VOICE AND IMAGE RECOGNITION FOR SMART PET APP

By

LEE YEN LONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

January 2019

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: VOICE AND IMAGE RECOGNITION FOR SMART PET APP

Academic Session: Jan 2019

I LEE YEN LONG

(CAPITAL LETTER)

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

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

Verified by,

_____ _____

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

Address:
No 9, Jalan Jelutong,
Kluang,
86000 Kluang, Johor.

Supervisor's name

Date: 8th April 2019 **Date**: _____

VOICE AND IMAGE RECOGNITION FOR SMART PET APP

By

LEE YEN LONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

January 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled "**VOICE AND IMAGE RECOGNITION FOR SMART PET APP**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : Lee Yen Long

Date : 8nd April 2019

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Saw Seow Hui, who has given me this bright opportunity to engage in voice and image recognition project. It is my first step to establish a career in voice and image recognition field. A million thanks to you.

To my groupmate, Ooi Shion Yeing, for her patience, unconditional support and help me a lot in this final year project. Finally, I must say thanks to my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

Nowadays, there are many people feel depressed and lonely. The propose of this project is to develop a Smart Pet Application which is an Android application that can interact with people to reduce their loneliness. There are two parts in the Smart Pet Application:

1. Game design and logics.
2. Artificial intelligence (AI) engine to perform voice and image recognition.

This report will focus on second part, which is descript how the AI engine work.

In the propose system, TensorFlow Sequence-to-Sequence (tf-seq2seq) Model is used to be the machine learning model to let the Smart Pet can communicate with its owner. tf-seq2seq is a general-purpose encoder-decoder framework for TensorFlow that suitable for Conversational Modelling.

On the other hand, Baidu Cloud face recognition SDK is used for the system to let the Smart Pet perform face recognition in order to recognize its owner. Baidu Cloud face recognition is a component on Baidu Cloud Computing Platform that suitable for real-time object recognition.

We will combine the game design and logics together with the AI engine to become a complete Smart Pet Application.

TABLE OF CONTENS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WHO	World Health Organization
AI	Artificial Intelligence
AR	Augmented Reality
GUI	Graphical User Interface
ASR	Automatic Speech Recognition
Siri	Speech Interpretation and Recognition Interface
NLP	Natural Language Processing
DARPA	Defence Advanced Research Projects Agency
CNNs	Convolutional Neural Networks
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
IoT	Internet of Things
API	Application Programming Interface
AIML	Artificial Intelligence Markup Language
Alice	Artificial Linguistic Internet Computer Entity
XML	Extensible Markup Language
RBCD	Rule-Based Closed-Domain
GPU	Graphics Processing Unit
GCP	Google Could Platform
vRAM	Video Random Access Memory
SDLC	Software Development lifecycle
ML	Machine Learning
STT	Speech-to-Text
Seq2Seq	Sequence-to-Sequence
TTS	Text-to-Speech
GRU	Gated Recurrent Units

Chapter 1: Introduction

1.1 Introduction

In today's busy society, stress is one of the biggest problems that everyone faces (Twenge 2014). If the stress is not relieved in time, it will cause depression over time. According to the World Health Organization (WHO)'s reports, among the cause of ill health and disability worldwide, depression is the main cause. From the WHO's estimates, there are more than 300 million people are suffering from depression and it is increasing about 20% for every 10 years (Chaib 2017). Depression is a disease that can interfere with motivation, attention, and many other aspects of everyday functioning. Depression can cause insomnia and can affect someone's appetite, in some cases leading to weight loss and in other cases gaining weight (Dysthymia 2014). Due to its severity, depression is a serious problem that should not be ignored.

In addition to depression, loneliness is one of the problems faced by people nowadays. Of course, loneliness is also the main factor of depression, and there is a close relationship between the two. Depression can manifest as a symptom of loneliness, but it does not mean that depression will automatically make people alone (Ross 2015). Although depression and loneliness were prevalent problems, what many people does not know that it can actually be overcome.

One of the most effective ways to overcome depression and loneliness is to keep a pet. Pets can make the owner feel comfortable and a special bonding with the owner. If someone feel lonely and worthless every day, pets can break the circle (Robinson 2017). Depression may make someone want to avoid others, but pets can open their world. Studies have shown that pets can help people to meet others, inspire friendships, and build a people's support network (Wood et al. 2015). But not everyone is able to keep a pet, so there is an alternative way which is keeping a virtual pet.

Compare to keeping a real pet, virtual pet has many advantages, one of the most significant advantages is the owner of real pet need to spend a lot of extra money for their pet. First, the owner could be spent an initial cost to get their pet, depending on the source

of the pet. If the owner buys a pet of the famous species, such as Munchkin cat or Pomeranian dog, the cost may require thousands of ringgits. Even though adoption from shelters are cheaper, but also cost more than 50 ringgits (Siew May 2016). Not only that, pet food is also a big expense for the owner, the owner can spent up to RM 150 per month on pet food (Emmanuel 2015).

Table 1.1.1 One-time expenses for keeping a dog and cat

One-time pet expenses	**Dog**	**Cat**
Spaying/Neutering	RM200	RM150
Medical	RM100	RM130
Accessories	RM125	RM100
Litter box		RM75
Scratching post		RM35
Crate	RM150	
Carrying Crate	RM100	
Training	RM100	
TOTAL	**RM775**	**RM490**

Table 1.1.2 Annual expenses for keeping a dog and cat

Annual pet expenses	**Dog**	**Cat**
Food	RM1,800	RM1,200
Medical Exams & Vaccination	RM150	RM150
Toys & Treats	RM240	RM240
Litter		RM180
License	RM10	
Grooming	RM975	RM690
Miscellaneous	RM100	RM85
TOTAL	**RM3,275**	**RM2,545**

Table 1.1.1 and Table 1.1.2 shown the expenses for keeping a dog and cat (Emmanuel 2015). We can see the, a pet owner needs to spent up to RM 3000 per year to keeping a pet, that is a huge amount of expenses, although some virtual pet application also

requires some fees (in-app purchase), but compare with real pet, that is just a small amount. So, virtual pet can save a lot of money compare to real pet.

Second, some students' families forbid them to keep pets, and also in university, most students' dormitory do not allow students to keep pets. So, they can keep a virtual pet. Third, the owner of virtual pet no need to spent a lot of time on their pet, especially for the working adults.

Furthermore, Artificial Intelligence (AI) is a hot topic, especially on voice and image recognition. By integrating voice recognition with mobile application, we can let user do much more by using voice, it can increased interactions with user and also increase user experience (Krunal 2018). There are several successful applications such as Apple Siri, Microsoft Cortana, Amazon Alexa prove that voice recognition integrated with mobile application can bring convenience to users. On the other hand, image recognition also become more and more popular in mobile application. The two largest topics regarding to image recognition is object detection and face recognition. In general, face recognition is used in security aspects, such as face unlock, identification of criminals (Srirupa 2017). In fact, image recognition can also be used in other area, which is let the virtual pet recognise its owner and perform different responses with its owner or strangers.

We are going to develop a smart pet game android application using Augmented Reality (AR) and Artificial Intelligence (AI) technology. The purpose of develop this proposed application is to help people that feel lonely and depression and people that cannot has a real pet especially students to own a pet that can release their stress and find the someone they can talk to. The proposed app can roughly divide into two part which is the game component and AR Graphical User Interface (GUI) and the second part is AI engine that can make the pet smarter. The main focus things in this report is the AI engine. The AI technology in the proposed app include voice and image recognition. For the voice recognition, we focus on using deep learning method to let the virtual pet can communicate with its owner. For the image processing part, we use TensorFlow Object Detection API to train the virtual pet recognize its owner.

1.2 Problem Statement and Motivation

Although there are many existing virtual pet game applications, but there are also several problems for the existing system that need to be solved.

Lack of dual communication between the owner and the virtual pet

For the existing app, the owner can only keep the virtual pet like just playing a mobile game, they cannot communicate with the virtual pet. Furthermore, their implemented voice recognition for the virtual pet can only understand some simple pre-defined commands, this is obviously failed to meet the requirement of solving loneliness of people. In the proposed app, by using deep learning, the virtual pet is not limited in understanding simple commands, but it is able to communicate with its owner. So, the owner can talk with the virtual pet to pass the time and also relieves the feeling of loneliness.

Lack of image recognition features

The crucial feature of a virtual pet is that it is able to recognize the owner, it will get different responses when interact with its owner or strangers (Merola et al. 2012). But in the existing virtual pet application, the virtual pet is not able to recognize the owner, which cause the owner think this is just a machine without any bonding. In the proposed app, the virtual pet will be able to recognize the owner. Thus, it looks more realistic and the owner feels comfortable and safe with the virtual pet.

1.3 Project Scope

The proposed app is a smart pet game based on android platform. It is a pet simulation game using Augmented Reality (AR) technology that allows the interaction between user and the virtual pet. There are two parts in the proposed app:

1. Front-end and back-end of mobile application part which include the Graphical User Interface (GUI) of game design, game features, and create the virtual pet game by using augmented reality technique.
2. Artificial Intelligence (AI) engine includes conversation component using Automatic Speech Recognition (ASR) to let the virtual pet able to communicate with its owners, face recognition component using Baidu Cloud face recognition SDK to let the virtual pet can recognize its owner and facial expression recognition component using Affectiva SDK to let the virtual pet can detect the owner emotion and based on the face expression to make some animation.

