AN EDUCATIONAL ANDROID APP FOR IDENTIFYING ANIMALS IN ZOO

VOON CHERNG JYH

UNIVERSITI TUNKU ABDUL RAHMAN

AN EDUCATIONAL ANDROID APP FOR IDENTIFYING ANIMALS IN ZOO

VOON CHERNG JYH

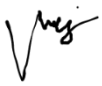
A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2020

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Voon Cherng Jyh

ID No. : 1601883

Date : 15 May 2020

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"AN EDUCATIONAL ANDROID APP FOR IDENTIFYING ANIMALS IN ZOO"** was prepared by **VOON CHERNG JYH** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : KHOR KOK CHIN

Date : 16 May 2020

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2020, Voon Cherng Jyh. All right reserved.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Khor Kok Chin for his invaluable advice, guidance and his enormous patience throughout the development of the project.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement, motivation and inspiration for the project.

ABSTRACT

Zoo has been one of the favourite learning spots for children when it comes to animal study. In the outdoor learning session, teachers will often find it difficult to tutor their students while ensuring their safety. The advent of modern technology aided teachers in educating students without their presence. The project explored the possibility of using smartphones as a medium to help kids to learn about animals in zoos, even without the guidance of teachers. The objective of this project is to build an Android mobile application for kids that automatically recognizes images of an animal using Convolutional Neural Network and Transfer Learning.

Phased development methodology was used for application development. The developed application consists of three major modules: animal recognition module that is able to recognize 18 different animals, gamification module that attracts the interest of kids and lastly quiz module that broadens the kid's knowledge on animals.

Apart from that, experiments were performed on three pretrained models, which is MobileNet, EffNet and NASNet to determine the most suitable model to be used for Transfer Learning. The models were analyzed from the aspect of accuracy, output model size, speed and power consumption. MobileNet was observed to be the least power-hungry (38.2 million FLOPs), fastest (0.018 Million parameters) and smallest in size (12.859 MB) among the three. Considering the time spent in zoo, Mobilenet was integrated into the system for image recognition.

The mobile application was successfully designed, developed and tested throughout the project. Three pretrained models were compared and MobileNet was selected as the image recognition model. It is believed that the product will aid zoos and educators in teaching the kids about wildlife.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

Dk	spatial width and height of a square kernel
D_F	width and height of a square feature map
M	number of input channels
N	number of output channels
α	width multiplier
ρ	resolution multiplier
ADM	Agile Development Methodology
API	Application Programming Interface
app	application
AUC	Area Under the Curve
CNN	Convolutional Neural Network
DB	Database
EffNet	EfficientNet
EXP	Experience points
FLOPs	Floating Point Operations
NASNet	Neural Architecture Search Network
RAD	Rapid Application Development
ROC	Receiver Operative Curve
WDM	Waterfall Development Methodology

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The chapter provides an overview of the project. Section 1.1 will be a study of the project background. In section 1.2, a description of the problems that inspired the project will be discussed. Section 1.3 provides description of the primary objectives of the project. Solutions to the problems stated in section 1.2 will be proposed in section 1.4. Subsequently, section 1.5 will briefly describe the development methodology used and how the solutions were implemented. Lastly, section 1.6 will provides a description of the project scope.

1.2 Project Background

Zoo has always been a place for children to observe and learn about non-domesticated animals physically. At a zoo, Kids are able to get engaged with the animals and study what they are not able to acquire from piles and piles of textbook. When they are provided a chance to observe animals' behaviour up close in such naturalistic environment, they will be motivated to learn more about those living creatures. (Ballantyne et al., 2007). While there are information signs regarding each animal in zoo, most of the kids will pay less to no attention to them. This is because most of the signs contain scientific terms or complicated phrases that are difficult to be understood by kids.

In the process of learning and studying animals, it would be absurd to assume that a teacher will be around whenever and wherever his or her student is currently at. Thus, self-learning is inevitable. With current technology, children are able to learn without the involvement of a human being. Image recognition with less human interaction had made possible with data mining. Hence, a system which integrated data mining technology will be able to provide information on any objects or organisms to its user without text input.

In the technology-driven era, people seem to be inseparable with technology gadgets, especially smartphones. According to a handphone survey by Malaysian Communications and Multimedia Commission (2017), users of smartphone had increased from 68.7% to 75.9% from the year 2016 to 2017. Compared with the other

age groups, the youngsters rely more on their smartphones. Most of the parents in Southeast Asia allows their children to have the device solely for educational reasons (Anon, 2014). Therefore, it can be inferred that smartphones, along with an image recognition system, can be used in tandem to facilitate the process of learning among children.

1.3 Problem Statement

In the current market, most of the android apps focus on plants recognition instead of animal recognition. Besides, there is no animal recognition android app made specifically for zoo animals and for kids. With the dataset retrieved from Kaggle, it was found out that there are only 147 educational apps in Google Play Store that are related to zoo, animal and image recognition. Nonetheless, in the research, apps that combine the concept of zoo, animal and image recognition is absent. The lack of the animal recognition app causes parents to have limited options in selecting an educational app that favours their need.

Children are always encouraged to learn more about the things that are taught. However, it is difficult for children to get their questions answered when their teachers or parents are not around. It is also impractical for kids to rely on their teachers and parents for knowledge. Thus, kids are often demotivated to learn more and expect to be spoon-fed.

On top of that, it is rather inefficient and time consuming to search for books or look up on the Internet. This causes the learning process to be slow and dull.

1.4 Project Objectives

The primary objectives of the projects are:

i. To develop an educational Android app that recognises animals in zoos for kids.

ii. To train a convolutional neural network for automating animal recognition using transfer learning based on the collected images.

1.5 Project Solution

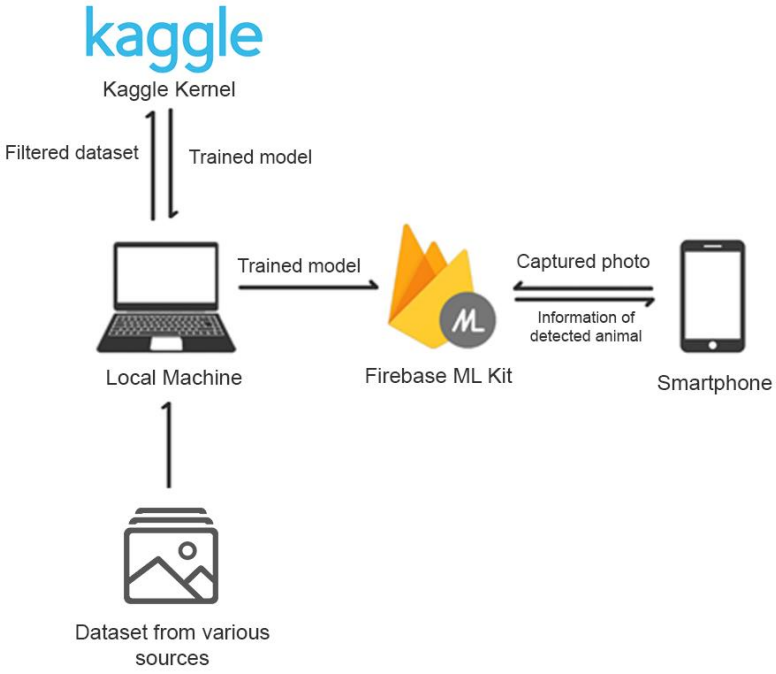


Figure 1.1 Proposed solution overview

Deep learning technology was applied in the app to aid the kids in self-learning without involvement of teachers or parents. The app is able to automate the process of animal recognition and display the animal's information based on the photos captured by the kids. Figure 1.1 shows how a trained model is accessed by the kids' smartphone to classify animals. In the first stage, animal images from various sources will be downloaded and filtered on the local machine. Next, a fine-tuned image classification model which is retrained in Kaggle Kernel will be uploaded to Firebase ML Kit. When the kids captured an image of animal, the image will be sent to Firebase ML Kit. Subsequently, Firebase ML Kit will analyze the received image and perform image recognition with the hosted model. Lastly, output of the analysis will be sent to the kids' smartphone.

By building this app, the problem of shortage in animal recognition app can be resolved. The project will focus on building an app that is specialized for kids.

Besides, the utilization of gamification concept in the app will attract kids into learning. Gamification is applied by assigning different challenges to kids. An example of the challenges would be capturing three images of mammals. The app will not

require text input but merely image input from the user to recognize the observed animals.

1.6 Project Approach

Phased development methodology will be implemented in the development of this application. By using phased development, flaws from the system and possible improvements can be determined in the early stage. In this project, the fundamental function, which is the image recognition system, will be delivered at the early stage so that further improvements can be made.

To make the app kids-friendly, a user interface will be designed with bright colors and simple buttons. Cartoonish images will be incorporated into the app's user interface as well to capture the children's interest.

Machine learning library, Tensorflow, will be used to pre-process batches of images. A classification algorithm will then be applied to gain new insight from the pre-processed data and produce a model. Subsequently, the model will be used to classify the image obtained from the Android app.

To get the users engaged with the app, unlockable badges will be included in the app. The badges act as a reward when kids completed challenges given by the system. Furthermore, users will interact with the app with buttons and camera rather than text input.

1.7 Scope of the Project

This project is aimed to create an educational Android app named Animalia that helps kids to recognize and study animals in a zoo. The project is targeted to deploy on Android platform because there is a lack of Android apps that assist kids in studying zoo animals via image recognition. The targeted users of the application will be kids who are studying primary school. This is because parents are more likely to let their six to eight years-olds children to use smartphones. (Anon, 2014)

The app will promote self-learning by letting the kids identify animals by themselves instantaneously. Photos snapped by kids with their smartphones will be analysed using a model built with data mining technology. The system will then output the name of the animal snapped and its profile such as its common names, scientific names, characteristics and habitat. A logbook that records where and when an animal is captured will be created so that users can revise the animals that they had seen in

zoo. Through capturing different animals and completing challenges, the users are able to earn badges. For instance, when the kids completed the challenge that requires them to capture 3 photos of mammals, they will be rewarded with a badge in the app as a sign of achievement. The purpose of designing a badges-collecting system is to increase the interest of kids in learning.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a review of literature was done on writings with relevance to four of the following topics:

1. Software Development Methodology
2. MobileNet
3. Existing mobile applications
4. Number of existing apps in Google Play Store

Comparison between the characteristics of different Software Development Methodologies will be discussed in the first part of the chapter. Convolutional Neural Network is a subset of deep learning which specializes in image recognition. The third part of the chapter focuses on MobileNet, a depthwise separable convolution network that is typically used in machine vision. Subsequently, a review of the existing mobile applications in the market will be studied in the final part of the chapter. On top of that, brief research on the number of existing apps in Google Play Store will be discussed in the last subsection.

2.2 Literature Review on Software Development Methodology

Software development methodology is a model that is used in constructing, coordinating and outlining an information system project (Centers for Medical Care & Medicaid Services, 2008). Generally, a software development methodology is composed of three phases in Software Development Life Cycle, which are: planning phase, analysis phase, design phase and implementation phase. According to Pavaloaia (2012), the domain of a software project decides which software development methodology to be used. In this literature review, we will focus on some of the predominant methodologies that are commonly used. The methodologies that will be discussed are:

- Waterfall Development Methodology (WDM)
- Rapid Application Development (RAD)
- Agile Development Methodology (ADM)

2.2.1 Waterfall Development Methodology (WDM)

The Waterfall Development Methodology refers to a continuous software development process which developer carry out a series phases of activity sequentially before completing an information system (Bassil, 2012). This implies that a preceding phase should be completed before moving on to the next phase. Thus, the project progress forwards in the same manner as a waterfall. Although it is possible to go back to the previous phase, reversing back to the previous phase is costly in time and funds. Therefore, when the WDM is applied, it is rather vital for project managers to ensure that the project is thoroughly planned and designed. It is also important to ensure every requirement is specified in the planning stage.

2.2.2 Rapid Application Development (RAD)

Rapid Application Development was first introduced by James Martin (1992 cited in Daud, Bakar and Rusli, 2010) in his book with the same title. Martin described RAD as a methodology that improves the traditional methodology in quality and time consumed. RAD achieves faster speed in analysis, design and implementation phase by using particular skill and tools. An evident characteristic of RAD is that developers focus on one major requirement at a time and deliver a minimum viable system to the users in order to receive feedback. Thus, RAD has a shorter implementation time and it is able to improve user satisfaction.

2.2.3 Agile Development Methodology (ADM)

According to Dennis, Wixom and Roth (2012), Agile Development is a collection of programming-centric models that reduce majority of the modelling and documentation and focuses on direct communication. As opposed to the WDM, Agile emphasizes on short and iterative development lifecycle. Some of the renown methodologies that are derived from Agile Methodology are Scum and Extreme Programming. These methodologies are usually applied by project managers in a team project. Because of frequent customer involvement, ADM allows precise requirements to be collected. Apart from that, an alteration in requirements is possible without having a major delay (Pavaloaia, 2012). Hence, ADM is highly flexible as compared with the conventional approach such as Waterfall Methodology.

2.3 Literature Review on MobileNet

Deep learning is an advanced approach of Machine Learning that is able to extract data features with its multi-layered processing layers (LeCun, 2015). Conventional machine-learning techniques require expert knowledge from professionals to perform feature engineering. On the other hand, Deep Learning is capable of recognizing features from image automatically. However, to find relevant features from raw data, Deep Learning requires huge dataset to search for optimal features. LeCun, et al. (2015), stated that Deep Learning had outperformed machine learning techniques in image recognition and speech recognition. An example of the difference between the two methods is shown in the figure below.

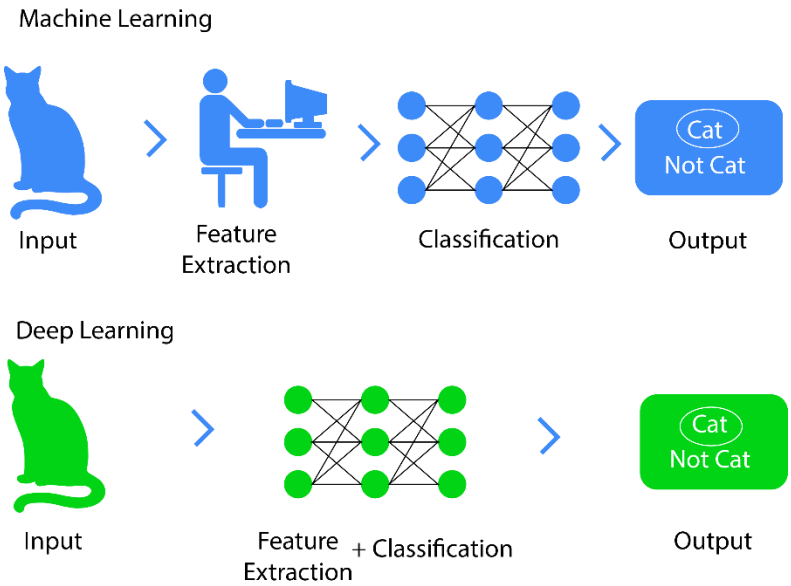


Figure 2.1 Comparison between machine learning and Deep Learning (adapted from Feunteun, 2019)

Deep learning was chosen for the image recognition module in the app because of the limited knowledge in designing feature extraction layer. Besides, there are many existing models that were built with Deep Learning. By retraining the existing model with a custom dataset, time to train the neural network can be shortened.

Convolutional Neural Network (CNN) is a deep learning algorithm that is primarily used for image recognition and object detection. Filters or kernels are used in CNN to extract certain feature from an input image. Process in CNN can be broken down into two parts: feature extraction and classification. To extract the features of an input image, the kernels were ‘shifted’ or convolved over the input image. In each

convolution operation, the dot products of the input image and kernel matrix were calculated. The figure below shows a visual representation of the process.

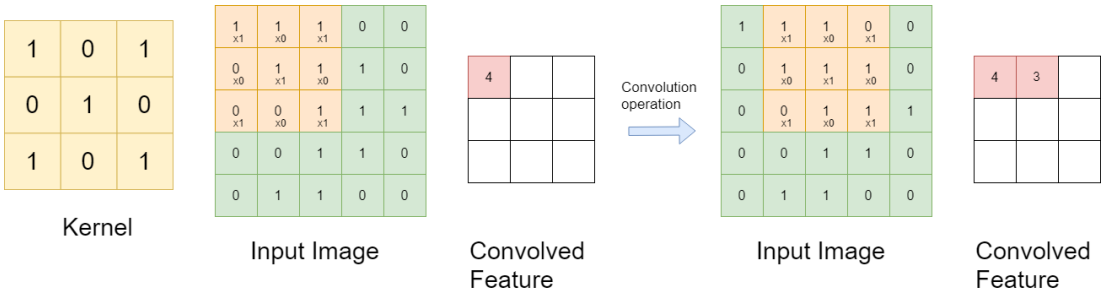


Figure 2.2 Convolution operation (adapted from Saha, 2019)

After the convolution operation is completed, the output will be flattened to allow classification of image features. The flattened layer will then be used as the input of a feed-forward neural network and backpropagation will be performed. Using a classification technique known as Softmax Classification, the model is able to differentiate the major features of the input images.

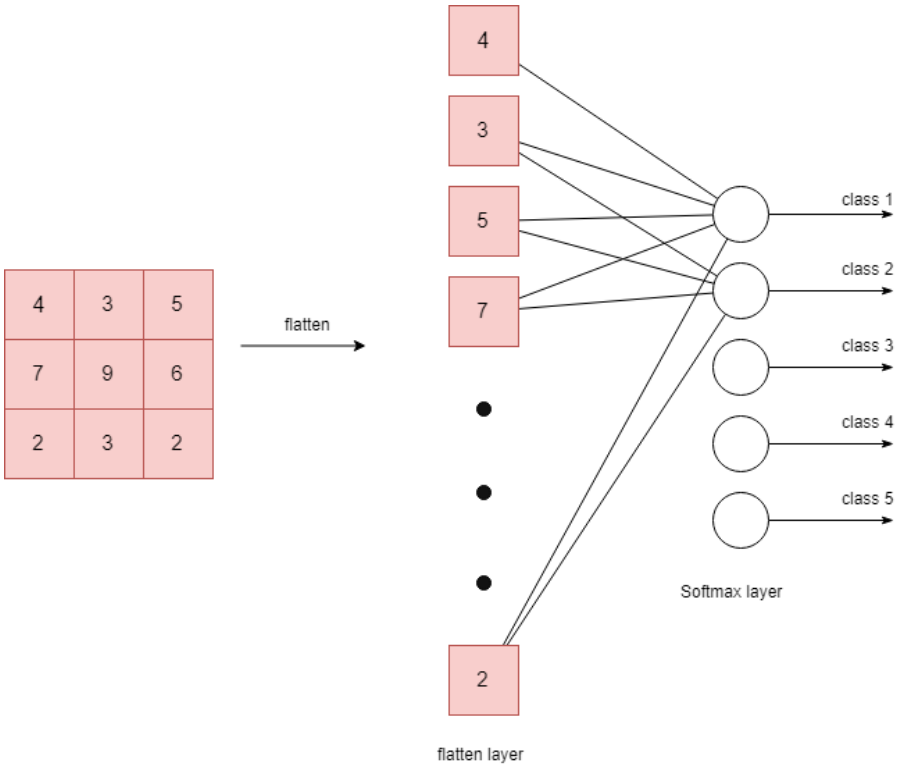


Figure 2.3 Illustration of classification process (adapted from Saha, 2019)

The advent of AlexNet in 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) revolutionized computer vision (Alom, et al., 2018). Since then, deep Convolutional Neural Network is widely used in neural networks such as the

Inception model developed by Google. In the current trend, accuracy is the major aspect to focus on when the developers train their models. However, efficiency in terms of speed and size is neglected in most of these models. This literature review will be focusing on MobileNet, a streamlined pre-trained model that emphasizes in both accuracy and efficiency.

In the research paper about MobileNet written by Howard, et al. (2017), it is stated that MobileNet is built on depthwise separable convolution except for its first layer. Unlike standard convolution neural network, depthwise separable convolution utilizes lower computational power to image convolution. The approach divides a standard convolution into two processes: depthwise convolution and pointwise convolution. The reason for separating the standard convolution into two is to reduce the occurrence of multiplicity, which will affect the computational cost.

The computational cost for a standard convolution can be denoted as:

$$D_F \bullet D_F \bullet M \bullet N \bullet D_k \bullet D_k$$

i. D_k indicates a square kernel's dimensional width and height.
ii. M indicates the input channels amount.
iii. N indicates the output channels amount.
iv. D_F indicates a square feature map's dimensional width and height.

The total sum of depthwise separable convolution cost:

$$M \bullet N \bullet D_F \bullet D_F + D_k \bullet D_k \bullet M \bullet D_F \bullet D_F$$

i. D_k indicates a square kernel's dimensional width and height.
ii. M indicates the input channels amount.
iii. N indicates the output channels amount.
iv. D_F indicates a square feature map's dimensional width and height.

The decrease in multiplication is able to reduce the computation by 8 to 9 times having a significant decrease in accuracy. In this stage, the size and speed of the base network are already relatively smaller and faster. However, in order to tailor the network for a smaller application, the network needs to be enhanced further. Thus, α, a width multiplier is introduced to lessen the network in each layer. α is commonly set at 1, 0.75, 0.5 and 0.25. To reduce the computational power even further, a resolution multiplier, ρ, will be multiplied with the spatial size of the feature map. The resolution

multiplier is ranged between zero to one to produce an input resolution of 224, 192, 160 and 128. After resolution multiplier is applied, the total computational cost of MobileNet can be concluded as:

$$\alpha M \bullet \alpha N \bullet \rho D_F \bullet \rho D_{F+} D_k \bullet D_k \bullet \alpha M \bullet \rho D_F \bullet \rho D_F$$

i. D_k indicates a square kernel's dimensional width and height.
ii. M indicates the input channels amount.
iii. N indicates the output channels amount.
iv. D_F indicates a square feature map's dimensional width and height.
v. α indicates the width multiplier.
vi. ρ indicates the resolution multiplier.

This approach was proven by Howard, et al. (2017) to be effective and efficient as compared with standard convolution. Full convolution MobileNet was found out to have an accuracy of 71.7% while a MobileNet without full convolution achieved a percentage of 70.6%

In MobileNet V2 architecture, linear bottlenecks and inverted residuals are introduced to enhance the performance of the base depthwise separable convolution network. According to Sandler, et al. (2019), it was found out that a ReLu transformation is correlated with a linear transformation if the output volume of the input manifold remains non-zero after the transformation. Secondly, ReLu is only able to conserve information of an input manifold which is in a low-spatial subspace. The diagram below shows how low-dimensional subspace alter the input manifold.

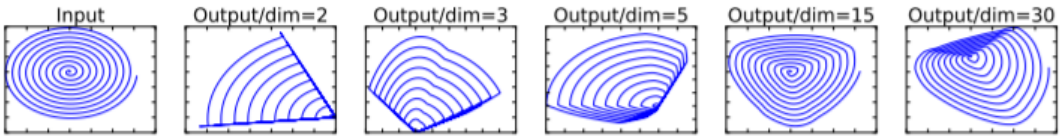


Figure 2.4 Illustration of ReLU transformations of low-spatial manifolds (adapted from Sandler, et al.)

To prevent non-linearities from corrupting large amount of information in low-dimensional subspace, a linear bottleneck layer is inserted into the convolution block. The comparison between depthwise separable convolution with and without linear bottleneck is shown in figure 2.5 and figure 2.6 respectively.

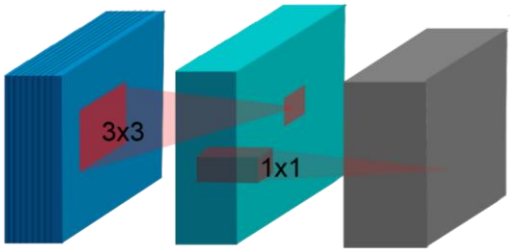


Figure 2.5 Depthwise Separable Convolution (adapted from Sandler, et al., 2019)

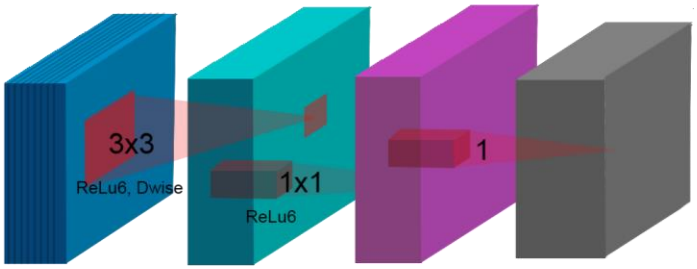


Figure 2.6 Depthwise Separable Convolution with a linear bottleneck (adapted from Sandler, et al., 2019)

Through the experiments, Sandler, et al. (2019) found out that a shortcut can be positioned directly between bottlenecks as the bottlenecks are where most of the information resides on. Expansion layer, on the other hand, is just implementation detail that coexists with a non-linear alteration. The connection between bottlenecks is shown in figure 2.7.

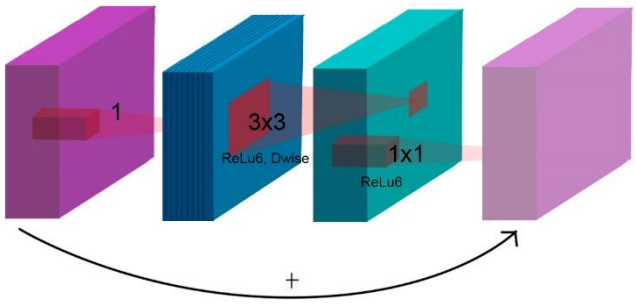


Figure 2.7 Bottleneck with expansion layer (adapted from Sandler, et al., 2019)

MobileNet V2 was proven to outperform its predecessor in the experiment of object detection conducted by Sandler, et al. (2019). The result of the experiment is shown as follows:

Table 2.1 Comparison between the performance of MobileNet V1 and Full Convolution MobileNet V2

Network Name	Top One Accuracy	Paramaters (Million)	Million Mult-Add	CPU Usage (milli)
MobileNet V1	70.6	4.2	575	113s
MobileNet V2	72.0	3.4	300	75s

2.4 Literature Review on EfficientNet

EfficientNet or EffNet is another lightweight CNN that uses similar approach as MobileNet V2. Instead of only applying Depthwise Separable Convolution, the neural network also uses Spatial Separable Convolution. Conceptually both of these convolution algorithms are similar to each other. However, Spatial Separable Convolution, as explained by its name, only deals with spatial dimension and "depth" dimension is neglected. A common example of Spatial Separable Convolution will be performing a convolution with a 3x1 kernel followed by a 1x3 kernel.

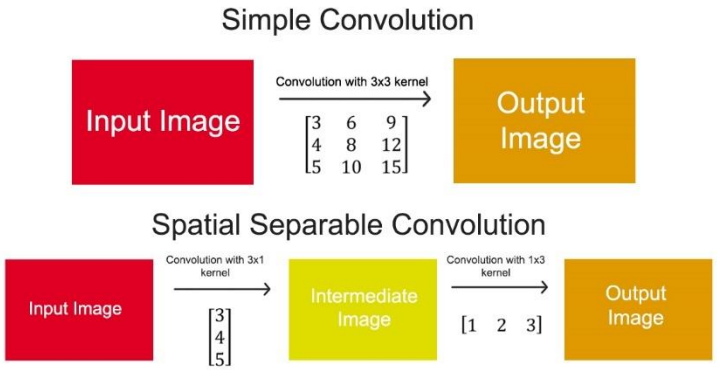


Figure 2.8 Difference between Simple Convolution and Spatial Separable Convolution (adapted from Wang, C.F., 2018)

Combining the concept of depthwise and spatial convolution, the computational cost can be reduced to half of normal depthwise convolution (Douillard, 2018). Besides, seperable pooling is also applied. Pooling is a technique usually used in CNN to reduce

the spatial size of an input image by extracting the features in a region using a filter kernel.

