

**TREASURE HUNT MOBILE APPLICATION
WITH OBJECT RECOGNITION
BY
THEW THUNG LIANG**

**A PROPOSAL
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Perak Campus)**

May 2018

REPORT STATUS DECLARATION FORM

Title: _____

Academic Session: _____

I _____
(CAPITAL LETTER)

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

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

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

Supervisor's name

Date: _____

Date: _____

**TREASURE HUNT MOBILE APPLICATION
WITH OBJECT RECOGNITION
BY
THEW THUNG LIANG**

**A PROPOSAL
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Perak Campus)**

May 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled “**TREASURE HUNT MOBILE APPLICATION WITH OBJECT RECOGNITION**” 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 : _____

Date : _____

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this challenging project. A special gratitude I would like to give to my supervisor, Dr. Ng Hui Fuang, who provided me thought-provoking feedbacks and encouragement throughout this project life span.

Not to forget my course mate, Ng Kean Sean who provided his valuable insights on training the Faster R-CNN algorithm.

Lastly, I would like to thank my family for their unconditioned love and monetary support throughout my study.

Abstract

Location-based mobile application such as treasure hunt game have great potential to enable users to explore the environment in an engaging and immersive way. There are many existing mobile application based on this idea, like Actionbound, but are not designed as meaningful and engaging experiences. Generally these application and frameworks are just merely asking player to perform some tasks such as taking a short quiz and taking a group photo at a specified location which don't really deliver the joy and excitement to the user for looking an unknown physical object by the hints.

With the emergence of Deep Learning technologies, recognition of physical objects can be done in short amount of time; higher accuracy; less sensitive to different angles and illumination. Besides, the increment computational power of mobile devices opens up the possibility of running complex object detection algorithm.

Therefore, this project proposes a real-time object recognition mobile application to provide the real fun of treasure hunting and also enhance the learning experience at an unfamiliarized place. The novelty of this project is by combining object detection algorithm with mobile application. To achieve this, this project will be using Faster Regional based Convolutional Neural Network architecture which runs on Inception network Version 2.

Table of Contents

Title Page	i
Declaration of Originality	ii
Abstract	6
Table of Contents	7
List of Tables	9
List of Figures	10
List of Abbreviation	11
Chapter 1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Background and motivation	3
1.3 Project Objectives	5
1.4 Proposed Approach	6
1.5 Report Organization	8
Chapter 2 LITERATURE REVIEW	9
2.1 Review of Existing System	9
2.1.1 Actionbound	9
2.1.2 Huntzz – Treasure Hunt	11
2.1.3 Geocaching	13
2.2 Comparison Analysis	15
Chapter 3 PROPOSED METHOD/APPROACH	16
3.1 Design Specification	16
3.1.1 Methodology and General Work Procedure	16
3.1.2 General Work Procedures	17
3.2 Tools to use	19
3.2.1 Software Requirements	19
3.2.2 Programming Languages	20
3.3. Requirement Specifications	21
3.3.1 Functional Requirements	21
3.3.2 Non-Functional Requirements	21
Chapter 4 System Design	22
4.1 System Architecture	22
4.2 Tensorflow Object Detection API	24

4.2.1 Object Recognition Architecture	24
4.2.2 Base network.....	32
4.3 Use-case diagram	37
4.4 System Flowchart.....	38
4.5 System Flow with UI and detailed explanation	40
4.6 Entity-Relationship Diagram	47
4.7 Timeline	48
Chapter 5 Implementation and Testing.....	49
5.1 Implementation Issues and Challenges	49
5.2 Testing.....	50
5.2.1 Different test scenarios.	50
Chapter 6 CONCLUSION	52
6.1 Project Review and Discussion.....	52
6.2 Future Work.....	53

List of Tables

Table 2-1 Comparison analysis.....	15
Table 5.2.1.1 Display user's current location	50
Table 5.1.2.3 Click on object detection feature	51
Table 5.1.2.4 Prompt out the detected object info when it's detected	51
Table 5.1.2.5 Showing user current scores	51

List of Figures

Figure 1-1Faster R-CNN architecture overview.....	6
Figure 2-1-1-1 Actionbound home page.....	10
Figure 2-1-1-2 List of treasure hunt missions.....	10
Figure 2-1-1-3 Find out the location.....	10
Figure 2-1-1-4 Complete mission	10
Figure 2-1-2-1 List of treasure hunt locations	11
Figure 2-1-2-2 Details of treasure hunt.....	11
Figure 2-1-2-3 Follow maps to find virtual treasure hunt.....	12
Figure 2-1-2-4 Answer clues to gain guide	12
Figure 2-1-2-5 player will receive badge after reaching the next level	12
Figure 2-1-3-1 Green icons indicate it's available for non-premium member.....	13
Figure 2-1-3-2 Way on getting to the treasure.....	14
Figure 2-1-3-3 Details of treasure	14
Figure 2-1-3-4 English description	14
Figure 2-1-3-5 Malay descption	14
Figure 3-1Evolutionary Prototyping	166
Figure 4-1System Architecture	22
Figure 4-2-1 Faster R-CNN Architecture	165
Figure 4-2-2 RPN.....	26
Figure 4-2-3 Output of RPN	27
Figure 4-2-4 Anchor	28
Figure 4-2-5 Refine anchor based on subject	28
Figure 4-2-6 ROI pooling	31
Figure 4-2-7 Inception V2 architecture.....	32
Figure 4-2-8 Factorization into smaller convolutions.....	33
Figure 4-2-9 Spatial Factorization	34
Figure 4-2-10 Higher Dimension Representation.....	35
Figure 4-2-11 Relationship between 3 different Inception Module.....	36
Figure 4-2-12 Inception V2 architecture in tabular form.....	36
Figure 4-3 Use-case diagram	37
Figure 4-4 System Flowchart.....	38
Figure 4-5-1 Homepage of THOR.....	40
Figure 4-5-2 Recognizing Object.....	41
Figure 4-5-3 Congratulations page	42
Figure 4-5-4 Treasure Details	43
Figure 4-5-5 AR Geolocation of FICT	44
Figure 4-5-6 Treasure Lists.....	45
Figure 4-5-4 User Profile.....	46
Figure 4-6 Entity Relationship Diagram.....	47
Figure 4-7 Timeline	48

List of Abbreviation

THOR	Treasure Hunt mobile application with Object Recognition
POI	Point of Interests
AR	Augmented Reality
Faster R-CNN	Faster Region-based Convolutional Neural Network
COCO	Common Objects in Context
CNN	Convolutional Neural Network
Region Proposal Network	RPN
Intersection over Union	IoU
Entity Relationship Diagram	ERD

Chapter 1 INTRODUCTION

1.1 Problem Statement

Creating a memorable and nostalgic experience to visitor has always been an annoying problem to management echelon when managing a huge area of property such as universities, shopping mall, big firms who have a large workplace, tourist destination, museum and etc. These places have one thing in common which is too big to be discovered. While some places like historical sites or museum can easily get the visitors' attraction as these places are place where human culture for posterity has been preserved. Visitors can find their own identity, learning other's culture and history, discover the life of ancestors'. However, places that are very common and not bound with a location, culture or that is not historic such as shopping mall or universities might be hard in getting visitor's attention and time for them to familiar themselves. Taking university as an example, it's very typical to see most of the universities will held campus guide prior to the beginning of new semester for incoming freshmen to make themselves familiar with the campus that they are going to study in next couple of years. However, do all the visitors really remember all the places they have visited or an exit route to escape from a building if an emergency occurs? Once they have lost their interest, they would prefer to stand at the ground playing with their mobile phone and having a bored impression on that place. Without a gamified outdoor exploration activity for visitors, they would easily lose motivation to discover the place. This really matter as if university is able to create some fun and interactive activities during campus tour, students might get themselves accommodate with the campus environment quickly.

This leads to the **existing imperfect solution** which is by organizing a **traditional treasure hunt** in shopping mall or universities that can encourage visitors to explore and discover different places and also promote interaction between them. Visitors will be asked to install a treasure hunt mobile game application on their mobile phone. The way of how the treasure hunt mobile game application works is by asking players to fill up some quiz and scan the QR codes when they have reached the hinted location. **Such scenarios don't really bring out the true essence and excitement** when playing treasure hunt without looking for a specific real life object.

Chapter 1 INTRODUCTION

How would this kind of treasure hunt mobile game application really helps player in remembering the places, building or the facilities. How players can easily remember the coolest and iconic object that they have seen if they are not asked to look for it. It's bored and players wouldn't remember much about the specific real life object/place after the game ends. The true essence of treasure hunting is to look for any physical objects that are identical and can represent for that place. Just like the treasure hunt movies or video games story line, the character would need to look for a specific treasure in order to achieve the mission. It is really exciting when seeing the characters overcome all the obstacles to find and get the treasure.

Moreover, current treasure hunt mobile games rely on GPS to determine whether players have reached a certain location. Such mechanism pose a big issue to players and mobile app developers when playing in indoor as **GPS doesn't work well within and near buildings** (Jacob, 2015).

1.2 Background and motivation

Background

Gamification has become a new word on few years back ago which was 6/30/2014. The fundamental concept of this new word isn't new, but the term is a 21st-century addition to the English lexicon (Anon., 2018). The word allude to the embodiment of game elements, like reward systems and point, to task or mission to lure people participate in. Gamification is really efficient because it taps into people's nature desires for competing with others and getting rewards after accomplish a mission.

A common example that could be seen in real life is the process of "levelling up" at a coffee shop as customer buy more drinks from them. Every purchase would get a stamp on a small punch cards. Once the stamps has accumulated enough, they can exchange for a free drink. This is the reason being for Gamification. It could encourage user to reach their goals by making the whole mundane process into addictively game-like (BADERTSCHER, 2018).

There are a lot of studies prove that the development of gamified learning activities are able to increase student engagement in familiarize themselves in a new environment. In these recent years, location-based applications have shown a great potential to let users explore the environment in an immersive and engaging way.

Having said that, most of the current treasure hunt mobile application are almost the same which is by creating a multi-staged quests such as taking a quiz, taking a groupie, when players have reach the specified location. For all these kind of mobile games will involve a lot of calculation at the server for analyzing the submitted quests from players and requires a large database storage to store players' picture.

Motivation

Humans are all curious creatures. We love to explore new places and discover new things. When Gamification is integrated into learning, we tend to perform much better and able to connect new information with game contents (Roland, 2016). Therefore, treasure hunting is always seen as a best tools to be used in outdoor and indoor learning.

Now, with the integration of powerful object recognition technology developed with Deep Learning. Treasure hunt mobile game application has never been so adventurous. Object recognition technology opens up treasure hunt mobile game application with numerous of possibilities. Treasure hunt mobile game application can now be played in indoor and outdoor. It can be used in wide areas such as education, retail, tourism and outdoor games. Retailers can hold shopper's long-term concentration, shoppers will be given discount when found a treasure in shopping mall, students would love to have biology treasure hunt, companies can organize treasure hunt to boost up teamwork culture and etc.

This leads to my novel idea which create a location-based treasure hunt mobile game application that inject with the long lost story-based and playful element which was finding a physical object in the real world. This mobile game application is the first in the world that integrate object recognition into location-based treasure hunt mobile game. This idea can bring a positive impact to the current location-based treasure hunt mobile game application and also to the player.

Therefore, I would like to develop a location-based treasure hunt mobile game that named as THOR which is an abbreviation of Treasure Hunt with Object Recognition.

THOR aims to deliver an interactive treasure hunt mobile game that is running on Android platform. Deep learning technology will be used for recognizing 3D physical objects in real life. To uplift the gaming experience, Augmented Reality will be used to display virtual information board at certain locations to let players know more about their surroundings.

1.3 Project Objectives

- I. To provide an interactive and fun learning environment for players to learn about their surroundings.
- II. To explore and extend the use of object recognition and AR in outdoor and indoor game field.
- III. To bring playing treasure hunt mobile game to a whole new level.
- IV. To enable the possibility of playing treasure hunt in indoor.

1.4 Proposed Approach

This project can be divided into two parts which are recognizing physical objects in real world and also providing virtual information about user surrounding.

Recognizing physical objects

Recently, there are a lot of people talking about Artificial Intelligence, Machine Learning, and Deep Learning. But what is the relationship between them? To put it simply, Deep Learning is a subset of Machine Learning, and Machine Learning is a subset of Artificial Intelligence. (Skymind, 2017)

There are dozens architecture of Deep Learning and for this application we will be using Faster R-CNN which is well suited to object recognition with images where high accuracy and low latency is desired.