Responsibility of Team Member

1. Ooi Shion Yeing
    i) Graphical User Interface (GUI)
    ii) AR technology
    iii) Facial Expression Recognition
2. Lee Yen Long
    i) Artificial Intelligent (AI) engine
        - Smart Chat Component
        - Face Recognition Component

Deliverables

1. An AR smart pet game android application
2. Markerless AR application

1.4 Project Objectives

The project objectives of the proposed application are:

i. To develop a virtual pet application that let everyone can have a pet.
   Keeping a real pet need to spend more than RM 3000 per year (Emmanuel 2015), by using this virtual pet application, people can save a lot of money. Some students' families forbid them to keep pets, and also in university, most students' dormitory does not allow students to keep pets. So, they can keep a virtual pet. Next, the owner of virtual pet no need to spent a lot of time on their pet, especially for the working adults.

ii. To implement dual communication between the owner and the virtual pet using voice recognition.
   In order to let the smart pet application more intelligence, voice recognition will be implemented to allow the user to be able to talk to the virtual pet.

iii. To implement face recognition feature on the virtual pet.
   By implement the face recognition feature, the virtual pet will recognize its owner when the user interact with it and the virtual pet will more enthusiastic to the owner compare to strangers. This feature allows the virtual pet to look more similar to a real pet (Merola et al. 2012), it also allows user feel that the virtual pet is loyal to him.

1.5 Impact, Significance and Contribution

By the end of this project, the proposed app will able to help students pass the time. When students feel bored, they can launch this virtual pet application and interacting with the virtual pet. Besides, with the companionship of virtual pet, students are less prone to loneliness. This is because for students who lack of friends or trustworthy person talk to, the virtual pet can become the listener for them.

On the other hand, the proposed app also helps students who are not able to keep a real pet meet their wish to keep pets by keeping a virtual pet. The virtual pet will implement using AI techniques so it is smart and can give user an amazing experience.

1.6 Target Audience

The target audience of the proposed app is student. Based on (Harris 2015), 80% of the students own a smartphone which they use daily in school and home. Figure 1.6.1 shows the device usage of students.

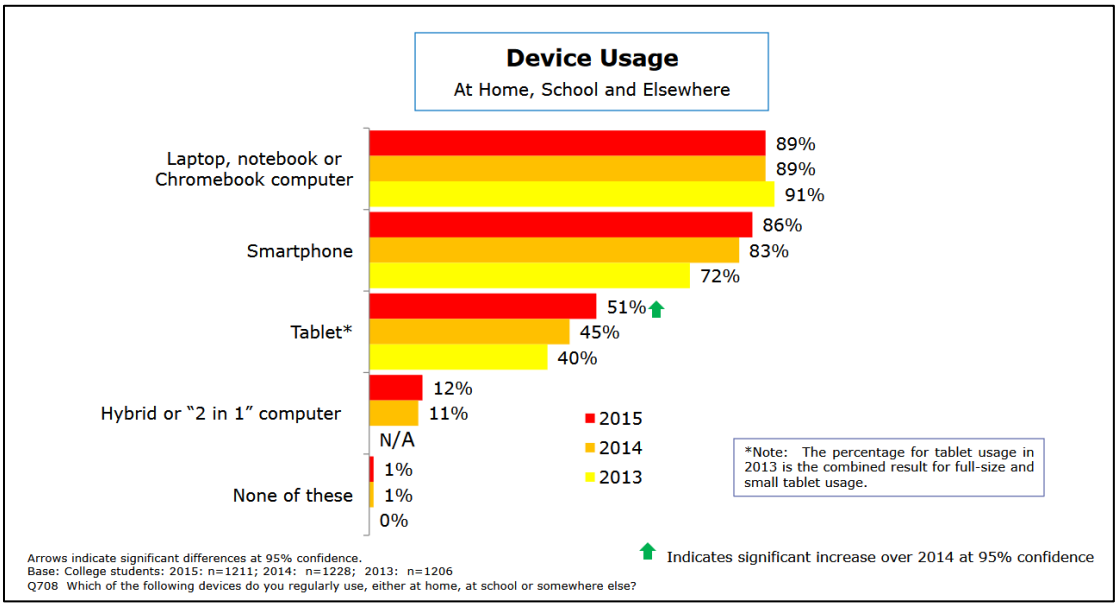


Figure 1.6.1 Device usage of students

The study proved that stress of a student should not be ignored. They have to find a way to release their stress. Next, students may be excluded from classmates at school. When they return home, their parents are less likely to communicate with them because they are busy with work. So, student need a pet to overcome their depression and loneliness. But, most of the students do not have the ability to keep a pet, thus the proposed app will be an alternative choice for the students.

1.7 Report organization

This report was divided into 6 chapter in total. The first chapter is the introduction of the project background, problem statement and motivation, project scope and objective, as well as the impact and target audience.

The second chapter of this report will review on the similar system and compare the previous work with our system. In chapter 3, describe the overall system design which included the proposed method and architecture to implement the proposed system.

In chapter 4, detailed steps of how the system be implemented is included. For chapter 5 is the testing and objectives evolution on the system.

The last chapter is the conclusion of the whole project and also the future improvement on the system.

Chapter 2: Literature Review

2.1 Review on Similar System - Smart Chat

2.1.1 Apple Siri

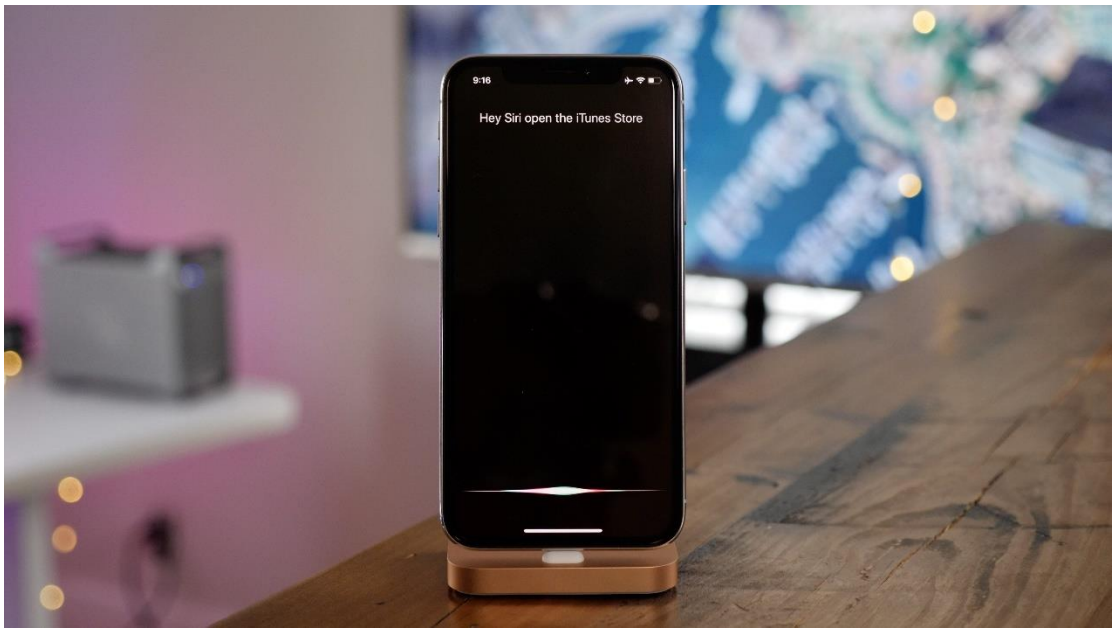


Figure 2.1.1 Siri running on iPhone x with iOS 12

Introduction

Siri (Speech Interpretation and Recognition Interface) is an artificial intelligent software built onto Apple's iOS and MacOS system. It uses Natural Language Processing (NLP) technology, allowing users to interact with their mobile phones using natural conversations, completing search, querying the weather, setting up a mobile calendar, setting alarms, and more (Apple 2018b).

Siri was originally a research project of the US Department of Defence's DARPA (Defence Advanced Research Projects Agency), which was positioned as a national-level virtual voice assistant (which may even become the default feature of Android smartphones, but was eventually sold to Apple). Siri is founded by Dag Kittlaus, Adam Cheyer, and Tom Gruber in year 2007. On April 28, 2010, Apple completed the acquisition of Siri (Scoble 2010). Prior to Apple's acquisition of Siri, Siri was an application in the iOS

App Store, and was also developed on the Blackberry and Android platforms; but after Apple completed the acquisition of Siri, Siri was removed from the App Store and all software developments except the iOS platform due to Apple introduced iPhone 4S which build in Siri (Kumparak 2011).

Siri's speech recognition engine is provided by Nuance Communications (Bostic 2013). The speech recognition systems used sophisticated machine learning techniques, including convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) (Levy 2016). LSTM is a units of recurrent neural network (RNN), due to its unique design structure, LSTM is suitable for processing and predicting important events with very long intervals and delays in time series (Hochreiter and Schmidhuber 1997).

The first Siri prototype was implemented using the Active platform (Guzzoni 2008), which is the focus of a PhD thesis led by Siri's chief scientist, Didier Guzzoni. The Active platform is also a joint project between the Artificial Intelligence Center of SRI International and the Vrai Group.

Why LSTM?

Long Short-Term Memory (LSTM) is an RNN similar with the hidden Markov model. But there are a significant different which is the construction and calculation on parameter (Eugine Kang 2017). The advantage of the LSTM is it is insensitive to the gap length. So, for other models, it will forget the starting point if the gap length is big, but for the LSTM, it will still remember. Although the LSTM is good, but it required more data to train compare to other models (Angus 2016).