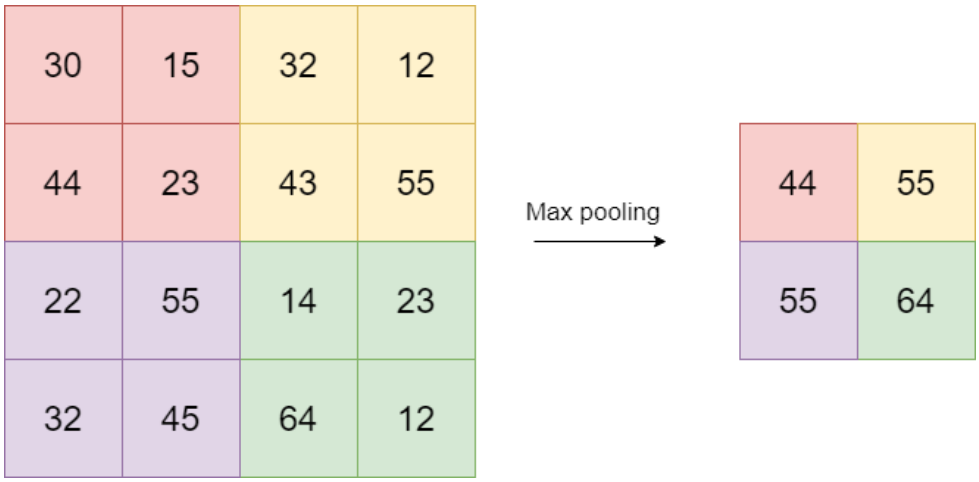


Figure 2.9 Example of max pooling (adapted from Dertat, 2017)

To perform seperable pooling, a 2x1 pooling kernel is applied after the first spatial convolution layer. The second pooling phase will then be carried out after the last pointwise convolution.

Another significant difference between EfficientNet and MobileNet is the replacement of first layer. As mentioned earlier, the first layer of MobileNet was not built with the concept of Depthwise Separable Convolution. Based on the research by Freeman, et al. (2018), the size of first layer increased after all the other layers are optimized. Hence, the first layer was replaced with EffNet block as well. As a result, around 30% of the respective layers computational power was reduced.

2.5 Literature Review on NASNet A-Mobile

NASNet or Neural Architecture Search Net took a different approach compared with MobileNet and EffNet. Neural Architecture Search (NAS) is a deep learning technique that is utilized to automate the process of artificial neural network design. Opposingly, MobileNet and EffNet models were hand-designed architectures that requires human intelligence. In this subchapter, the basic concept of Recurrent Neural Network (RNN) and the procedure of building a neural network with NAS will be briefly explained.

RNN is a neural network that takes the output from previous step as the new input of current step. RNN, unlike any other neural network, has hidden state that enable it to “remembers” all the information that was calculated previously. In traditional neural network, different parameters are used for each input, which means that each of the input layers are independent of each other. On the other hand, RNN

provides the same parameters for each input layer and join every input layer into a single recurrent layer. By joining the layers, the inputs are now dependent to each other and the “memory” of the network is able to be passed from previous step to current step.

The steps of creating a neural network with NASNet is listed below:

1. A hidden state is selected from the set of hidden states created in previous blocks.
2. A second hidden state is selected.
3. An operation is selected to apply to the hidden state chosen in Step 1.
4. An operation is selected to apply to the hidden state chosen in Step 2.
5. The outputs of Step 3 and 4 are combined to create a new hidden state.

A set of common operations used in CNN is used for step 3 and 4. The list of operations is shown below:

- identity
- 1x3 then 3x1 convolution
- 1x7 then 7x1 convolution
- 3x3 dilated convolution
- 3x3 average pooling
- 3x3 max pooling
- 5x5 max pooling
- 7x7 max pooling
- 1x1 convolution
- 3x3 convolution
- 3x3 depthwise-separable conv
- 5x5 depthwise-seperable conv
- 7x7 depthwise-separable conv

As a result, NASNet A, B and C are formed. The best CNN created with NAS is NASNet A according to an experiment performed by Zoph, et al. (2018).

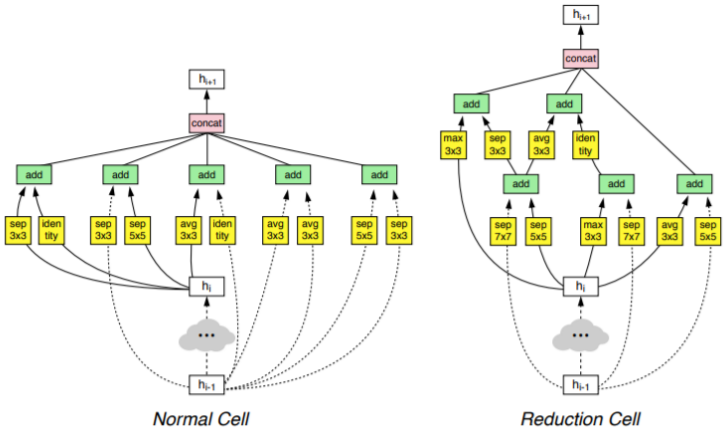


Figure 2.10 Architecture of NASNet A (adapted from Zoph, et al.)

In the diagram shown above, different operations were selected and combined together to produce a new hidden layer. To make the architecture scalable, two types of convolutional cells were designed. The normal cells output a feature map with unchanged dimension whereas the reduction cell return a feature map which height and width is decreased by a factor of two.

2.6 Literature Review on Existing Mobile Application

Three of the existing Android mobile application that performs animal image recognition will be reviewed in the following analysis.

2.6.1 Seek

Seek is a mobile application that was developed by iNaturalist for identifying organism including animals and plants. It spans across multiple mobile platform such as Android and iOS.

Registration is not required to use the app. In fact, a user can access all the feature of the app without creating an account. The app is capable of real-time image recognition, which means that the target can be analysed before the photo is captured. After an image of a plant or animal is captured, the data, such as name, species and description, of the captured target will be shown. The location of the captured species will be collected by the app. With the locations collected, the app is able to show nearby places where the same animal is discovered by another user. Another

fascinating feature of the app is its badge collecting system. Badges can be unlocked by completing requested task such as observing one bird.

Overall, the app provides a simple and attractive user interface that is suitable for all ages. Its badge collecting system is captivating for children as well as adults. With the large dataset provided by iNaturalist, Seek is able to identify most of the subject with high accuracy. The app also provides an informative description of the observed subject. However, the description of the subject is retrieved from Wikipedia which might not be a reliable source. Besides, some of the scientific terms in the description might not be understandable by the children.

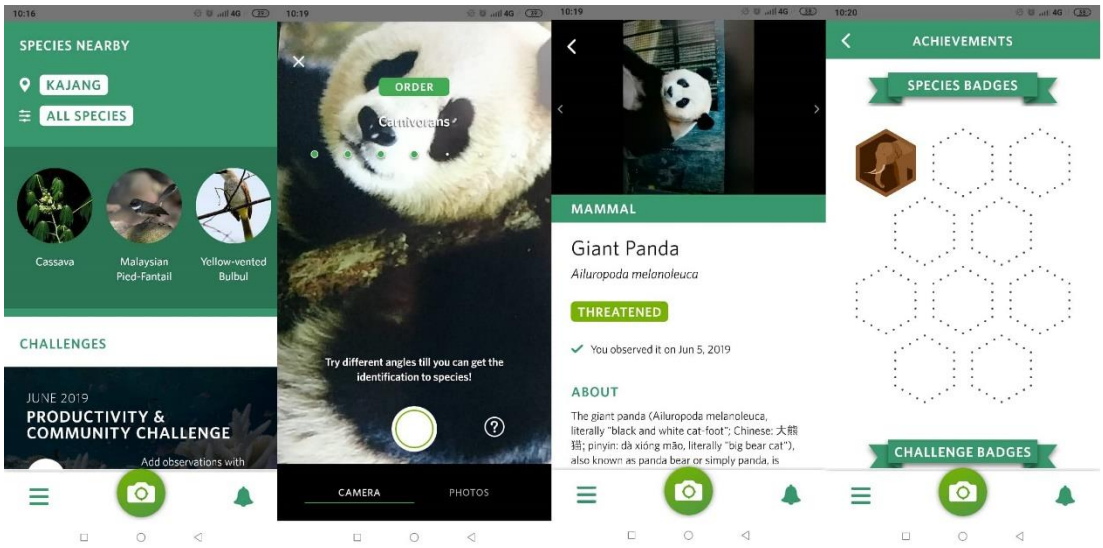


Figure 2.11 Features of Seek (screenshots taken from iNaturalist, 2019)

2.6.2 Google Lens

Google Lens is a powerful image recognition app that allows users to perform a visual search. It is capable of scanning text, searching for similar products, recognizing animals and identifying popular landmarks. Google Lens can be accessed with most of the Android smartphones through Google Assistance or Google Photos.

To use Google Lens, users will have to first launch Google Assistance by pressing the home button. When Google Lens detects an object, a coloured dot will be shown on the subject of the image. Users can view the details of the subject by tapping on the symbol. The information of the objects varies based on the type of object detected. If animal is detected, a snippet of the animal's characteristics will be shown. If a product is observed, the price of the product will be shown. There are also other features such as copying text in the image to the users' devices.

One of the advantages of Google Lens is that it is preinstalled in most of the Android devices along with Google Assistance. Hence, manual installation from the App Store is not required. However, it is inconvenience for user to launch Google Assistance or Google Photo if their main intention is to use Google Lens. Although the standalone version of Google Lens is available for download, it is incompatible for most of the Android devices, according to Liao (2018). Another drawback of Google Lens is its data source. Similar to Seek, Google Lens retrieve its information of animals and plants from Wikipedia.

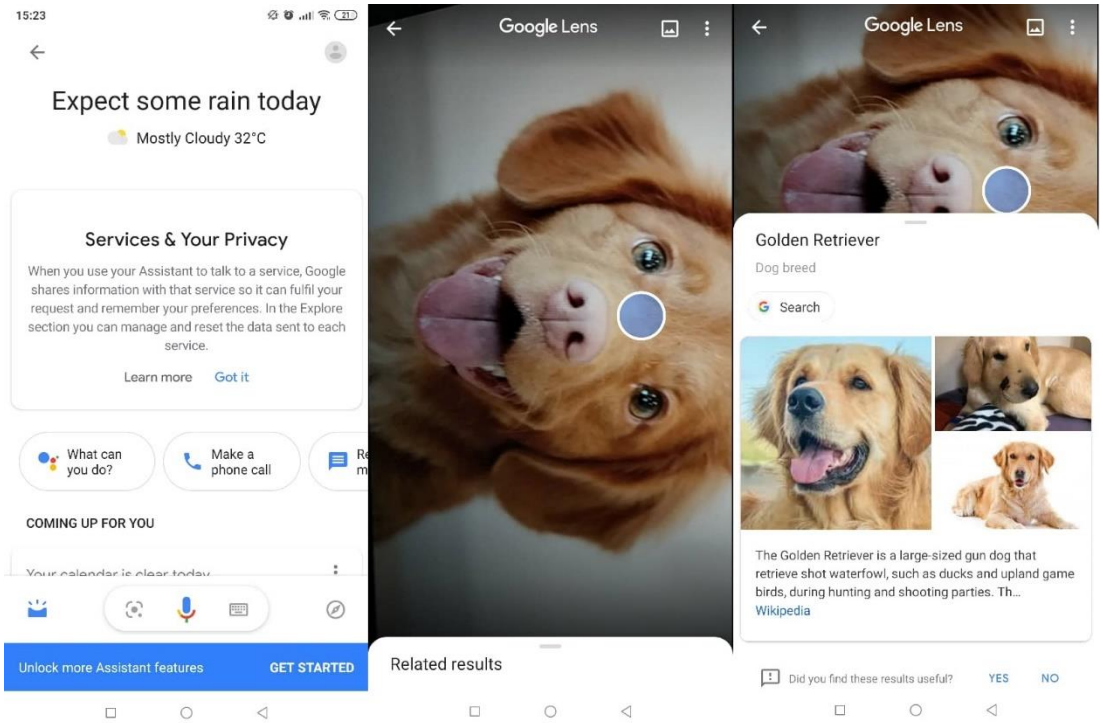


Figure 2.12 Features of Google Lens (adapted from Google, 2019)

2.6.3 Image Recognizer

Image Recognizer is an Android app published by LD Studio in Google Play Store. As described by its name, Image Recognizer classifies the image inputted by the user and allows users to explore further on the output through Wikipedia, YouTube or Amazon. Its primary functions include label detection, image attributes and Optical Character Recognition (OCR).

The home screen of Image Recognizer consists of three options for users to submit their input image. The input can be submitted from the camera, existing photo or a randomly downloaded picture. Once a submission method is selected, the app will begin to analyse the image and displays a text output. The app provides function that

allows users to search the information of the output through Wikipedia, YouTube or Amazon. Besides, users can translate the output instantaneously with the translation feature.

Image Recognizer has a unique attribute identifying function. Apart from the name of the subject, it also describes its properties, for instance, colour and size. Nevertheless, the attribute identifying function increase the time for the system to produce the output. As a matter of fact, Image Recognizer takes a longer time to analyse the output compared to the apps discussed previously. Furthermore, Image Recognizer also possesses the same drawbacks, which is, using Wikipedia as its data source. Apart from that, it would be an inconvenience for the user to obtain the information by browsing Wikipedia manually.

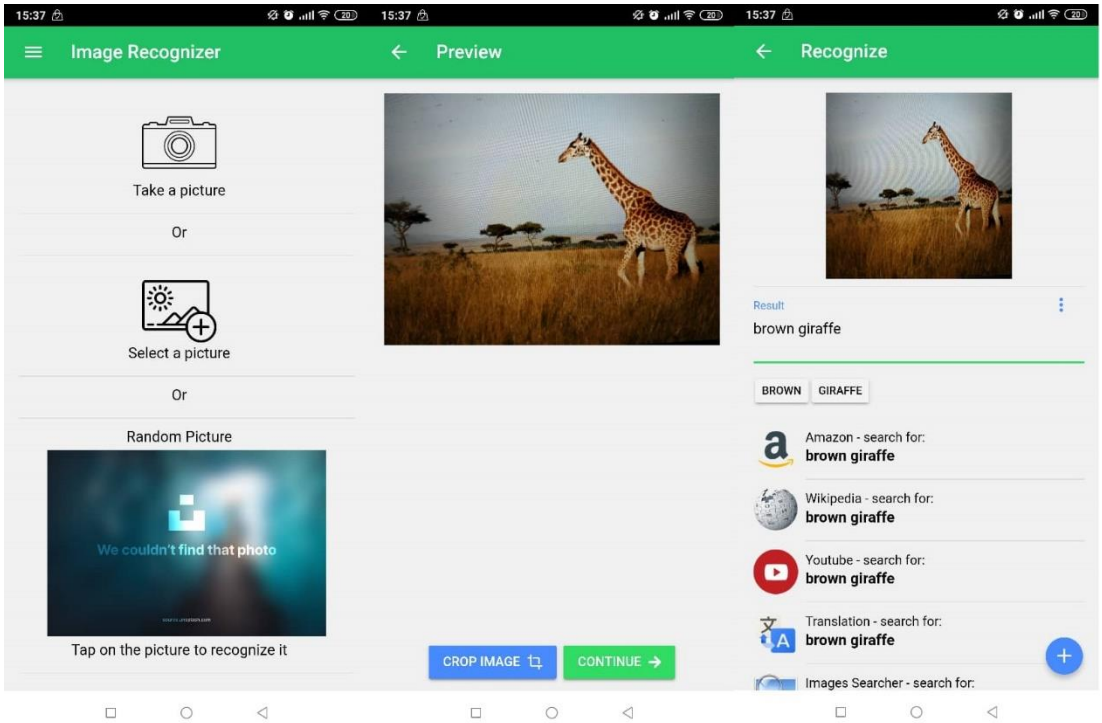


Figure 2.13 Features of Image Recognizer (adapted from LD Studio, 2017)

2.7 Research on Related App in Google Play Store

A sample data of 267,051 apps from Google Play Store is retrieved from Kaggle (Prakash, G., 2019). Descriptions such as app name, category and content rating are provided in the dataset. The data was analysed using Kaggle Kernel with Python script.

Before analysis of the data began, the data was transformed into a compatible format for data extraction to be carried out. For instance, spaces in the dataset's column names were replaced with an underscore. This step is crucial to allow queries to be carried out on the dataset.

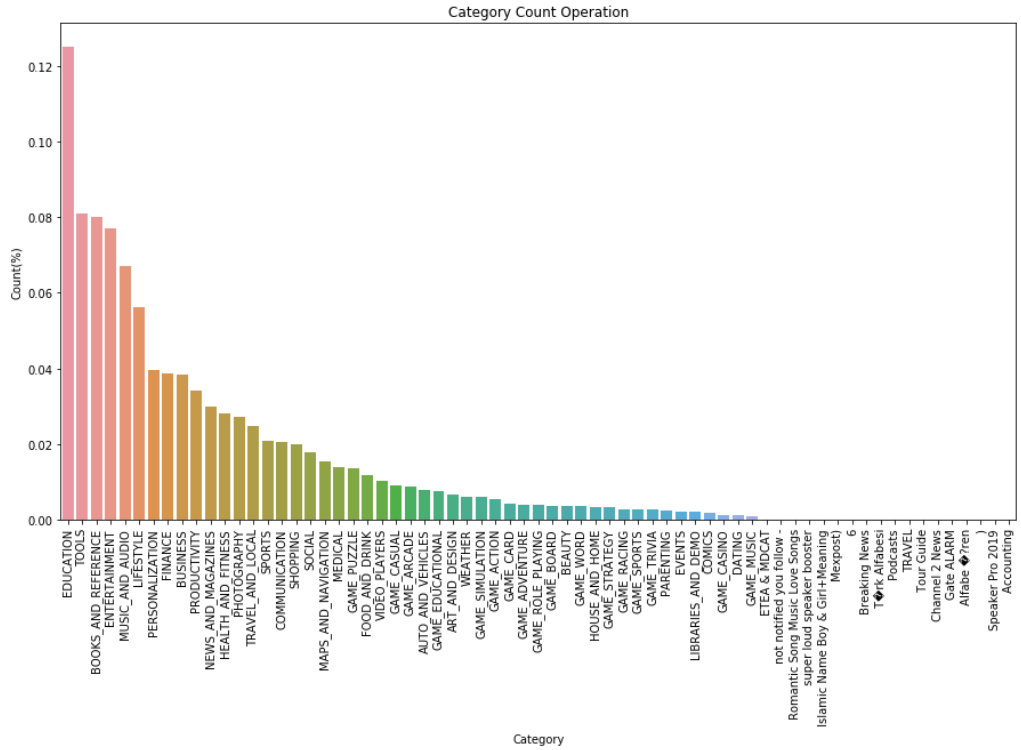


Figure 2.14 Comparison of the number of apps in different categories (based on dataset by Prakash, G. 2019)

Among the apps in the dataset, it was found out that educational apps have the highest amount as compared to other apps. In fact, 0.125% of the apps are composed of educational apps. Followed by educational apps are apps that are categorized in tools and books and reference which are both around 0.08%.

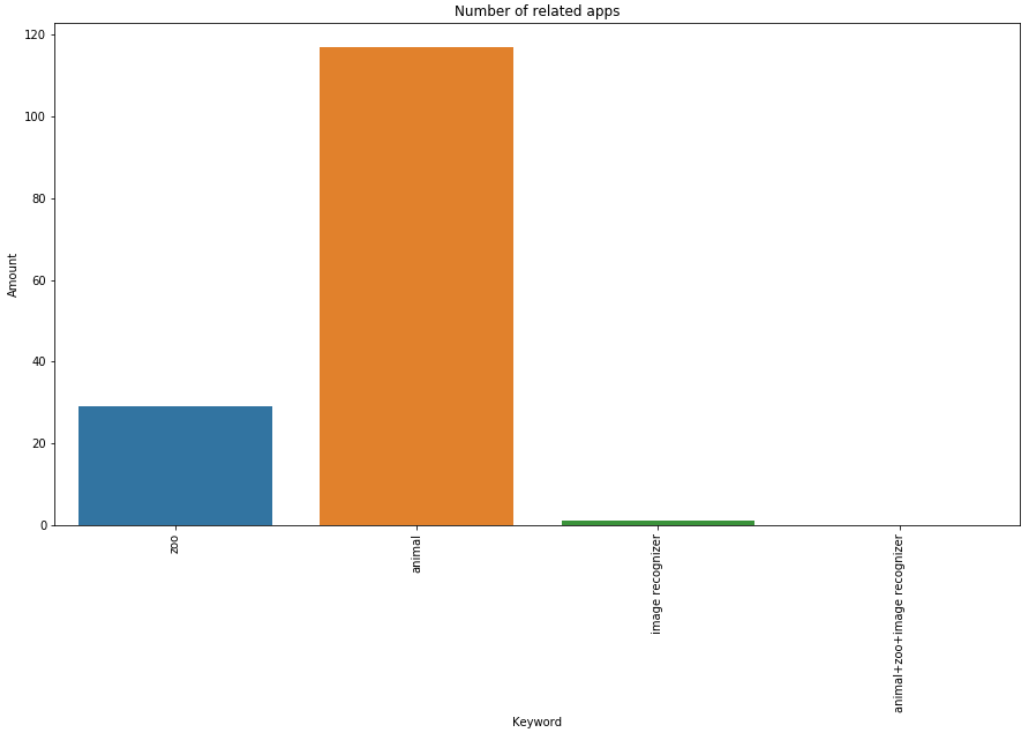


Figure 2.15 Number of zoo animal related apps in Google Play Store (based on dataset by Prakash, G. 2019)

To search for the number of apps related to this project, keywords such as zoo, animals and image recognition are used. Furthermore, the search scope was limited to educational apps and apps that were listed as “Everyone” in their content rating. The results showed that there are none of the apps that are related to three of the major elements. However, there are 117 of the apps that are related to the keyword ‘animal’ 29 apps that are related to ‘zoo’ and one app that is related to ‘image recognition’. In conclusion, it can be inferred that the number of zoo animals image recognition apps is limited.

2.8 Summary

In the first part of the literature reviewed, three of the major software development methodology is discussed. Section 2.2.1 is summarized in the form of a table in the following.

Table 2.2 Comparison between three of the reviewed methodologies

Software Development Methodology	Characteristics
Waterfall Development Methodology	- Unidirectional - Non-iterative
Rapid Application Development	- An enhanced version of traditional approach - Iterative - User-centric - Fast development - Early user involvement
Agile Development Methodology	- Programming-centric - Emphasize on face-to-face communication with clients - Iterative - Flexible - Widely used in team projects

In comparison to the other two methodologies, Waterfall Development Methodology requires more time and cost because of its non-iterative characteristics. Opposingly, RAD and ADM provide a faster and flexible approach to software development. One of the unique features of ADM is its compatibility on team projects. ADM is widely used in team projects by project managers to manage the team's workflow. On the other hand, the key feature of RAD is the early involvement of user in the project. Prototypes are delivered to the users periodically to receive feedback in the early stages. In this project, RAD was utilized to develop the application as early feedback is utterly important to verify requirements. Besides, the project did not involve any team collaboration.

In Section 2.2.2, an overview of MobileNet V1 and MobileNet V2 is shown. MobileNet V1 utilizes depthwise separable convolution as its base network to ensure

a network that is smaller in size is produced. Furthermore, a width multiplier and a resolution multiplier are included as hyper-parameters for network thinning and input image reduction. The network is proven to have an insignificant difference in accuracy when compared to traditional Convolution Neural Network. In the journal by Sandler, et al. (2019), it was found out that ReLu is unable to conserve the manifold of interest in an input image in low dimensional subspace. Therefore, a linear bottleneck was introduced in MobileNet V2 to overcome this limitation. Sandler, et al. (2019) also discovered that shortcuts can be used as a connection between bottlenecks to further improve the network. This is because most of the useful information exists in the bottlenecks. With two of the techniques, the accuracy of MobileNet was proven to increase in an experiment conducted by Sandler, et al. (2019).

In the final section, three existing Android mobile apps are reviewed. Based on three of the apps, the characteristics of the applications can be sorted into 7 categories. A matrix of the apps and their characteristic is shown below:

Table 2.3 Comparison between three of the reviewed apps

	Seek	Google Lens	Image Recognizer
Internet Access Reliability	High	High	High
Pricing	Free	Free	Free
Data Source	Wikipedia	Wikipedia	Wikipedia
Speed (1 for fastest, 3 for slowest)	2	1	3
Simplicity	High	High	High
User Interface Appearance	Attractive	Attractive	Average
Unique Feature	Badge and challenges assigning system.	Ability to differentiate between objects and organisms.	Image attribute analysis.

From the review, it can be inferred that the data source used by three of the apps might not be reliable. Therefore, in the development of the application, the reliability of data source was taken into account. Another common characteristic of the apps is high reliability on Internet access. By using ML Kit, this shortcoming can be overcome as ML Kit allow image recognition to be done offline. However, the speed of the analysis will be compromised when image recognition is done offline. On top of that, the badge and challenges assigning system in Seek can be taken as an inspiration for the gamification of our app.

To determine the population of apps related to zoo animal and image recognition, a brief analysis of Google Play Store apps was carried out. In the research it was observed that among 267 thousand of apps, 0.125% of the apps are educational app. However, only 147 of the apps are related to zoo animal and image recognition.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

Phased development-based methodology was implemented in the development of the apps. Major considerations that had been taken into account when selecting development methodology were time and clarity in requirements. By using Phased Development Methodology, the core function was delivered in the first version. Other functions apart from the core function were refined or appended into the system in the future versions. In this manner, requirements could be refined in each phase of loop in the early stage.

In this chapter, the implementation of the methodology will be explained. Besides, the process of model training will be elaborated. Lastly, A list of tools and hardware used will be discussed.

