will crash when it wants to use the object detection model.

4. SELF EVALUATION OF THE PROGRESS

In progress.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 7**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Success in finding the cause of the problem encountered last week.
- Success in finding a solution to solve the problem.
- Retrain the object detection model.

2. WORK TO BE DONE

- Perform testing on the model before incorporating it into the application.
- Improve and add some features to the application.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 8**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Develop a multiple attractions recognition function.
- The real-time multiple attractions detection function is successfully developed.

2. WORK TO BE DONE

Perform testing on the overall features of the application.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 9**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

Perform refinement on some features of the application.

2. WORK TO BE DONE

- Perform on-site testing at the temples.
- Start to write the report.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 10**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Draw the diagrams that illustrate the design and infrastructure of the application.
- Compile the processes of training the models and developing the application.

2. WORK TO BE DONE

Record and write the testing results.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 11**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

- Write and describe the testing results.
- Write the conclusion and recommendation of the project.

2. WORK TO BE DONE

Compose the report.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	**Study week no.: 12**
Student Name & ID: Teh Kai Wen 19ACB01435	
Supervisor: Dr Liew Soung Yue	
Project Title: Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples	

1. WORK DONE

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

Check and finalize the report.

2. WORK TO BE DONE

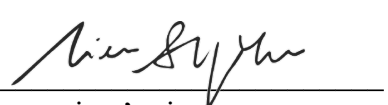
Prepare presentation and demonstration.

3. PROBLEMS ENCOUNTERED

No.

4. SELF EVALUATION OF THE PROGRESS

In progress.