Features

- Let user stay connected without lifting a finger.
- Find the song that user wants to hear.
- Control smart home devices.
- Can answer all kind of questions.

Strength

- Can perform basic task such as make a phone call without internet.
- Support up to 37 languages (Apple 2018a).

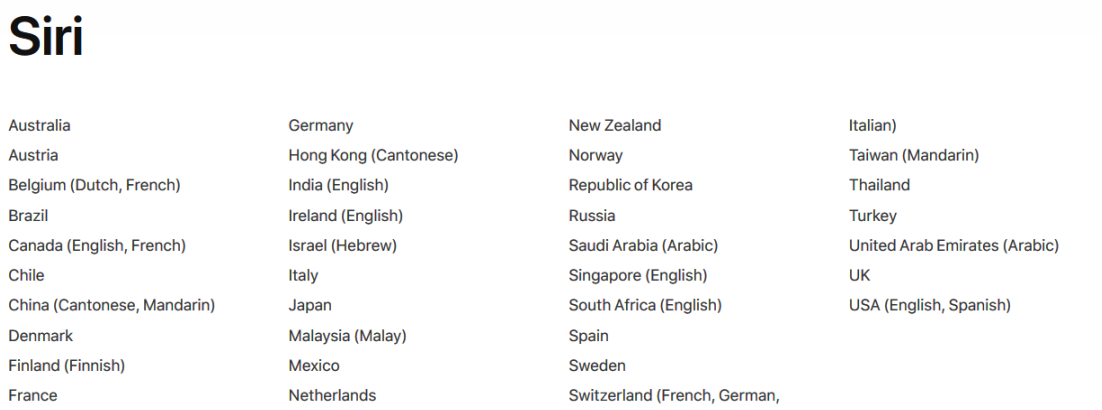


Figure 2.1.2 Languages supported by Siri

Weakness

- Only compatible with those devices running with iOS and MacOS

2.1.2 Xiaomi XiaoAi（小米小爱）

Figure 2.1.3 XiaoAi running on Mi 8 with MIUI 10

Introduction

XiaoAi is a smart personal assistant published by Xiaomi Technology on July 26, 2017. It is used for Xiaomi smart speaker and also Android smartphone running with MIUI (MI User Interface, a custom make OS based on Android) and is built on the database of Xiaomi Water Droplet Platform (小米水滴平台), now renamed as XiaoAi Open Platform (小爱开放平台). It can be directly linked to the MiJia Internet of Things (IoT) system. So, it is able to control smart furniture provided by Xiaomi, such as lamp, TVs, rice cooker and fan (HuiWen 2017).

XiaoAi uses the Automatic Speech Recognition (ASR), which is able to convert user’s voice into text through acoustic model, with recognition rate up to 97%. After recognised user’s voice content, same with Siri, XiaoAi is also using Natural Language Processing (NLP) technology to process user’s request on cloud (Xiaomi 2018b).

However, Xiaomi did not state that what is the model they used to train the XiaoAi personal assistant.

Features

- Let user stay connected without lifting a finger.
- Find the song that user wants to hear.
- Control smart home devices (MiJia IoT).
- Can answer all kind of questions.

Strength

- Can perform basic task such as make a phone call without internet.

Weakness

- XiaoAi can only understands Chinese. If user use the language interface other than Simplified Chinese to launch XiaoAi, system will prompt a message say that XiaoAi temporarily support only Chinese (Thorben 2018).

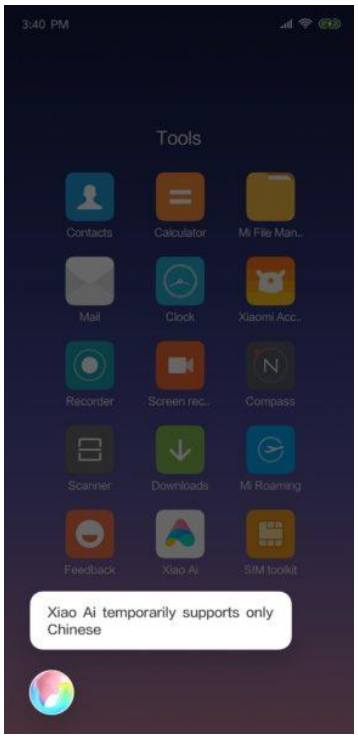


Figure 2.1.4 XiaoAi can only work in Chinese

- XiaoAi only compatible with those devices running with MIUI and smart furniture provided by Xiaomi. If somebody want to try XiaoAi on non-Xiaomi smartphone, they need to flash their smartphone with Xiaomi Flash Tool (Xiaomi 2018a).

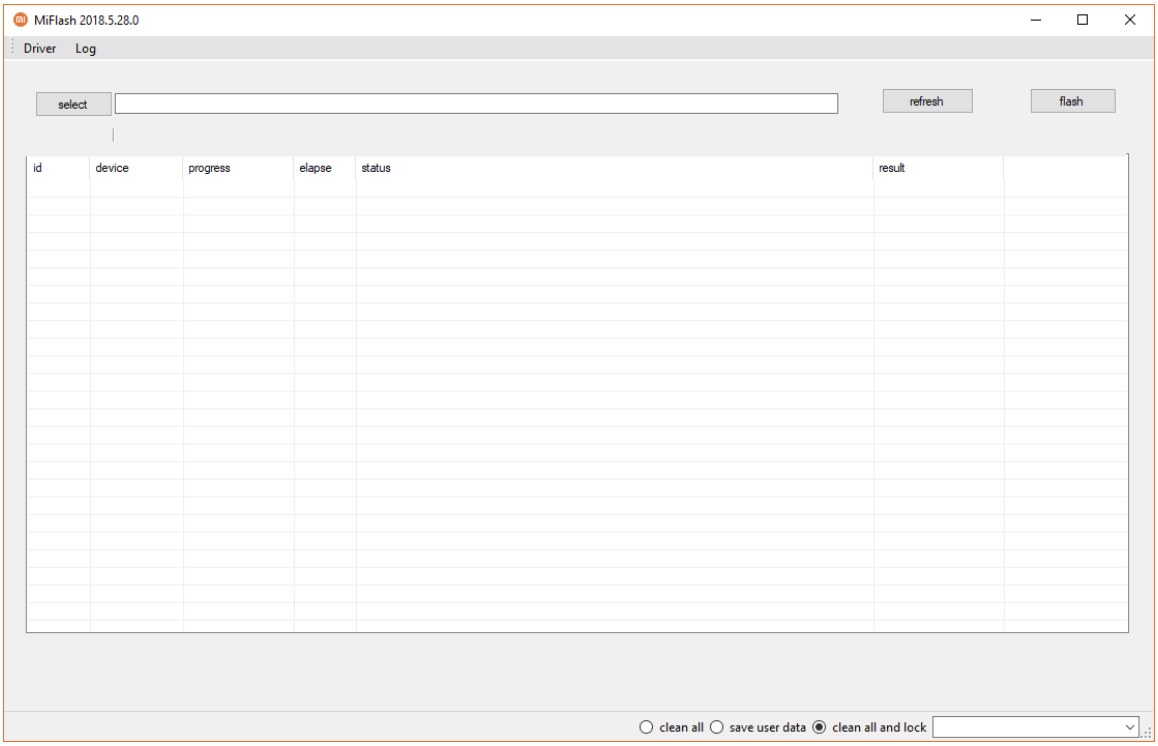


Figure 2.1.5 User Interface of Xiaomi Flash Tool

2.1.3 SimSimi

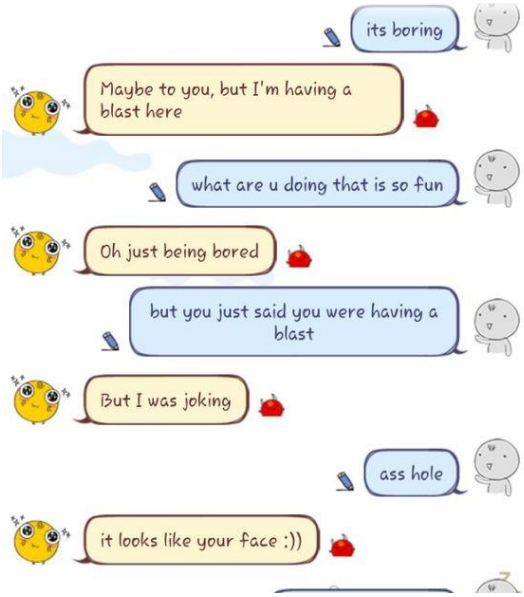


Figure 2.1.6 User Interface of SimSimi

Introduction

SimSimi is a popular artificial intelligence chatting program created by ISMaker in year of 2002 (Sheets 2012). Unlike Siri and XiaoAi, which is positioned as a smart personal assistant, SimSimi is closer to what we want to do, which is a smart pet that can interact with the user to relieve their stress. The name of SimSimi is from Korean word simsim (심심), which means “bored”.

Besides, developers are able to develop their own chatting robot using the conversation Application Programming Interface (API) provided by SimSimi at low cost (SimSimi 2013). Developer also can try the trial version of SimSimi API for one week.

Feature

- User can teach SimSimi answers to particular question, but because of this, user can also abuse this feature by teaching SimSimi some vulgar words.
- SimSimi has a filter for these vulgar words, user can decide the level of vulgar in their conversation.