Faster R-CNN comes with 2 interlinked modules: a region proposal network (RPN) for generating region proposals to be used in the second module, Fast R-CNN detector to detect objects. In the RPN module, there's a small window slides over the convolution feature map with multiple anchors at each sliding window location. The RPN outputs a bounding boxes (region proposals) and predicted class as Fast R-CNN module does. (Ren, et al., 2015)

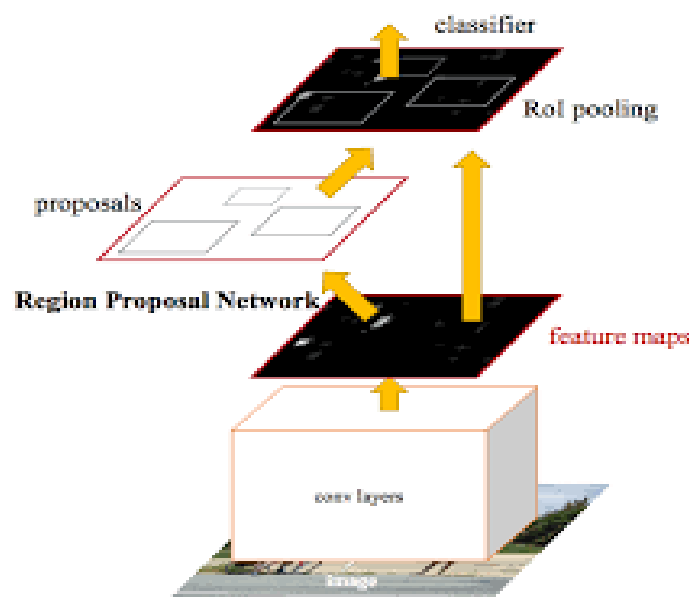


Figure 1-1Faster R-CNN architecture overview

Providing virtual information about user surrounding

To achieve this function, it is required to use a mixture of (AR) Augmented Reality and geospatial data determining mechanism. The virtual information can be called as geo markers or POI which is an abbreviation of points of interests.

The placement of virtual information is placed at the specified location and would appear when user is at nearby. The exact position of the smartphone is determined through the device's GPS. When user is near to the specified location, the virtual information would appear on screen when the device camera is opened.

1.5 Report Organization

This project has 6 chapters in total. Chapter 1 gives a brief introduction about the project existence, its objectives and the approach that has been used to achieve its objectives. Chapter 2 is to research the similar technologies or solutions that provide the similar features and a detailed comparison table between this project with existing solutions.

Moreover, chapter 3 will talk about the detailed description of this project architecture in words and charts. Chapter 4 tells about the development methodology and the tools that were used in implementing the ideas to an actual solution.

Chapter 5 describes the challenges and the testing method to validate the system performance and chapter 6 draws a final conclusion on this project achievement and some discussion on it's future work.

Chapter 2 LITERATURE REVIEW

2.1 Review of Existing System

There are some similar existing treasure hunt apps in the market and 3 of it have been taken for review.

2.1.1 Actionbound

Launched in 2012 at Germany and it is similar to a classical scavenger hunt or geocaching that allows players to play in real life but it's more than that.

Players will have to login to actionbound.com first to create or edit a bound. What bound means over here is a treasure to be looked for. So in this app, players can set a few stages where each stages come with different difficulties and form such as finding a physical object based on the given photo/hint, look for a QR code which serve as a bound and stucked on physical object, complete a mission like taking a groupfie, answering a quiz, or taking a short survey. User can choose to share their result to their social media account after completes each stage. After the game ends, each user will be shown with their own scores and ranking.

From the main page, user can choose to find a bound, scan for the QR code or search for the nearby bounds. User can utilize the search function to look for a bound or search for nearby location but the results are shown in German language which degrades the user experience. Bounds which are within the radius of 50km from user will appear automatically when user press for the bounds nearby icon.

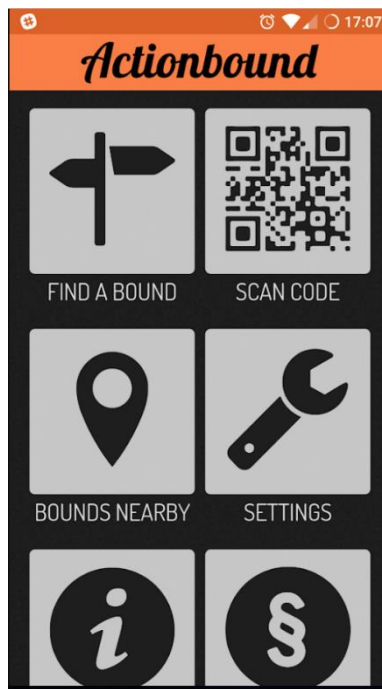


Figure 2-1-1-1 Actionbound home page

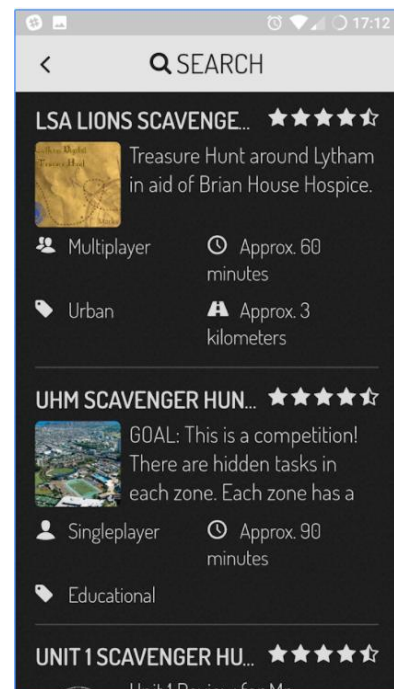


Figure 2-2-1-2 List of treasure hunt missions

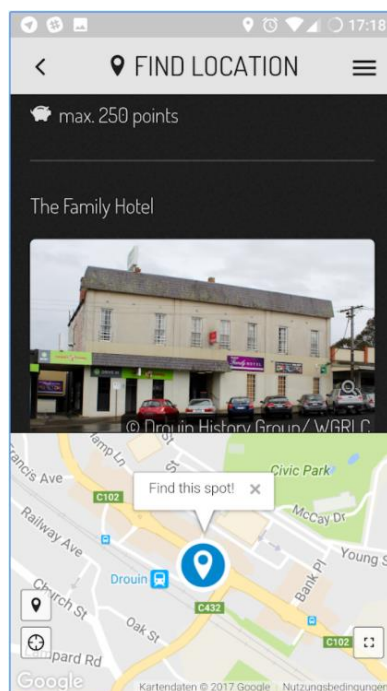


Figure 2-1-1-3 Find out the location

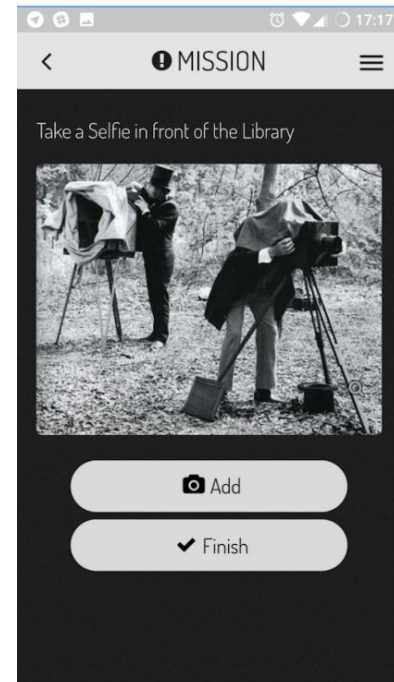


Figure 2-1-1- 4 Complete mission

2.1.2 Huntzz – Treasure Hunt

As featured in BBC Click, this is another treasure hunt app that comes with 2 themes: modern and classic. By entering into the first page, around 50 renowned places around the world are shown as where the treasures are located.

By entering into one of the places, it will show the distance needed to take and with some hints which gives some brief introduction about the place. Not all the places are free to played, some will need to pay for 1.25 pound before allowed to play. Besides, user can open the map that shows where the places are located.

This app only performs a very simple interaction with user and lack with some interesting features such as restricting user to customize the treasure location by themselves and will have to wait for the app developer to approve user request. Besides, this app will automatically play classical background music when the app starts that might make user feel uncomfortable. This app requires user to purchase for the treasure before they are allowed to play.

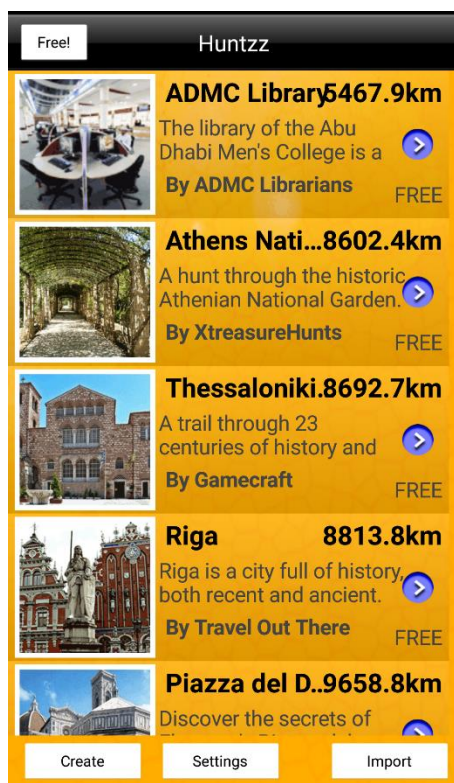


Figure 2-1-2-1 List of treasure hunt locations

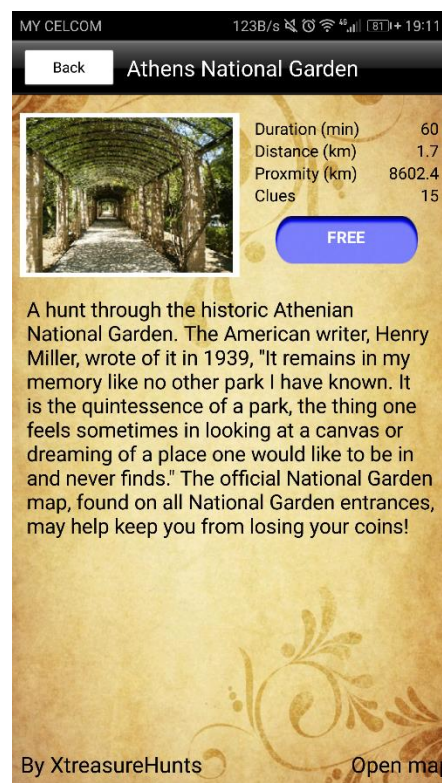


Figure 2-1-2-2 Details of treasure hunt

Chapter 2 LITERATURE REVIEW

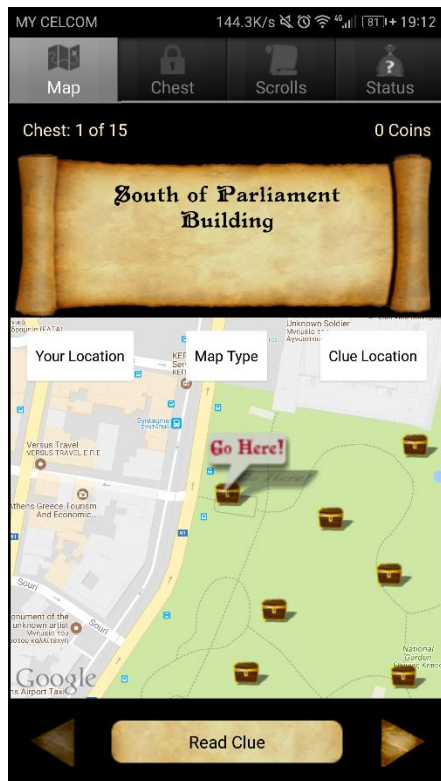


Figure 2-1-2-3 Follow maps to find virtual treasure hunt

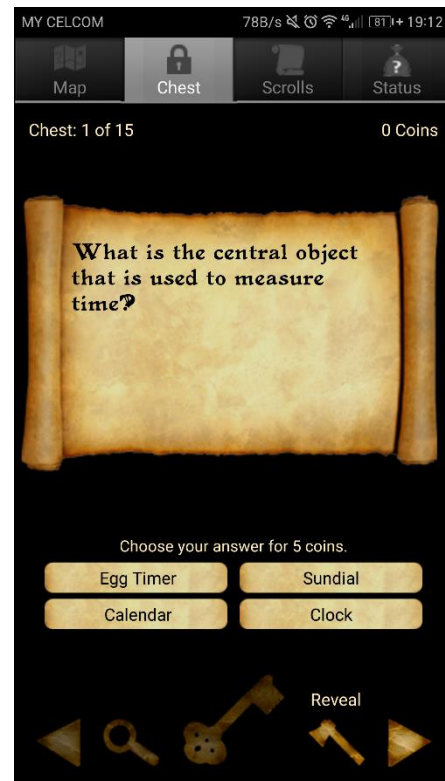


Figure 2-1-2-4 Answer clues to gain guide



Figure 2-1-2-5 player will receive badge after reaching the next level

2.1.3 Geocaching

The world largest treasure-hunting community. It has friendly user interface and easy to use. There are multiple types of treasure such as moving treasure but most are requiring players to be a premium member before playing.