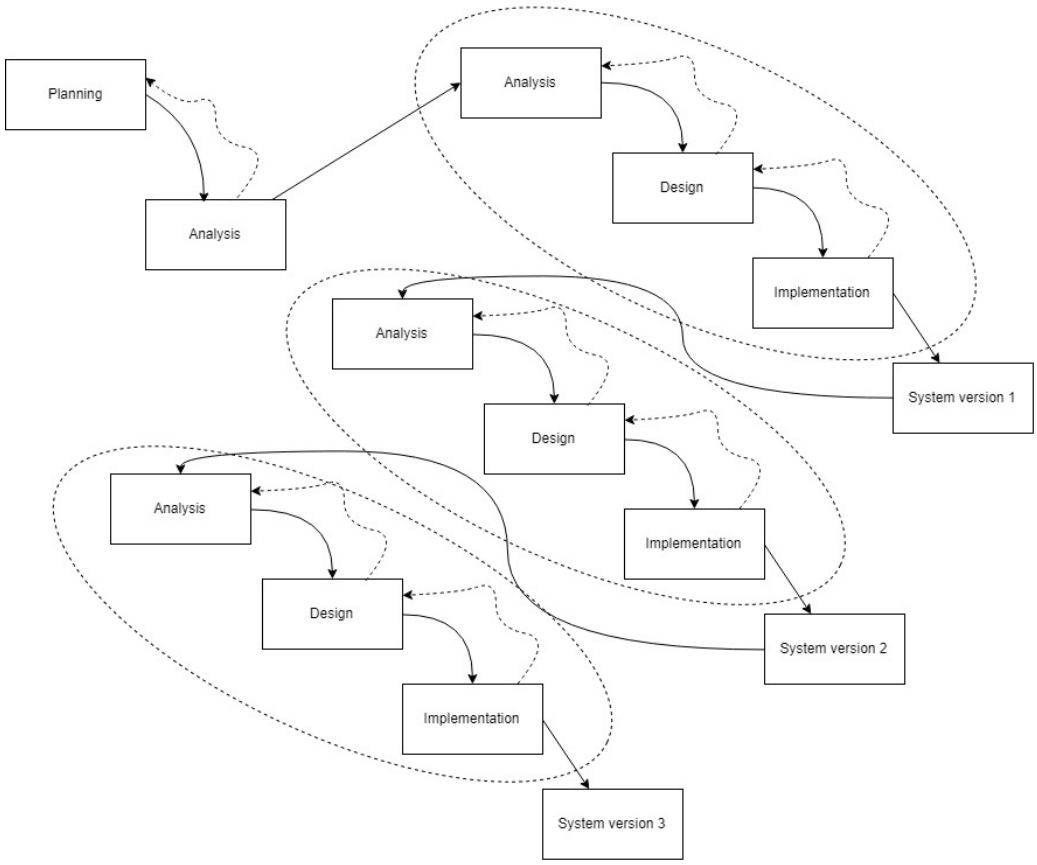


Figure 3.1 Phased Development Methodology (adopted from Dennis, A. et al., 2015)

3.2 General Work Procedure

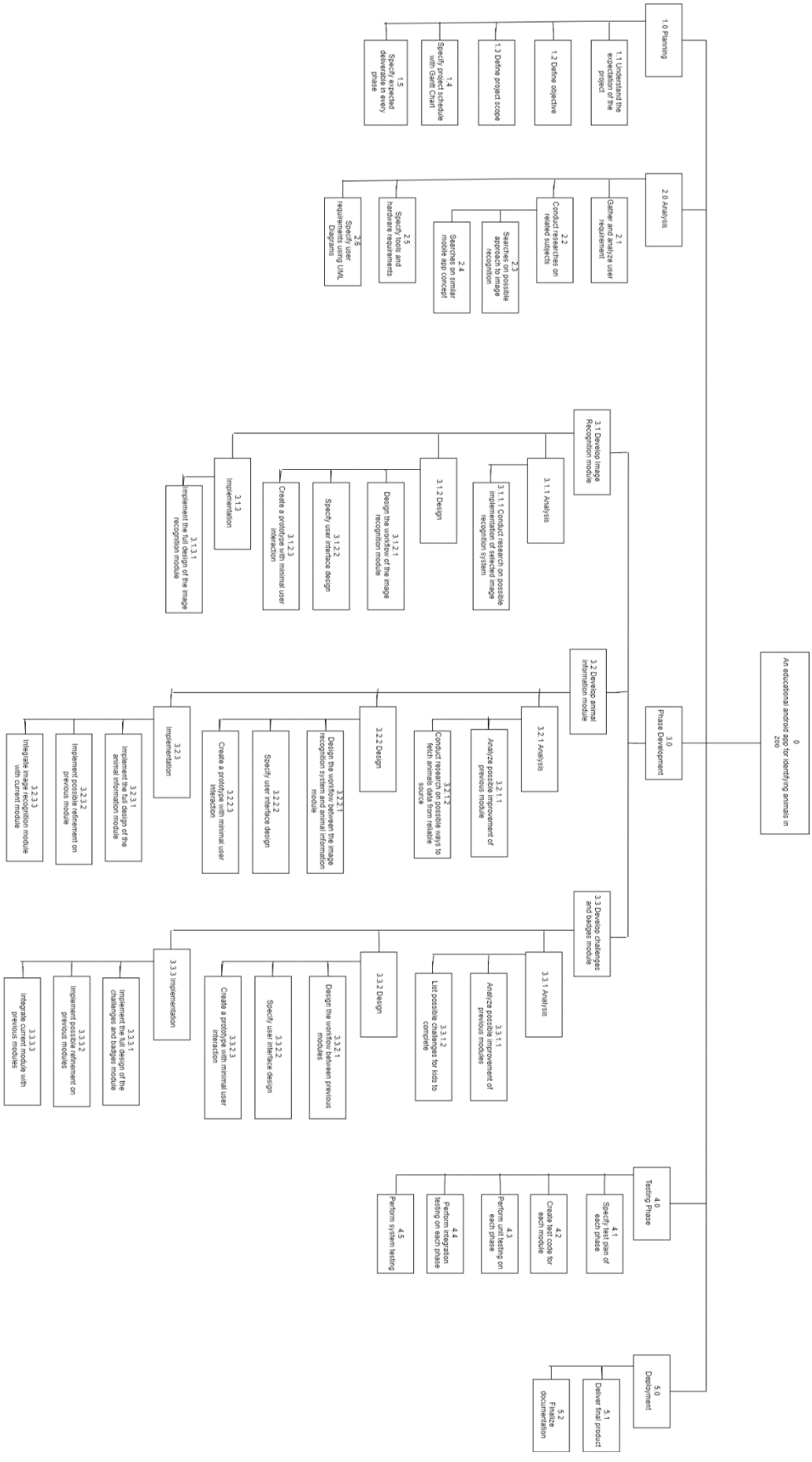


Figure 3.2 Overall work breakdown structure

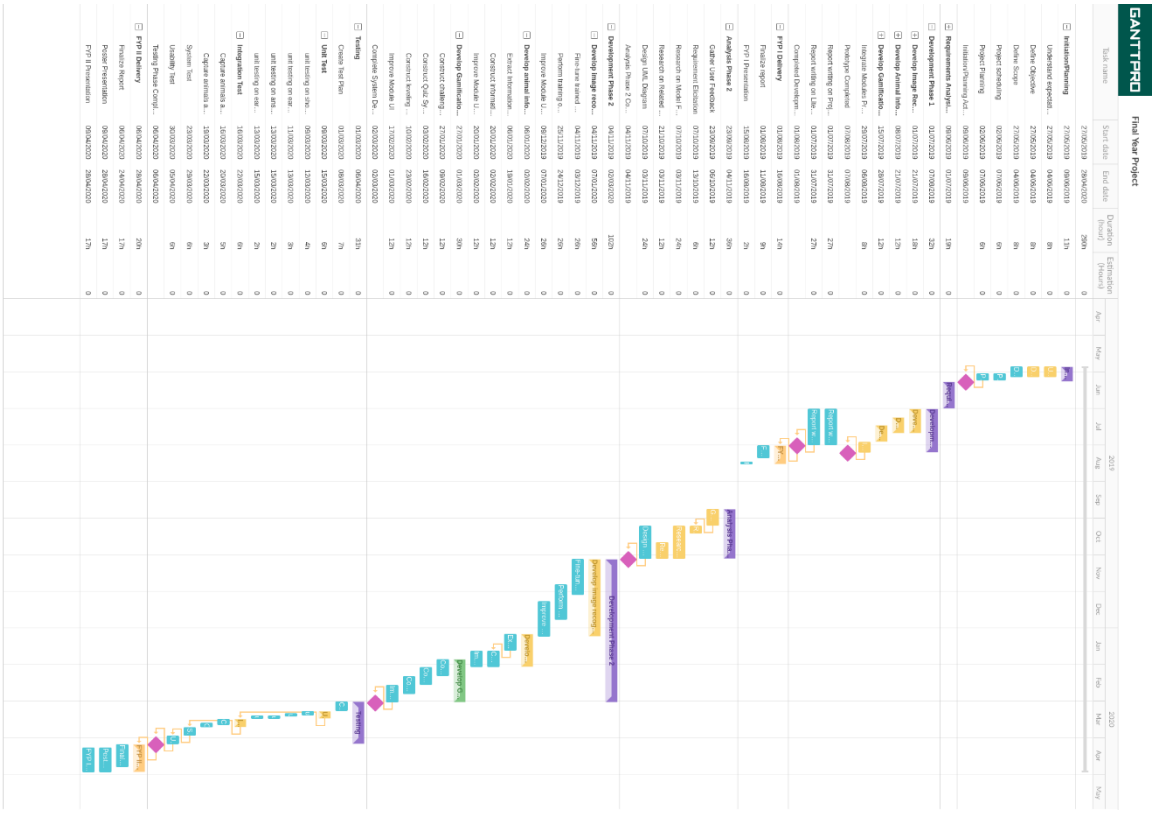


Figure 3.3 Project Gantt Chart

The project was organized into five main stages. In the planning stage, the project objectives were discussed and expected deliverables in each phase were decided. Requirements elicitation and analysis were performed in the next stage, which is the analysis stage. Next, an analysis was carried out again to determine the possible solution for the first development phase. Workflow, user interface design and prototype were created in the design stage of each phase. The actual development of the module occurred in the implementation stage. A similar process was iterated in three of the phases. After three of the modules were developed, test code was designed for each module. On top of that, in the deployment stage, documentation and the product were finalized.

In the first phase, the core function, which is the image recognition module, was developed. In the analysis stage of stage one, the algorithm of Tensorflow and MobileNet V2 image classification model were studied to determine possible ways that these tools can be used. Next, in the design phase, a workflow of the module was designed based on the information found in previous stage. An example of the workflow would be how the android app was connected to the classification model on

a cloud. A minimal prototype was created in the design stage as well. In the development phase, the full module, which consists of the complete user interface and functionality, was developed.

Animal information retrieval module was the focal point of phase two. Analysis stage of phase two was focusing on how the system retrieves information of each captured animal from a reliable source. Possible refinement on the first module was discovered as well. Similar to phase one, the main outcome of the design stage was a workflow, prototype of the module and the user interface. In the implementation stage, module one was refined and integrated with module two.

In the last phase, the gamification module was developed. The main outcome of the analysis stage was a list of challenges for kids to complete. The design and implementation stage of phase three was identical to phase two. By the end of phase three, all module was combined to form a fully functional system.

Testing was carried out at the end of each phase to detect any defect. Three levels of testing: unit testing, integration testing and usability testing were executed. The planned usability test documents can be viewed in Appendix A of this report. Simple languages were used in the post-test interview as the subjects were children. After every stage was accomplished, deliverables were finalised and the app was deployed.

3.3 Image Classification Model Training

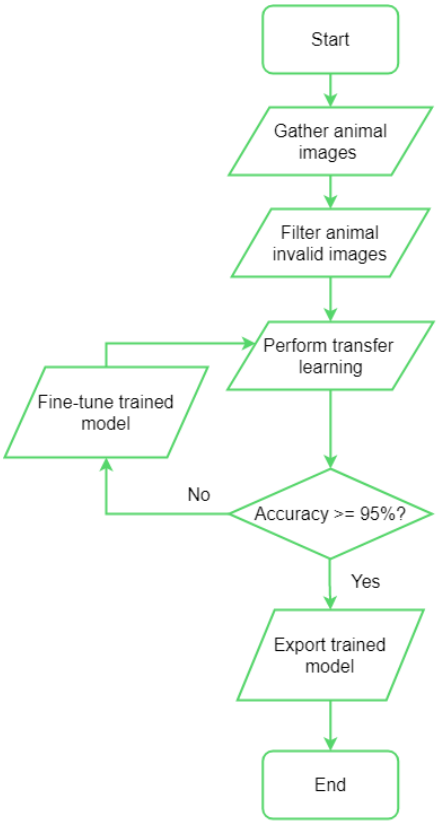


Figure 3.4 Workflow of image classification model training

The first step that was taken to train an image classification model was searching animal images dataset. A dataset published by Afkham (2008) on KTH Royal Institute of Technology site was utilized. The dataset contains 100 images for every 19 classes of animals. Sinnot, et al. (2018) inferred that quantity and quality of raw images are crucial for the accuracy of an image recognition model. Although dataset provided by Afkham (2008) was segmented and high in resolution, the amount was insufficient. Therefore, another batch of images was obtained from WSID-100, a dataset created by Yao, Shen and Zhang (n.d.). WSID-100 provide 2000 images for every 100 categories of objects and animals. As opposed to Afkham's dataset, WSID-100 excellent in quantity but not quality. Images from WSID-100 contain watermarks and objects that are unrelated to the subject of image. Thus, filtering images was vital before using the datasets as input.

Figure 3.5 Examples of inappropriate image (adapted from Yao, Shen and Zhang, n.d.)

Figure 3.6 Example of image cropping (adapted from Yao, Shen and Zhang, n.d.)

Images that were removed when filtering the data were images that contain watermarks, unrelated subjects, occluded animals and blurry images. Some of the images were segmented to remove unrelated image features.

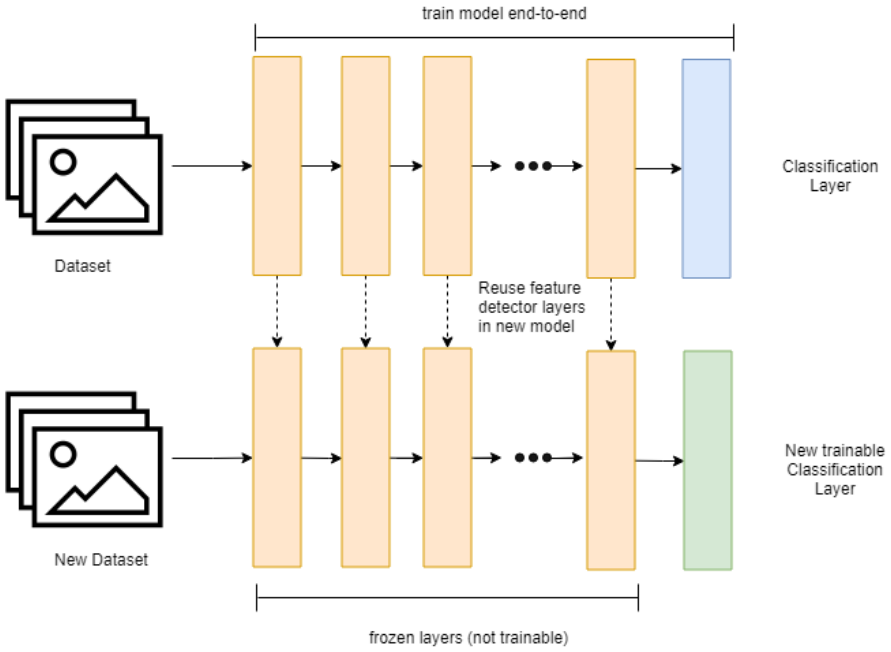


Figure 3.7 Illustration of transfer learning (adapted from Cruz, 2019)

Next, transfer learning was carried out. Reddy and Juliet (2019) stated that transfer learning is a process of conveying the features learned from a pretrained network to a second network to train on a new dataset. In the figure above, it can be observed that the image feature extraction layers are unchanged when they are transferred to a new network. In spite of that, the classification layer can be replaced. By applying transfer learning, a model was built without training from scratch. The approach largely reduces the amount of time to train a neural network. In this project, the base network that was employed was MobileNet V2. After the base network was specified, a classification layer was appended on top of the base network. The actual training of images was begun after compiling the classification layer and the base network. In each epoch of training, validation accuracy, as well as validation loss of input data was displayed. The training was stopped when convergence occurred.

Enhancement of the model was done by fine-tuning the network. Unfreezing the base network was necessary to allow fine-tuning. However, it would be inefficient to unfreeze and fine-tune every base-network layer. Therefore, the layers to be fine-tuned was specified. Two major parameters that were taken into consideration for fine-tuning were:

1. Activation function
2. Learning rate

An artificial neural network is a simulation of a biological neural network. Hence, to illustrate the usage of activation function, a comparison between a biological neuron and artificial neuron is shown below.

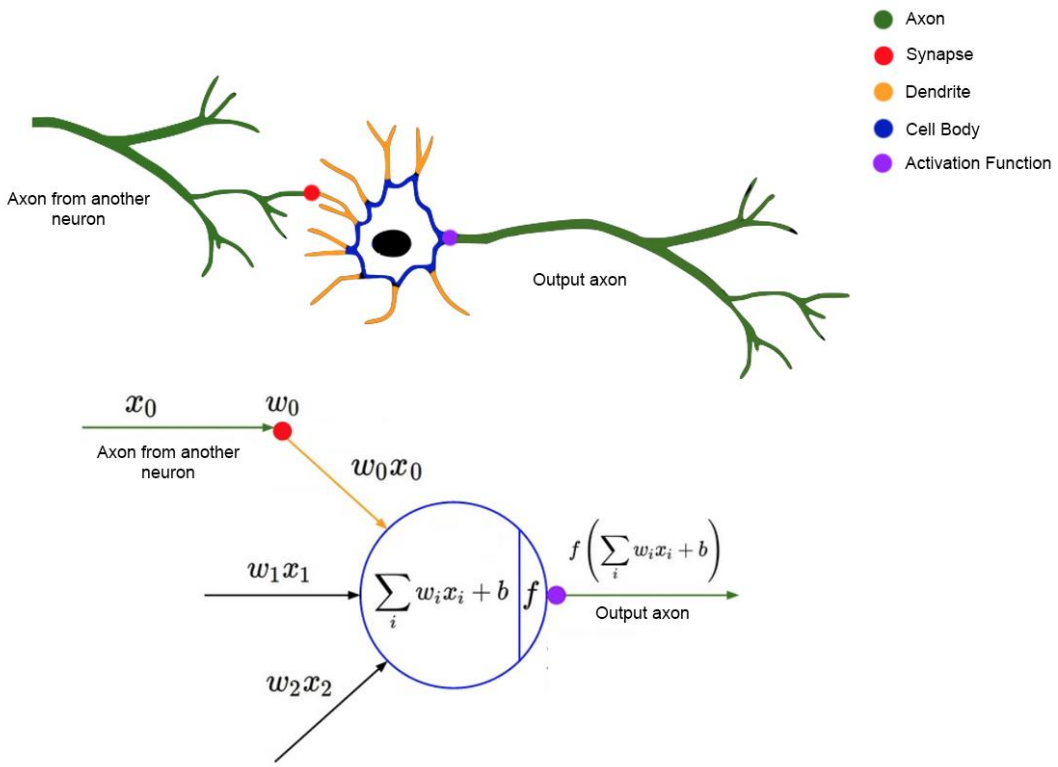


Figure 3.8 Comparison between a biological neuron and an artificial neuron (adapted from Jain, V., 2019)

In a biological neural network, a neuron sends an electrical signal with different weights to other neurons at the synapses of both neurons. When the magnitude of the electrical signals reaches a certain level, the recipient neuron will be activated and will transmit a response to the next neuron. Otherwise, the recipient neuron will remain inactive. To simulate the active state of a neuron, an activation function is added into the artificial neural network. As stated by Ding, Qian and Zhou (2018), an activation function is capable of enabling nonlinear classification in a neural network. Some examples of the common activation functions used in a neural network include Sigmoid, Rectified Linear Units (ReLu) and Tanh. In this project, the activation model used for the initial image classification model was ReLu. Fang, Wu and Sinnott (2018) implied that performance of ReLu was faster than the computationally expensive Sigmoid and Tanh.

In the paper by Fang, Wu and Sinnott (2018), it was observed that a higher learning rate might cause the model to miss the minimum loss in training. On the other hand, a small learning rate has a higher chance to determine the minimum loss in training. However, using a small learning rate is computationally expensive and can

cause the converging process to slow down. A recommendation provided by Fang, Wu and Sinnott (2018) was to apply momentum approach technique. The technique combines both of the advantages of high and low learning rate by reducing the learning rate from high to low periodically. With the mentioned approach, the global minimum can be quickly determined.

After fine-tuning the model, the accuracy of the model was evaluated by training the neural network with input images. Initially, in the prototype, an accuracy of at least 60% and above was aimed to be achieved. If accuracy of the model did not reach the specified percentage, the model would be fine-tuned and trained again. Otherwise, the model would be exported as an output graph and published on Firebase ML Kit.

3.4 Tools to Use

3.4.1 Development Tools

1. React Native

   React Native is a cross-platform mobile framework that uses Javascript. Multiple open-source tools are available for developers to integrate into their Android and iOS mobile apps.

2. Visual Studio Code

   Visual Studio Code is a lightweight source-code editor that provides multiple features such as code completion, syntax highlighting and Git version control.

3. Tensorflow

   Tensorflow is an open-source library widely used in training Machine Learning and Deep Learning models. In this project, we used Tensorflow Lite which is a lightweight version of Tensorflow, designed specifically for mobile devices. Tensorflow library was used for image recognition in this project.

4. Docker

   Docker is a software that utilises container. A container is a software unit that wraps up the software code and the software dependencies with operating system-level virtualisation. With Docker, Tensorflow library and all of its dependencies are packaged up and can be used instantly.

5. Firebase ML Kit

   Firebase ML Kit is a mobile Software Development Kit that uses the functionality of Google's machine learning algorithm. (Google, n.d.) Firebase ML Kit was used to host the Tensorflow Lite model and perform classification on the cloud.

6. Kaggle Kernel

   Kaggle Kernel is a free cloud computational environment that allows the user to perform data analysis. Process of transfer learning can be accelerated with better hardware. Hence, with the powerful GPU and CPU in Kaggle Kernel, amount of time to complete transfer learning can be largely reduced.

7. Sparx System Enterprise Architect

   Sparx System Enterprise Architect is a UML diagram modelling tool. Use case diagram and flowchart was drawn using this tool.

3.4.2 Hardware Used

1. Lenovo IdeaPad 700-15isk

   Processor: Intel® Core™ i7-6700HQ CPU @ 2.60GHz 2.59GHz

   Installed memory (RAM): 12.0GB

   System type: 64-bit Operating System, x64-based processor

   Graphics Card: NVIDIA GeForce GTX 950M

2. Redmi Note 5

   Operating System: Android 9

   CPU: Octa-core Max 1.80GHz

   RAM: 4.0GB

   Camera: 12MP

3.5 Summary

In this chapter, we reviewed the development methodology, overall work procedure, tools that were used and the requirements specification of the project.

As discussed previously, the project was developed with the implementation of phased development methodology. The project was broken down into three phases, in which each phase would produce a module. The modules that were produced, according to sequence, were image recognition module, animal information retrieval module and gamification module.

In the process of image classification model training, image selection and filtering were first carried out. Image datasets from WSID-100 and KTH Royal Institute of Technology were filtered by clipping the images and eliminating inappropriate images. Next, the actual model training was performed. Transfer learning technique was applied to reduce the amount of training time. The base model that was utilized was MobileNet V2. After the training, the accuracy of the model was evaluated and fine-tuned if it did not meet the expected accuracy. Eventually, a graph output was exported when the specified accuracy was reached.

The primary tools used in the project were Tensorflow and React Native. The main purpose of using Tensorflow library is to produce an animal images classification model that categorizes animals based on images snapped. Apart from that, React Native was used to facilitate the process of Android application development. Besides, Kaggle Kernel was used to perform transfer learning on the dataset.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter discusses the requirement specifications and use case of the mobile app. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications was referred to when documenting the software requirements specifications.

4.2 Requirement Specifications

4.2.1 Functional Requirements

1. The system shall be able to recognise animal through a camera.
2. The system shall display the information of the captured animals.
3. The system shall perform image recognition on the cloud.
4. The system shall allow the user to view the list of animals that were captured.
5. The system shall record the activity log of the user.
6. The system shall assign challenges for the user.
7. The system shall award the users a badge after they completed a challenge.
8. The system shall allow the user to check the rewarded badges.
9. The system shall display fun facts about random animals.
10. The system shall provide quizzes for the users.
11. The system shall allow the user to level up.

4.2.2 Non-functional Requirements

Compatibility

1. The system must be able to work on Android devices with Android 4.1 and above.

Availability

1. The system must be able to recognise images when an internet connection is not available.

Performance

1. The system must display the image recognition output within three seconds after an image is taken.

Usability

1. The system must use bright colours in its user interface.
2. The system must use simple English language.

4.2.3 Operating Environment

The system will operate on Android smartphones with at least Android 4.1 installed. A camera will be needed for capturing animals' photo.

4.2.4 Design and implementation constraint

Camera - To perform the core function of the application, a camera acts as the most important feature to allow further operations. Without a camera, the core function of the application cannot be initiated.

Supported platform - While the application is able to compile to both Android and iOS platforms with a single code base, some modules are unable to work on both platforms. Due to time limitations, the priority will be given to the Android platform.

4.2.5 Assumptions and dependencies

This project will develop under the dependencies of third-party packages and is assumed no breakage when React updates. Packages will include or may subject to change:

- React Native Camera
- Rn Mlkit Custom Model
- Open Trivia Database API
- Firebase ML Kit
- React Native Vector Icons

4.3 System Features Requirement

4.3.1 Capture and recognize photo

4.3.1.1 Description and priority

This feature allows users to capture and receive the animal's name. It should have high priority.

4.3.1.2 Stimulus/ Response Sequence

1. Children taps on the camera button.
2. The system launches the camera.
3. Children take a picture of an animal.
4. The system sends the picture to the model at Firebase server.
5. The system sends the output of the model to the users' device.
6. The system updates user experience and achievements.

4.3.1.3 Associated functional requirements

REQ-1: The system shall be able to recognise animals through a camera.

REQ-2: The system shall display the information of the captured animals.

REQ-3: The system shall perform image recognition on the cloud.

4.3.2 View activity log

4.3.2.1 Description and priority

This feature allows user to browse through the list of animals' photos that were taken. It should have medium priority.

4.3.2.2 Stimulus/ Response Sequence

1. Children taps on the logbook icon.
2. The system retrieves and displays the list of animals that the child had captured.

4.3.2.3 Associated functional requirements

REQ-4: The system shall allow the user to view the list of animals that were captured.

REQ-5: The system shall record the activity log of the user.

4.3.3 View animals' fun fact

4.3.3.1 Description and priority

This feature allows the user to read animals' fun fact. It should have medium priority.

4.3.3.2 Stimulus/ Response Sequence

1. Children navigate to the main screen.
2. The system retrieves and displays a random animal fun fact.

4.3.3.3 Associated functional requirements

REQ-6: The system shall display fun facts about random animals.

4.3.4 View challenges and badges

4.3.4.1 Description and priority

This feature allows user to view unlockable badges and their corresponding challenges. It should have medium priority.

4.3.4.2 Stimulus/ Response Sequence

1. Children navigate to the achievement screen.
2. The system displays unlockable badges.
3. Children taps on one of the badges.
4. The system retrieves and displays a challenge for children to complete.

4.3.4.3 Associated functional requirements

REQ-7: The system shall assign challenges for users.

REQ-8: The system shall award the users a badge after they completed a challenge.

REQ-9: The system shall allow the user to check the rewarded badges.

4.3.5 View and complete quiz

4.3.5.1 Description and priority

This feature allows the user to view and answer quizzes. It should have medium priority.

4.3.5.2 Stimulus/ Response Sequence

1. Children navigate to the main screen.
2. The system retrieves quizzes from Open Trivia Database.
3. The system displays the retrieved quizzes.
4. Children taps on an answer.
5. The system displays the correct answer.
6. The system updates users experience and achievements.

4.3.5.3 Associated functional requirements

REQ-10: The system shall provide quizzes for the users.

4.3.6 Earn experience points

4.3.6.1 Description and priority

This feature allows user to earn experience point through capturing animals' photo, answering quiz and completing challenges. Once the user earned sufficient experience points, the user's level will increase. It should have medium priority.

4.3.6.2 Stimulus /Response Sequence

1. Children perform capture animal photo use case, answer a quiz or complete a challenge.
2. The system updates the user's experience point.
3. The system increases user's level if the total experience points reach 10 or 20.

4.3.6.3 Associated functional requirements

REQ-11: The system shall allow users to level up.