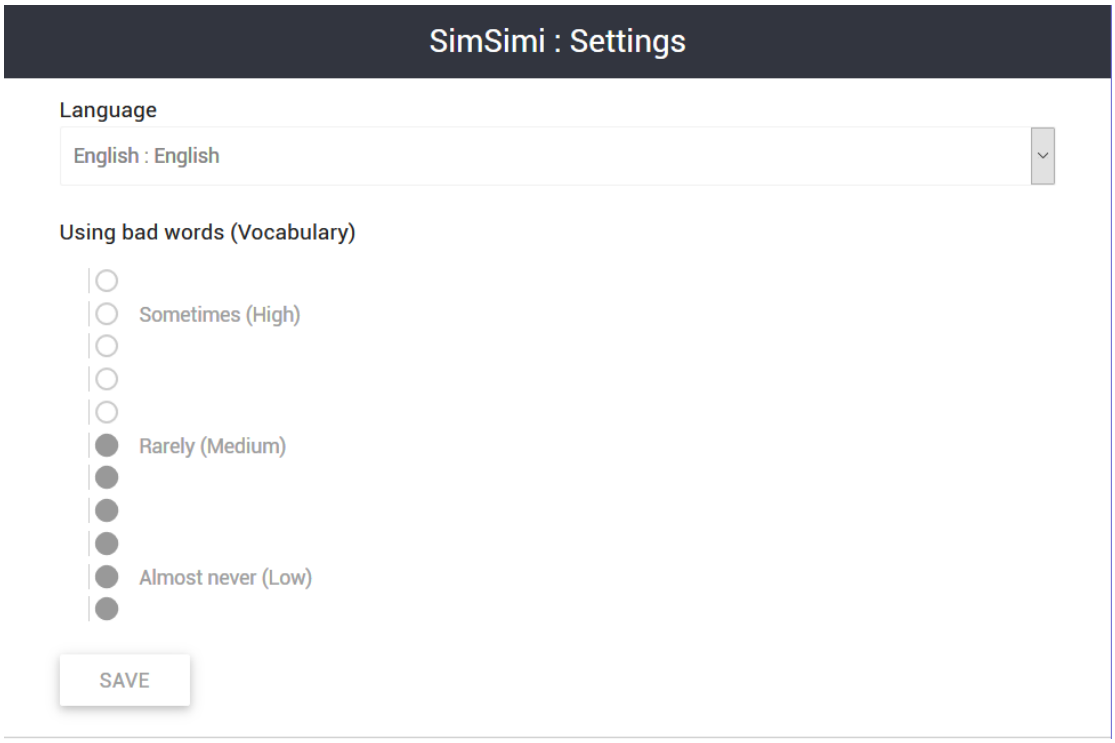


Figure 2.1.7 Setting page of SimSimi web version

Strength

- Support up to 81 languages.
- Support multi-platforms such as Android, iOS, Windows Phone and the web version.

Weakness

- Unable to interact with user via voice-recognition interface, user can only chat with SimSimi by typing.

2.1.4 Mitsuku

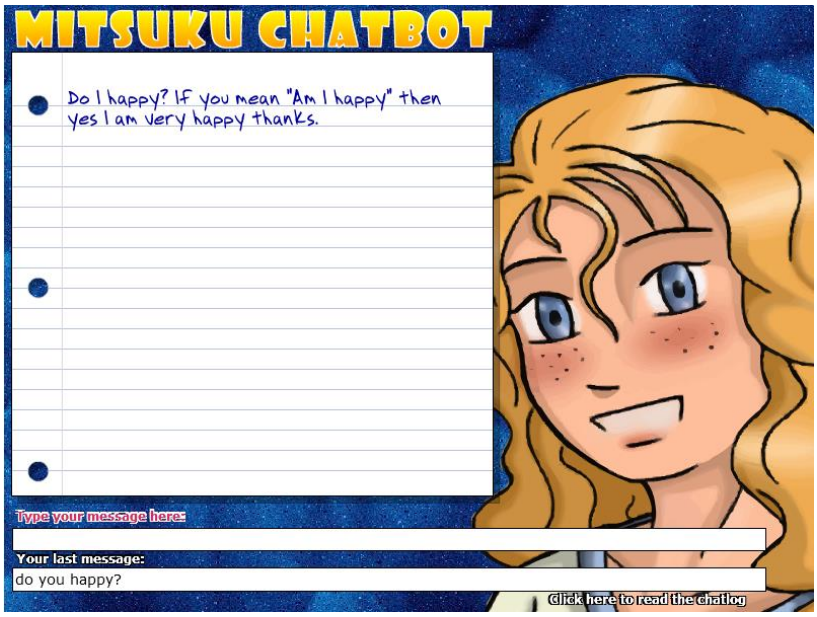


Figure 2.1.8 Mitsuku Chatbot flash version

Introduction

Mitsuku is a smart chatbot written by Artificial Intelligence Markup Language (AIML) created by Steve Worswick. It has won the Loebner Prize, which is awarded to the most "human-like" chatbot in the year of 2013, 2016 and 2017. In year 2018, Mitsuku lost to "Tutor" by 2 points and ranked second (AISB 2018).

Mitsuku claims to be a female chatbot from Leeds with the aged of 18. It took 9 to 10 years to develop. It consists of Artificial Linguistic Internet Computer Entity (Alice)'s AIML files. In addition, it also add in many user generated conversations that enable Mitsuku to reply accurately (Dreams 2013).

Mitsuku is not an application, it is just a chatting bot engine, it includes Mitsuku's Skype account and Messenger account, people can talk with Mitsuku by using those account. If someone just need to try Mitsuku without download anything, they can try the flash version of Mitsuku, the only tool needed is just a web browser. There are also Mitsuku IRL app which can be found on Google Play Store but not many downloads due to the flash version is more convenient for people to chat.

Why AIML?

AIML is an Extensible Markup Language (XML) file that encodes based on a list of Question-Answer rules. It is a kind of Rule-Based Closed-Domain (RBCD) Artificial Intelligence, developer can input the questions and specified its answers in the AIML script. So, the conversation flow and logic need to map very carefully.

The advantage of using AIML is the developer can specify the reply of the chatbot, so the chatbot can understand and reply the pre-defined question perfectly. But it may cause the chatbot cannot understand some complicated question which does not defined in the AIML file (Eric Arthur 2016). Another advantage of AIML is the developer does not need to prepare a large dataset to train the AI because it does not request much machine learning compare to other forms of AI.

Strength

- Mitsuku is the most human like chatbot among all the existing system (TechBubble 2016).
- Support multi-platforms.

Weakness

- Only support English.
- Unable to interact with user via voice-recognition interface, user can only chat with Mitsuku by typing.

2.1.5 Summary of Comparison Similar System - Smart Chat

The following table shown that the comparison between four similar system which is Apple Siri, Xiaomi XiaoAi, SimSimi and Mitsuku with proposed app.

Table 2.1.1 Comparison between similar system (Smart Chat).

Name / Features	**Apple Siri**	**Xiaomi XiaoAi**	**SimSimi**	**Mitsuku**	**Proposed App**
Platform Supported	iOS & MacOS	MIUI	Cross Platform	Cross Platform	Android
Interaction	Voice-based	Voice-based	Text-based	Text-based	Voice-based
Method / Features	**LSTM**		**AIML**		**Proposed App**
Conversation	Learn for a large dataset.		Defined in an XML file.		Learn for a large dataset.
Memory	Can reply base on previous question.		No memory on previous question.		Can reply base on previous question.
Answer Specific pre-defined Question	No		Yes		No

2.2 Review on Similar System - Face Recognition

2.2.1 IObit Applock

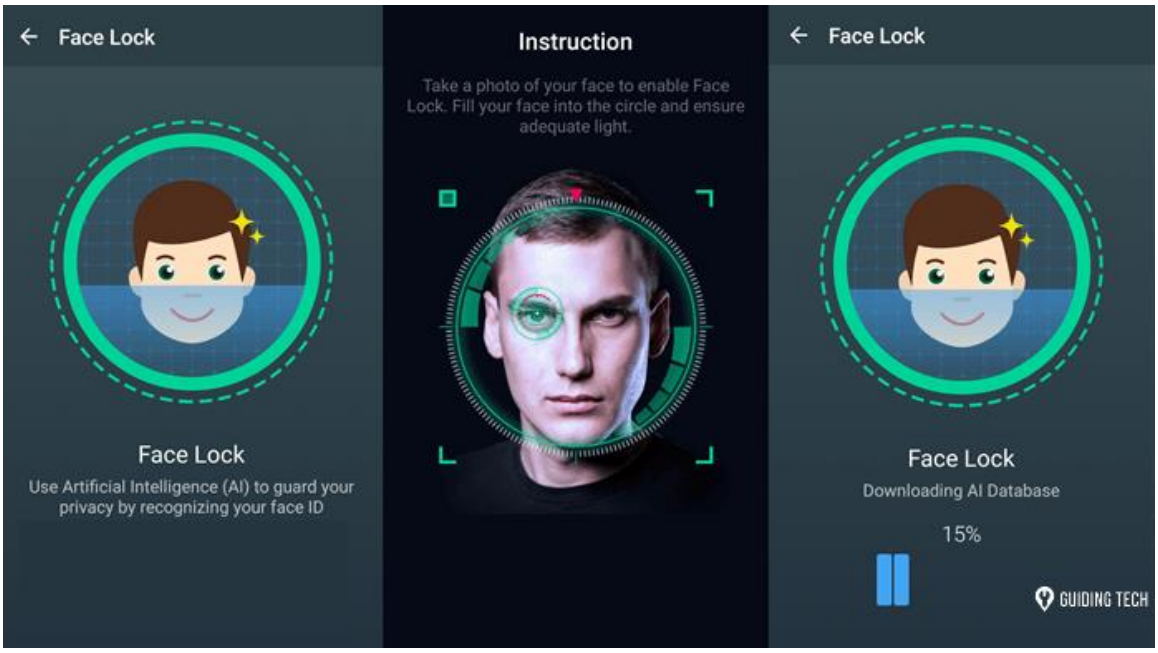


Figure 2.2.1 User Interface of IObit Applock

Introduction

IObit Applock is a security software for Android smartphone, it allows user to lock any application on their phone to avoid unauthorize people seeing their private data such as chat record. IObit Applock also user lock their phone settings and purchase feature in Google Play Store to avoid children accidentally messing up phone settings or buy somethings in from Google Play Store. If someone try to unlock the phone but failed up to 3 times, IObit Applock will take a photo of that people secretly and send it to phone owner's email address (IObit 2018).