Non-premium members can only look for physical container which placed by other player, and also able to look for nearby events. Premium members can have much choices.

When click on the physical container on the map, it will show the ways to there and showing the details for looking the treasure such as difficulty, terrain and the treasure size. Clicking on the navigation button will open up Google Maps and can guide players to there. The description comes with Malay and English which suits for Malaysian.

Once the treasure has been found, player can click on the found button. If the treasure couldn't be found, players can report it to the app.

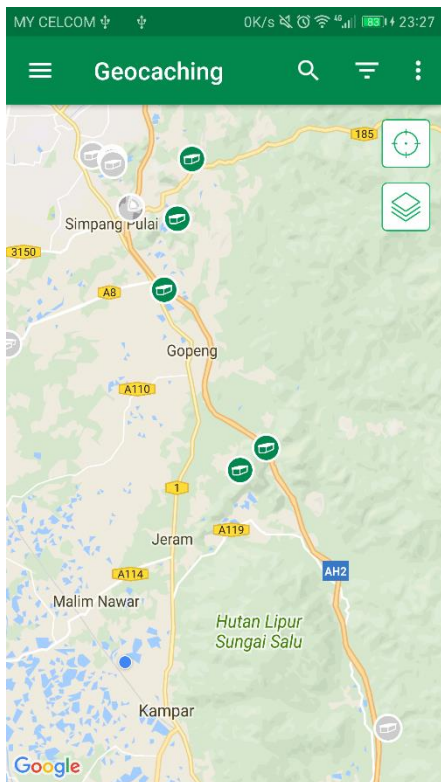


Figure 2-1-3-1 Green icons indicate it's available for non-premium member

Chapter 2 LITERATURE REVIEW

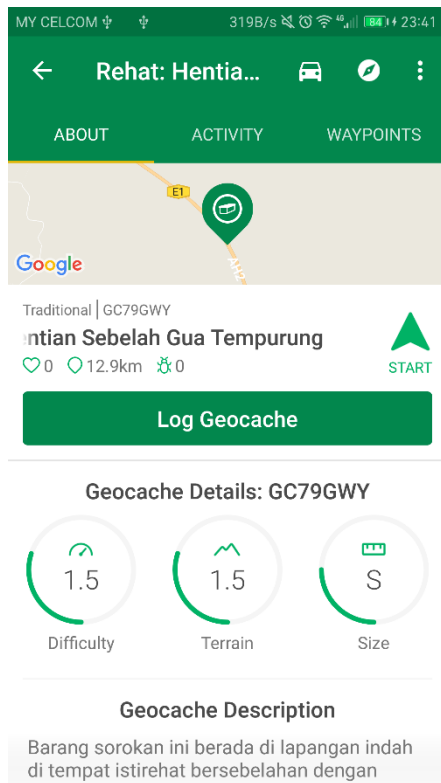


Figure 2-1-3-2 Way on getting to the treasure

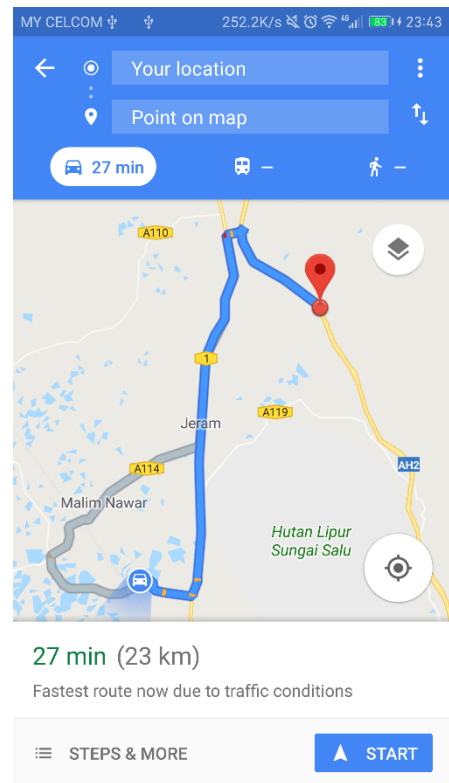


Figure 2-1-3-3 Details of treasure

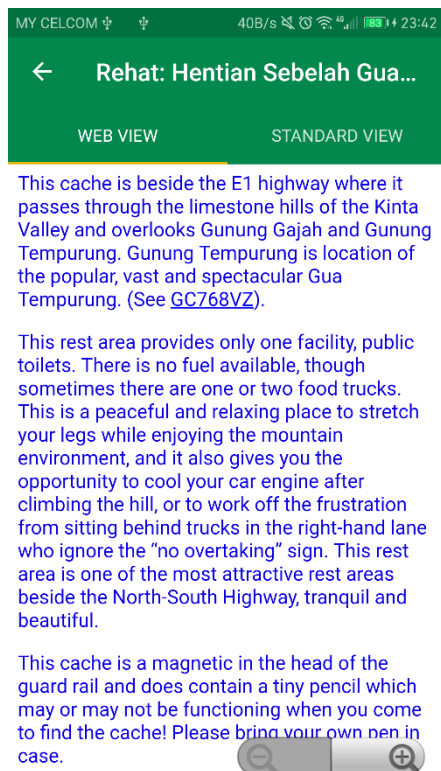


Figure 2-1-3-4 English description

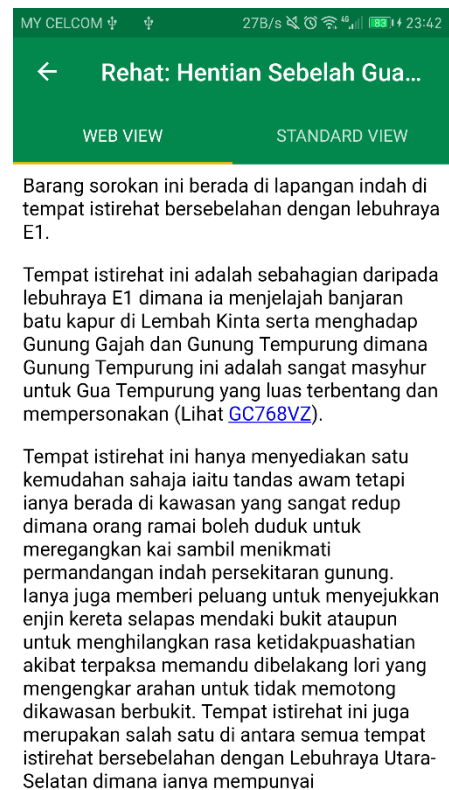


Figure 2-1-3-5 Malay description

2.2 Comparison Analysis

Apps Requirements	Actionbound	Huntzz	Geocaching	Proposed App (THOR)
Support for English	No	Yes	Yes	Yes
Camera	Yes	Yes	Yes	Yes
Internet Access	High	High	Low	Low
GPS	Yes	Yes	Yes	Yes
Hardware Requirement	Low	Low	Low	Moderate
Multi player	Yes	None	None	None
Simplicity	Low	Low	High	High
Pricing	Free	Paid	Free for 5 teams	Free
Object Recognition	None	None	None	Yes

Table 2-1 Comparison analysis

Chapter 3 PROPOSED METHOD/APPROACH

3.1 Design Specification

3.1.1 Methodology and General Work Procedure

The development approach to be used for THOR will be evolutionary prototyping that is under Prototype model given the consideration of THOR will have a lot of interaction with users and when not all of the project requirements are known in detail ahead of time. It allows developers to create partial of the application either with real data or just a few screens with sample data to demonstrate the core functionality and make any necessary refinements before developing the actual application. Since a working model of the system is displayed, the users can get a better understanding of the system being developed. Quicker user feedback always lead to better solution as missing functionality or errors can be detected much earlier and easily when developer and user are in sync with each other's expectations continuously. As such, developer can provide the expected work in the short amount of time.

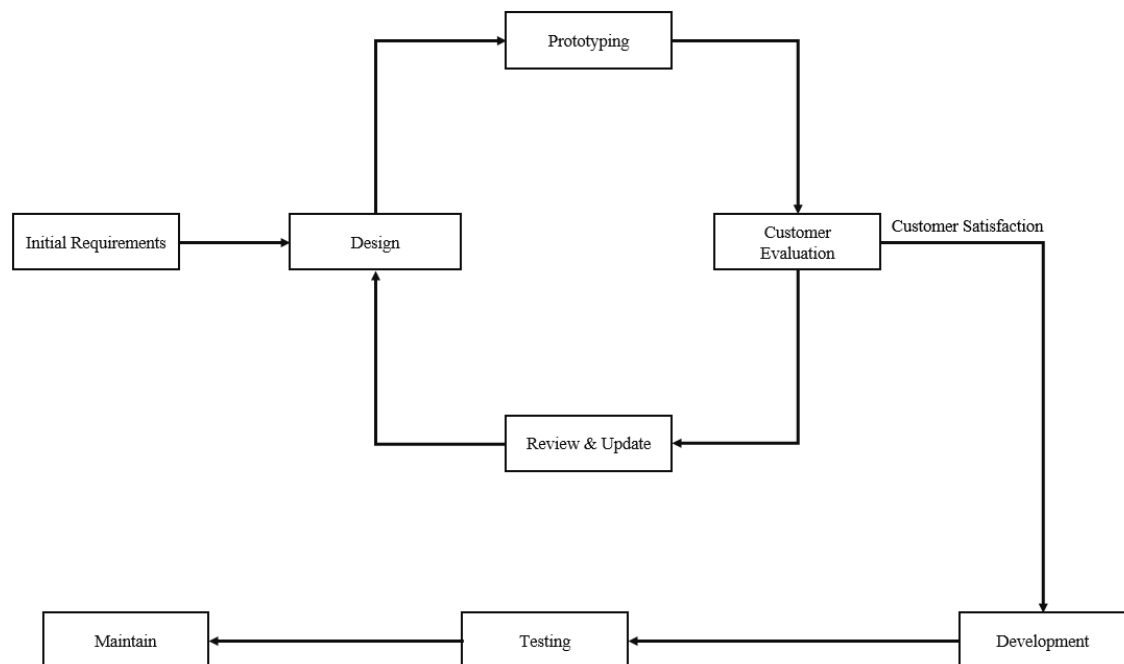


Figure 3-1 Evolutionary Prototyping

3.1.2 General Work Procedures

1. Initial Requirements

The goal of this phase is to define application functionalities from user's requirements with a greater level of detail to begin with. The simplest approach is to host an interview with user (interviewee).

Selecting the interviewee such as manager or key stakeholder who could make the decisive choice is very critical. However, it is recommended to collect different perspective from other users as well. Once interviewee has been chosen, then the next procedure is to conduct the interview. Few important aspects that has to put in mind during conducting interview are go through the application requirements again which is discussed before, record all the information, and allow interviewee to ask questions. The interviewing questions can be designed in either closing-ender questions, open-ended questions or probing questions.

2. Design

A preliminary design is created for the new application according to user requirements that is gathered in the first phase or after reviewing from user's further request. Activity diagram will be drawn out to present a series of actions performed by user and the flow of control in this application.

3. Prototyping

The goal of this phase is to write the necessary code to implement the applications specified in the design. Along with the code or minimal UI design, the level of ambiguity decreases as both developer and user have a clear picture on how should the application looks like. Tools such as Android Studio, Wireframe and necessary libraries will be utilized for developing prototype.

4. Customer Evaluation

The goal of this phase is to present the prototype to end user for examine the prototype and collect feedback if there is any.

5. Review and Update

The goal of this phase is to update the design based on user's review and enhance the prototype.

Take note that phase 2-5 are iterated as many times as necessary until the prototype has met user's requirement or represents the final product desired.

6. Development

The application will then be developed in thoroughly manner based on the final prototype.

7. Testing

The goal of this phase is to validate the developed application performs properly according to the user's requirements in phase 2-5. Checklist is provided to display the features that are or aren't able to be carried out. This is the last stage where users might request developer to make any final changes to the application as most of the changes had been done in phase 2-5.

8. Maintenance

Finally, the workable application has been deployed to user. However, this doesn't mean developer is completely break away from developing this application as it's unavoidable for application to have some bugs or user's might request for further improvement. Instead, developer should carry out routine maintenance on a continuing basis to prevent vital break down.

3.2 Tools to use

3.2.1 Software Requirements

1. Android Studio

It is the official Integrated Development Environment (IDE) from Google that most developers use for android application development and it provides a lot of handy and useful tools to fasten the development process.

2. Google Map API

Google Maps is a useful tool for showing the places and also the direction to guide user to there. Besides, user can choose the varying of transportation mode such as by walking or cycling.

3. Tensorflow

TensorFlow™ is an open source Machine Learning framework which was developed by researchers and engineers working within the Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. (Anon., n.d.). We will be focusing on its object detection feature.

4. Wikitude

Wikitude SDK is an easy to use software library and framework to develop amazing augmented reality experience on mobile devices.