4.4 Use Case Diagram

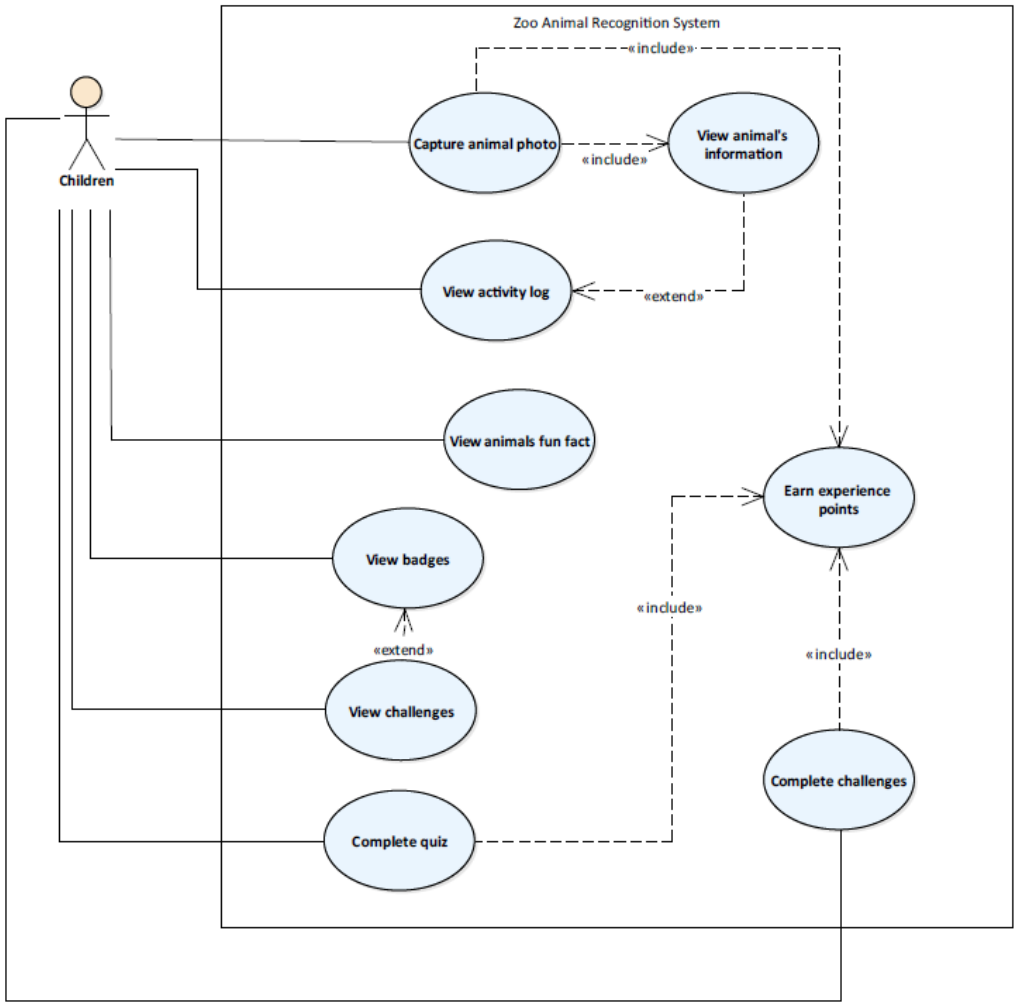


Figure 4.1 Use case diagram of the system

4.4.1 Use Case Description

Table 4.1 Use case description of capture animal photo

Use case name:	Capture animal photo
Primary Actor:	Children
Normal flow of events: 1. Children taps on the camera button. 2. The system launches the camera. 3. Children take a picture of an animal. 4. The system sends the picture to the model at Firebase server. 5. The system sends the output of the model to user. 6. The system updates user experience and achievements.	

7. The system performs view animal’s information use case.

Table 4.2 Use case description of view animal's information

Use case name:	View animal’s information
Primary Actor:	Children
Normal flow of events: 1. Children perform capture animal photo use case. 2. The system displays the information on the animal.	

Table 4.3 Use case description of view activity log

Use case name:	View activity log
Primary Actor:	Children
Normal flow of events: 1. Children taps on the logbook icon. 2. The system retrieves and displays the list of animals that the child had captured.	
Alternative flow: 2a. Children taps on one of the logs. 2a1. The system performs view animal’s information use case.	

Table 4.4 Use case description of view animal fun fact

Use case name:	View animal fun fact
Primary Actor:	Children
Normal flow of events: 1. Children navigate to the main screen. 2. The system retrieves and displays a random animal fun fact.	

Table 4.5 Use case description of view badges

Use case name:	View badges
Primary Actor:	Children
Normal flow of events: 1. Children taps on the trophy icon. 2. The system retrieves and displays the list of badges that can be earned.	

Alternative flow: 2a. Children taps on one of the badges 2a1. The system displays the progress of the challenge that is linked to the badge.

Table 4.6 Use case description of view challenges

Use case name:	View challenges
Primary Actor:	Children
Normal flow of events: 1. Children navigate to the achievement screen. 2. The system displays unlockable badges. 3. Children taps on one of the badges. 4. The system retrieves and displays a challenge for children to complete.	

Use case name:	Complete quiz
Primary Actor:	Children
Normal flow of events: 1. Children navigate to the main screen. 2. The system retrieves quizzes from Open Trivia Database. 3. The system displays the quizzes retrieved. 4. Children taps on an answer. 5. The system displays the correct answer. 6. The system updates users experience and achievements.	

Use case name:	Complete challenges
Primary Actor:	Children
Normal flow of events: 1. Children complete one of the challenges in to earn a badge. 2. The system updates the badge progress. 3. The system performs earn experience points use case.	

Use case name:	Earn experience points
Primary Actor:	Children
Normal flow of events:	

4. Children perform capture animal photo use case. 5. The system updates user experience point.
Alternative flow: 1a. Children answer a quiz. 1b. Children complete a challenge. 2a. The system updates user level if experience points reach 20 or 30

4.5 Summary

In this chapter, the requirements of the application were reviewed. The app has three major functions: perform image recognition using deep learning technology, assist children in learning by displaying animals' fact and gamify the learning process. This chapter also discussed the constraint of the app. A camera is needed for the app to run as it is necessary for photo taking.

The prototype consists of 5 screens, which are the main screen, logbook screen, achievement screen, camera screen and information screen.

Table 4.7 Summary of screen functions

Screen Name	Functions
Main Screen	- Display fun facts. - Display challenges. - Show navigation bar.
Log Book Screen	- Display the captured animal photo.
Camera Screen	- Capture images of animals. - Send images to Firebase ML Kit for image classification.
Badges Screen	- Display unlockable badges. - Display challenges associated with the badges. - Display progress of challenges
Information Screen	- Display animal's information.

CHAPTER 5

DESIGN

5.1 Introduction

This chapter focuses on the design of the system architecture, database, data flow and user interface. In section 5.2, the system architecture was discussed. In section 5.3 and 5.4, the database details and overall data flow of the system were shown. Section 5.4 consists of the activity diagram of the overall system. Lastly, the user interface details were explored in the last subchapter.

5.2 System Architecture Design

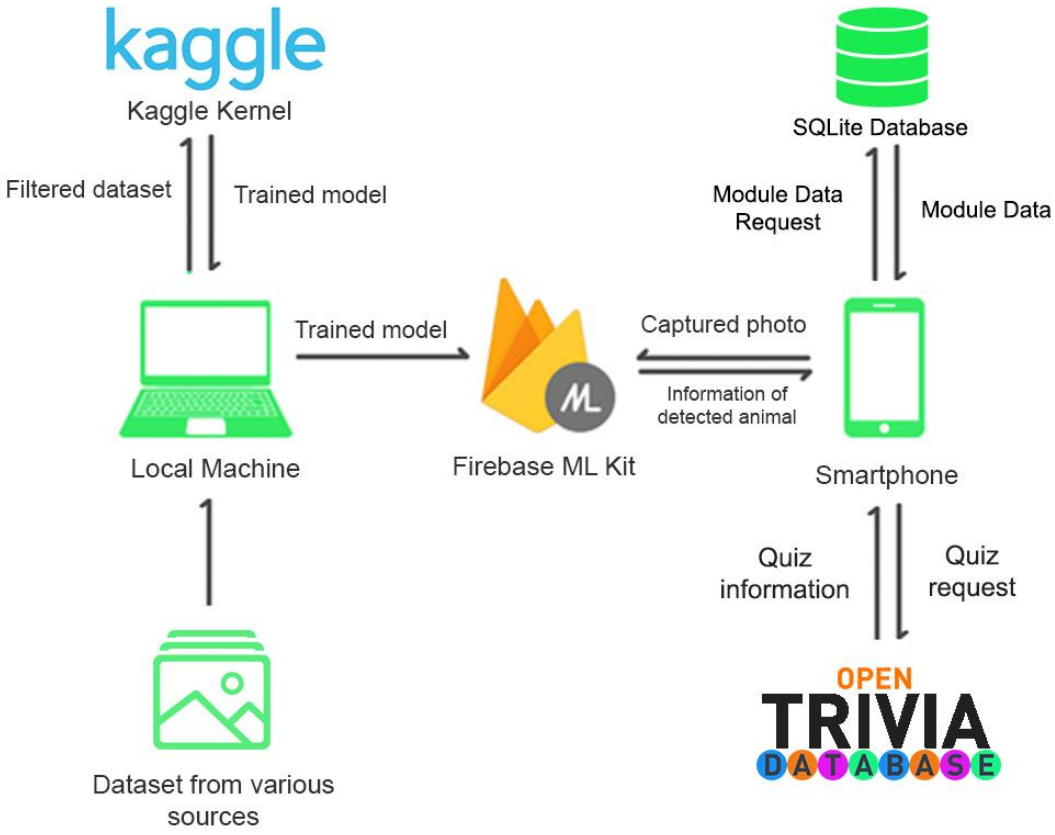


Figure 5.1 Animalia system architecture design

As shown in the figure above, animal datasets were collected from WSID-100 and KTH School of Electrical Engineering and Computer Science dataset. As mentioned in subchapter 3.3, the images were filtered and cropped to improve training effectiveness. Kaggle Kernel was used to perform transfer learning and image model training. To enable the trained model to be used on mobile devices, it was converted

from ProtoBuf format to Tensorflow Lite format on a local machine. The trained model was then uploaded and hosted on Firebase ML Kit. When an Internet connection is available, the image captured by users will be sent to Firebase ML Kit. Subsequently, the uploaded image will be analysed by the trained model hosted on Firebase and response with predicted animal accuracies will be sent back to the users' device. Alternatively, if an Internet connection is unavailable, the captured photo will be analysed by the model that was integrated into the mobile app. Animals information, badge and logs details are stored locally using SQLite database to ensure that the app can be run even without Internet access. Quizzes are retrieved from Open Trivia Database using its API.

5.3 Database Design

5.3.1 Entity Relationship Diagram (ERD)

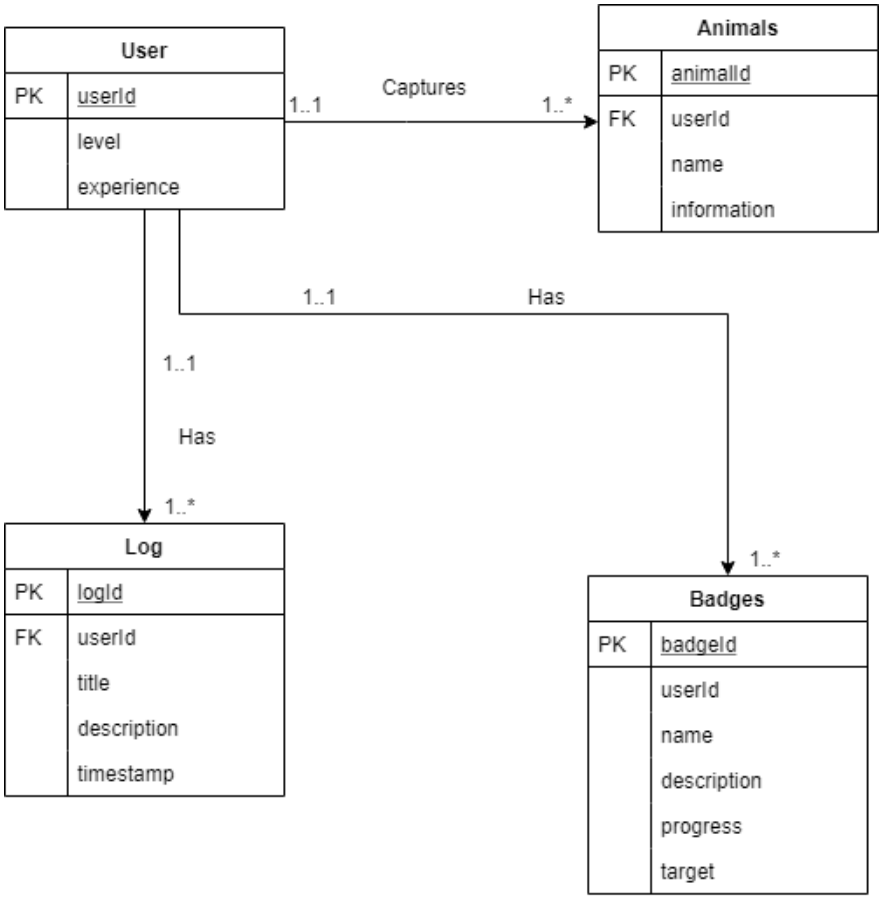


Figure 5.2 Physical entity relationship diagram

The ERD of the system consists of four tables which are User, Animals, Log and Badges.

Table 5.1 Description of each table

Tables	Description
User	Store the information of the users
Animals	Store the information of all the animals
Log	Store the log record of the animals caught.
Badges	Store all the achievements and badges.

5.3.2 Data Dictionary

Table name: User

Table 5.2 Data dictionary for user entity

Column	**Description**	**Data Type**	**Primary Key**	**Foreign Key**	**FK reference table**
userId	Unique identifier for users	Integer	Yes	No	-
level	User's current level	Integer	No	No	-
experience	User's experience points	Integer	No	No	-

Table name: Log

Table 5.3 Data dictionary for log entity

Column	**Description**	**Data Type**	**Primary Key**	**Foreign Key**	**FK reference table**
logId	Unique identifier for log	Integer	Yes	No	-
userId	Unique identifszier for users	Integer	No	Yes	user
title	Log's title	Text	No	No	-
Description	Log's description	Text	No	No	-
timestamp	The time when the log was recorded	Text	No	No	-

Table name: Badges

Table 5.4 Data dictionary for badges entity

Column	Description	Data Type	Primary Key	Foreign Key	FK reference table
badgeId	Unique identifier for badge	Integer	Yes	No	-
userId	Unique identifier for user	Integer	No	Yes	user
name	Badge's name	Text	No	No	-
description	Badge's description	Text	No	No	-
progress	User's progress of achieving a badge	Integer	No	No	-
target	The target that needs to be reached to acquire the badge	Integer	No	No	-

Table name: Animals

Table 5.5 Data dictionary for animals entity

Column	Description	Data Type	Primary Key	Foreign Key	FK reference table
animalId	Unique identifier for animals	Integer	Yes	No	-
userId	Unique identifier for users	Integer	No	Yes	user
name	Animal's name	Text	No	No	-
information	Animal's information	Text	No	No	-