Feature

- Face Lock – Lock apps and phone settings.
- Fake Lock – Up to 6 kinds of fake covers to let people hard to detect that they need to unlock the phone to used it. The fake cover can be just a photo chosen by user or an unknown incoming call cover to disguise locked apps.
- Notification Lock – Lock content of notification from social apps.
- Intruder Selfie – Take a photo of who failed to unlock up to 3 times.

Strength

- Face recognition without show the camera display.
- Fast recognition speed.

Weakness

- Only compatible with Android devices which version is higher than Android 4.0.3.
- Need to purchase Pro version to unlock features other than face lock.

2.2.2 Selphi

Figure 2.2.2 User Interface of Selphi iOS version

Introduction

Selphi is a software framework designed by Facephi to let banks verify their clients when they are preforming online banking. Selphi can integrate with other application such as bank's mobile app, so the mobile app can recognise client's face by take some picture of them with their smartphone. Not only that, Selphi also able to learn user's face when user using it, this will help to improve the recognition accuracy and speed. Because of the high accuracy and recognition speed, most American banks and government agencies have chosen Selphi become their partners (SelPhi 2018).

Feature

- Able to recognize user's face even in darkness by using the light from smartphone screen.
- Provide anti-spoofing feature by asking user perform certain action such as blink their eyes when recognising their face.

Strength

- Support multi-platform such as iOS, Android and HTML5.
- Learn user's face when user using it.
- Fast recognition speed.

Weakness

- Expensive.

2.2.3 Summary of Comparison Similar System – Face Recognition

The following table shown that the comparison between two similar system which is IObit Applock, Selphi with proposed app.

Table 2.2.1 Comparison between similar system (Face Recognition)

Name / Features	IObit Applock	Selphi	Proposed App
Platform Supported	Android	Cross Platform	Android
Face Recognition Without Show the Camera Display	Yes	No	No
Learn User's Face When Using	No	Yes	No
Completely Free	No	No	Yes

Chapter 3: System Design

3.1 Design Specifications

This propose app would be a native mobile app which build for Android. It is because the market share of Android is larger than others mobile operating system (Merola et al. 2012).

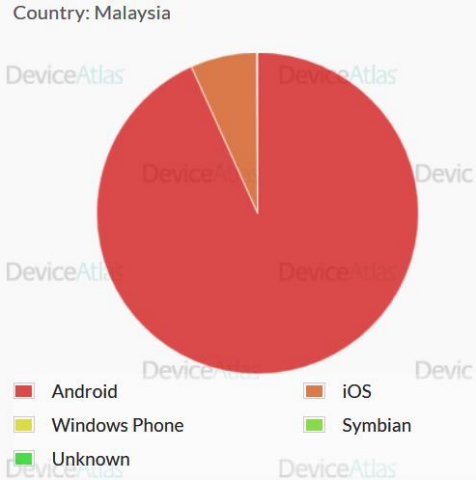


Figure 3.1.1 Mobile operating system's market share in Malaysia

The AI engine of proposed app is implemented in Python language with open source software library, TensorFlow Sequence-to-Sequence and TensorFlow Object Detection API with TensorFlow backend which can use Graphics Processing Unit (GPU) to perform acceleration during the training process.

TensorFlow is an open-source software library developed by Google Brain for machine learning which support Python, C++, Java and Go programming language (Abadi et al. 2016). But the most convenient way to do machine learning is using Python language, so AI engine of the proposed app will be using Python as well.

The training process is done on Google Could Platform (GCP) and accelerate by using cloud virtual compute engine with dedicated Nvidia Tesla K80 GPU with 16 GB of GDDR5X of video Random Access Memory (vRAM).

3.1.1 Proposed Method

Based on the project scope that we have mention, the module in the proposed app as Figure 3.1.2.

Figure 3.1.2 Modules in Smart Pet Game Application

In this report, we will focus on Smart Chat module and Face Recognition module.

3.1.2 Methodology

Instead of using traditional software development lifecycle (SDLC), the proposed app will follow the machine learning model for the development. The processes of machine learning (ML) model consists of problem and goals definition, data collection, data preparation, model learning, model deployment and integration, and model management (Thanaki 2018).

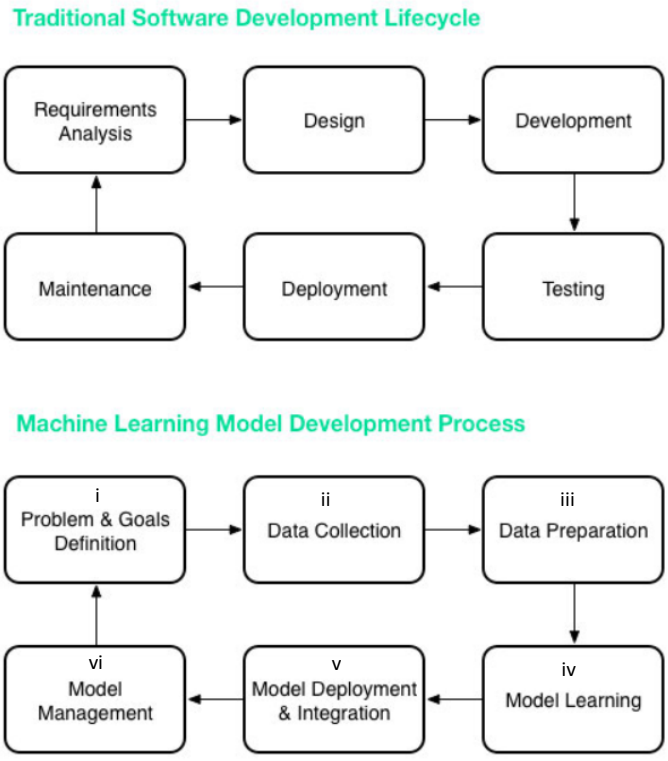


Figure 3.1.3 SDLC vs ML Model Development Process

Figure 3.1.2 illustrates the process of traditional software development lifecycle and machine learning model development process. By using traditional SDLC, developer need to design the system flow based on the requirements before develop the system. After finish the development process, developer will use some software testing method such as black box testing to ensure the correctness of software logic. By using ML model development process, developer need to collect data based on the problem definition. Next, developer will use the collected data to train the model. After finish the training process, the trained model will integration with other app to perform its function.

3.2 System Design / Overview

There are two parts for the implementation of AI engines:

3.2.1 Smart Chat Engine

3.2.2 Face recognition component

These are explained in detailed using the machine learning model development process in the following section.

3.2.1 Smart Chat Engine

Smart chat engine allows virtual pet to interact with the owner. There are several components in this engine:

a. Speech-to-Text (STT)

STT is used to convert user's voice into corresponding text, it also known as Automatic Speech Recognition (ASR). We need to train the speech recognition network before using it. The TED-LIUM form OpenSLR was selected to become the training dataset. The TED-LIUM dataset is provided by Laboratory of Informatics, University of Le Mans (LIUM), which is a 21GB voice data from TED talks (Rousseau et al. 2012).

b. Natural Language Processing (NLP)

After the conversion, NLP is used to allow the program to understand the input text. But before that, a suitable model is selected to perform deep learning together with NLP. The sequence-to-sequence (Seq2Seq) model is the most suitable model to be used in chatbot. It uses variant of this model (tf-seq2seq) to perform deep learning.

c. Text-to-Speech (TTS)

TTS is used to convert back the output text into voice. The Google Cloud Text-to-Speech provided by Google will use in this section.

a. Speech-to-Text (STT)

ii. Data Collection & iii. Data Preparation

The Speech-to-Text (STT) model was trained using the TensorFlow deep learning framework, Seq2Seq model neuron network. The following data is selected to become the training dataset:

1. TED-LIUM form OpenSLR (Rousseau et al. 2012).

The TED-LIUM is a 21GB voice recognition training data from TED talk and it is created by Laboratory of Informatics, University of Le Mans (LIUM).

iv. Model Learning

Recurrent Neural Network (RNNs) was selected to become the training model. The input of the model is a sequence of waves and the output is a sequence of words.