5. Draw.io

It's a free online diagram software for making workflow, BPM, org charts, UML, ER, and network diagrams. Grant chart diagram will be drawn by using this software.

6. SQLite

SQLite is an in-process library that implements a self-contained, zero-configuration, serverless, transactional SQL database engine. (Anon., n.d.) This will be used for storing user information to support application backend.

Chapter 3 PROPOSED METHOD/APPROACH

3.2.2 Programming Languages

1. Java

Java is an object-oriented language that is used to develop the functionalities of this Android application. Java language can be used to create mobile applications that can run on any Android device.

2. XML

XML that stands for extensible markup language which is used in Android studio as a markup language that works similar to HTML to describe data. It's mainly purpose is to implement the UI design to make it workable.

3. HTML and CSS

HTML or hypertext markup language is the standard markup language that is used in develop website. However, at here it's used to create and design point of interest (POI) through Wikitude SDK.

4. Javascript

An object-oriented scripting language that works on multiple platform that enable user to interact with the POI.

3.2.3 Hardware Requirements

1. Laptop

Model: Lenovo Flex 2

Processor: Intel® Core™ i5-4210 CPU @ 2.40GHz, Quad-core

RAM: 8GB

System Type: 64-bit operating system

Graphics Card: NVIDIA GeForce 840M 4.0GB

2. Mobile Device

Model: Huawei Mate 9 Pro

Platform OS: Android OS, v5.0.2

Processor: Octa-core (4x1.7 GHz Cortex-A53 & 4x1.0 GHz Cortex-A53)

GPU: Adreno 405

Camera: Primary 20 MP, Secondary 12 MP

3.3. Requirement Specifications

3.3.1 Functional Requirements

1. Players should be able to view their personal details.
2. Players should be able to access camera to detect for treasure.
3. Players should be able to access GPS to determine their location.
4. Players should be able to see only one treasure icon on the map.
5. Players should be able to see their icon moving on the map if the GPS is switched on.
6. All the buttons should be working.
7. Players should see the congratulation page after finding the treasure.
8. Players should be able to enter and exit from the object detection feature.
9. The system shall pop up virtual information about the place when the device is at nearby within 100m.
10. The system should not crash when it's running.
11. The system shall not mistake non-trained custom object as treasure.

3.3.2 Non-Functional Requirements

1. Players should be able to log in with their Facebook account.
2. There should be a Help and About to guide players along the way.
3. Players' personal information shall not be shared with any third party apps.

Chapter 4 System Design

4.1 System Architecture

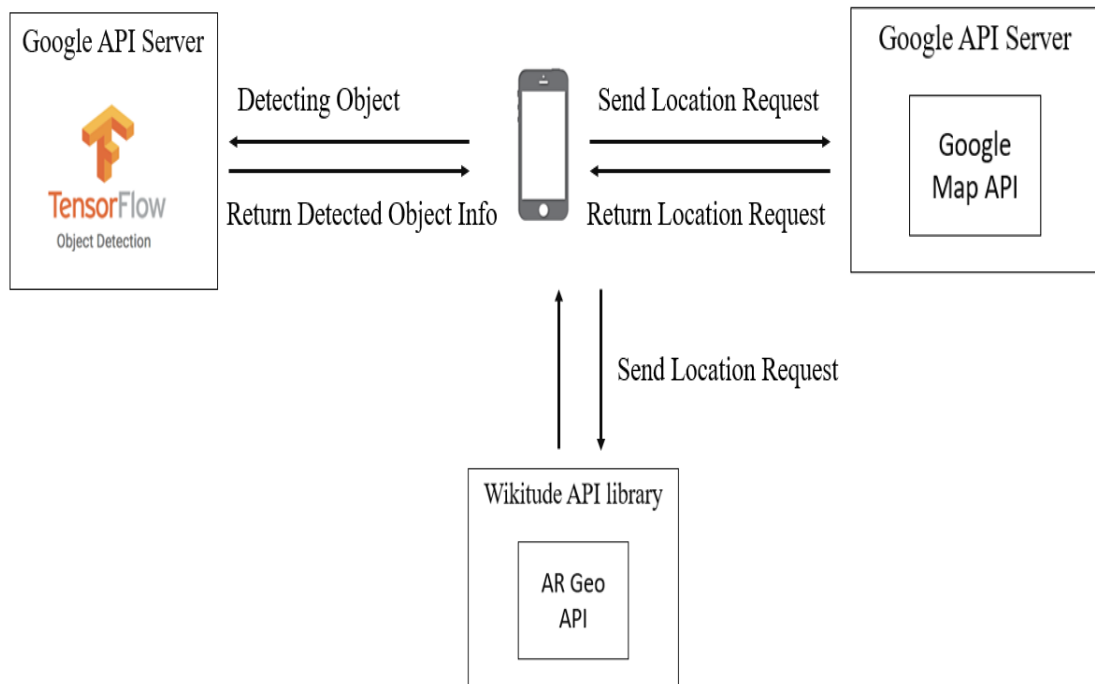


Figure 4-1 System Architecture

Above is the system architecture of THOR. THOR runs on Android devices and requires the OS version to be at least 6.0 (Marshmallow version).

To break it down, THOR consists of three modules which are object detection, Augmented Reality Geo location and positioning user on the map.

The object detection feature is utilizing TensorFlow's Object Detection API which is an open source framework released by Google that is built on top of Tensorflow. It's released in few years back ago and has quickly become one of the most favorite machine learning API due to its simplicity and yet powerful tool. It's pretty convenient to use and well documented for a non-machine learning expertise to construct, train and deploy their own object detection models. It's recommended to train the custom model on the laptop or on cloud and to speed it up the machine shall equipped with GPU. Once the custom model has been well trained, it will then be exported to Android devices to run on the detection when the camera is turned on.

Following would be the Wikitude API library which enables the user to view the location info in Augmented Reality. Once user has reach the specified location, the device will call upon the Wikitude Geolocation framework. Once user has switched on the camera, the virtual information will then be rendered on the device screen.

Last comes to the Google Maps API for Android. It's an easy to use tool to provide maps data to the user. The API automatically handle access to Google Maps servers, download the map data, displaying the map details and treasure location, and handle user's response to map gestures.

4.2 Tensorflow Object Detection API

There are numerous of pre-trained model on Tensorflow model zoo. There's no straightforward answer to determine which model is the best. The choice must be made based on the tradeoff between speed and accuracy. In this project, Faster R-CNN Inception V2 which trained on COCO dataset is chosen. Faster R-CNN is the network architecture and Inception is the base architecture. Below we will take about Faster R-CNN and Inception network.

4.2.1 Object Recognition Architecture

In Chapter 1, we have briefly talked about the object detection model that we are going to use in this application which is Faster R-CNN. To recap, Faster R-CNN consists of two modules which are the Region Proposals generated by the Region Proposal Network and a Fast R-CNN module for detecting object (Shaoqing Ren, 2016). The Faster-RCNN is an end-to-end model which accept the image as input and then output an image which has some bounding boxes with the label assigned and a score for each bounding box and label.

To fully understand what Faster R-CNN model is, we will have to know the reason for its existence. Before it's discovered, its predecessor is Fast R-CNN which is much slower and heavy consumptive on processing power due to it is using selective search to generate Regions Proposal. In the mid of 2015, a team at Microsoft Research that is led by Jian Sun discovered a novel way to reduce the cost of generating Region Proposal to almost costless. (Parthasarathy, 2017)

An instinctive speedup solution for this is to combine the production of Region Proposal into the Convolutional Neural Network model (CNN). The time cost of this is much smaller than using selective search to generate Region Proposal. Faster R-CNN will construct a single, unified model composed of Region Proposal Network and Fast R-CNN classifications which shares the most parameters (javier, 2018).

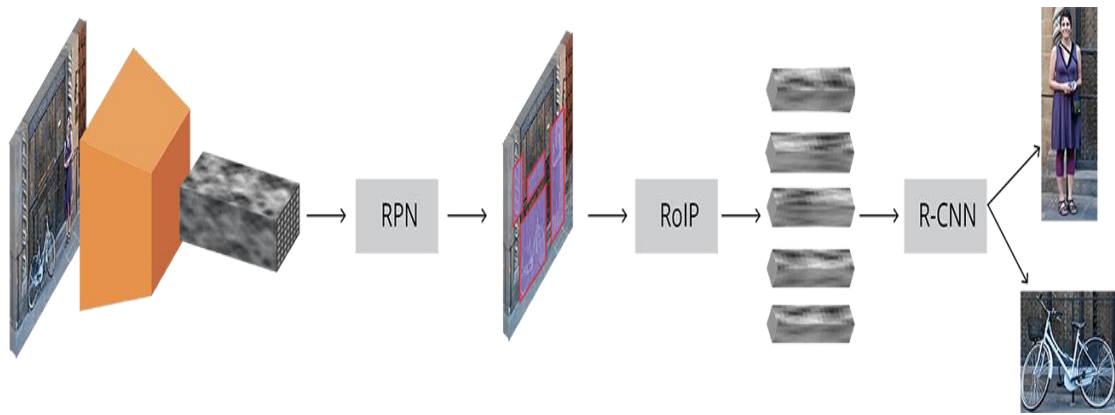


Figure 4-2-1 Faster R-CNN Architecture

Through this way, we could obtain Region Proposal almost without any additional computation power which is generated from convolutional feature maps through the CNN.

Therefore, compared with Fast R-CNN, the input to Faster R-CNN is a just an image without Region Proposals (as it could be generated from the CNN model) and the output from Faster R-CNN is a classification and the coordinates of the detected object's bounding box in the image.

Region Proposal Network

Region Proposal Network is used to speed up the inference process by predicting the object bounds and the objectness probability simultaneously. It will generate region proposals which are then send to Fast R-CNN object detection network. Both Region Proposal Network and Fast R-CNN are sharing the same convolution features map.

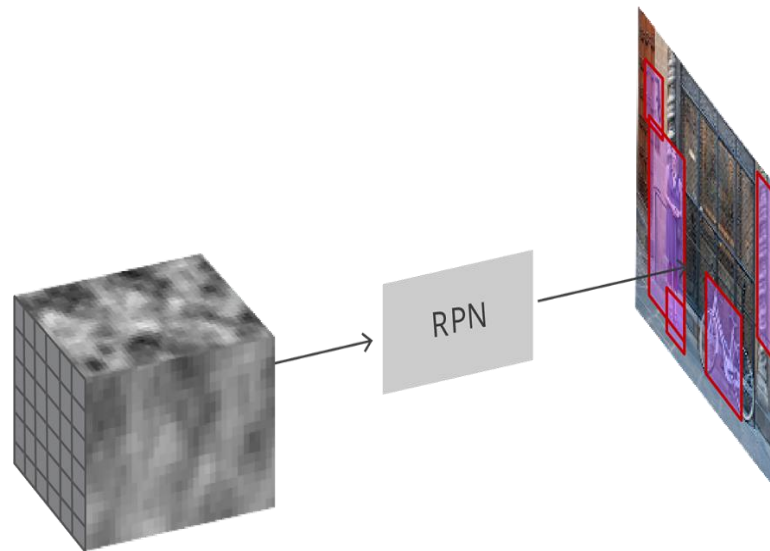


Figure 4-2-2 RPN

In order to generate Regions Proposal, Faster R-CNN outputs a convolutional feature map which is generated by the last shared convolutional layer and use a small sliding window to slide a small network on the convolutional feature map. The small network receives input of $n \times n$ of the sliding window which is then mapped to a lower dimension feature. This lower dimension feature is then fed into 2 fully-connected layer which are a box-classification layer and a box-regression layer.

At each spatial sliding-window, the maximum number of proposals is indicated as k . Since the classification layer will only predict whether the bounding box contains an object or it's just a background. Therefore the output for classification layer is $2k$. Whereas the regression layer will predict the bounding box coordinates, therefore the output for the regression layer is $4k$. At here, the k proposals can be said as anchors.

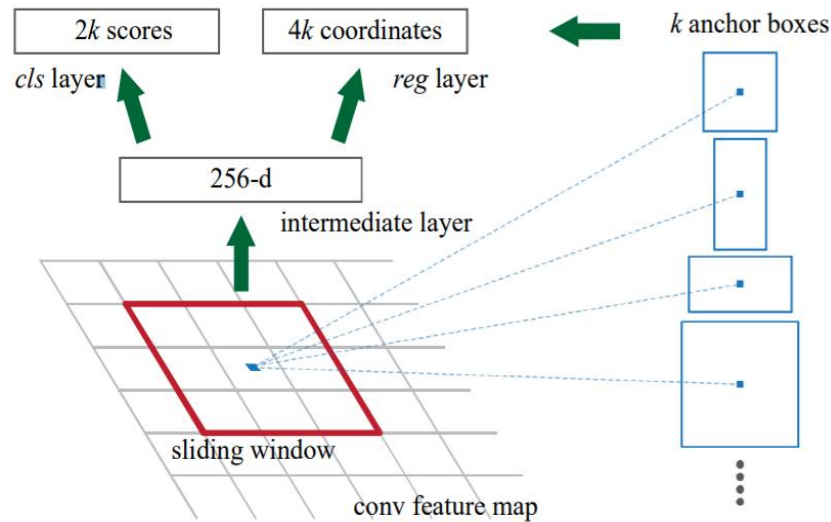


Figure 4-2-3 Output of RPN

In conclusion, the input to the Region Proposal Network is a CNN feature maps and the output is a bounding box per anchor which comes with a score which indicates whether the bounding box is an object or not.