5.4 Data Flow Diagram

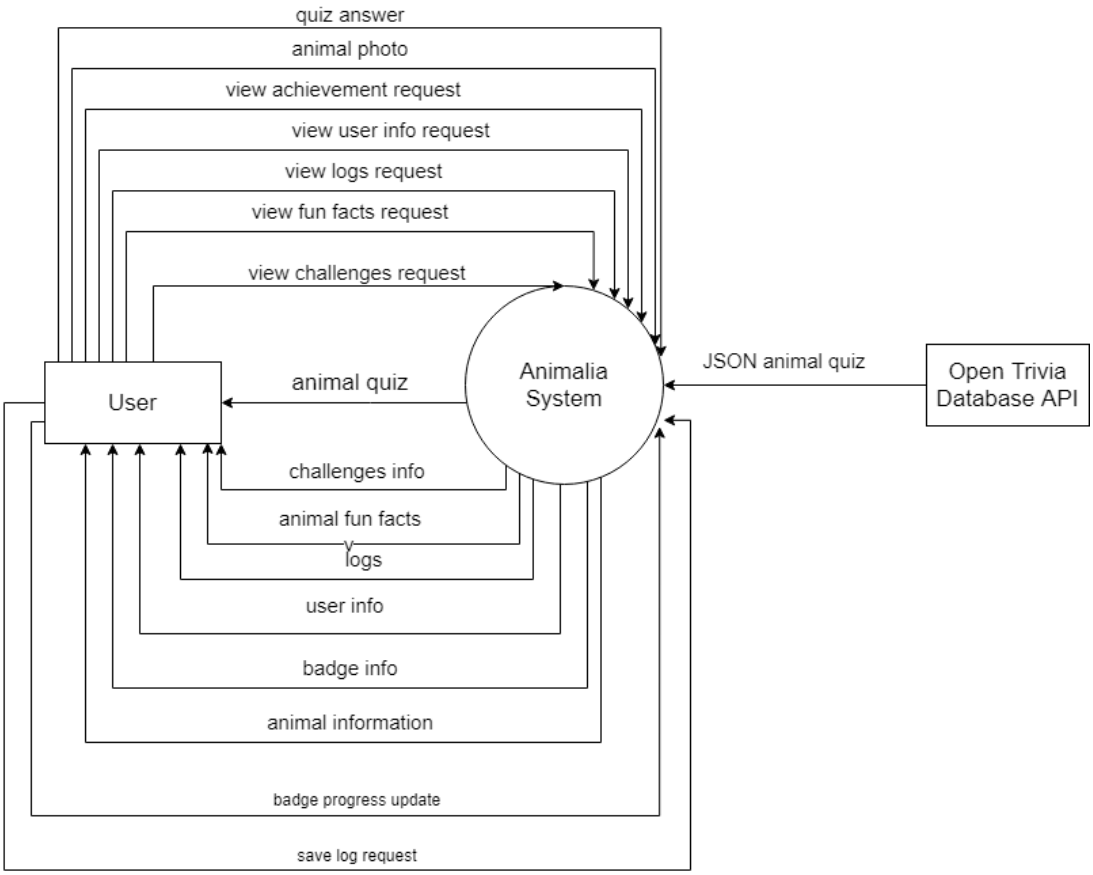


Figure 5.3 Context data flow diagram

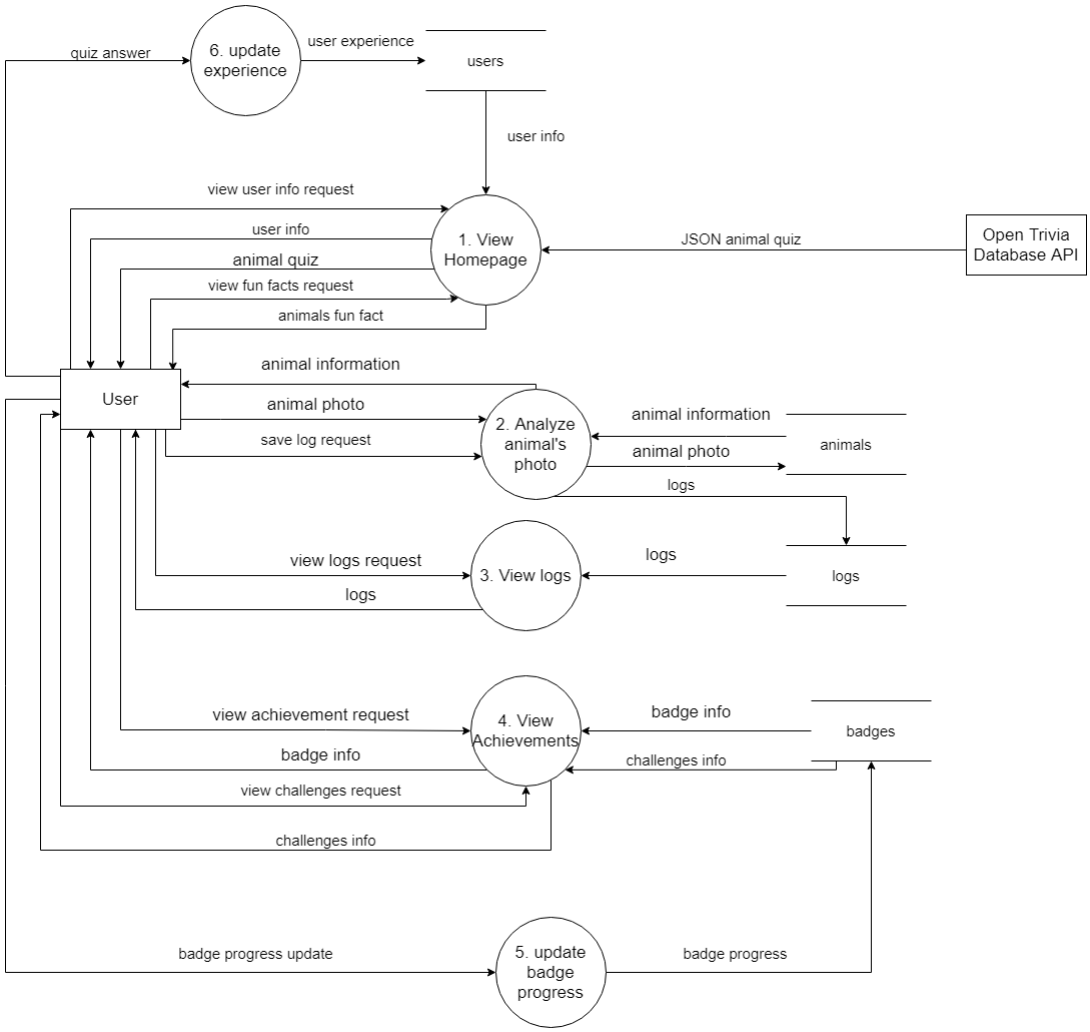


Figure 5.4 Level 0 data flow diagram

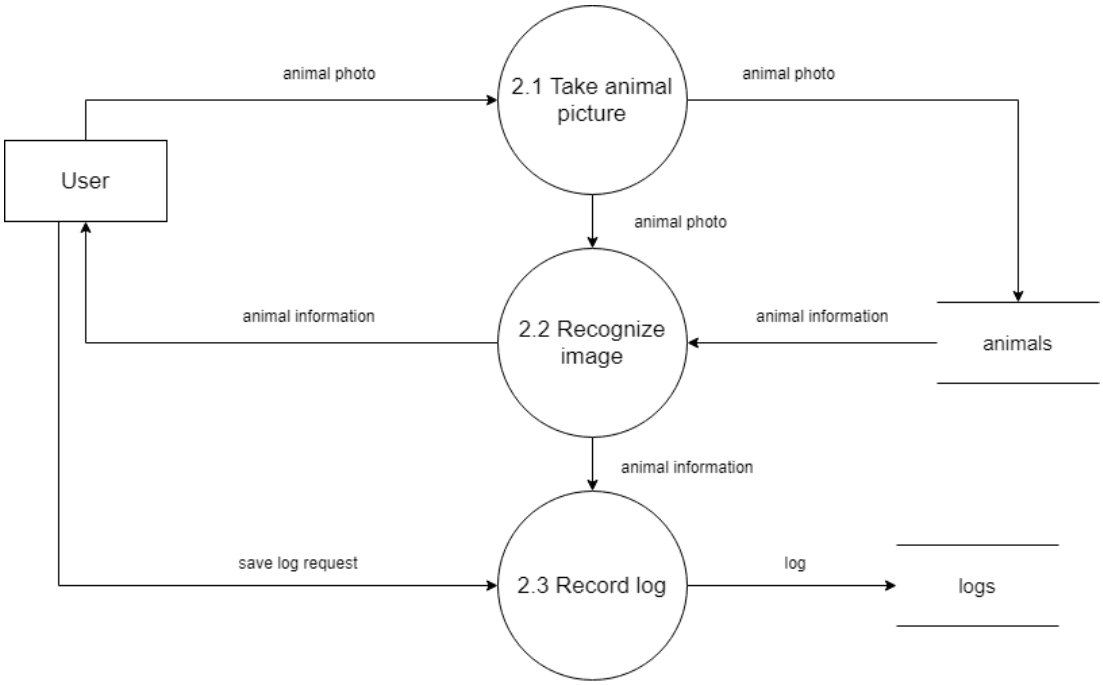


Figure 5.5 Level 1 data flow diagram for analyse animal's photo

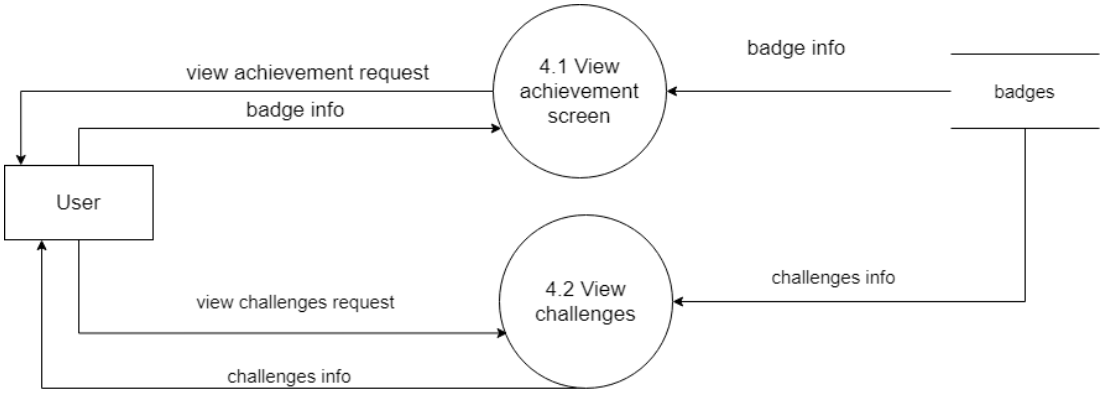


Figure 5.6 Level 1 data flow diagram for view achievements

5.5 Activity Diagram

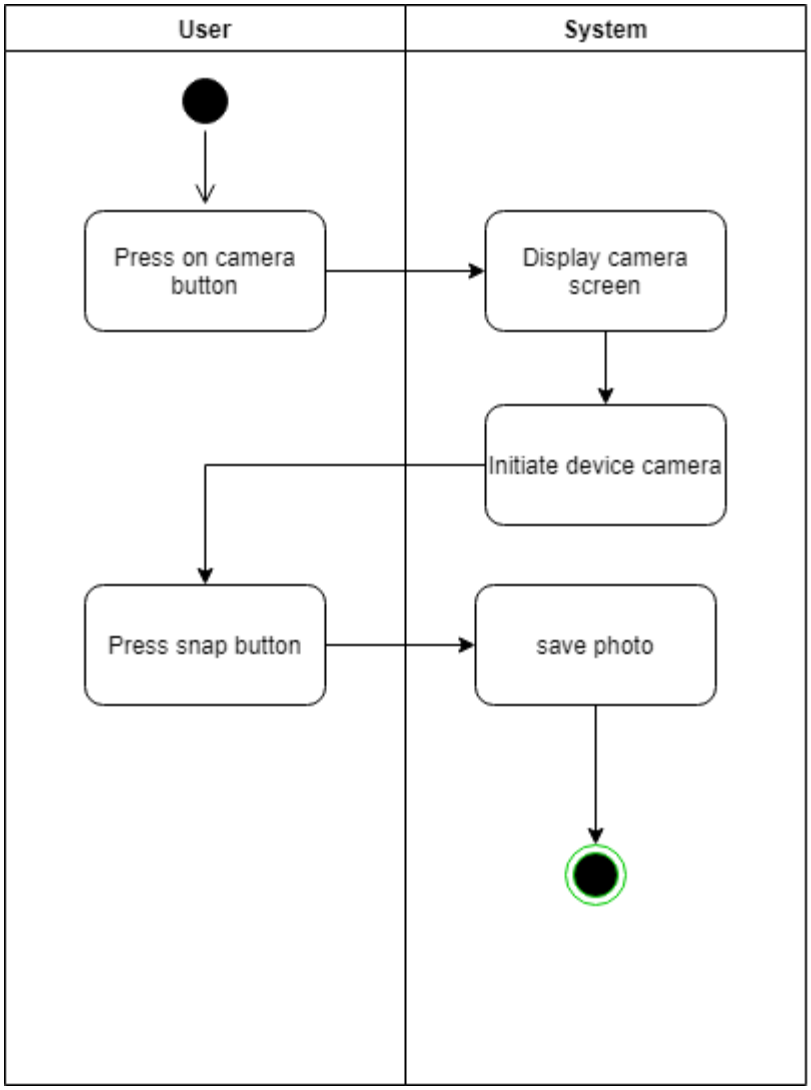


Figure 5.7 Activity diagram for capture animal photo

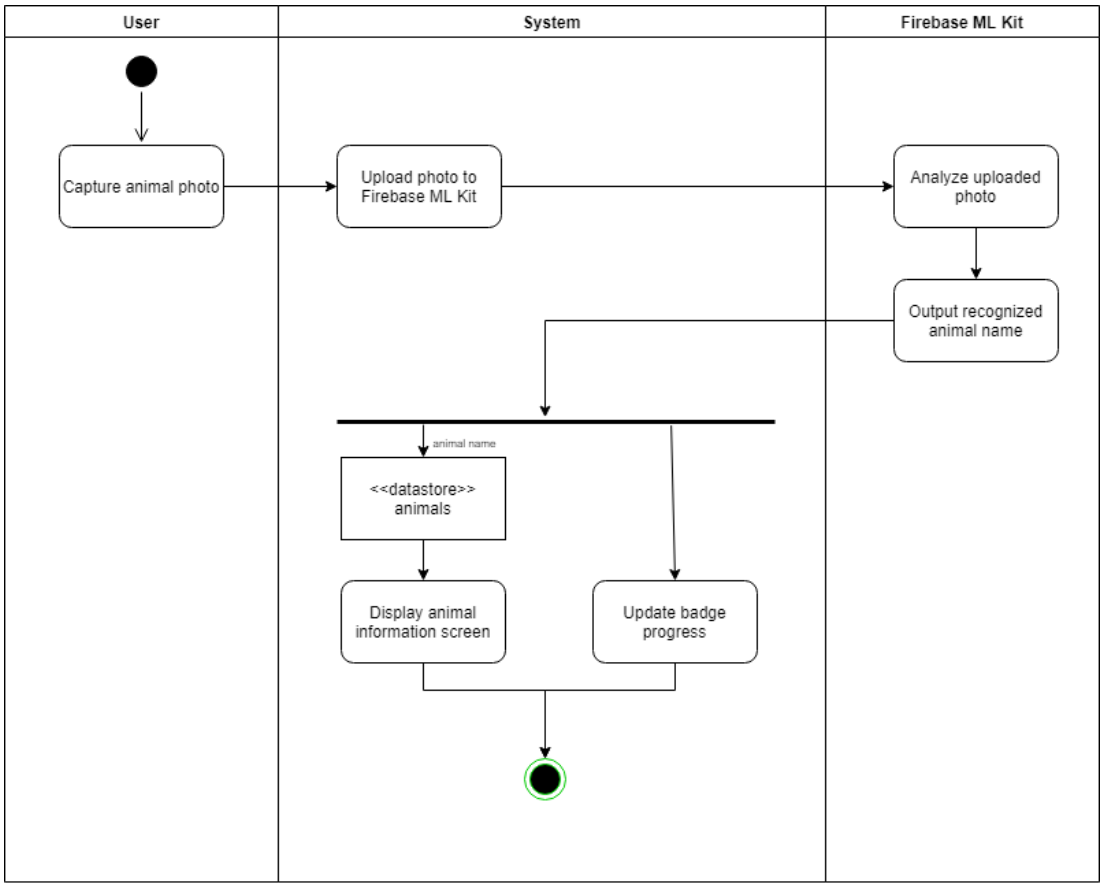


Figure 5.8 Activity diagram for analyse animal photo

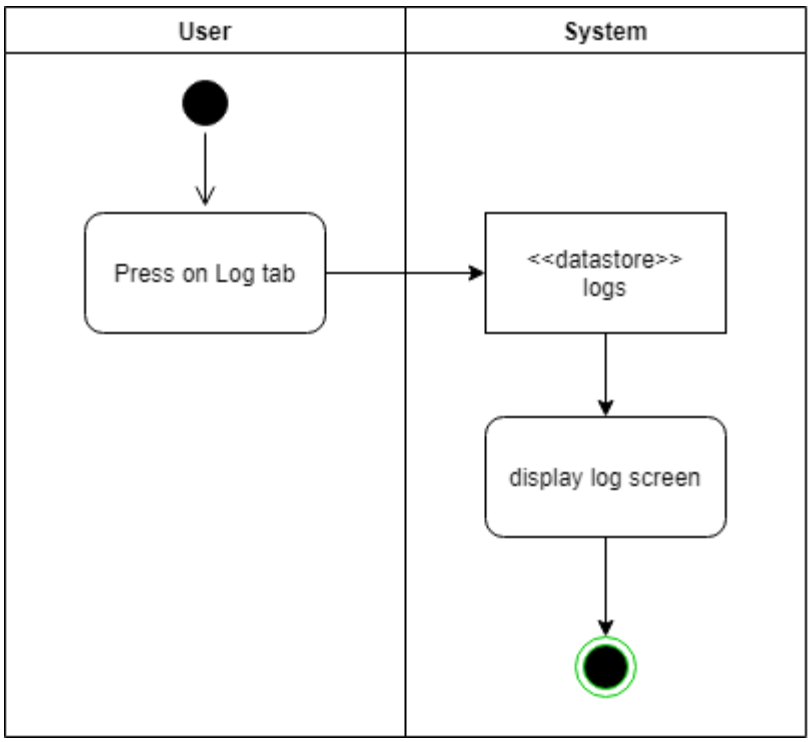


Figure 5.9 Activity diagram for display logs

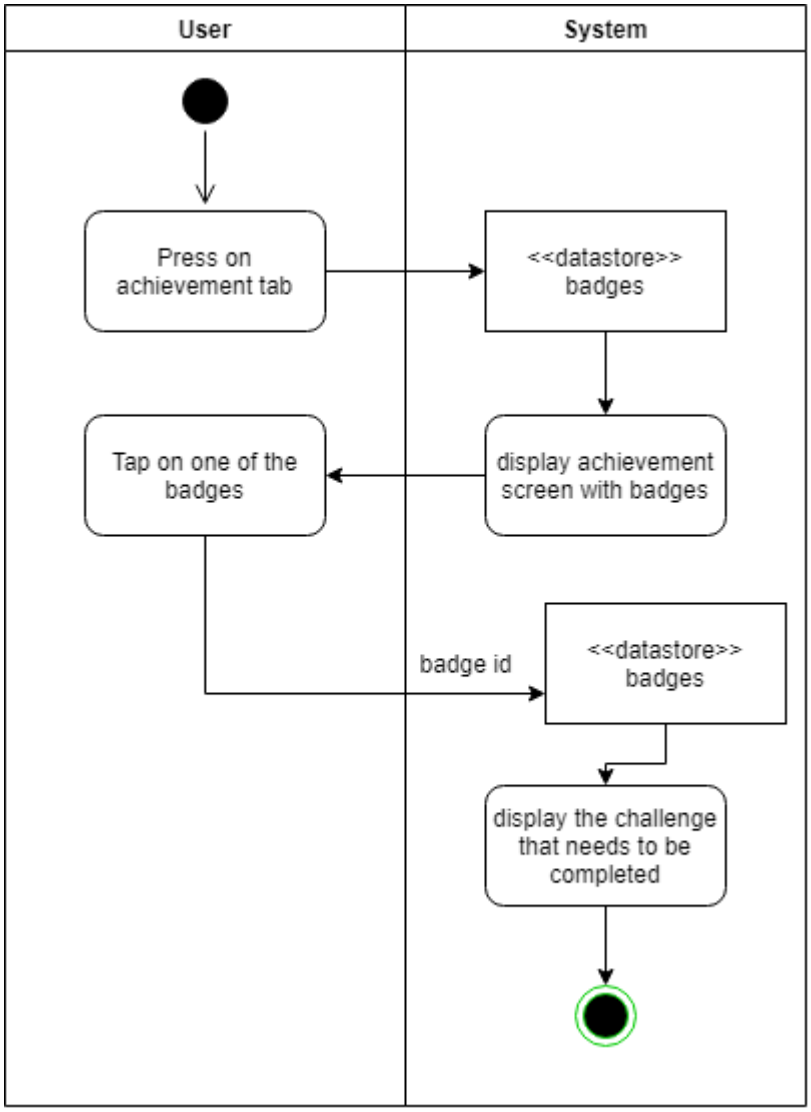


Figure 5.10 Activity for view badges and challenges

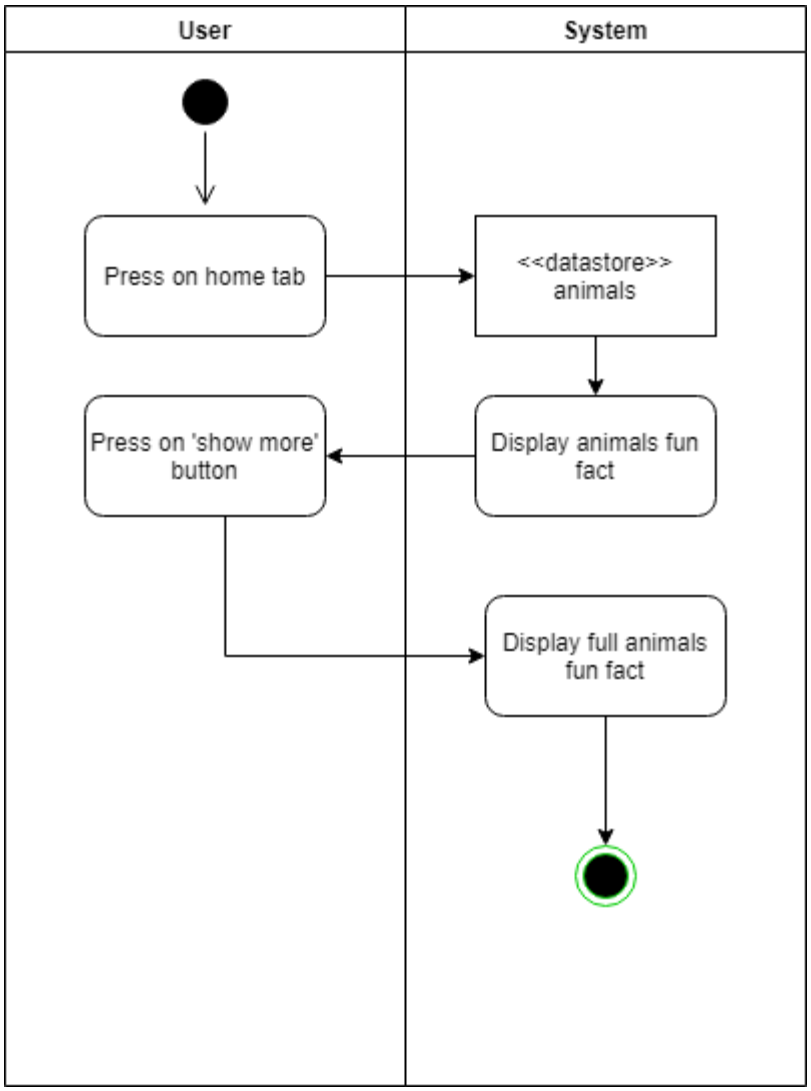


Figure 5.11 Activity diagram for view animals' fun fact

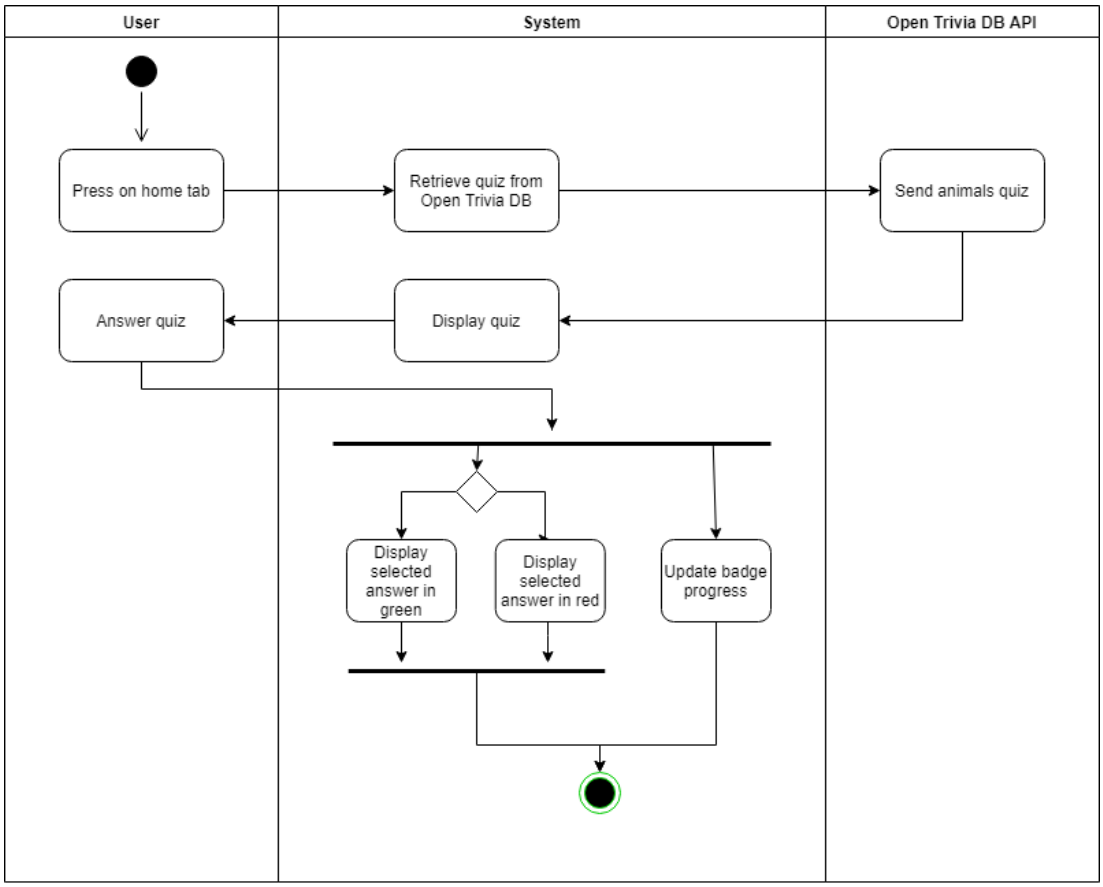


Figure 5.12 Activity diagram for answer quiz

5.6 User Interface Design

The user interface of the app is converted from the low-fidelity prototype which can be viewed in Appendix D.

5.6.1 Screen Flow

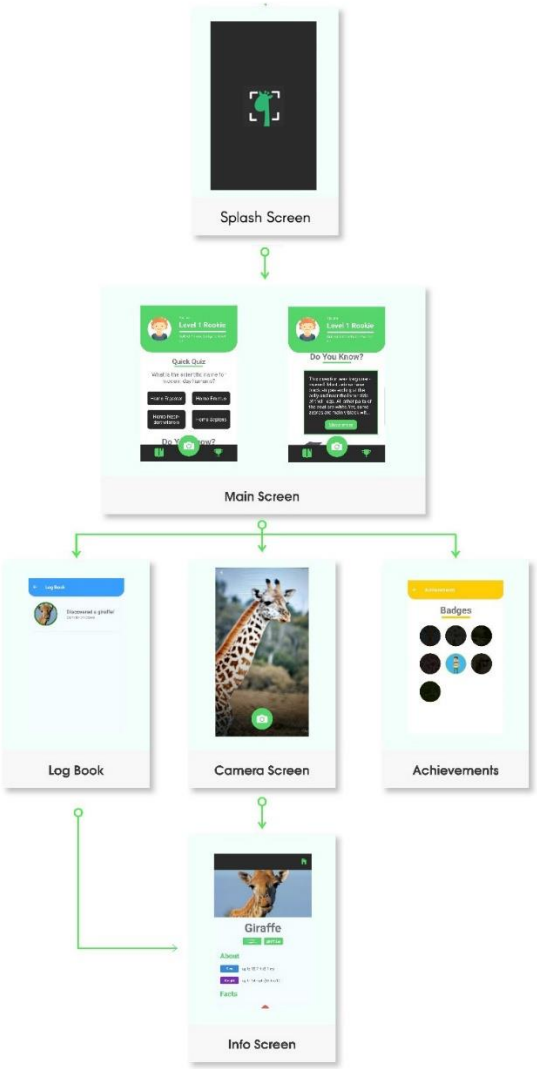


Figure 5.13 Screen flow of Animalia

The first screen that the user will encounter is the splash screen which is linked to the home screen. From the main screen, user can navigate to logbook screen, camera screen or achievement screen. After capturing the photo of an animal, the system will navigate the user to the information screen. Alternatively, the user can access the information screen by tapping one of the logs in log screen.

5.6.2 Splash Screen

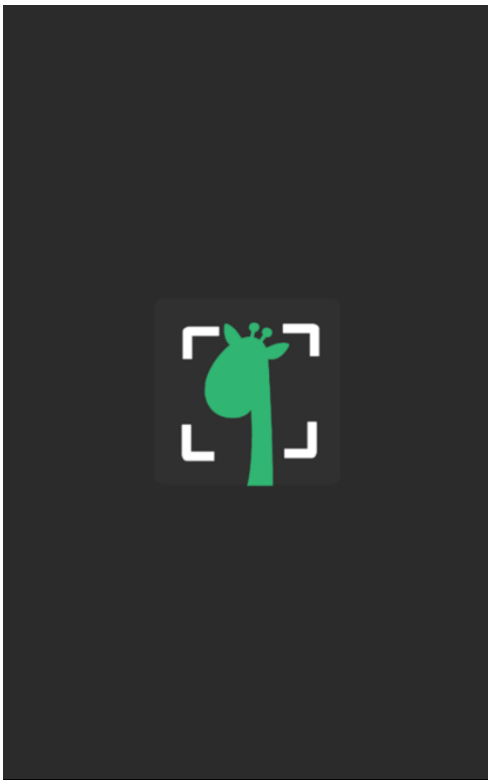


Figure 5.14 Screenshot of the splash screen

Before the homepage is shown, the splash screen with the app icon will be shown for a second. The icon shows camera marks and a silhouette of a giraffe.

5.6.3 Main Screen

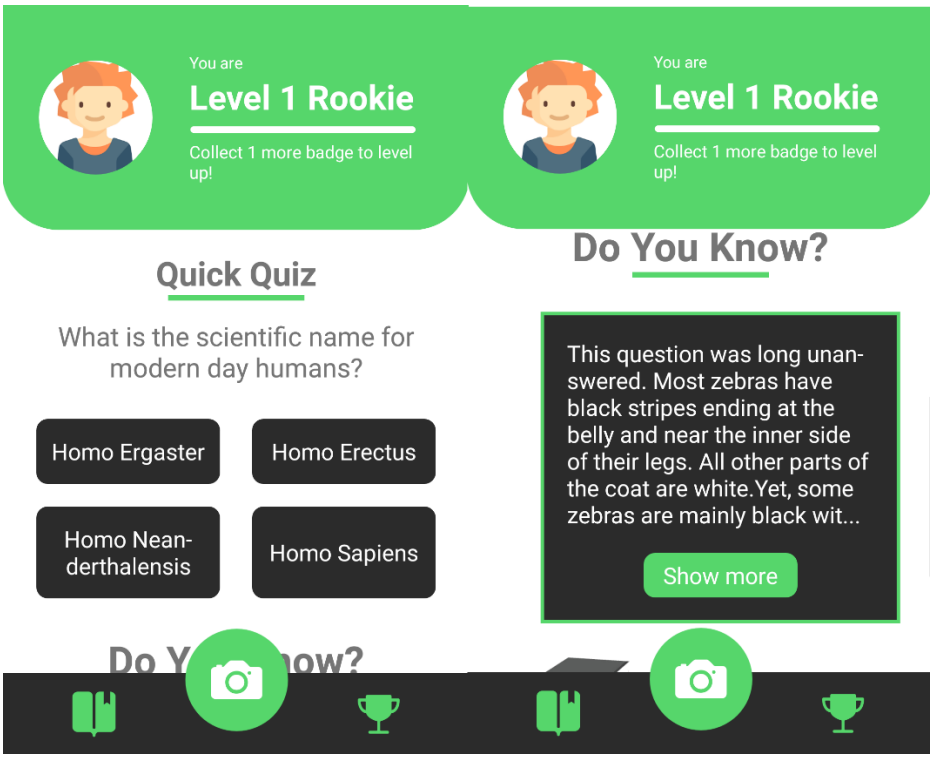


Figure 5.15 Screen capture of the main screen

The main screen displays the user current level, challenge and fun fact of random animals. Once an answer is chosen, the system will show the correct answer by showing it in green. A lengthy fun fact can be shown fully by tapping of the show more button. The bottom of the main screen shows the navigation bar which allows user to navigate to a linked screen. Logbook will be displayed by tapping the leftmost icon. The middle camera icon is linked to the camera screen. The trophy on the right side is connected to the badges screen.

5.6.4 Log Book Screen

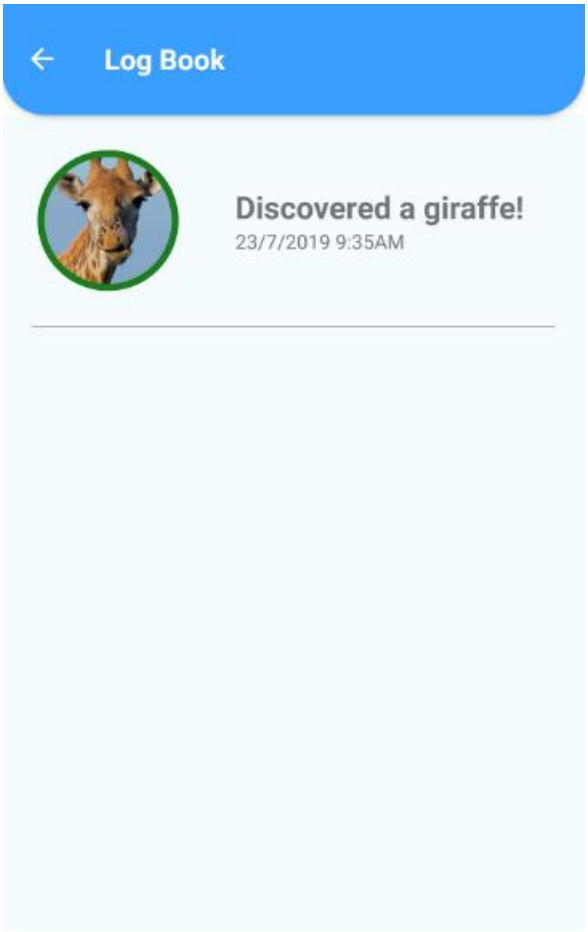


Figure 5.16 Screen capture of logbook screen

The logbook screen displays the activity log of users. Picture of animals that were taken by users will be shown in this screen, along with the date, time and name. By tapping on the log, users can view the animal's information in the information screen.

5.6.5 Camera Screen

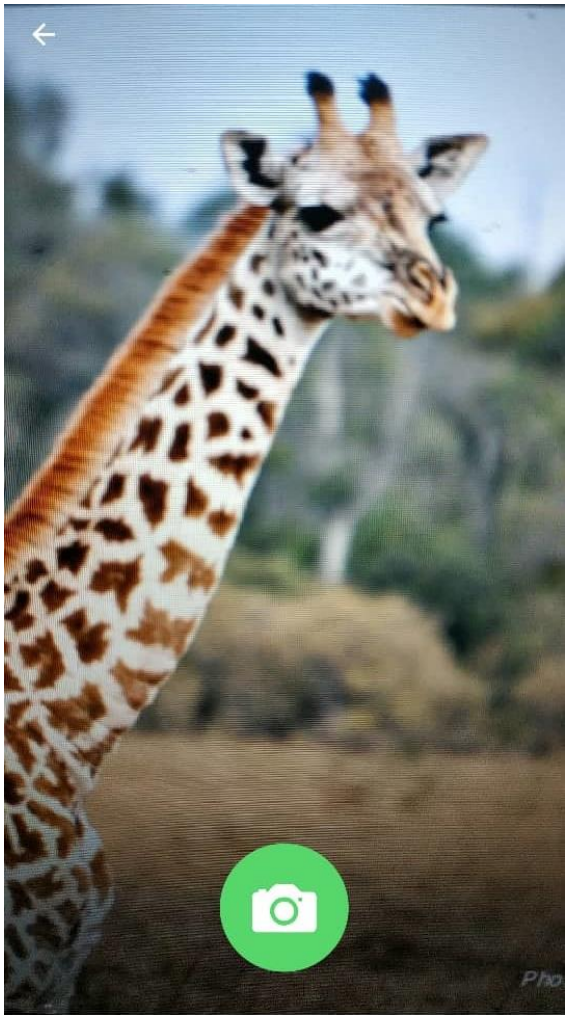


Figure 5.17 Screen capture of animal screen

The camera screen allows the user to capture the photo of animals by pressing the button with a camera icon.

5.6.6 Badges Screen

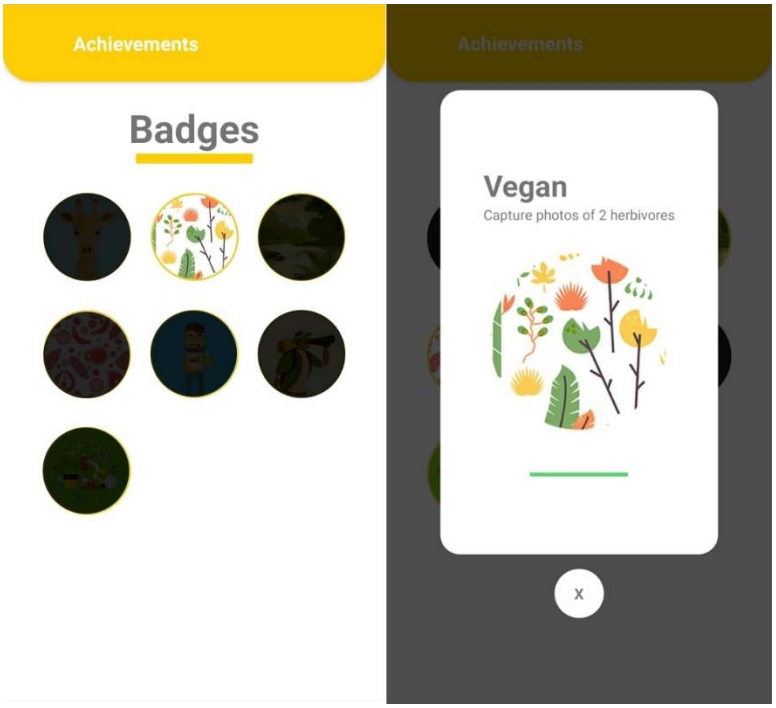


Figure 5.18 Screen capture of achievements screen

The badges screen shows unlockable badges and rewarded badges. User can tap on each of the badges to identify the challenges that are associated with the badges and the progress of the challenges.