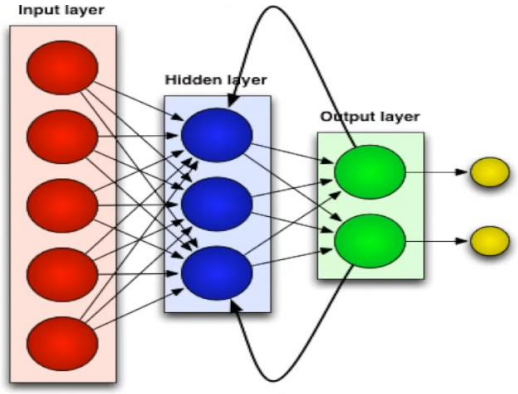


Figure 3.2.1 Recurrent Neural Network (RNN)

v. Model Deployment and Integration

When the model is trained, it will integrate with the Smart Pet App using Unity ML Agents toolkit. Unity is written in C# language, but TensorFlow cannot provide a local C# API, so the TensorFlow Sharp will be used to complete the integration.

b. Natural Language Processing (NLP)

ii. Data Collection & iii. Data Preparation

Seq2Seq model requires a large number of conversation logs in order to train the system. So, during data collection and data preparation phase, there are three datasets as the data source:

1. Cornell Movie Dialog Corpus (Mizil 2018) (Main Source).
2. The Ubuntu corpus (Lowe 2018).
3. Microsoft's Social Media Conversation Corpus (Sordoni 2016).

will used for the training phase.

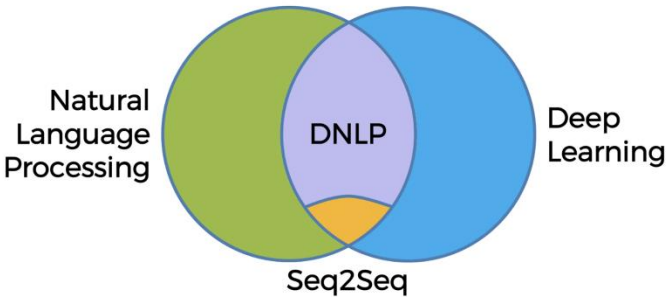


Figure 3.2.2 Seq2Seq model for deep learning with NLP

iv. Model Learning

The Seq2Seq model consists of two Recurrent Neural Networks (RNNs):

1. Encoder RNN that process the input.
2. Decoder RNN that generate the output (Cho et al. 2014).

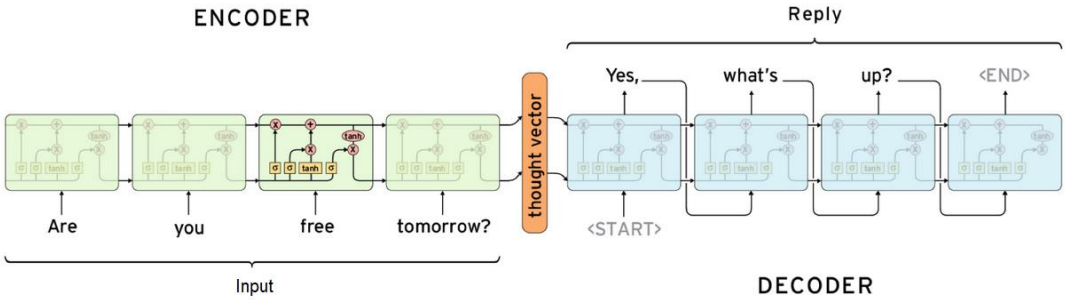


Figure 3.2.3 Encoder RNN and Decoder RNN

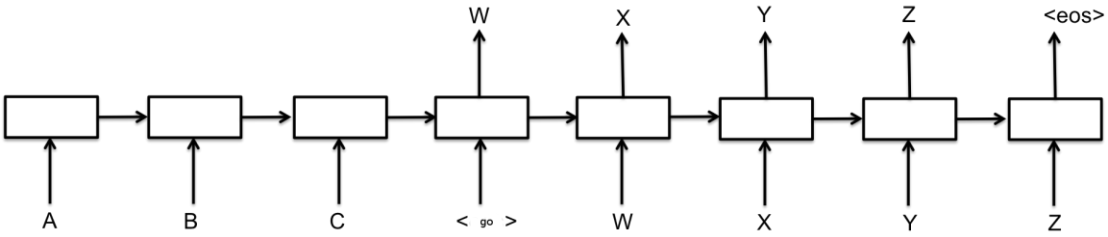


Figure 3.2.4 Basic architecture of Seq2Seq model

Figure 3.2.3 shows that the basic architecture of Seq2Seq model, each rectangle represents a cell of RNN, most commonly called Gated Recurrent Units (GRU) cell or Long Short-Term Memory (LSTM) cell. The goal of the LSTM is to estimate the following conditional probability (Deshpande 2017):

$y_1, \ldots, y_{T'}$ = Output Sequence

$x_1, \ldots, x_T$ = Input Sequence

$\vec{v}$ = Vector Representation

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} P(y_t | \vec{v}, y_1, \ldots, y_{t-1})$$

(Left) (Right)

The left side of the equation is probability of the output sequence under the condition of the input sequence. The right side of the equation is the multiplication of vector of probabilities of all the words under condition of vector representation and output from the previous iteration (*t - 1*).

For example, the input of “Are you free tomorrow?” shown in Figure 3.2.2 will lead to an answer “Yes”, “Yeah” or “No”. After the training for RNN is completed, the probability $p(y_1|\vec{v})$ will be resulting as follows:

$p(y_1|\vec{v})$ = [0 0.01 0 0.40 0.25 0.34]

Apple … But … Dog … No … Yeah … Yes

The probability is produced from the trained model based on training dataset.

The word ‘No’ has the highest probability, and thus this word will be the first unit in the decoder RNN.

As compared to the traditional machine learning method (e.g.: SVMs, linear regression) and deep learning method (e.g.: Convolutional Neural Networks (CNNs)) require fix input size and the fix output size. In other words, the length of the input must be set beforehand. Hence, these methods are not suitable to be used for the unknown input size and to generate the variable length output, such as speech recognition. On the other hand, Seq2Seq model allows flexibility, thus, the training for the smart chat will use Seq2Seq model in TensorFlow.

c. Text-to-Speech (TTS)

After that, the model should be able to communicate with people by replying a text message. Text-to-Speech (TTS) is implemented to convert the output text back to a sound wave. In this phase, Google Cloud Text-to-Speech API is selected because it have the WaveNet neural network which able to generate a more natural sound that similar to human speak (Oord et al. 2016).

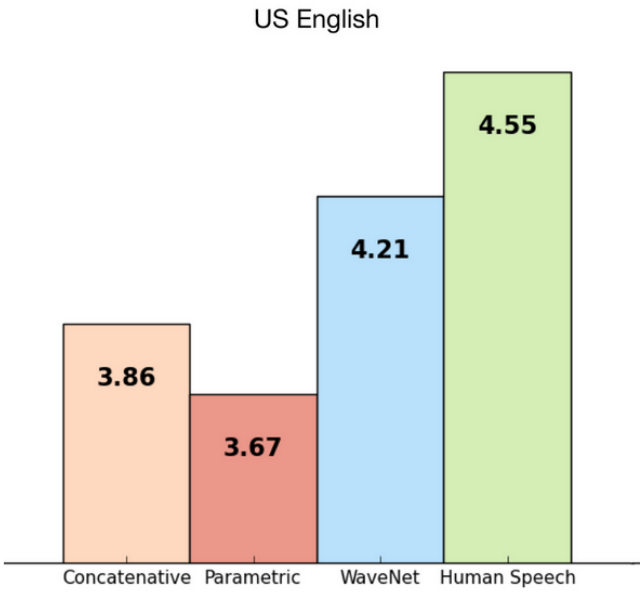


Figure 3.2.5 Voice Generated by WaveNet is more similar to human speech

v. Model Deployment and Integration

After the model training frame, the system is done but it is only in computer. Unity ML Agents toolkits will be used to integrate the Smart Chat Engine with the main Smart Pet App, which is written in C# using unity. All the model was trained in TensorFlow, but TensorFlow cannot provide a local C# API, in the older version of Unity ML Agents, TensorFlow Sharp will be used to complete the integration. Starting from version 0.7, Unity Inference Engine can let the Unity ML Agents directly communicate with the TensorFlow model, and it is included by default.

When the model integrated with the Smart Pet App, it should able to run on Android platform.

3.2.2 Face Recognition

The face recognition engine allows virtual pet to recognize the owner. Baidu Cloud face recognition SDK is selected to perform face recognition in Unity. The reason why we did not use a machine learning method on face recognition is, the Smart Pet App is written in Unity, but Unity does not allow perform model training on Android, it only can integrate a trained model with the mobile app. If we select a machine learning method such as using TensorFlow model, the model can only be trained in PC, and it will only able to recognize the people during training.

ii. Data Collection & iii. Data Preparation

First, the system needs to capture user's face from their smartphone's secondary camera (front camera) to become the training dataset. 10 pictures are sufficient for the training phase to enable the virtual pet to recognize its owner because it just need to learn one people. These pictures required is better to captured from different angle to let the Smart Pet easy to recognise its owner. All the taking pictures will upload and stored in face library on Baidu Cloud.

iv. Model Learning & v. Model Deployment and Integration

After the training images uploaded to the face library on Baidu Cloud, it will automatically start the training. When a person interacts with the virtual pet, the app will capture the user's face from camera and search it on the face library to compare the similarity of that person with the owner. If the similarity is more than the threshold, the person will be verified as the owner. Also, Baidu Cloud face recognition SDK are able to check whether the user is a living person before recognize its face.

3.3 Implementation Issues and Challenges

1. Difficult to get a high-performance machine for training the model.

   Training a model for Artificial Intelligence required high-performance machine with dedicated GPU to speed up the process. But a high-end GPU is very expensive, it normally cost few thousand ringgits.