Anchors

At each sliding window location, there would be 3 different kinds of anchor boxes for different scales which are 128x128, 256x256, and 512x512. Each of the anchor boxes will have 3 aspect ratios which are 1:1, 2:1 and 1:2 (Gao, 2017). So to sum up, there are 9 boxes at each window location on which Region Proposal Network predicts the score of it being foreground of background.

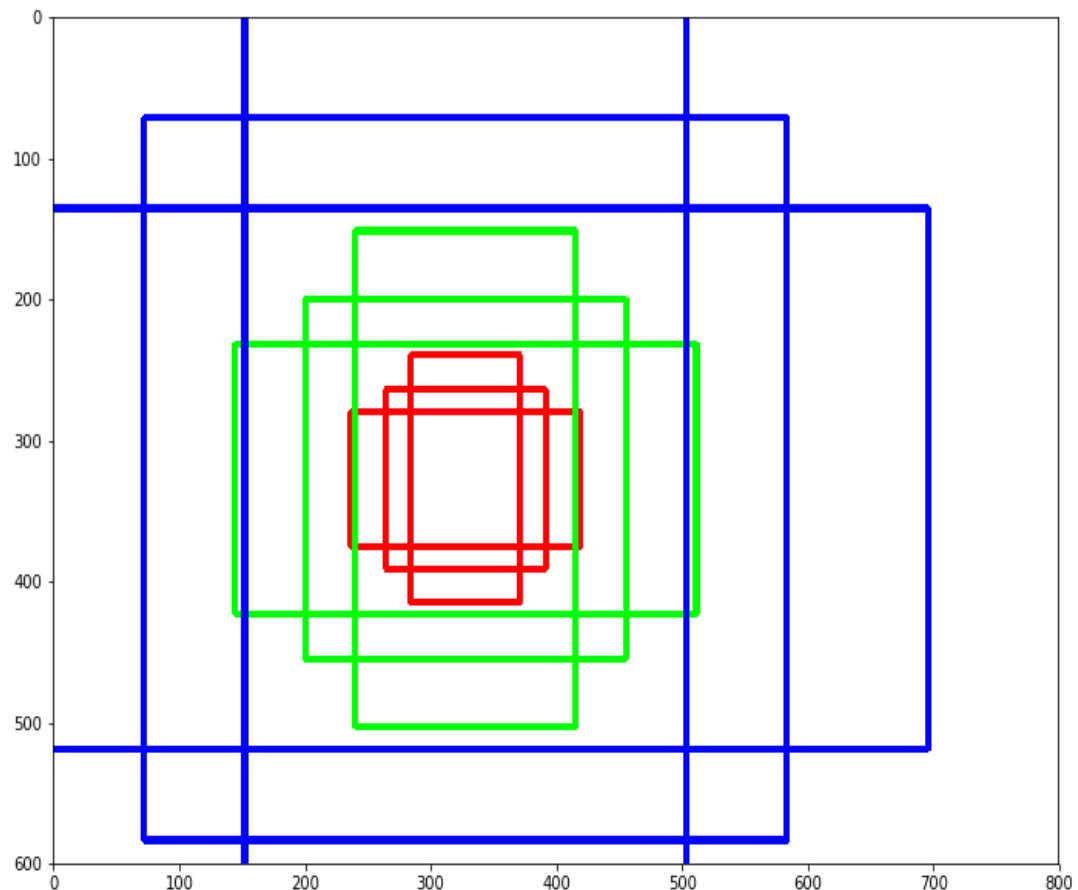


Figure 4-2-4 Anchor

To optimize the speed, Faster R-CNN allows us to refine the anchor boxes' aspect ratios to fit the assumption of the object that we are going to detect. For example, if we were designing an object detection model to detect human. We could just skip through those anchors which are too short or too large. This is because we know in real world, most of the human shapes can be bound a rectangular box.



Figure 4-2-5 Refine anchor based on object

In RPN, each of the anchor is predicted with the possibility of containing an object and it's coordinates to resize the anchor into the right position (Fathi, 2017).

Training

In modern machine learning, training is part of the pipeline to improve the model accuracy.

For training the RPN, a binary class is labeled on each anchor to indicate whether it's an object or not. (Mirjalili, 2017)

- i. Positive label
 - a. Anchors that are overlapping with a ground-truth box that have the highest Intersection over Union (IoU)
 - b. Anchors that are overlapping with a ground-truth box that have the highest Intersection over Union (IoU) over 0.7
- ii. Negative label
 - a. Anchors that do not overlap with any ground-truth box
 - b. Anchors that are overlapping with a ground-truth box that have the highest Intersection over Union (IoU) below than 0.3

A score p^* is calculated to indicates the overlapping of IoU with ground-truth box

$$p^* = \begin{cases} 1 & \text{if } IoU > 0.7 \\ -1 & \text{if } IoU < 0.3 \\ 0 & \text{otherwise} \end{cases}$$

Whereas the formula for IoU is defined as below:

$$IoU = \frac{Anchor \cap Ground - Truth Box}{Anchor \cup Ground - Truth Box}$$

With these, our loss function for RPN is defined as below:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

- i. i represents the index of the anchor in a mini-batch.
- ii. p_i represents the probability of anchor i being a foreground (object).
- iii. p_i^* represents the overlapping of IoU with ground-truth box.
- iv. t_i represents the location of the predicted object bounding box.
- v. L_{cls} represents the classification loss which a logarithm loss over 2 classes (object or none-object).
- vi. L_{reg} represents the regression loss. The loss function that being used in here is smooth-L1 loss function which is explained at below.
- vii. $p_i^* L_{reg}$ represents the regression loss will only be activated should anchors are positive ($p_i^* = 1$) and will be disabled otherwise ($p_i^* = 0$).
- viii. The summation of classification loss and regression loss will then be normalized by N_{cls} and N_{reg} respectively. Then weighted by lambda λ

Smooth-L1 loss function

A loss function is to penalize the output scores from a higher one to a lower one. The reason for selecting Smooth L1 loss function is because it produces sparser solution and less prone to outliers. A sparser solution is good for selecting features when the data is in high dimensional space and also good for prediction speed.

The loss function of L1 is defined as below:

$$\sum_i |\hat{y}_i - y_i| = \|\hat{y} - y\|$$

The loss function will generate an absolute loss. Since L1 loss function is not differentiable in 0, so it's possible to use a Smooth L1 loss function:

$$|d|_{\text{smooth}} \begin{cases} 0.5d^2, & \text{if } |d| \leq 1 \\ |d| - 0.5, & \text{otherwise} \end{cases}$$

In short, RPN loss consists of classification loss and the regression of the bounding box loss.

The error rate will be used in back-propagation and stochastic gradient descent (SGD) to improve it's accuracy. This is how the deep learning model started with bad performance but can be improved till achieve with the highest accuracy,

ROI pooling

So far, we have explained on how to train a network to get the Region Proposal, but without considering the Fast R-CNN that will use these proposal to detect object.

Both Region Proposal Network and the detection network Fast R-CNN which are trained independently will have their own convolutional layers. Therefore, to allow sharing for convolutional layers between them, the author developed a technique called Non-approximate Joint Training which is also named as RoI pooling.

Once the RPN is generated, it will then create different sizes of proposed regions without any class assigned to them. This means there are different size of CNN feature maps as well and this will make the entire architecture inefficient. To overcome for this issue, Region of Interest pooling plays a vital role.

It reduce the CNN feature maps into the equivalent size. Same size of CNN feature maps are necessary in order for Fast R-CNN to classify them into a certain number of classes. The ROI pooling layer receives the CNN feature maps and the bounding box as input, and then splits it into a certain number (for example m) of almost equivalent size of regions. Then applying Max-Pooling to get the final feature maps for each region.

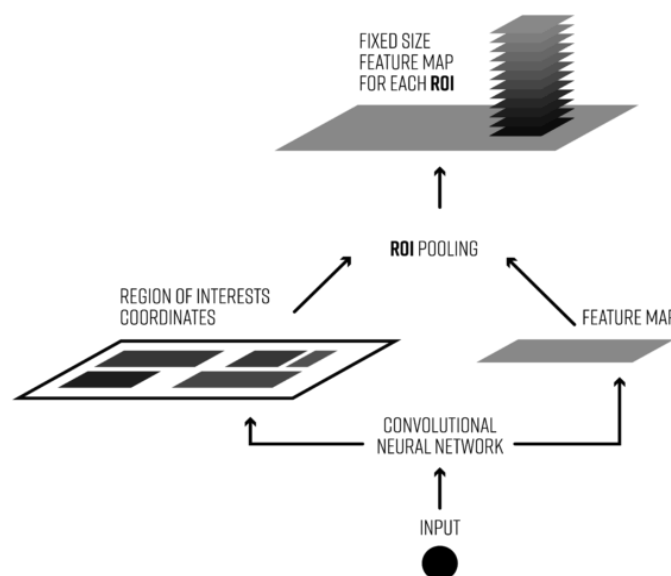


Figure 4-2-6 ROI pooling

4.2.2 Base network

The pre-trained CNN network that is chosen for this project is the Inception network which is developed by Google's researcher. Inception Network also can be called as GoogLeNet. Prior to the Inception network, very deep CNN networks has gained popularity through yielding considerable performance in various benchmarks. However, this poses a problem of requiring huge amount of memory and time even the network is deployed on the modest GPU. The major advancement of Inception network is using 12 x lesser parameters than the AlexNet but still able to achieve higher accuracy (Deshpande, 2016).

Since the creation of Inception network, it has evolved into several versions. The fundamental concept of Inception network is that it didn't follow the conventional way which is stacking more convolutional layers, increasing the network depth and width, and the number of units at each level. This means that the architecture will have a higher number of parameters which will cause the network prone to over fitting. Another drawback is the increased use of computation resources.

Therefore, Inception overcomes the problems by making the architecture "wider" rather than "deeper". Google's researches notice that by introducing sparsity into the network, it could reduce the amount of time and computational needed in training. (Valigi, 2016).

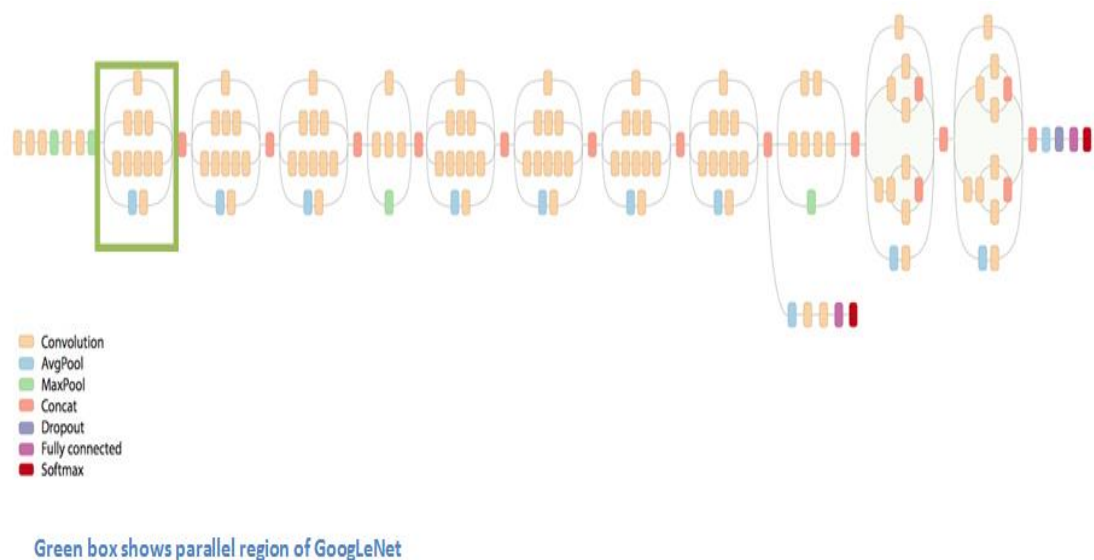


Figure 4-2-7 Inception V2 architecture

The bounding green box represents the inception module in the Inception network. There are 9 inception modules being stacked linearly (Raj, 2018). Overall, Inception V2 has some slight differences than Inception V1. In the Inception V2, the authors used smart factorization methods to improve the convolution efficiency.

Three different Inception modules are used in Inception V2 (Christian Szegedy, 2015).