5.6.7 Information Screen

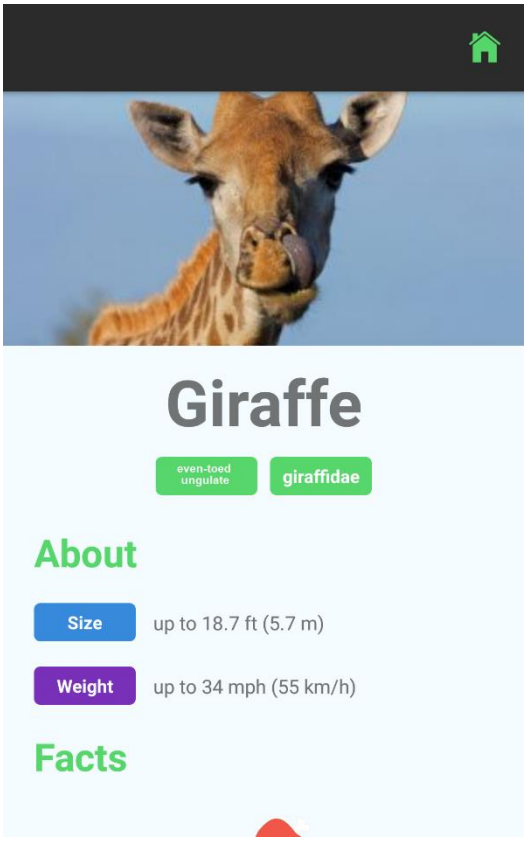


Figure 5.19 Screen capture of the information screen

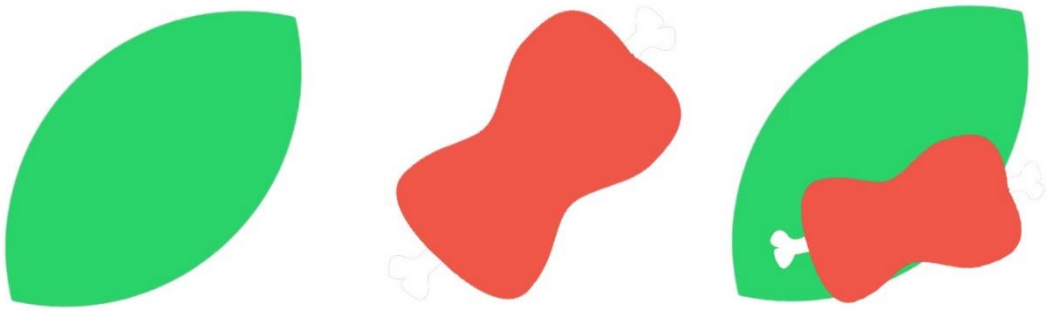


Figure 5.20 Icons that symbolize different diet

The information screen displays the image and information about the animals, including its species, size and weight. Besides, different icons will be shown in the screen. The icons shown depends on the animal's diet. A piece of leaf will be shown if the animal is a herbivore whereas meat will be shown for carnivores. Omnivore is symbolized with both meat and leaf.

CHAPTER 6

RESULTS AND DISUSSION

6.1 Introduction

This chapter shows how the three models were trained, analysed and compared among each other. Besides, this chapter also includes explanations of how the app modules were implemented.

6.2 Training results of pretrained models

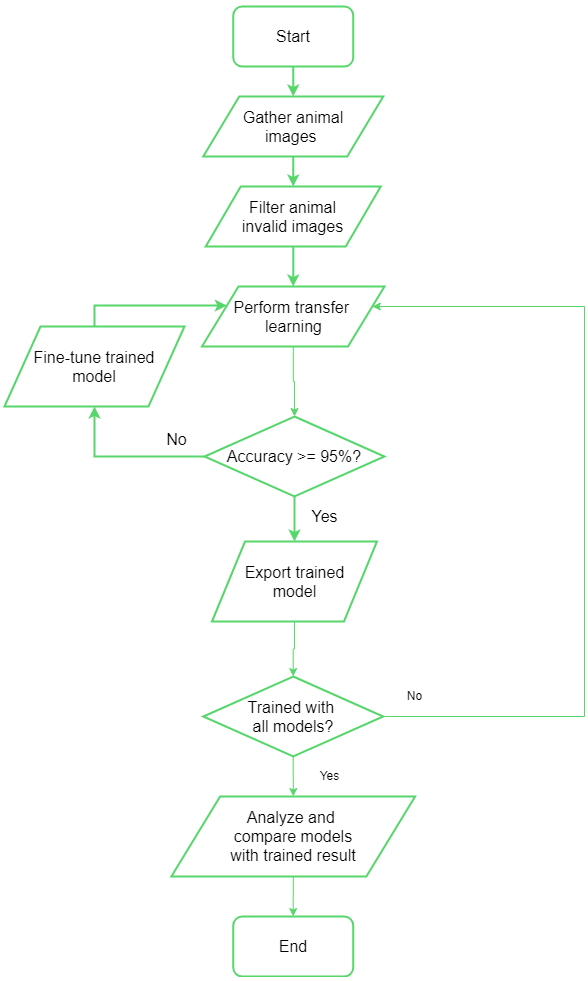


Figure 6.1 Complete models training workflow

As shown in the figure above, the process mentioned in figure 3.4 was repeated with three different models, which are MobileNet, EffNet and Nasnet. After transfer learning was carried out for the three models, the results of training which include their accuracy, Area Under the Graph and Receiver Operating Characteristic (AUC-ROC),

output graph size, Floating Point Operations (FLOPs) and the number of parameters are analyzed and compared. Besides, the accuracy of each animal's classes was calculated.

6.2.1 Comparison between trained model attributes

To select the most suitable model for the app, the models were compared from multiple aspects, including their accuracy, speed, memory and power consumption. These aspects can be measured through metrics mentioned above. The complete results of training can be found in Appendix B.

In the training process, each of the models was fed with the same set of input image dataset and the training steps was limited to 500. By the end of the training, the accuracy of final training was recorded. The result had shown that EffNet has achieved the highest accuracy (98.4%) among the three models, followed by NASNet and MobileNet.

Area Under the Graph and Receiver Operating Characteristic (AUC-ROC) is one of the metrics that was computed during the training process to verify the models' accuracy. AUC-ROC curve determines a model's ability to discriminate between classes. A higher AUC implies better differentiation ability of a model. To plot a ROC curve, the true positive rate should be plotted against the false positive rate of a model. True-positive rate determines the percentage of actual positives that are identified correctly as such. For instance, the proportion of sick patients that are correctly determined as having a condition. False-positive rate, on the other hand, is the percentage of misidentified actual positives. True positive rate or sensitivity can be found using the following equation:

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

i. TPR indicates true positive rate.

ii. TP indicates number of True Positive.

iii. FN indicates number of False Positive.

False positive rate was calculated using the equation:

$$FPR = \frac{FP}{TN + FP}$$

i. FPR indicates False Positive Rate.

ii. FP indicates number of False Positive.

iii. TN indicates number of True Negative.

After the training, it was found out that three of the models achieved AUC value of over 0.99. Among three of the models, EffNet has achieved the highest AUC, which was around 0.9975, followed by Nasnet and Mobilenet. This indicates that EffNet has the highest accuracy among the models.

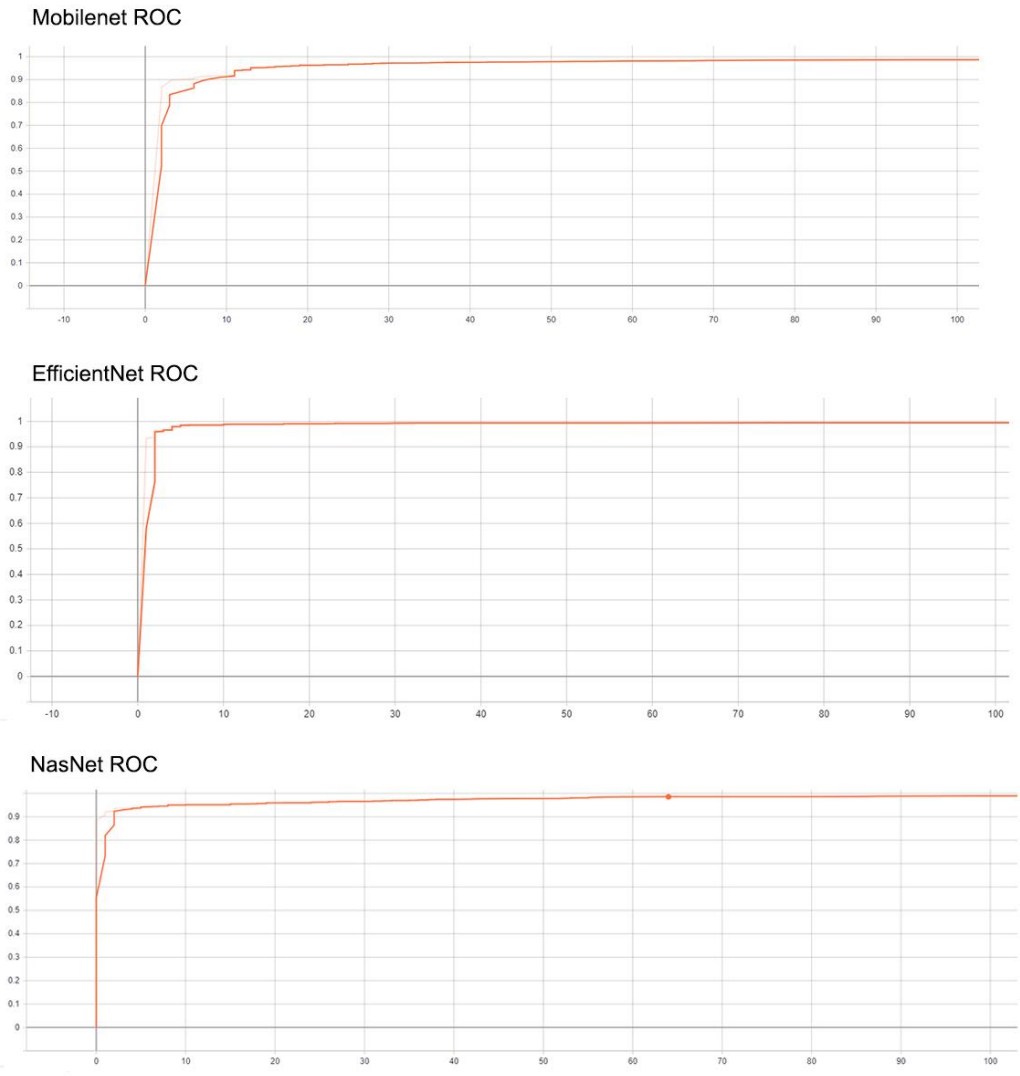


Figure 6.2 ROC graphs of MobileNet, EfficientNet and NASNet

When training is completed, the trained model is said to be ‘frozen’ as a ProtoBuf file. Freezing a model is equivalence to saving a model by producing a singular file that contains the weights and parameters. The size of the output model is crucial in this project as mobile devices have limited storage. Besides, Firebase limits the uploaded custom model to 40MB. Hence, the size of the selected pretrained model

should be below 40MB. By comparing the models, it was shown that Mobilenet has the lowest output size among all.

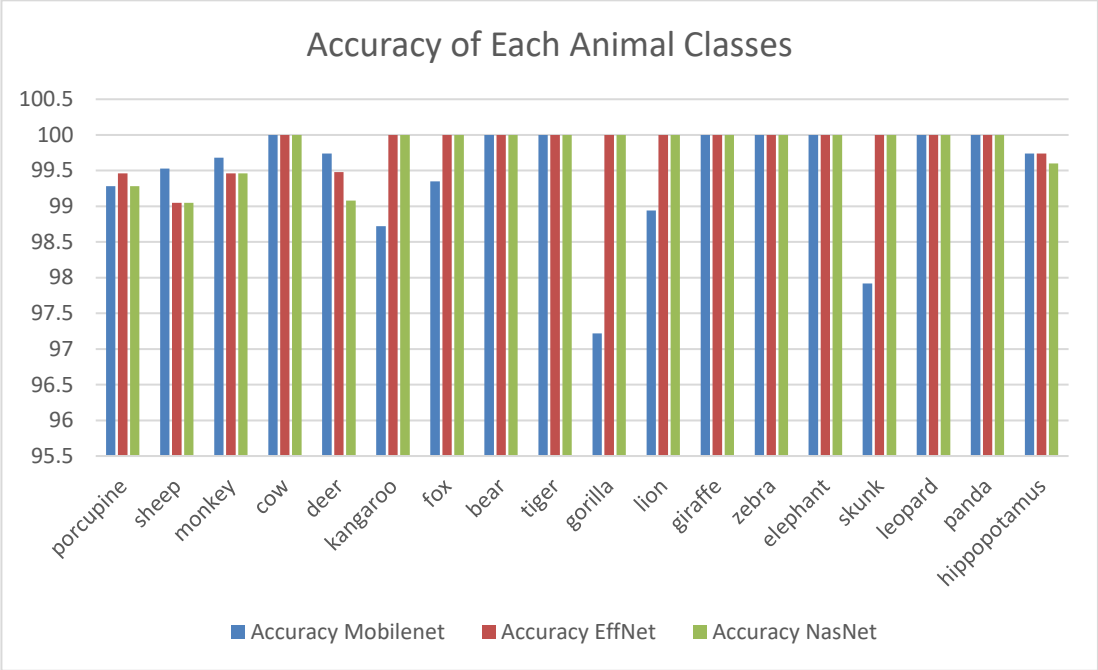
To compute the sped of models, the number of Floating-Point Operations (FLOPs) and parameters were analysed. FLOPs are any mathematical operations that involve floating point in a neural network. Parameters in this context are the weights between neural network connections. Both of the elements are commonly used in deep learning to identify the complexity of a model. However, Tang et. al (2018) proposed that the number of FLOPs is stronger at predicting the computational power usage of a model. The experiment between different models suggested that EffNet and Nasnet have a similar amount of FLOPs during the training, whereas Mobilenet has the lowest amount of FLOPs.

In consideration of the performance of the app, MobileNet was chosen as the image recognition model. The difference in accuracy between the three models was insignificant and, hence, carries the least weightage. EffNet, even though with the highest accuracy, is inappropriate for the app because of its relatively high number of FLOPs and output size. Furthermore, the size of EffNet model had exceeded the maximum size allowed by Firebase. The performance of Nasnet model is comparable with Mobilenet's. The reason for selecting Mobilenet is primarily because of its low energy consumption. A visit to a zoo usually takes up to half a day or even more. Moreover, despite the battery energy consumed by the image recognition process, the energy used by phone camera and an internet connection was also taken into account. Saving the energy consumption prevents drained battery during an emergency, for instance, when a kid is lost at the zoo and need to call others with his or her phone. Hence, considering all the factors, MobileNet is more suitable for the mobile app.

Table 6.1 Training results of pretrained models.

	Pretrained Models		
Attributes	Mobilenet	EffNet	Nasnet
Final training accuracy	95.6	98.4	96
AUC-ROC	0.99005705	0.99753076	0.9965985
Output Frozen Model Size (MB)	12.859	42.545	17.366
Million FLOPs	38.20	42.52	42.08
Million Parameters	0.018450	0.027666	0.019026

6.2.2 Comparison of accuracy between animal classes



Graph 6.1 Accuracy of animal classes with different pretrained model

Table 6.2 Accuracy of animal classes with different pretrained model

Animal	Accuracy		
	Mobilenet	EffNet	NASNet
bear	100	100	100
cow	100	100	100
elephant	100	100	100
giraffe	100	100	100
leopard	100	100	100
panda	100	100	100
tiger	100	100	100
zebra	100	100	100
fox	99.35	100	100
lion	98.94	100	100
kangaroo	98.72	100	100
skunk	97.92	100	100
gorilla	97.22	100	100
hippopotamus	99.74	99.74	99.60
deer	99.74	99.48	99.08
monkey	99.68	99.46	99.46
porcupine	99.28	99.46	99.28
sheep	99.53	99.05	99.05

Besides the models, the accuracy of each animal classes were also identified. The results showed that hippopotamus, deer, monkey, porcupine and sheep have lower accuracy compared to other animals when trained by all three of the models.

6.3 Modules implementation

As mentioned in subchapter 3.2, the app consists of three main major modules, that are animal recognition module, animal information module and gamification module. In this subchapter, the implementation of animal information and gamification module will be discussed.

A reliable and kids-friendly data source is undeniably important for the animal information module. The data source used for the module was scrapped from animalfunfacts.net, with the consent of the owner. Animalfunfacts.net is an animal encyclopedia made specifically for children that received multiple awards and quality seals since its deployment. The open-source JavaScript library, Cheerio, was used to scrape the data from the website. The scrapped data was output in JSON format and stored inside SQLite database for retrieval. Examples of data scrapped include size, diet and habitat.

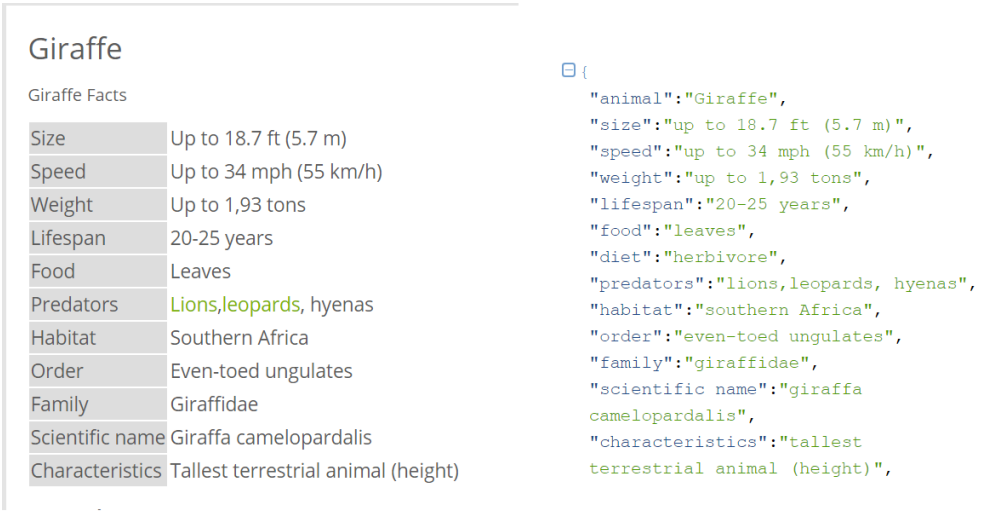


Figure 6.3 Data scrapped from animalfunfacts.net

Quizzes is retrieved from Open Trivia Database, an open-sourced trivia API. It allows users to configure the category and difficulties of questions and generate an

endpoint. To fetch the data from the endpoint, Fetch library was utilized. An example of the fetched JSON data is shown below.


```
"results": [
	{
		"category": "Animals",
		"type": "multiple",
		"difficulty": "easy",
		"question": "What is the fastest land animal?",
		"correct_answer": "Cheetah",
		"incorrect_answers": ["Lion","Thomson&rsquo;s
		Gazelle","Pronghorn Antelope"]
	},
```


The last module implemented was gamification module. The module can be broken down further into two parts: the badge system and level up system. Badges will be awarded to the user once the given mission is completed. The details of the 7 badges are shown in the table below.

Table 6.3 Description of badges

Badge name and image	Description
Skyscraper 	Capture photo of 3 giraffes
Into the wild 	Reach level 2
Adventurer 	Reach level 3

Vegan 	Capture photos of 2 herbivores
Carnist 	Capture photos of 2 carnivores
Foodie 	Capture photos of 2 omnivores
Biologist 	Answer 5 pop quizzes correctly

To gain level, kids have to collect experience points (EXP) by interacting with the application. The table below shows the ways to gather EXP and the amount of EXP given for each activity.

Table 6.4 Activities and EXP allocated

Activity	EXP allocated
Capture an animal's photo	1
Make progress in earning a badge	2

The amount of EXP needed for kids to gain a level is shown below.

Table 6.5 Amount of EXP needed to level up

Level	Name	EXP needed
1	Rookie	0
2	Trainee	10
3	Adventurer	20

CHAPTER 7

TESTING

7.1 Introduction

This chapter describes different types of testing which have been carried out to ensure the consistency of the system performance. The chapter includes the test plan and the implemented test cases. Test implemented includes unit test, integration test, system test and usability test. Besides, static code analysis is implemented using JavaScript source code analysis library.

7.2 Static Code Analysis

Plato was used for static code analysis in this project. Plato is a JavaScript source code visualization, static analysis, and complexity tool. According to the result, the source code in the project has achieved an average maintainability score of 75.82.

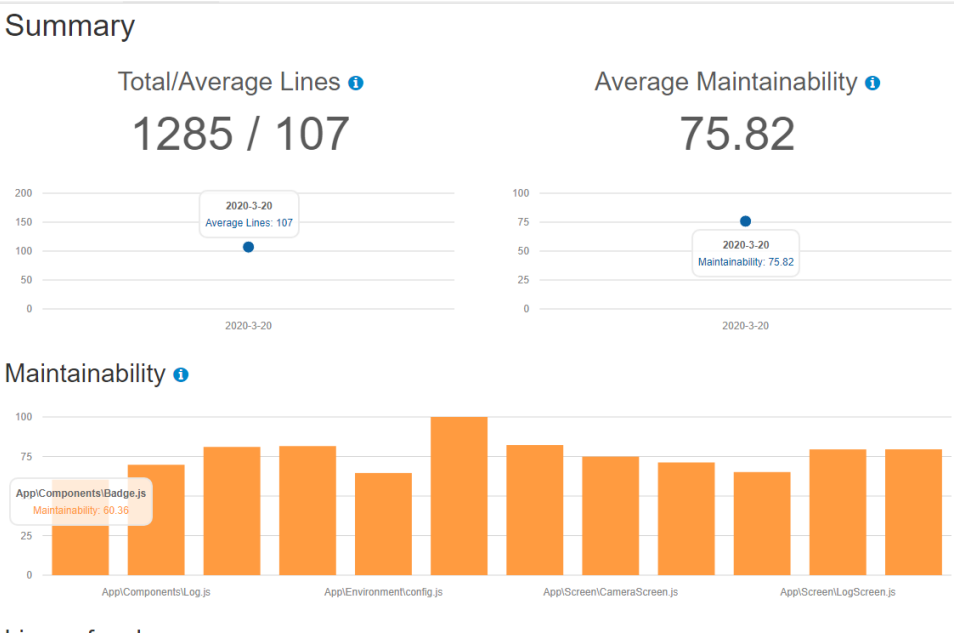


Figure 7.1 Result of static code analysis

7.3 Test Planning

7.3.1 Test Items

The test items covered in this test plan includes:

- AchievementScreen.js
- CameraScreen.js
- HomeScreen.js

- InfoScreen.js
- LogScreen.js

7.3.2 Features to be tested

The features that will be tested includes:

- Show captured animals' information
- Earn badge
- Answer quiz
- Earn experience points
- View logs
- Capture animals and view logs
- Capture animals and earn experience points

7.3.3 Features not to be tested

The features that will not be tested includes:

- Test run the app on iOS device
- External dependencies such as Open Trivia DB API and Firebase ML Kit

7.3.4 Approach

The test will be carried out on Android device with Android version 9.0 and device with MIUI 11.0.3. For virtual Android devices, Android Virtual Device should be installed in the local machine along with emulators with Android version 9.0.

Use case testing technique is recommended when testing is carried out. The technique ensures that all business case and requirements can be executed flawlessly.

7.3.5 Item pass/fail criteria

An item is considered as pass if the expected output is achieved when the testing is completed. An item is considered as fail if there was a dissimilarity between the expected and actual result or the requirements were not met.

7.3.6 Testing tasks

The major tasks for this test session are test plan creation, unit test, integration test, system test and usability test.

Before the unit test and integration test are carried out, at least an Android emulator with Android version 9.0 should be installed in the local machine. Installation of React Native CLI should not be neglected before the tests.

Before usability tests are carried out, testers should provide the participants with a consent form and post-test survey form. At least six observation notes should be prepared for testers to record the process. During usability testing, any significant action done by the participants should be recorded in details using the observation note.

7.3.7 Environmental needs

For testing purpose, an Android emulator or physical device should be prepared. The Android version should be at least 4.1. The test is recommended to be carried out on a device with Android 9.0. Besides, React Native CLI should be installed on the local machine. For physical device, a USB cable is needed to connect the local machine with the mobile device.

7.3.8 Schedule

The implementation of test planning and test execution is expected to be completed within 39 days, that is, from 1st March 2020 to 6th April 2020. The date of completion of each testing tasks is shown in the table below.

Table 7.1 Date and duration for each testing tasks

Task	Date of initiation	Date of completion	Duration (days)
Create a test plan	1st March	8th March	8
Unit Test	9th March	15th March	7
Integration Test	16th March	22nd March	7
System Test	23rd March	29th March	7
Usability Test	30th March	5th April	7

7.4 Test cases and results

7.4.1 Unit Test

Table 7.2 Test case for show captured animal information

Test Form Number	1

Test Designed By	Voon Cherng Jyh		Test Designed date	20 March 2020	
Test Executed by	Voon Cherng Jyh		Test Execution Date	1 April 2020	
Test Case					
Test case name	Show captured animal information				
Module Name	HomeScreen.takePicture				
Pre-conditions	User is in homepage				
Dependencies	SQLite, RNMlKitCustomModel				
Test Activities					
Test case summary	Test steps	Test data	Expected Results	Actual Results	Status
Show captured animal information	1. capture a giraffe.	Giraffe picture	User is navigated to animal information screen and information about giraffes is shown	User is navigated to animal information screen and information about giraffes is shown	Success

Table 7.3 Test case for earn badge

Test Form Number	2				
Test Designed By	Voon Cherng Jyh		Test Designed date	20 March 2020	
Test Executed by	Voon Cherng Jyh		Test Execution Date	1 April 2020	
Test Case					
Test case name	Earn Badge				
Module Name	CameraScreen.getAnimal				
Pre-conditions	User is in homepage				
Dependencies	SQLite				
Test Activities					
Test case summary	Test steps	Test data	Expected Results	Actual Results	Status
Earn Skyscrapper Badge	1. Take picture of a giraffe	Giraffe picture	'Skyscraper' badge progress increase by 1.	'Skyscraper' badge progress increase by 1.	Success

			‘Vegan’ badge progress increase by 1. User experience increase by 2.	‘Vegan’ badge progress increase by 1. User experience increase by 2.	
Earn Vegan badge	1. Take picture of a panda	Panda picture	‘Vegan’ badge progress increase by 1. User experience increase by 2.	‘Vegan’ badge progress increase by 1. User experience increase by 2.	Success
Earn Carnist badge	1. Take picture of a lion	Lion picture	‘Carnist’ badge progress increase by 1. User experience increase by 2.	‘Carnist’ badge progress increase by 1. User experience increase by 2.	Success
Earn Foodie Badge	1. Take picture of a bear	Bear picture	‘Foodie’ badge progress increase by 1. User experience increase by 2.	‘Foodie’ badge progress increase by 1. User experience increase by 2.	Success

Earn Into The Wild Badge	1. Increase level to level 2	user.experience: 10 user.level: 1	User's level increase to 2 User experience increase by 2 'In to the wild' badge progress increase by 1	User's level increase to 2 User experience increase by 2 'In to the wild' badge progress increase by 1	Success
Earn Adventurer Badge	1. Increase user level to level 3	user.experience: 20 user.level: 2	User's level increase to 3 User experience increase by 2 'Adventurer' badge progress increase by 1	User's level increase to 3 User experience increase by 2 'Adventurer' badge progress increase by 1	Success

Table 7.4 Test case for answer quiz

Test Form Number	3		
Test Designed By	Voon Cherng Jyh	Test Designed date	20 March 2020
Test Executed by	Voon Cherng Jyh	Test Execution Date	1 April 2020

Test Case					
Test case name	Answer quiz				
Module Name	HomeScreen.checkAns				
Pre-conditions	User is in homepage				
Dependencies	SQLite, Open Trivia DB API				
Test Activities					
Test case summary	Test steps	Test data	Expected Results	Actual Results	Status
Chose the correct answer	1. Select the correct answer	-	The selected answer is shown in green colour.	The selected answer is shown in green colour.	Success
Chose the incorrect answer	1. Select an incorrect answer	-	The selected answer is shown in red colour. The correct answer is shown in green colour.	The selected answer is shown in red colour. The correct answer is shown in green colour.	Success

Table 7.5 Test case for view fun fact

Test Form Number	4				
Test Designed By	Voon Cherng Jyh		Test Designed date	20 March 2020	
Test Executed by	Voon Cherng Jyh		Test Execution Date	1 April 2020	
Test Case					
Test case name	View fun fact				
Module Name	HomeScreen.getFacts				
Pre-conditions	User is in homepage				
Dependencies	SQLite				
Test Activities					
Test case summary	Test steps	Test data	Expected Results	Actual Results	Status
View fun fact	1. Scroll to ‘do you know’ section 2. click on show more	-	The box that shows the fun fact expands when show	The box that shows the fun fact expands when show more is tapped.	Success

			more is tapped.		