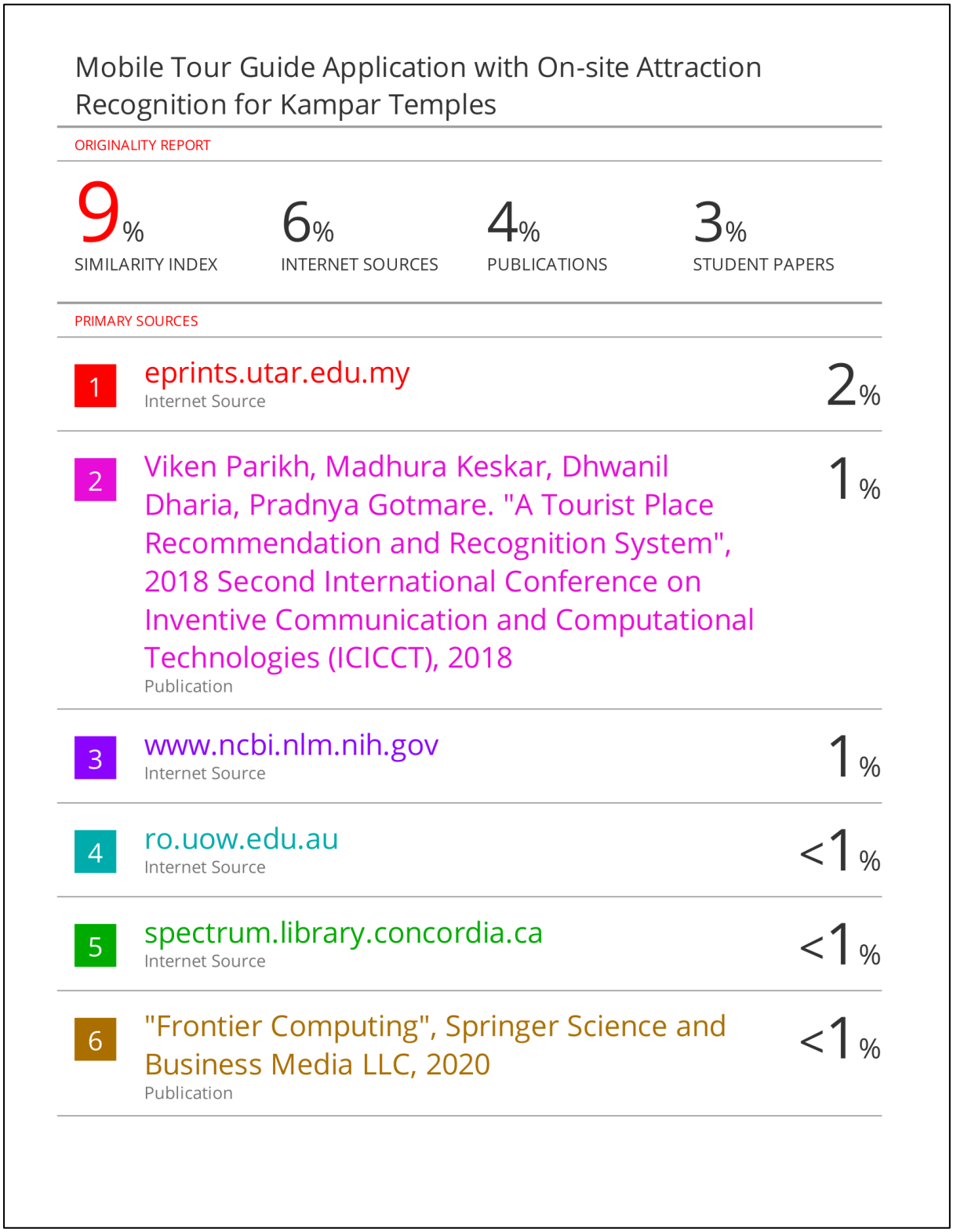
Supervisor's signature

Student's signature

POSTER


MOBILE TOUR GUIDE APPLICATION WITH ON-SITE ATTRACTION RECOGNITION FOR KAMPAR
DONE BY:
TEH KAI WEN
SUPERVISED BY:
DR. LIEW SOUNG YUE
Introduction
A mobile tour guide application with on-site attraction recognition for Kampar. Currently, the tourism downturn in Kampar will cause some issues. Hence, this application is developed to improve tourism in Kampar. Due to time constraints, the project will focus on recognising 2 temples in Kampar.
Methods
• Attractions catalogue module
• Attractions recognition module
• Real-time attractions detection module
Image Classification
Application
Object Detection
Discussion & Conclusion
The concept of this project is proven to be feasible. The application will enhance the on-site experience of the tourists. It provides an interesting and interactive way for letting tourists get to know about Kampar. In short, the project is hoped to have good contributions to the tourism and development of Kampar.
jupyter
TensorFlow
Flutter
Visual Studio Code
Android Studio

PLAGIARISM CHECK RESULT



Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples

ORIGINALITY REPORT

9%	6%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Viken Parikh, Madhura Keskar, Dhwanil Dharia, Pradnya Gotmare. "A Tourist Place Recommendation and Recognition System", 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018 Publication	1%
3	www.ncbi.nlm.nih.gov Internet Source	1%
4	ro.uow.edu.au Internet Source	<1%
5	spectrum.library.concordia.ca Internet Source	<1%
6	"Frontier Computing", Springer Science and Business Media LLC, 2020 Publication	<1%

Plagiarism Check Result

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

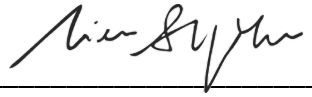
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Teh Kai Wen
ID Number(s)	19ACB01435
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONOURS)
Title of Final Year Project	Mobile Tour Guide Application with On-site Attraction Recognition for Kampar Temples

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 9 % **Similarity by source** Internet Sources: 6 % Publications: 4 % Student Papers: 3 %	Within the required range.
Number of individual sources listed of more than 3% similarity: 0	Within the required range.
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: Liew Soung Yue

Date: 25/4/2023

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB01435
Student Name	Teh Kai Wen
Supervisor Name	Dr Liew Soung Yue

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.
NA	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)
NA	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
NA	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: 25/4/2023