1. Factorization into smaller convolutions

A 5 x 5 filter (which is being used in Inception V1) is factorized into two 3 x 3 filters. Although the input and outputs remain the same size, but the computation task was reduced to $\frac{9+9}{25}$ which is around 28% of improvement.

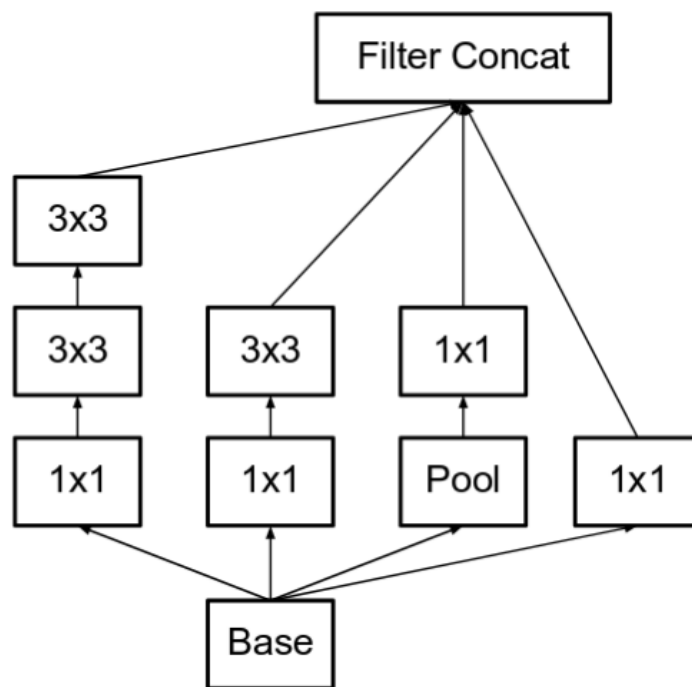


Figure 4-2-8 Factorization into smaller convolutions

2. Spatial Factorization into Asymmetric Convolutions

Replacing convolutions of filter size from $n \times n$ to the mixture of $1 \times n$ and $n \times 1$ convolutions. This method can reduce the computational load as n grows.

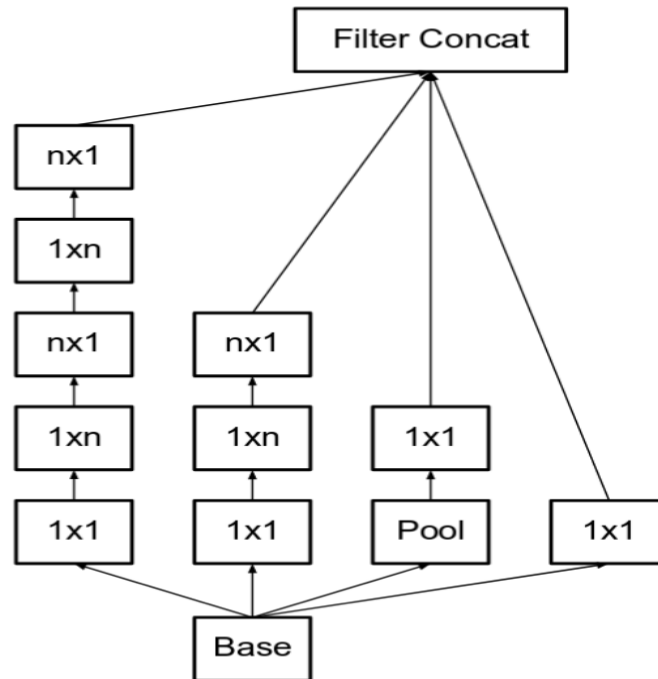


Figure 4-2-9 Spatial Factorization

3. Higher dimensional representation

Expand the filter banks to remove the representational bottleneck and allowing for more features. This will increase the training speed.

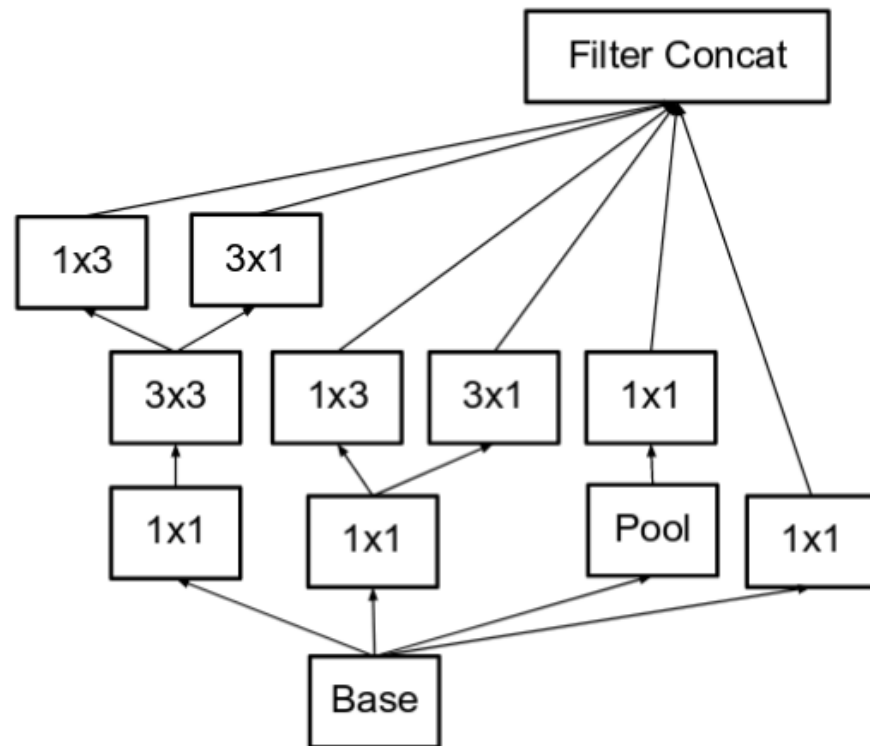


Figure 4-2-10 Higher Dimension Representation

The above 3 Inception modules are used in the network and below is the overall diagram that shows these three different modules working together (yeephycho, 2018).

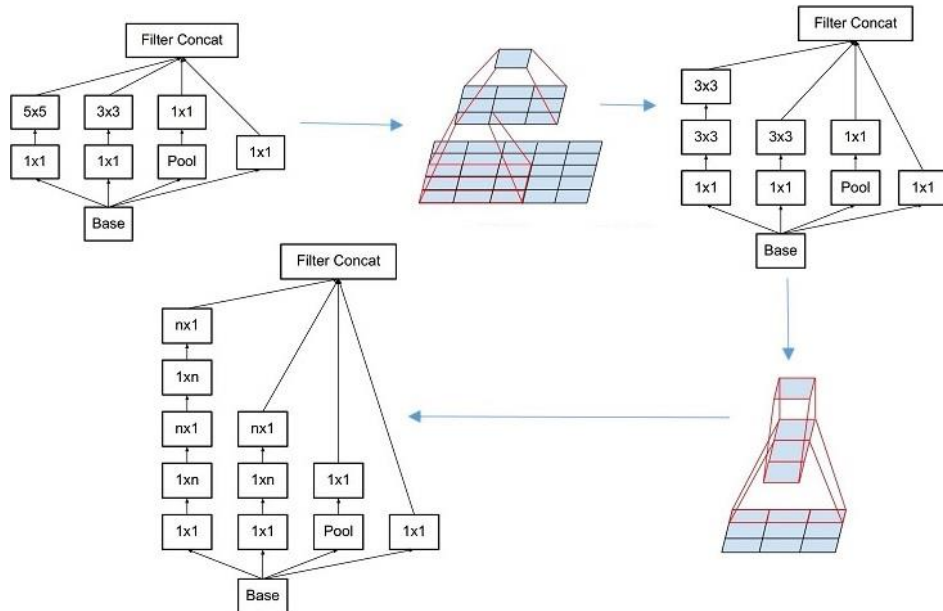


Figure 4-2-11 Relationship between 3 different Inception Module

type	patch size/stride or remarks	input size
conv	$3 \times 3 / 2$	$299 \times 299 \times 3$
conv	$3 \times 3 / 1$	$149 \times 149 \times 32$
conv padded	$3 \times 3 / 1$	$147 \times 147 \times 32$
pool	$3 \times 3 / 2$	$147 \times 147 \times 64$
conv	$3 \times 3 / 1$	$73 \times 73 \times 64$
conv	$3 \times 3 / 2$	$71 \times 71 \times 80$
conv	$3 \times 3 / 1$	$35 \times 35 \times 192$
$3 \times$ Inception	As in figure 5	$35 \times 35 \times 288$
$5 \times$ Inception	As in figure 6	$17 \times 17 \times 768$
$2 \times$ Inception	As in figure 7	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

Figure 4-2-12 Inception V2 architecture in tabular form

Above diagram is the official Inception V2 architecture which the module 1 is referred from figure 5, module 2 is referred from figure 6, and module 3 is referred from figure 7.

4.3 Use-case diagram

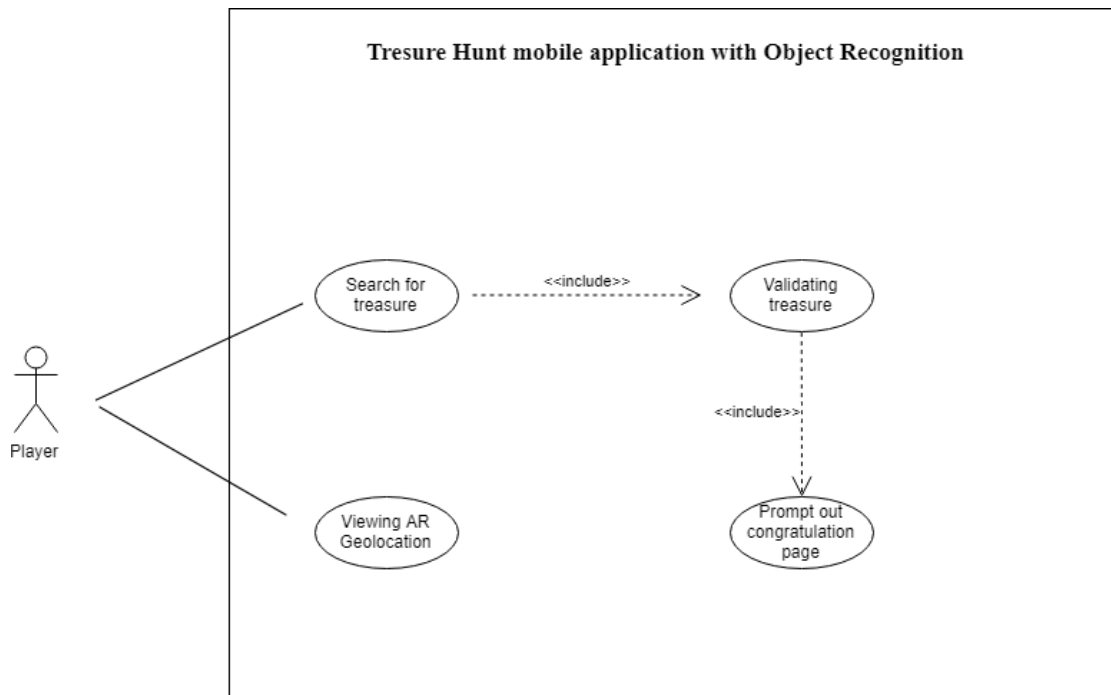


Figure 4-3 Use-case diagram

Above diagram explicitly shows that the player is able to search for the treasure and while he/she is passing to the specified location, an AR Geo location would be prompted out.