Table 7.6 Test case for view logs

Test Form Number	5				
Test Designed By	Voon Cherng Jyh		Test Designed date	20 March 2020	
Test Executed by	Voon Cherng Jyh		Test Execution Date	1 April 2020	
Test Case					
Test case name	View logs				
Module Name	LogScreen				
Pre-conditions	The user had captured an animal’s picture				
Dependencies	SQLite				
Test Activities					
Test case summary	Test steps	Test data	Expected Results	Actual Results	Status

View logs	1. Navigate to logs screen.	-	Log of captured animal is shown	Log of captured animal is shown	Success

7.4.2 Integration test

Table 7.7 Integration test cases and results

Modules involved	Test case	Test execution step	Expected output	Result
1. Camera module. 2. Logs module	The user views the logs after capturing an animal's picture	1. Touch on the camera button. 2. Take a picture of an animal. 3. Tap on the home button. 4. Tap on logs icon	The captured animal is shown in the log screen.	Success
1. Camera module 2. Badge module	The user unlocks an achievement and able to view the achievement progress	1. Touch on the camera button. 2. Take a picture of a giraffe. 3. Tap on the home button.	'Vegan' badge image is lightened. The progress of 'Vegan' badge increases.	Success

		4. Tap on achievement button.		
1. Quiz module. 2. User's level module.	The user answers a quiz correctly and earned experience point.	1. Tap on a correct answer in pop quiz section.	The progress bar in the homepage increases.	Success

7.4.3 System test

Table 7.8 Test cases of system testing

Steps	Test Step	Expected Results	Actual Results	Pass/fail
Use case: Capture animal photo				
1.	Tap on the app icon	The app is launched without any errors	The app is launched without any errors.	Pass
2.	Tap on the camera button	Camera screen is shown	Camera screen is shown	Pass
3.	Tap on the snap photo button	The animal photo is captured and saved. Animal information screen is shown. User experience points increases.	The animal photo is captured and saved. Animal information screen is shown. User experience points increases.	Pass
Use case: view activity log				
1	Touch on logbook icon	The system displays a list of activity logs.	The system displays a list of activity logs.	Pass
Use case: view animal's information				
1	Touch on logbook icon.	The system displays a list of activity logs.	The system displays a list of activity logs.	Pass
2	Touch on one of the logs	Animal information screen is shown. Information about the chosen animal is shown.	Animal information screen is shown. Information about the chosen animal is shown.	Pass
Use case: view animal fun fact				

1.	Navigate to the home screen	Animal's fun fact is shown at the bottom of the home screen.	Animal's fun fact is shown at the bottom of the home screen.	Pass
2.	Tap on show more button	The container expands and the full fun fact is shown.	The container expands and the full fun fact is shown.	Pass
Use case: complete quiz				
1.	Navigate to the home screen	A quiz is shown at the top.	A quiz is shown at the top.	Pass
2.	Tap on one of the answers	The correct answer is shown in green. User experience points increases.	The correct answer is shown in green. User experience points increases.	Pass
Use case: view badges				
1.	Tap on the trophy button from the home screen.	Achievement screen is shown. A list of badges is shown.	Achievement screen is shown. A list of badges is shown.	Pass
Use case: view challenges				
1.	Tap on one of the badges from the achievement screen.	A floating window is shown with the badge progress. The details of the challenge are shown.	A floating window is shown with the badge progress. The details of the challenge are shown.	Pass

7.4.4 Usability test

Usability test was conducted with 5 participants. Initially, the usability test is targeted on kids. Due to the implementation of Movement Control Order during the development of the project, the target of test participant was switched from children to university students. Furthermore, all the physical forms were replaced and were filled virtually. During the usability test, the steps taken by the participants were closely observed and the time taken for executing each task was recorded. A post-test survey was conducted with the participants after the session ended to collect their opinion on the application.

The summary of the usability test is shown below:

Table 7.9 Summary of usability test

Task	**Average Time Taken(s)**	**Number of tests conducted**	**Number of test success**	**Observation**
Snap A photo of an animal	9.6	5	5	Task executed smoothly
Check the animals captured in log screen	10.8	5	5	The participant took a while to find the home button
View acquired badges	4.2	5	5	Task executed smoothly
Answer quiz	10.2	5	5	The participants took a while to choose the correct answer
View animal fun fact	2.6	5	5	Task executed smoothly

From the post-test survey, it can be concluded that most of the participants enjoy the overall appearance of the app. Apart from that, the words and symbols used in the app are highly comprehensible. From the opinions of five university students, it can be deduced that the app is enjoyable and fun to use. Major elements in the app, especially the symbol and user-friendliness, are satisfactory. Lastly, all of the participants think that they would download the app if the app is on App Store. The summary of the post-test survey is shown below:

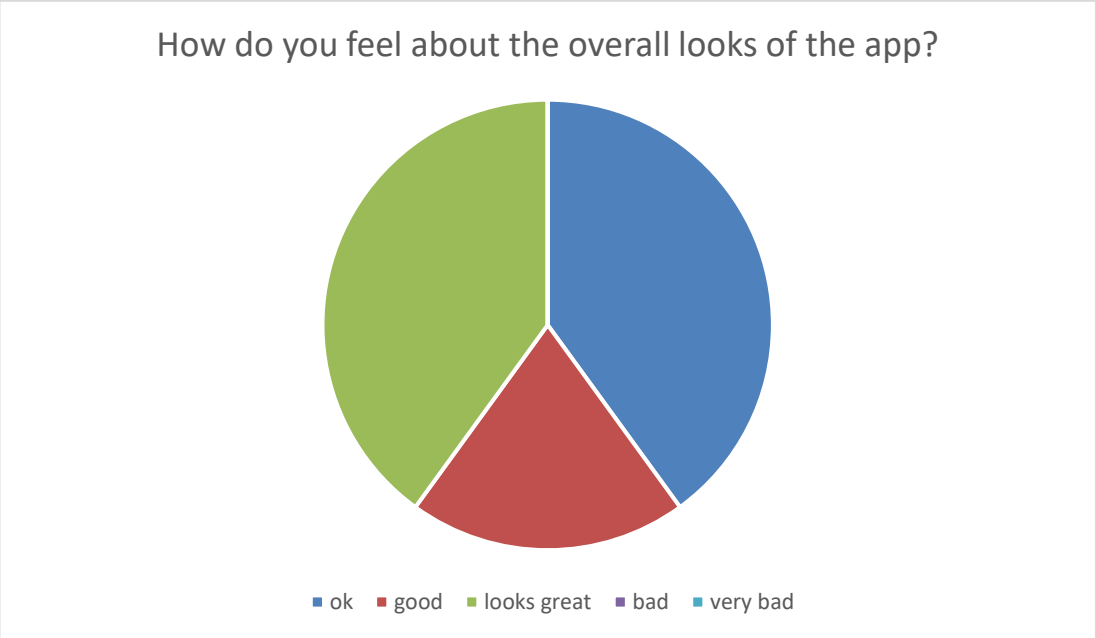


Figure 7.2 Opinion on the app user interface design

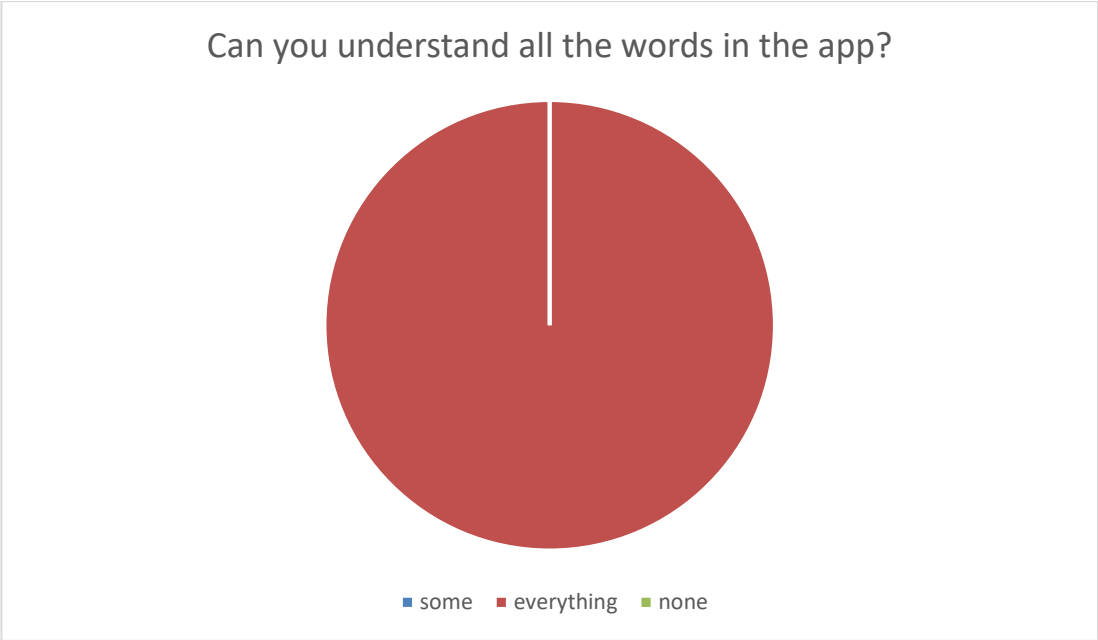


Figure 7.3 Readability of the app

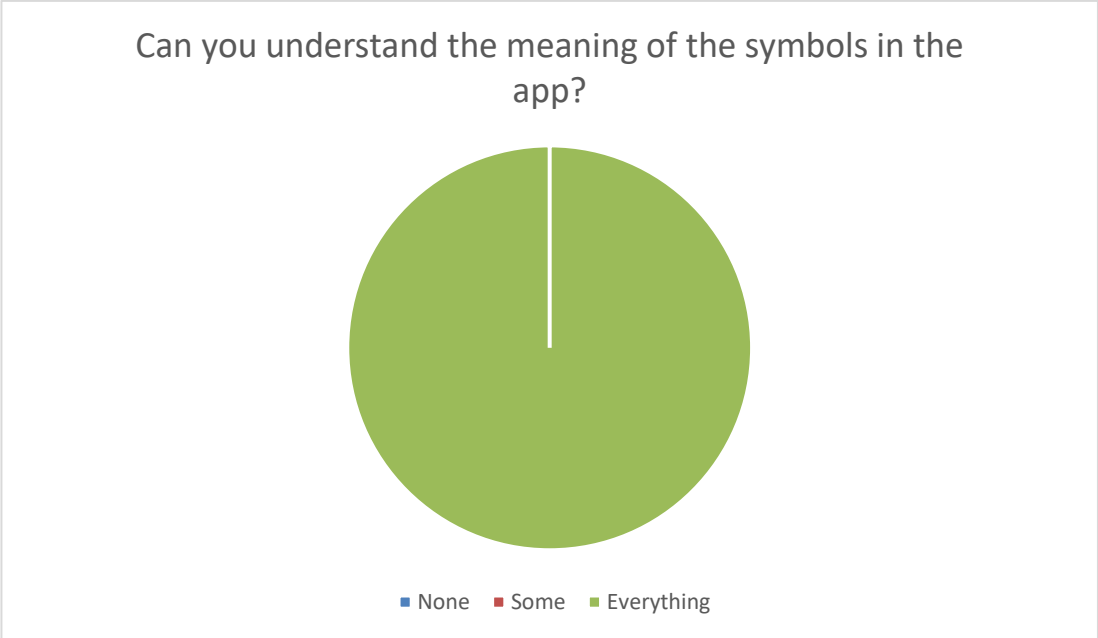


Figure 7.4 Readability of the app symbols

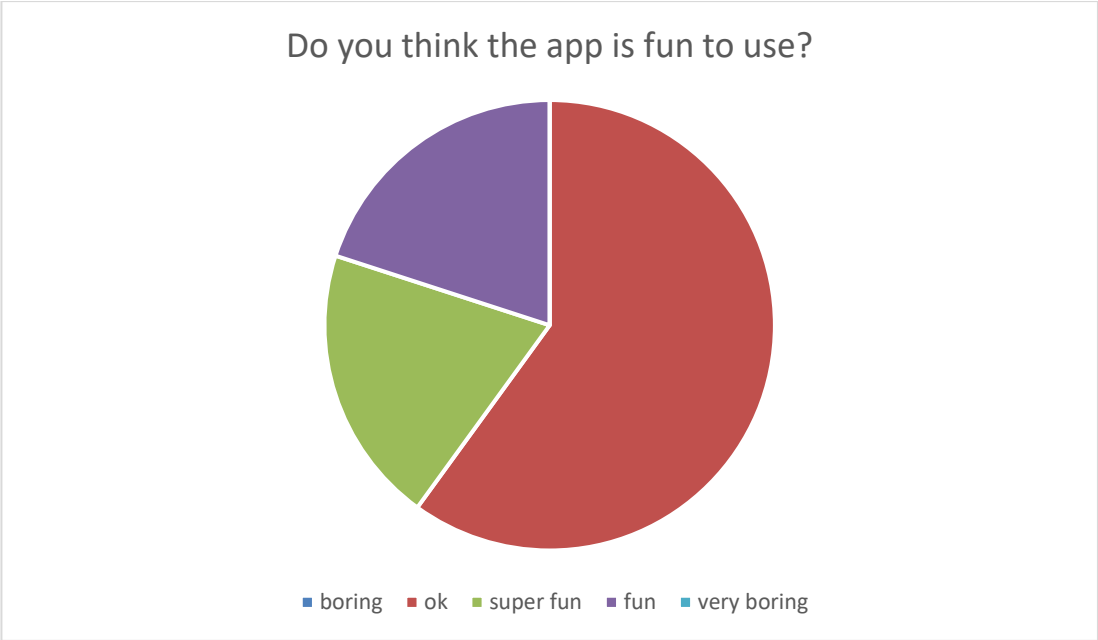


Figure 7.5 Satisfaction of the users

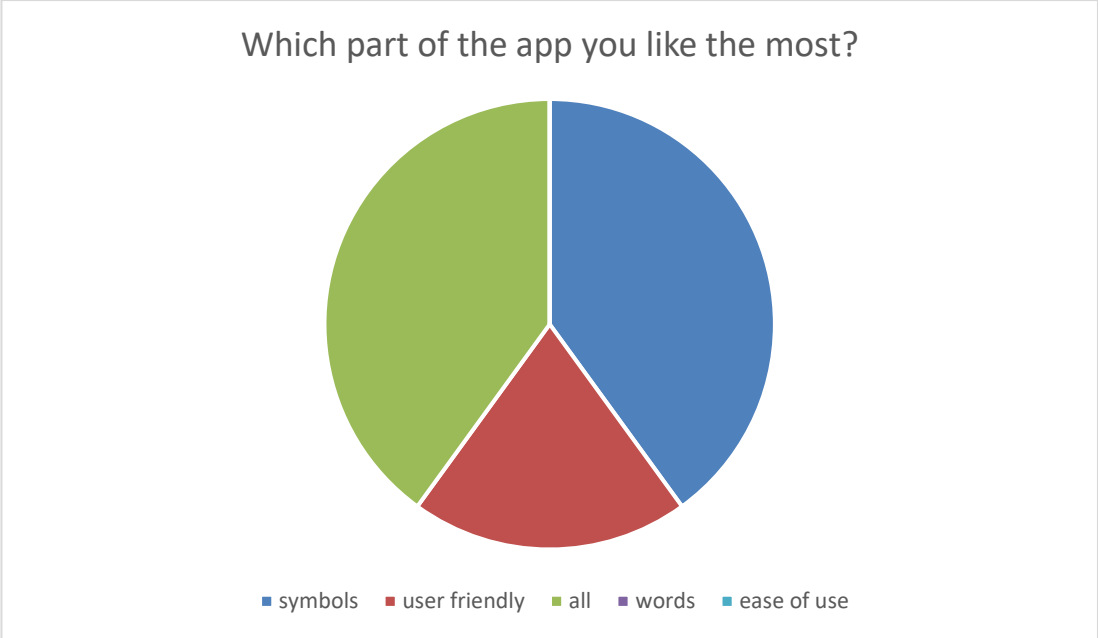


Figure 7.6 Favourite element in the app

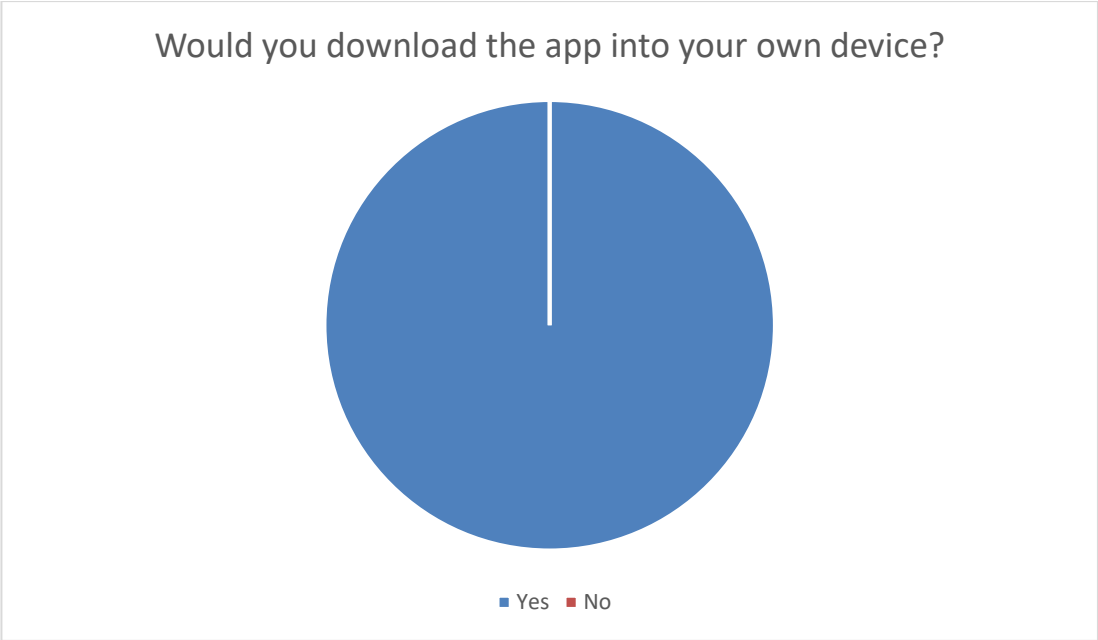


Figure 7.7 Opinion on downloading the app

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

The project combined the elements of education, technology and biology by developing a mobile application for kids to recognize animals in the zoo. The app promotes self-learning in kids when visiting a zoo. With just a picture of an animal, the app is able to infer the animal captured and provide its information. To fulfil the objectives, the project was broken down into two major parts: Android mobile application development and CNN model training.

During the 6-months project, the Android mobile application named 'Animalia' was developed. The mobile app consists of three major modules, animal recognition module, gamification module and quiz module. The animal recognition module receives an animal photo and performs image recognition. To attracts kids in using the app, gamification concept such as challenges, experience point and user level is utilized. Besides, the quiz and animal fun facts allow kids to discover more about the animals in a zoo. The modules were designed, implemented and went through unit testing, integration testing and usability testing.

Experiments were conducted on three different pretrained models, which are, MobileNet, EffNet and NASNet. Three of the models were analysed from the aspects of output model size, the computational power needed, speed of image recognition and accuracy of the model. MobileNet was chosen as the model for the image recognition because of its high efficiency. Among the three models, although MobileNet shows a low performance on accuracy, it consumes a lower amount of computational power and storage size.

8.2 Recommendations for future work

Despite the objective fulfilment, there is still room for improvement for the app in the future. The limitations and corresponding improvements were tabulated and shown below.

Table 8.1 Limitations and recommendations

Limitation	Recommendations
Unable to recognize animal species	The current animal recognition model is only capable of recognizing the common name of an animal. For instance, the image recognition is only able to tell if a photo contains a bear but it is unable to know if it is a grizzly bear or sun bear. To overcome this, the CNN model can be trained with multiple classes of different bear species.
Accuracy of each animal classes	The targeted accuracy of current CNN model is 95% and above. Achieving an accuracy of over 98% is believed to be attainable by improving the architecture and the hyperparameter of the CNN.
Number of animal classes	The trained model in the current system is only capable of recognizing 18 types of animals. The animals' number can be improved by feeding additional dataset of other animals.
Mobile app data is stored in local storage	SQLite, which is mobile local storage, is used for the current system. This implies that the data will be destroyed once the app is uninstalled. To overcome the limitation, cloud storage can be used.

REFERENCES

Afkham, H. M., 2008. Kth-animals.[online] Available at:<http://www.csc.kth.se/~heydarma/datasets/animal_database.tar.gz> [Accessed 7 July 2019]

Ballantyne, R., Packer, J., Hughes, K. and Dierking, L., 2007. Conservation learning in wildlife tourism settings: Lessons from research in zoos and aquariums. Environmental Education Research, 13(3), pp.367-383.

Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. [online] 2(5). Available at: <http://arxiv.org/abs/1205.6904>.

Board, I.S.S. and ICSSES Committee, 1998. Ieee recommended practice for software requirements specifications. In Institute of Electrical and Electronics Engineers.

Brahimi, S., Ben Aoun, N. and Ben Amar, C., 2017. Inception Recurrent Convolutional Neural Network for Object Recognition. [online] p.1034107. Available at:
<http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2268672>

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition, 30(7), pp.1145-1159

Centers for Medicare & Medicaid services, 2005. Selecting a development approach. [pdf]. Available at: <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf> [Accessed 10 July 2019]

Cruz, C., 2019. Easy Image Classification with TensorFlow 2.0. Medium. Available at: https://towardsdatascience.com/easy-image-classification-with-tensorflow-2-0-f734fee52d13 [Accessed August 10, 2019].

Daud, N.M.N., Bakar, N.A.A.A. and Rusli, H.M., 2010. Implementing Rapid Application Development (RAD) methodology in developing practical training application system. Proceedings 2010 International Symposium on Information Technology - System Development and Application and Knowledge Society, ITSim'10, 3, pp.1664–1667.

Dennis, A. et al., 2015. System analysis & design: an object-oriented approach with UML, Hoboken, NJ: John Wiley & Sons, Inc.

Dertat, A., 2017. Applied Deep Learning - Part 4: Convolutional Neural Networks. Medium. Available at: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 [Accessed March 20, 2020].

Ding, B., Qian, H. and Zhou, J., 2018, June. Activation functions and their characteristics in deep neural networks. In 2018 Chinese Control And Decision Conference (CCDC) (pp. 1836-1841). IEEE

Douillard, A., 2018. 3 Small but Powerful Convolutional Neural Networks. Arthur Douillard. Available at: https://arthurdouillard.com/post/3-small-but-powerful-cnn/ [Accessed March 20, 2020].

Feunteun, Y., 2019. Parallel and Distributed Deep Learning: A Survey. Medium. Available at: https://towardsdatascience.com/parallel-and-distributed-deep-learning-a-survey-97137ff94e4c [Accessed August 10, 2019].

Google LLC, 2019. Google Lens. Mobile app. Version 2.5.4. Available from: https://play.google.com/store/apps/details?id=com.google.ar.lens&hl=en. [Accessed 17 July 2019].

iNaturalist, 2019. Seek by iNaturalist. Mobile app. Version 2.5.3. Available from: https://play.google.com/store/apps/details?id=org.inaturalist.seek&hl=en. [Accessed 17 July 2019].

Jain, V., 2019. Everything you need to know about "Activation Functions" in Deep learning models. Medium. Available at: https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253 [Accessed August 10, 2019].

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.

Freeman, I., Roese-Koerner, L. and Kummert, A., 2018, October. Effnet: An efficient structure for convolutional neural networks. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 6-10). IEEE.

Liao, S., 2018. Google Lens is now available as a standalone app. [online] Available at: <https://www.theverge.com/2018/6/4/17426668/google-lens-app-play-store-android> [Accessed 17 July 2019]

LD Studio, 2017. Image Recognizer. Mobile app. Version 1.0.2. Available from: https://play.google.com/store/apps/details?id=com.ldqstudio.image_recognizer&hl=en. [Accessed 17 July 2019].

Malaysian Communication and Multimedia Commission, 2017. Hand phone users survey 2017. [pdf] Malaysia: MCMC. Available at: <https://www.skmm.gov.my/skmmgovmy/media/General/pdf/HPUS2017.pdf> [Accessed 7 June 2019]

Păvăloaia, V.-D., 2013. Methodological Approaches to Computer Modelling Possibilities in Financial Analysis. Annals of the Alexandru Ioan Cuza University - Economics 59. doi:10.2478/v10316-012-0026-5

Prakash, G., 2019. Google Play Store Apps. Dataset. Version 1. Available at: <https://www.kaggle.com/gauthamp10/google-playstore-apps> [Accessed 17 July 2019].

Reddy, A.S.B. and Juliet, D.S., 2019, April. Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0945-0949). IEEE.

Saha, S., 2018. A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. Medium. Available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 [Accessed August 10, 2019].

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2019. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).

Sinnott, R.O., Wu, F. and Chen, W., 2018, December. A Mobile Application for Dog Breed Detection and Recognition Based on Deep Learning. In 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT) (pp. 87-96). IEEE.

Tang, R., Wang, W., Tu, Z. and Lin, J., 2018, April. An experimental analysis of the power consumption of convolutional neural networks for keyword spotting. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5479-5483). IEEE.

theAsianparent, 2014. Mobile Device Usage Among Young Kids: A Southeast Asia Study. [pdf] Singapore: Tickled Media Pte Ltd. Available at: <https://s3-ap-southeast-1.amazonaws.com/tap-sg-media/theAsianparent+Insights+Device+Usage+A+Southeast+Asia+Study+November+2014.pdf> [Accessed 7 June 2019]

Wang, C.F., 2018. A Basic Introduction to Separable Convolutions. Medium. Available at: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728 [Accessed March 20, 2020].

Zoph, B., Vasudevan, V., Shlens, J. and Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8697-8710).

APPENDICES

APPENDIX A: Usability Test Documents

Participant: Date: Researcher: Voon Cherng Jyh				
Task	**Time Taken**	**Observation**	**Category** (Error, success …)	Route Taken
Snap A photo of an animal				
Check the animals captured in log screen				
View acquired badges				
Answer quiz				
View animal fun fact				

Document A-1: Observation notes template

In each question, select one of the answers which you think is the most suitable.