   Solution:

   Using compute engine (server) on Google Cloud Platform (GCP) to train the model. GCP is a platform provided by Google which allows developer rent a server from them, every new user of GCP can get a free USD 300 credit for one year. By using GCP, we can select some high-end GPU such as Nvidia Tesla K80 to perform acceleration on training process.

2. Difficult to set up the software environment.

   After get a compute engine on GCP, the setup of software environment such as install Nvidia CUDA graphic driver, install Jupyter Notebook, install TensorFlow also take a lot of time. This is because GCP do not provide GUI on compute engine, so we need to install that software by Linux command and also set up the firewall rules to let the compute engine can connect from local PC.

3. Difficult to combine the trained model with Android App

   After all the TensorFlow model has been trained, we found that it is difficult to integrate it with the Smart Pet App which created using Unity.

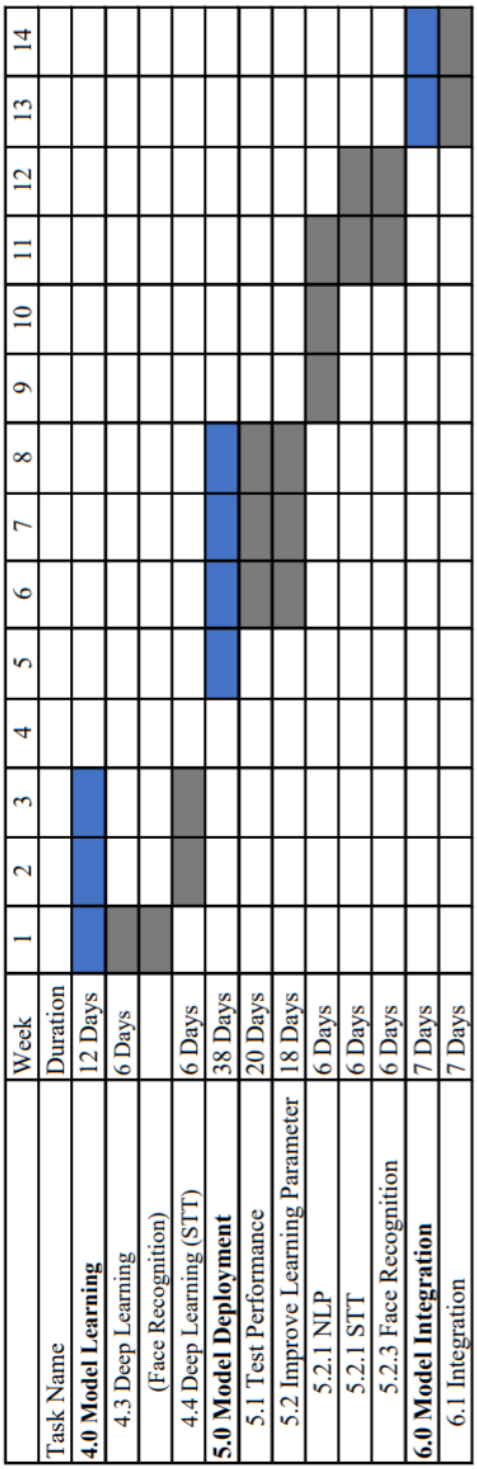
   Solution:

   Unity Inference Engine in Unity ML Agents is used to help the model integration.

3.4 Timeline

	Week	1	2	3	4	5	6
Task Name	Duration						
1.0 Problem & Goal Definition	9 Days						
1.1 Overview	1 Day						
1.2 Identify Problem Statements	1 Day						
1.3 Identify Objectives	1 Day						
1.4 Literature Review	2 Days						
1.4.1 Apple Siri							
1.4.2 Xiaomi XiaoAi							
1.4.3 SimSimi							
1.4.4 Mitsuku							
1.4.5 Iobit Applock							
1.4.4 Selphi							
1.5 Machine Learning Model Development	2 Days						
1.5.1 Problem & Goal Definition							
1.5.2 Data Collection							
1.5.3 Data Preparation							
1.5.4 Model Learning							
1.5.5 Model Deployment							
1.5.6 Model Integration							
1.6 System Design	2 Days						
1.6.1 Smart Chat Engine	1 Day						
1.6.2 Face Recognition	1 Day						
2.0 Data Collection	2 Days						
2.1 Collect Data (STT)	1 Day						
2.2 Collect Data (NLP)	1 Day						
3.0 Data Preparation	3 Days						
3.1 Create BPE Encoding & Dictionaries	3 Days						
4.0 Model Learning	10 Days						
4.1 Setup Software Environment	5 Days						
4.2 Deep Learning (NLP)	5 Days						

FYP1


Week
Duration
Task Name
4.0 Model Learning
12 Days
4.3 Deep Learning
6 Days
(Face Recognition)
4.4 Deep Learning (STT)
6 Days
5.0 Model Deployment
38 Days
5.1 Test Performance
20 Days
5.2 Improve Learning Parameter
18 Days
5.2.1 NLP
6 Days
5.2.1 STT
6 Days
5.2.3 Face Recognition
6 Days
6.0 Model Integration
7 Days
6.1 Integration
7 Days
1
2
3
4
5
6
7
8
9
10
11
12
13
14

FYP2

Chapter 4: System Implementation

4.1 Software implementation

The following table shows the software tools for the development of our proposed app:

Table 4.1.1 Software tools for development

Particulars	Software Tools	Logo
Operating System	GCP server: Ubuntu 18.04 (for model training)	
	Microsoft Windows 10 (for Android development)	
	Android (for app testing)	
Integrated Development Environment (IDE)	Anaconda Jupiter Notebook (for model training)	
	Unity (for model integration) - Android Software Development Kit (SDK) - Unity ML Agents	
Programming Languages	Python (for model training)	
	C# (for model integration) - Unity Inference Engine	
	Bash shell script (to process file in Ubuntu)	
Libraries, API and other tools	Google TensorFlow	
	Baidu Cloud face recognition SDK	
	Google Cloud Text-to-Speech API	

4.1.1 Environment Setup

Google Cloud Platform (GCP) was selected to become the model training platform of the system. During the model training, dozens GB of data will be used to train the model, it may consume a lot of time, so we need to use a powerful GPU to perform acceleration on training phase. Compute Engine on Google Cloud Platform provide some high-end GPU, such as Nvidia Tesla K80, it can let the model training become faster.

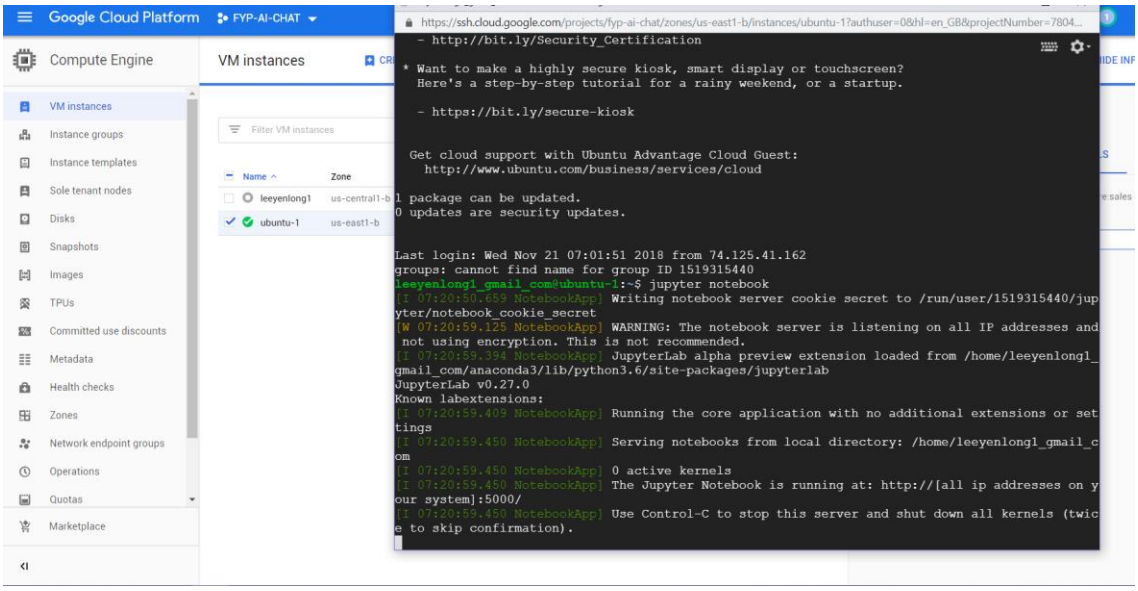


Figure 4.1.1 Connect to compute engine in GCP

After get a compute engine on GCP, we need to setup the software environment such as install Nvidia CUDA graphic driver, install Jupyter Notebook, install TensorFlow and all the necessary package. GCP do not provide GUI on compute engine, so we need to install all the necessary software by Linux command, it is very inconvenience if we need to use command to access source code file and data file on GCP, so VNC server is selected to enable the GUI on GCP, VNC server will be installed on the compute engine and VNC will be installed on local PC.

Although the compute engine was running at the cloud, but we still can use the browser on local PC to connect to Jupyter Notebook on the cloud. We need to set up the firewall rules to let the Jupyter Notebook on compute engine can connect from local PC and also other rules to let the VNC client on local PC able to connect to VNC server on GCP to enable the GUI.