4.4 System Flowchart

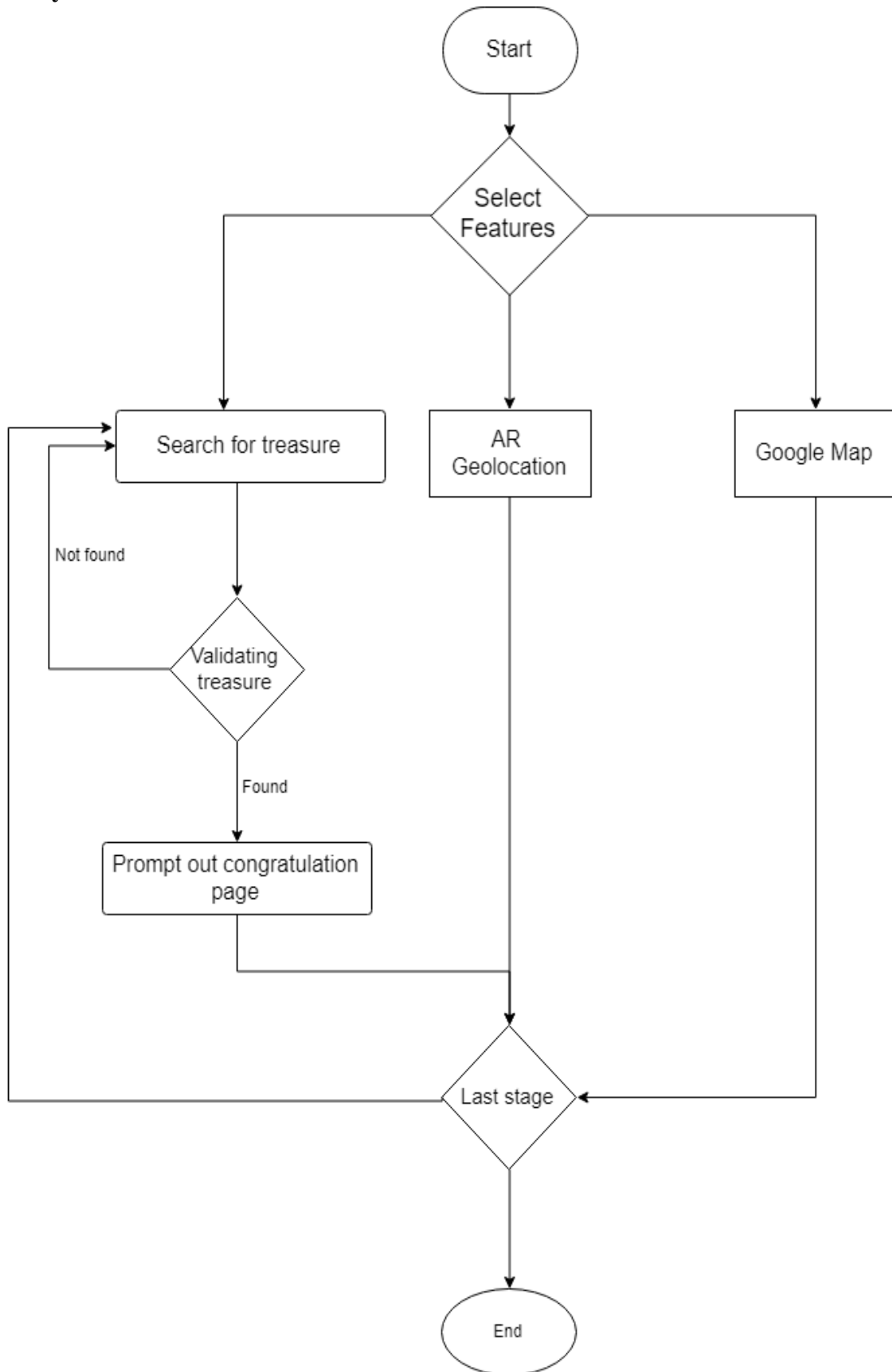


Figure 4-4 System Flowchart

Above diagram explicitly shows the system flowchart in details and below is the corresponding explanation.

Whenever users open the app, they can choose either detecting the object, using the Augmented Reality Geolocation feature or positioning themselves on Google Map.

If they click on the object detection feature, the system will detect all of the objects taken from each frame at the background. Once the object has been found, the system will then prompt out a congratulation page to congratulate user for putting in the effort in looking for the hinted treasure and reward them with some points, Then the system will explain to the user about the object that they had found, like the actual name of the location, it's purpose and what it can be used for. Next the system will validate whether the found treasure is the last treasure, if it's then the system will end the treasure hunting game and calculate the total scores that the user has accumulated.

If user click on the view map, the map will show the user current location and also the location of the next treasure with an icon. So user can easily identify where to look for the next treasure.

If user click on the Augmented Reality Geolocation viewer, they could see the virtual information that describe the place information in Augmented Reality.

4.5 System Flow with UI and detailed explanation

First of all, players will be given a list of hints to look for a treasure. The treasure can be a statue, a building, an animal, and etc. The game uses camera to identify whether the treasure is being found and GPS to show players the nearby treasure around them. Players will be given a score for each found treasure. At the end, the highest score of the players will be the winner.

1. After logging into THOR, player will see a map at the background. Their profile at the bottom left, a binoculars button at the middle bottom and a white bar at the bottom right.
 - a. Profile button – Once clicked into it, user is able to view and edit their details. User is also able to check for the score from it.
 - b. Binoculars button – Once clicked into it, player can scan the real-life object to see whether they found the treasure.
 - c. Information Icon – Hints to guide player to found for the next object.

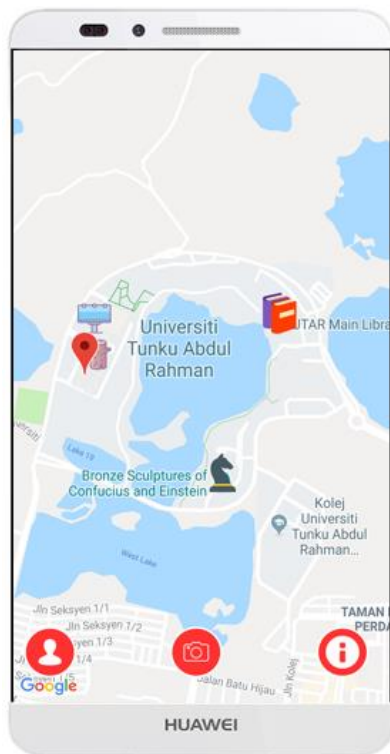


Figure 4-5-1 Home page of THOR

2. When clicked into binocular button, the device camera will start to look for object. If the player scans on the right object, then the player will be prompted to the congratulations page and be informed about the found object, A map button at the bottom will bring player to the first page (map).



Figure 4-5-2 Recognizing object

3. Once the treasure has been found, players will be prompted with a congratulation page. From there, players can choose to share the result or proceed to next round.

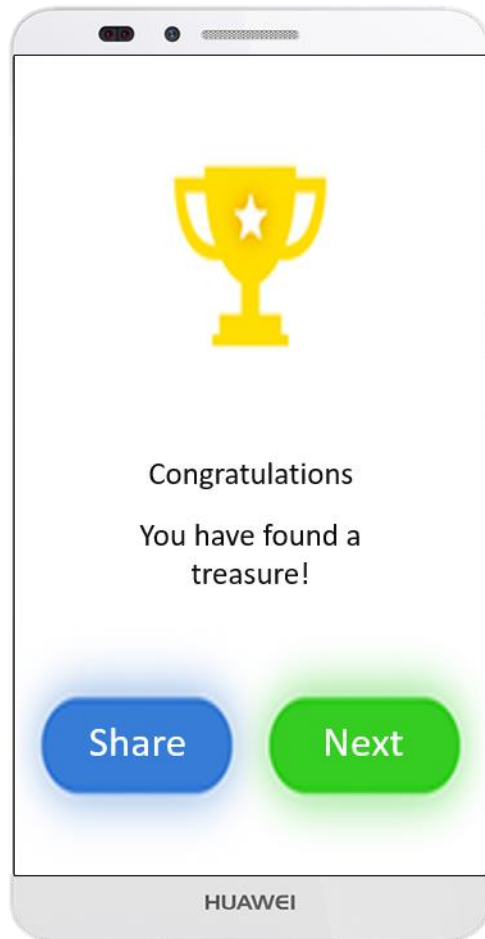


Figure 4-5-3 Congratulation page

4. Next, the system will tell the treasure info in details.



Figure 4-5-4 Treasure Details

5. If the player click on the Augmented Reality Geolocation feature, below is how it will look like when the player has reached a certain location. For example, if user has arrived nearby FICT, the icon which shows the name of the building will appear on the user screen automatically .



Figure 4-5-5 AR Geolocation of FICT

6. If the player click on the Icon button, he / she can view the hints for current treasure. However, the player can only view for the current or found treasure.

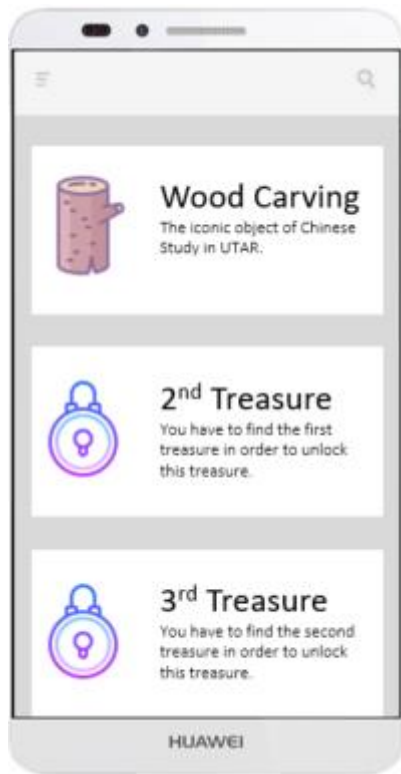


Figure 4-5-6 Treasures List

7. If the player click on the user icon, he / she can view his / her own personal details, scores being achieved, and some settings to the app.

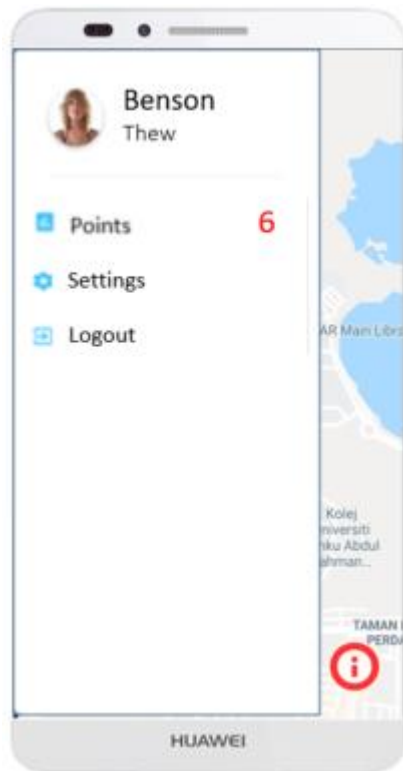


Figure 4-5-7 User Profile

4.6 Entity-Relationship Diagram

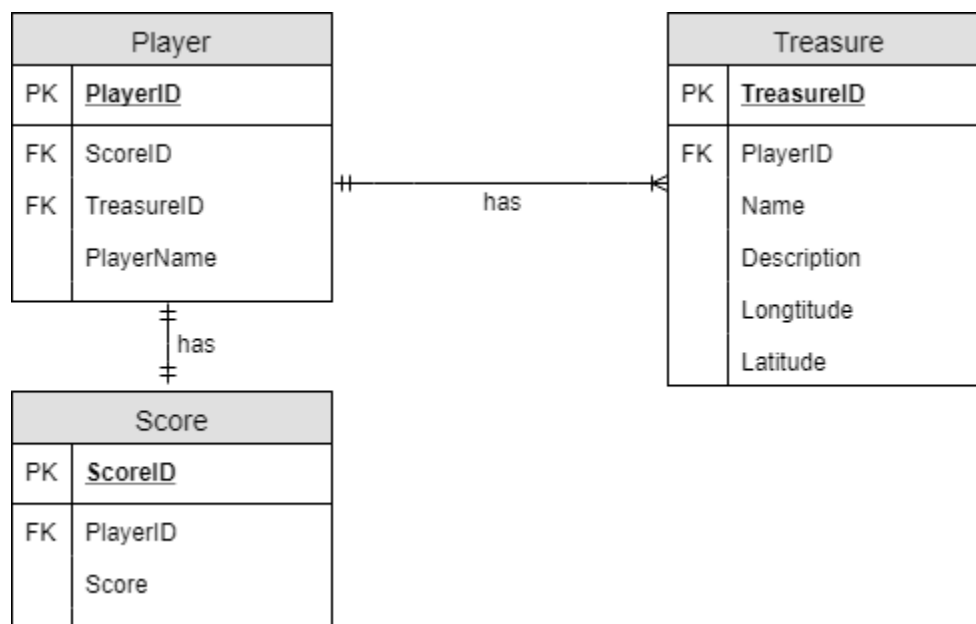
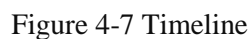


Figure 4-6 Entity Relationship Diagram

Above ERD shows three tables which are player, score and treasure table.

Each player can look for more than one treasure and each treasure that has been found will award user with one point. However, each treasure can only be found by one player since this is a single player treasure hunt mobile game.



Chapter 5 Implementation and Testing

5.1 Implementation Issues and Challenges

The development pipeline of this project is developed in the following order

1. Training the Faster R-CNN to detect for custom objects on laptop.
2. Export the Faster R-CNN model file into Android.
3. Integrate all the features on Android mobile app.

Throughout the development process, there are numerous obstacles occurred such as:

1. TensorFlow Object Detection incompatibility on Windows operating system
Since TensorFlow was developed in Linux, therefore there are some of errors shown up while training on the Faster R-CNN.
2. Exporting model to Android
There are several building alternatives that requires different configuration. So to figure out the most appropriate build alternatives require a lot of trial and error.
3. Integrating all the features
Running the object detection algorithm through the frames received from the camera is already troublesome but to integrate it with all the features require well-design of the mobile application architecture.

5.2 Testing

To ensure this project could give the most exhilarating and consistence performance, it has been gone through 360 degree of different test scenarios.