1. How do you feel about the overall looks of the app?

2. Can you understand all the words in the app?

3. Can you understand the meaning of the symbols in the app?

4. Do you think the app is fun to use?

Document A-2-1: Post-test questions

5. Do you think this app can help you in your study?

6. Which part of the app you like the most?

- ☐ Words
- ☐ Symbol
- ☐ Ease of use
- ☐ User friendly user interface
- ☐ All of the above

7. Would you download the app into your own device?

Document A-2-2: Post-test questions

APPENDIX B Usability Test Results

Participant: Yeoh Xing Yuan
Researcher: Voon Cherng Jyh
Date: 10 April 2020

Task	Time Taken	Observation	Category (Error, success…)	Route Taken
Snap A photo of an animal	11	Task executed smoothly	success	Home -> camera screen
Check the animals captured in log screen	18	Home button not obvious	success	Info screen -> home -> log screen
View acquired badges	3	Task executed smoothly	success	Log Screen -> home screen -> achievement screen
Answer quiz	8	Task executed smoothly	success	Achievement screen -> home screen
View animal fun fact	1	Task executed smoothly	success	Slide down home screen

Participant: Yap Ying Hao
Researcher: Voon Cherng Jyh
Date: 10 April 2020

Task	Time Taken	Observation	Category (Error, success…)	Route Taken
Snap A photo of an animal	9	Task executed smoothly	success	Home -> camera screen
Check the animals captured in log screen	25	Home button not obvious	success	Info screen -> home -> log screen
View acquired badges	3	Task executed smoothly	success	Log Screen -> home screen -> achievement screen
Answer quiz	13	Took some time in choosing the correct answer	success	Achievement screen -> home screen
View animal fun fact	9	Did not realize the home page is scrollable	success	Slide down home screen

Participant: Bryan Yap Seng Haw

Date: 10 April 2020

Researcher: Voon Cherng Jyh

Task	Time Taken	Observation	Category (Error, success…)	Route Taken
Snap A photo of an animal	6	Task executed smoothly	success	Home -> camera screen
Check the animals captured in log screen	3	Task executed smoothly	success	Info screen -> home -> log screen
View acquired badges	8	Task executed smoothly	success	Log Screen -> home screen -> achievement screen
Answer quiz	6	Task executed smoothly	success	Achievement screen -> home screen
View animal fun fact	1	Task executed smoothly	success	Slide down home screen

Participant: Voon Li Jiat

Date: 10 April 2020

Researcher: Voon Cherng Jyh

Task	Time Taken	Observation	Category (Error, success…)	Route Taken
Snap A photo of an animal	15	Task executed smoothly	success	Home -> camera screen
Check the animals captured in log screen	4	Task executed smoothly	success	Info screen -> home -> log screen
View acquired badges	2	Task executed smoothly	success	Log Screen -> home screen -> achievement screen
Answer quiz	15	Took some time in choosing the correct answer	success	Achievement screen -> home screen
View animal fun fact	1	Task executed smoothly	success	Slide down home screen

Participant: Tan Chuan Zhi

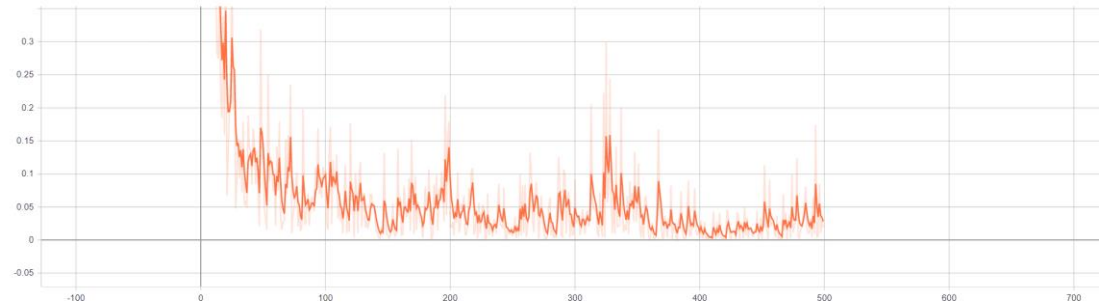
Researcher: Voon Cherng Jyh

Date: 10 April 2020

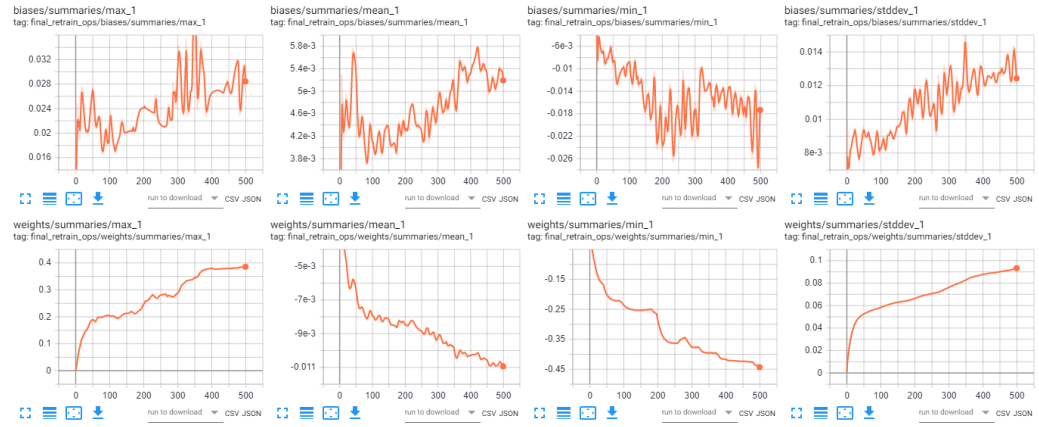
Task	Time Taken	Observation	Category (Error, success…)	Route Taken
Snap A photo of an animal	7	Task executed smoothly	success	Home -> camera screen
Check the animals captured in log screen	4	Task executed smoothly	success	Info screen -> home -> log screen
View acquired badges	5	Task executed smoothly	success	Log Screen -> home screen -> achievement screen
Answer quiz	9	Task executed smoothly	success	Achievement screen -> home screen
View animal fun fact	1	Task executed smoothly	success	Slide down home screen

APPENDIX C Models training results

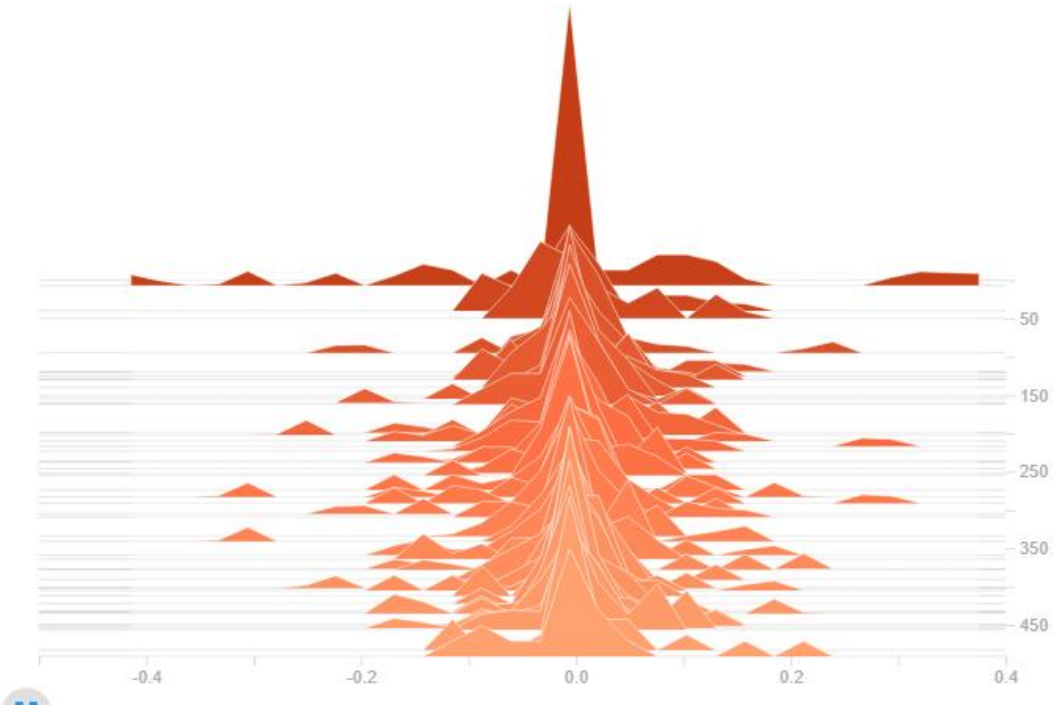
Appendix B-1-1 Graph of Mobilenet accuracy



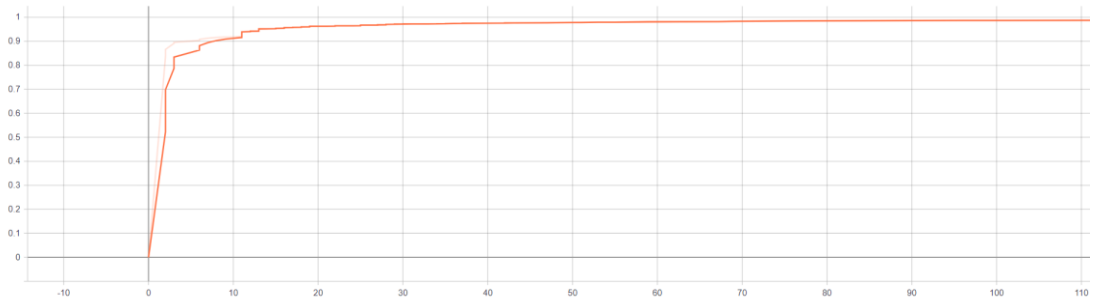
Appendix B-1-2 Graph of Mobilenet cross entropy



Appendix B-1-3 Graph of Mobilenet weights and biases

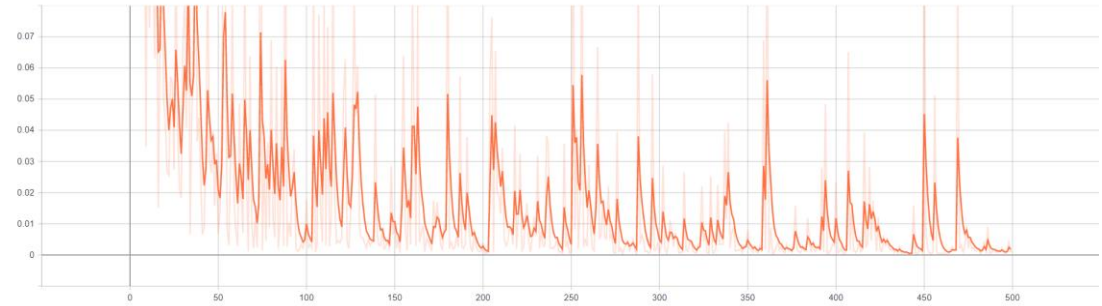


Appendix B-1-3 Graph of Mobilenet learning rate

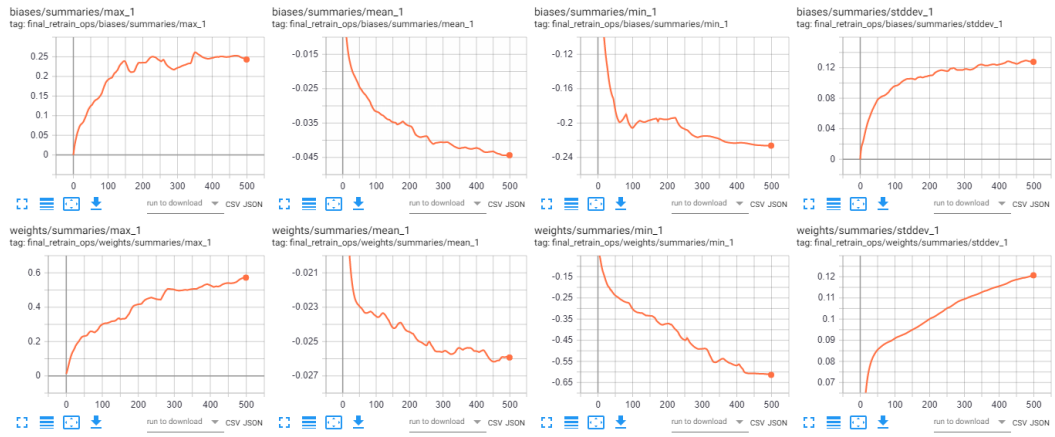


Appendix B-1-4 Mobilenet ROC curve

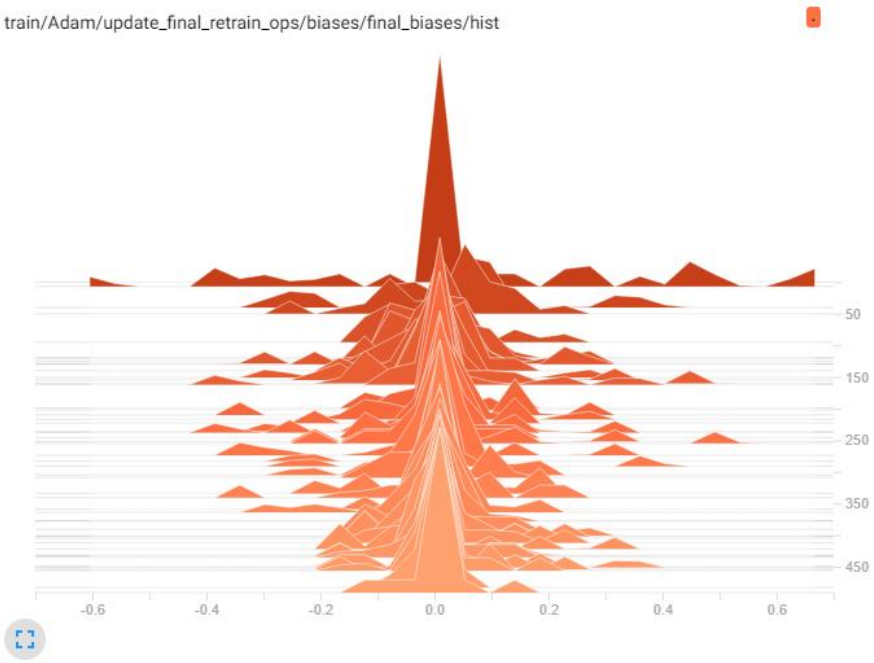
Appendix B-2-1 Graph of EffNet accuracy



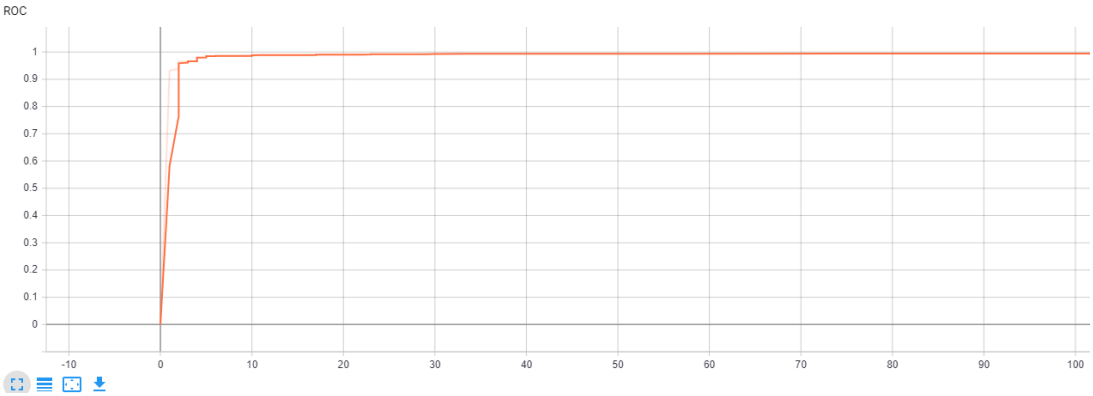
Appendix B-2-2 Graph of EffNet cross entropy



Appendix B-2-3 Graph of EffNet weights and biases

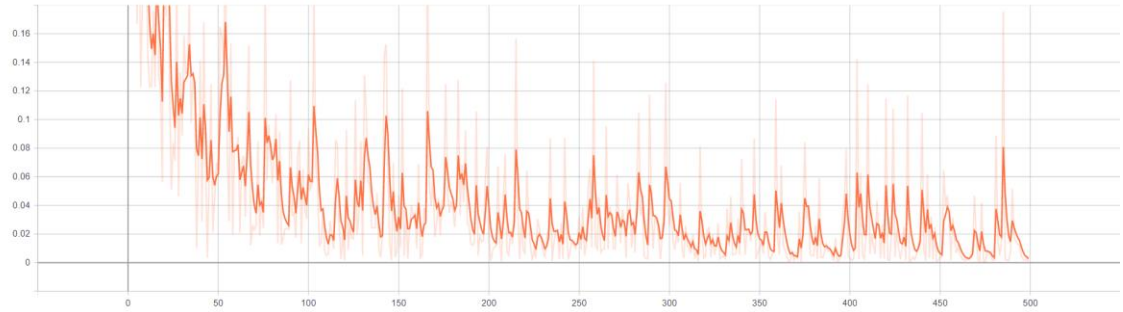


Appendix B-2-4 Graph of EffNet learning rate

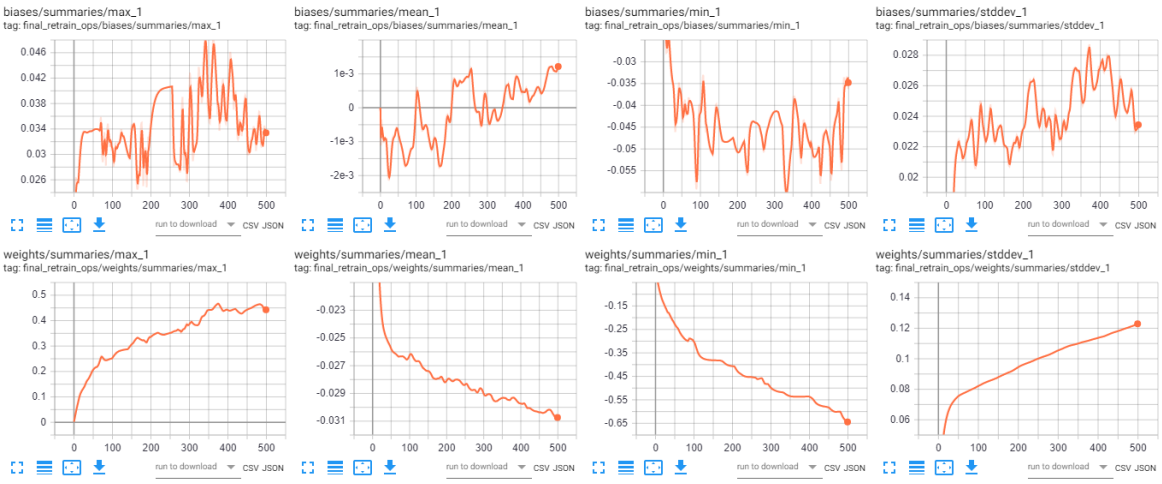


Appendix B-2-5 Graph of EffNet ROC curve

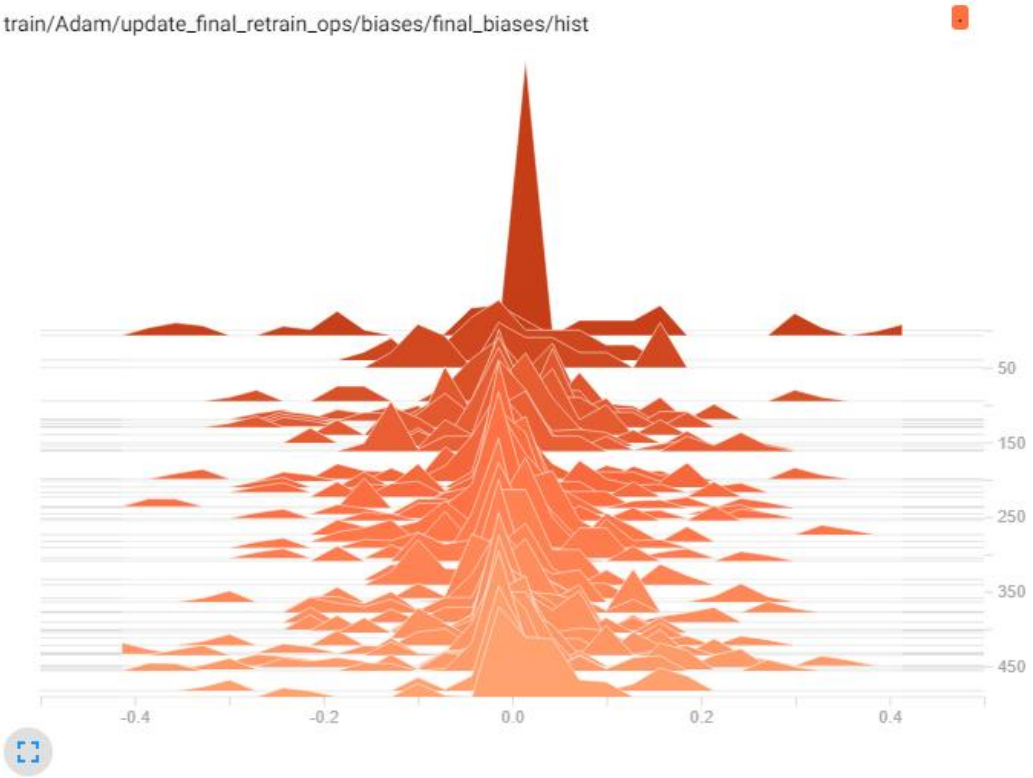
Appendix B-3-1 Graph of NASNet accuracy



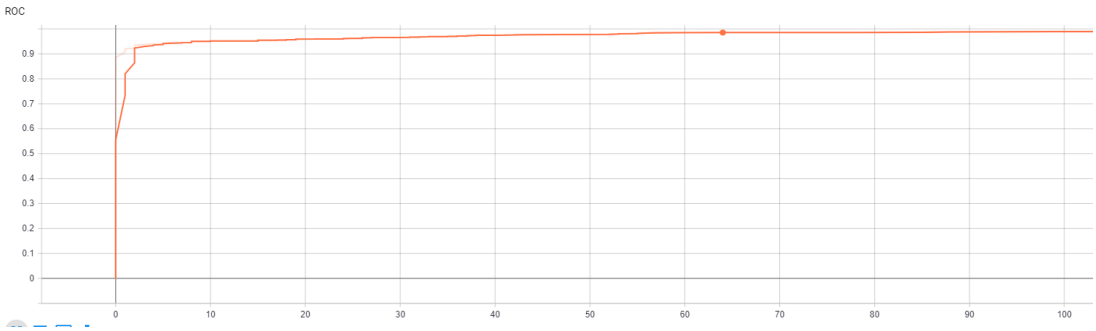
Appendix B-3-2 Graph of NASNet cross entropy



Appendix B-3-3 Graph of NASNet weights and biases

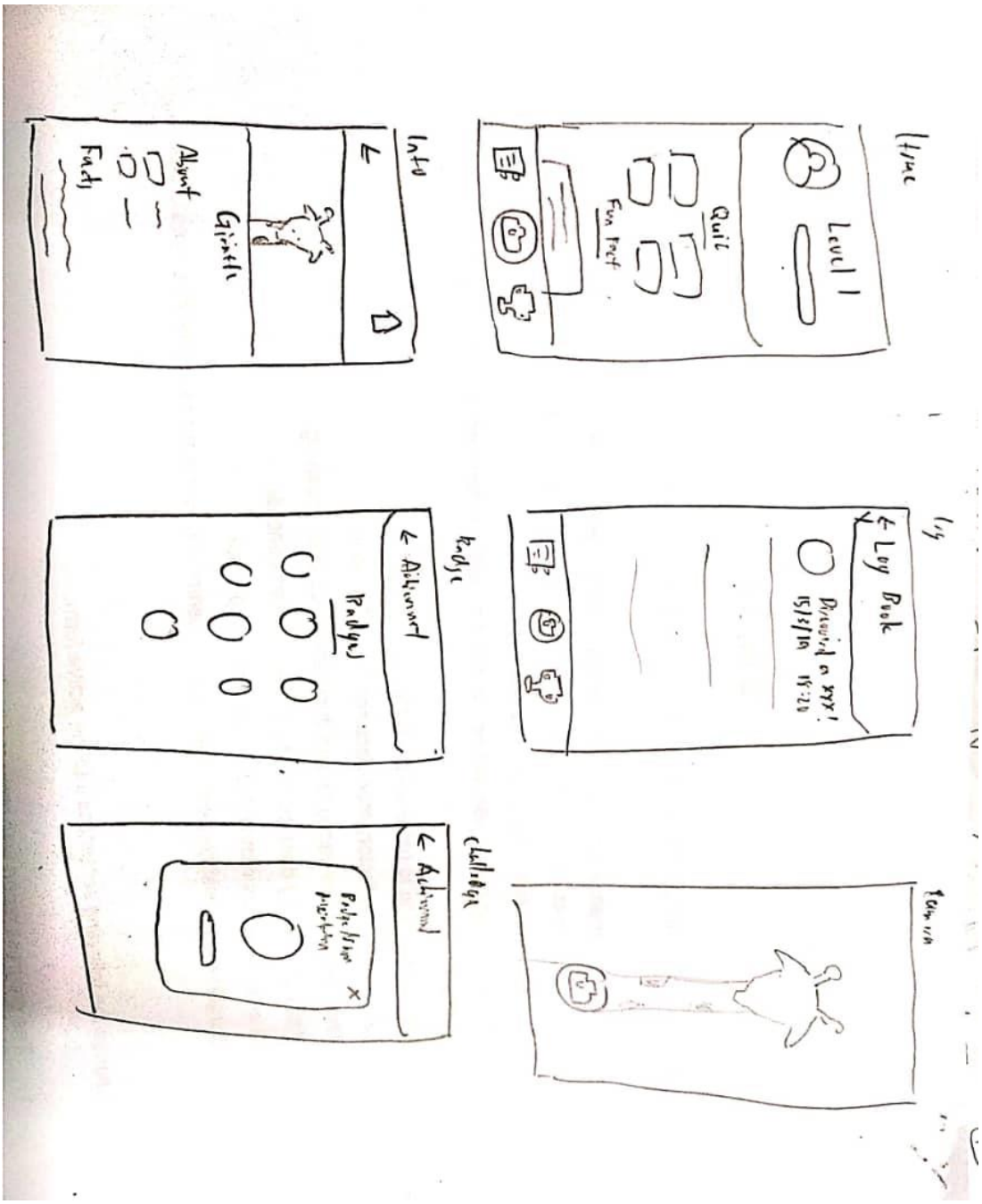


Appendix B-3-4 Graph of NASNet weights and biases



Appendix B-3-5 Graph of NASNet ROC curve

APPENDIX D Low-fidelity prototype



APPENDIX E Comments on FYP Report I

Project title:	An educational android app for identifying animals in zoo
Student Name	VOON CHERNG JYH
Supervisor	Dr Khor Kok Chin
Moderator	Dr Winnie Wong Whee Yen

Key Assessment for Project Proposal	Supervisor Comments/Remarks	Moderator Comments/Remarks
Project Description - Is the problem or need to be addressed clearly presented? - Is the proposed approach or solution clearly presented and justified?		Pg 12 in-text referencing for Asiaparent is not correct
Project Scope and Objectives - Is the scope of the project clearly defined? - Are the objectives of the project clearly specified? - Are the project scope and objectives appropriate for a final year project?		
Literature Review / Fact Finding for Benchmarking /		

Verification of Project - Are sources for literature review / fact finding appropriate? - Is information from literature review / fact finding relevant and adequate? - Is information from literature review / fact finding clearly presented and discussed?		
Research/Development Methodology and Development Tools - Is the methodology for the project clearly described and discussed? - Are the required development tools clearly described and discussed? - Are the stated methodology and development tools appropriate?		
Project Plan		Figure 3.2, reader unable to

- Are the phases and tasks of the project properly defined and planned? - Are the phases and tasks consistent with the methodology of the project?		read and view the content. Suggest to put it into landscape Figure 3.3, unable to read gantt chart. Project Spec and non-functional spec can be further improve by using PM template
Initial Deliverables - Are deliverables (e.g. use case diagrams and descriptions) of initial phases of the project plan included in the report?		Reference: Do remember to include ref for all Figures (e.g. 2.3, 2.11, 2.12 3.18,, etc…
Report Structure and References - Is the report organised in a logical structure? - Are references listed in accordance to Harvard format?		[redacted]
Language and Clarity of Writing - Are the sentences concise and		

understandable? - Are there spelling and grammar issues?		[redacted]