5.2.1 Different test scenarios.

1. Test Scenario: Display user's current location

Test Case	Expected Results	Actual Result	Status (Pass/ Fail)
GPS is switched on.	User's location is updated when he /she moves	User's location is updated when he /she moves	Pass
GPS is switched off.	User's location shouldn't be updated when he /she moves	User's location is updated when he /she moves	Pass

Table 5.2.1.1 Display user's current location

2. Test Scenario: Display AR when user reaches certain locations

Test Case	Expected Results	Actual Result	Status (Pass/ Fail)
Mobile device GPS is near to specified location within 100m.	AR will be prompted out.	AR will be prompted out.	Pass
Mobile device GPS is far from specified location.	AR will not be prompted out.	AR will not be prompted out.	Pass

Table 5.2.1.2 Display AR when user reaches certain locations

3. Test Scenario: Click on object detection feature

Test Case	Expected Results	Actual Result	Status (Pass/ Fail)
Camera permission is given.	Open front camera.	Open front camera.	Pass
Camera permission isn't given.	Unable to open front camera.	Unable to open front camera.	Pass

Table 5.1.2.3 Click on object detection feature

4. Test Scenario: Prompt out the detected object info when it's detected

Test Case	Expected Results	Actual Result	Status (Pass/ Fail)
Detect on trained custom object.	Explain on the detected object info.	Explain on the detected object info.	Pass
Detect on non-trained custom object.	Keep detecting object from the frame.	Keep detecting object from the frame.	Pass

Table 5.1.2.4 Prompt out the detected object info when it's detected

5. Test Scenario: Showing user current scores

Test Case	Expected Results	Actual Result	Status (Pass/ Fail)
User found the treasure.	Show the scores based on the number of treasures are found.	Show the scores based on the number of treasures are found.	Pass
User didn't found the treasure.	Show 0 scores.	Show 0 scores.	Pass

Table 5.1.2.5 Showing user current scores

Chapter 6 CONCLUSION

6.1 Project Review and Discussion

Outdoor learning has been gaining popularity recently due to its incomparable benefits to the students and the teachers. Outdoor learning is essential in building up a person awareness to new surroundings (Yancey-Siegel, 2017), learn how to be a great team player, confidence level and also a successful learner.

However, today modern technologies are much focusing on the indoor learning sector and ignore the need and growth of outdoor learning sector. Less research and development of outdoor learning mobile application has been done. Also outdoor learning mobile technology that could provide users with the real fun and unforgettable pleasure experience are much lesser.

Therefore, this project aims to deliver a solution for outdoor learning. This project tries to enter the outdoor learning market by providing treasure hunting experience. None of the existing treasure hunting mobile app in the market is using object recognition developed with Deep Learning. We have seen Deep Learning is extensively used in autonomous vehicles, smart video surveillance, facial detection and various people counting applications (Xu, 2017). However, its powerful object recognition technology is seldom seen from being applying in outdoor learning.

Changing to the perspective in education industry, THOR can be used to provide educational component to even common games, increasing their educational value and making them more engaging to the users (Yao-Ting Sung a, 2015).

6.2 Future Work

In this project, the custom object that are being trained are only 5 different objects. It would be good if the custom trained objects can go up to 20. However, this requires much time and a modern high end GPU to train.

In future, the training of the model can be fasten through training on Google Cloud Platform. The platform enables the training distributed to several worker with the high end GPU and merged it at the end of the training.

At last, it's worth a try to use the Single Shot Detector (SSD) object detection network to increase the detection speed. However, the downside of it is the lower accuracy. So hopefully more researchers can work hard on increasing the detection speed of SSD.

References

Anon., 2018. *merriam-webster*. [Online]

Available at: <https://www.merriam-webster.com/dictionary/gamification>

[Accessed 18 August 2018].

Anon., n.d. *SQLite*. [Online]

Available at: <https://www.sqlite.org/about.html>

[Accessed 21 March 2017].

Anon., n.d. *Tensorflow*. [Online]

Available at: <https://www.tensorflow.org/>

BADERTSCHER, C., 2018. *qnnect*. [Online]

Available at: <https://www.qnnect.com/blog/draft/how-gamification-leads-to-better-learning>

[Accessed 18 August 2018].

Christian Szegedy, V. V. S. I. J. S. Z. W., 2015. *Rethinking the Inception Architecture for Computer Vision*, Volume 3, p. 10.

Deshpande, A., 2016. *Adit Deshpande*. [Online]

Available at: <https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>

[Accessed 18 August 2018].

Fathi, M., 2017. *StackExchange*. [Online]

Available at: <https://stats.stackexchange.com/questions/265875/anchoring-faster-rcnn>

[Accessed 17 Aug 2018].

Gao, H., 2017. *Medium*. [Online]

Available at: <https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8>

[Accessed 18 August 2018].

Jacob, 2015. *Telenav*. [Online]

Available at: <https://telenav.zendesk.com/hc/en-us/articles/204508569-Why-does-GPS-not-work-near-buildings-and-other-tall-objects->

[Accessed 18 August 2018].

javier, 2018. *tryolabs*. [Online]

Available at: <https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/>

[Accessed 18 August 2018].

Mirjalili, V., 2017. *Quora*. [Online]

Available at: <https://www.quora.com/How-does-the-region-proposal-network-RPN->

in-Faster-R-CNN-work

[Accessed 18 August 2018].

Parthasarathy, D., 2017. *Medium*. [Online]

Available at: <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4>

[Accessed 14 8 2018].

Raj, B., 2018. *TowardsDataScience*. [Online]

Available at: <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202>

[Accessed 18 Aug 2018].

Ren, S., He, K., Girshick, R. & Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. p. 8.

Roland, J., 2016. *Insights*. [Online]

Available at: <https://insights.samsung.com/2016/07/19/does-gamification-education-really-improve-learning/>

Shaoqing Ren, K. H. R. G. a. J. S., 2016. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, 3(arxiv), p. 14.

SkyMind, 2017. [Online]

Available at: <https://deeplearning4j.org/ai-machinelearning-deeplearning>

Valigi, N., 2016. *nicolovaligi*. [Online]

Available at: <https://nicolovaligi.com/history-inception-deep-learning-architecture.html>

[Accessed 18 August 2018].

Xu, J., 2017. *TowardsDataScience*. [Online]

Available at: <https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>

[Accessed 1 3 2018].

Yancey-Siegel, W., 2017. *opencolleges*. [Online]

Available at: <https://www.opencolleges.edu.au/informed/features/outdoor-education-prepare-students-future/>

[Accessed 18 August 2018].

Yao-Ting Sung a, K.-E. C. T.-C. L., 2015. The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. p. 24.

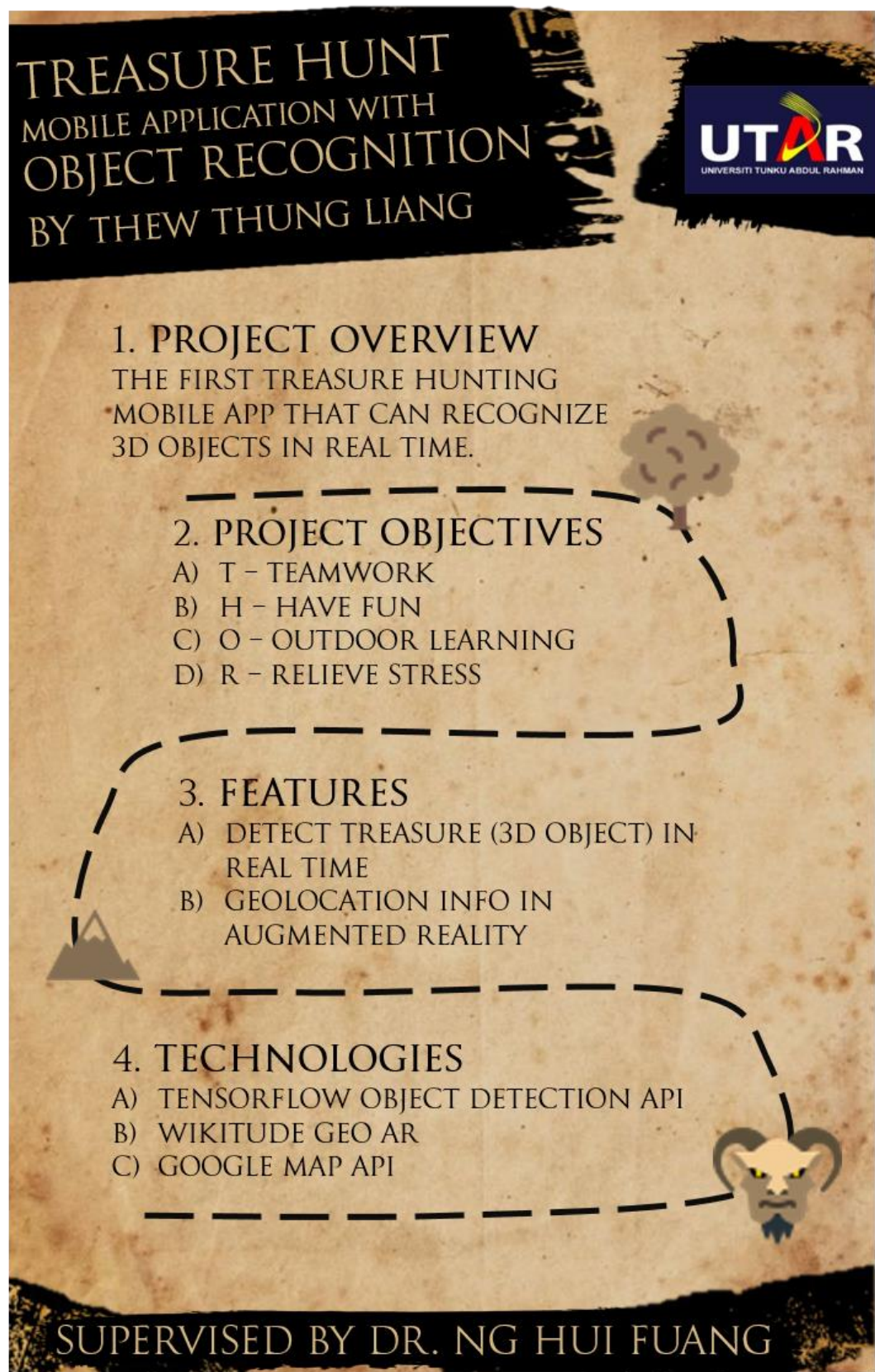
yeephycho, 2018. *github*. [Online]

Available at: <http://yeephycho.github.io/2016/08/02/A-reminder-of-algorithms-in-Convolutional-Neural-Networks-and-their-influences-II/>

[Accessed 18 August 2018].

BIS (Hons) Information Systems Engineering

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



Turnitin Originality Result

Feedback Studio - Google Chrome
Secure | https://ev.turnitin.com/app/carta/en_us?lang=en_us&is=&student_user=1&o=990964151&u=1072429149

feedback studio Benson Thew Thung Liang Check 2

Chapter 1 INTRODUCTION

1.1 Problem Statement

Creating a memorable and nostalgic experience to visitor has always been an annoying problem to management echelon when managing a huge area of property such as universities, shopping mall, big firms who have a large workplace, tourist destination, museum and etc. These places have one thing in common which is too big to be discovered. While some places like historical sites or museum can easily get the visitors' attraction as these places are place where human culture for posterity has been preserved. Visitors can find their own identity, learning other's culture and history, discover the life of ancestors'. However, places that are very common and not bound with a location, culture or that is not historic such as shopping mall or universities might be hard in getting visitor's attention and time for them to familiar themselves. Taking university as an example, it's very typical to see most of the universities will held campus guide prior to the beginning of new semester for incoming freshmen to make themselves familiar with the campus that they are going to study in next couple of years. However, do all the visitors really remember all the places they have visited or an exit route to escape from a building if an emergency occurs? Once they have lost their interest, they would prefer to stand at the ground playing with their mobile phone and having a bored impression on that place. Without

Match Overview

7%

1	cs231n.stanford.edu	Internet Source	1%
2	Shaoqing Ren, Kaiming...	Publication	<1%
3	Armin Hutzler, Rudolf ...	Publication	<1%
4	Yoshimasa Kawazoe, K...	Publication	<1%
5	Rui Hou, Chen Chen, M...	Publication	<1%
6	"Image and Graphics T...	Publication	<1%
7	Louise Barkhuus. "Emp...	Publication	<1%

Page: 2 of 51 Word Count: 7499 Text-only Report High Resolution On

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



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

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

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

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

Signature of Supervisor

Name: _____

Date: _____

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	
Student Name	
Supervisor Name	

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

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

I, the author, have checked and confirmed all the items listed in the table are included in my report. <div style="text-align: right; margin-top: 20px;">_____</div> (Signature of Student) Date: _____	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. <div style="text-align: right; margin-top: 20px;">_____</div> (Signature of Supervisor) Date: _____
---	---