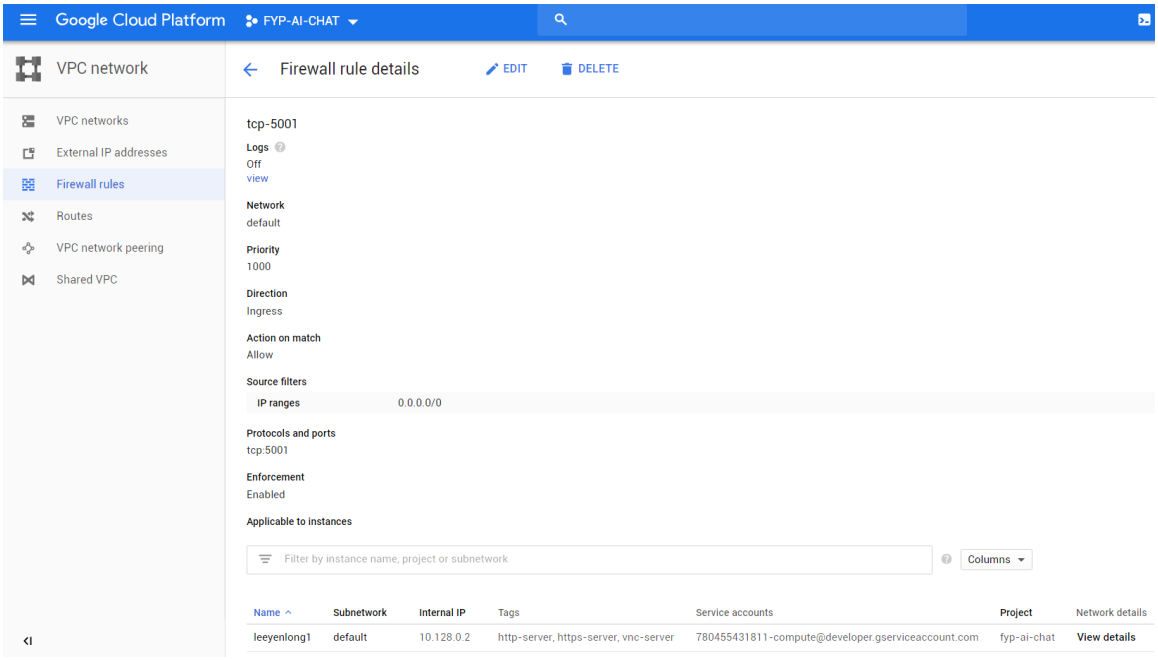


Figure 4.1.2 Firewall rules that allows access Jupyter Notebook on the GCP from local PC

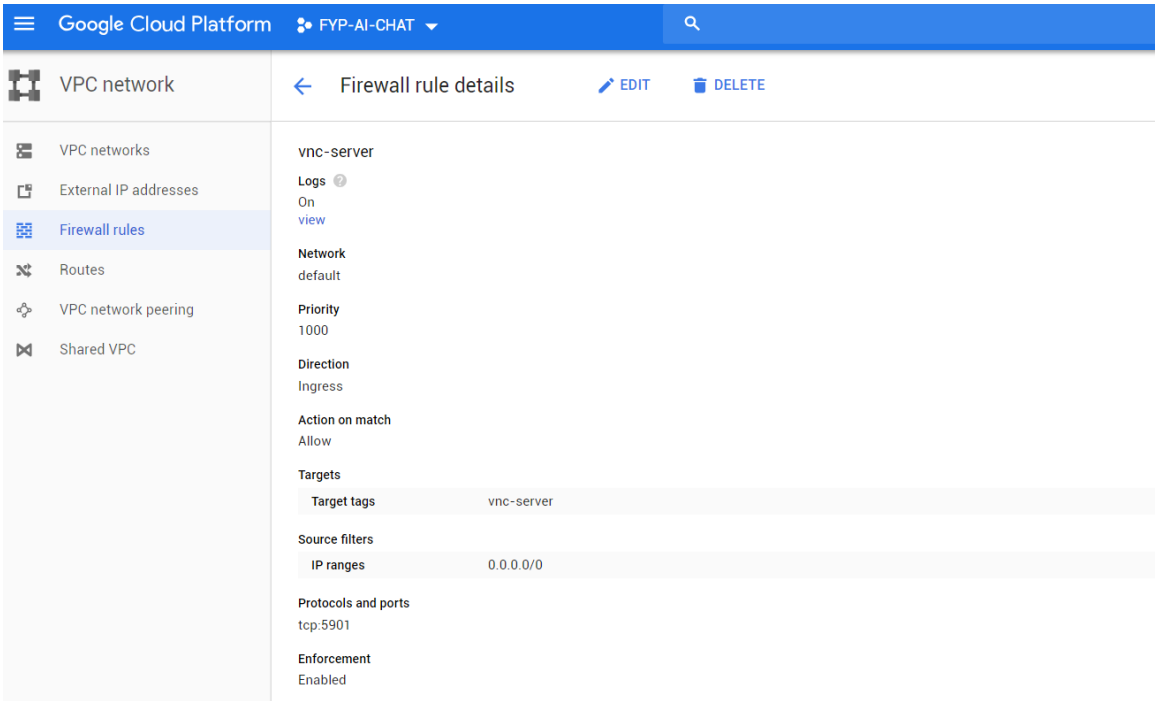


Figure 4.1.3 Firewall rules that open VNC server port

After setting up the firewall rules, the compute engine should able to connect from local PC.

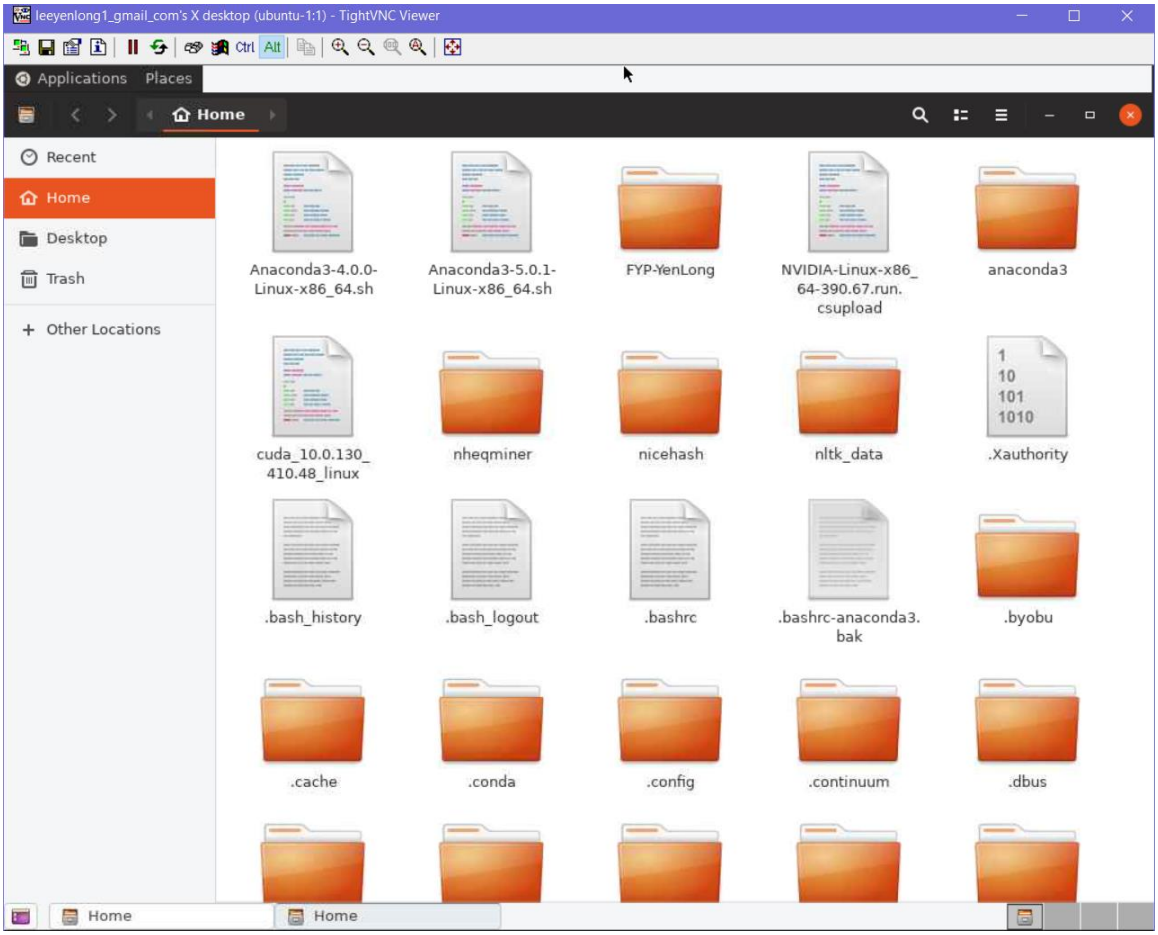


Figure 4.1.4 GUI enable by VNC server

After the VNC port (tcp: 5901) was opened using the firewall rules, by using VNC client (TightVNC Viewer) on local PC, we can connect to VNC server on the compute engine and enable GUI. The GUI will help us to control the GCP compute engine easier.

4.2 Deep Learning for Voice Recognition

4.2.1 Speech-to-Text

The source code of Speech-to-Text model training will be written in two files:

1. speech_data.py: To handle the training data preparation, such as define the path of the data.
2. training.py: Build the RNN and train the voice data.

The training dataset (TED-LIUM) was downloaded and saved into /data/ path, before the training start, system will check the dataset is exist or not using check_dataset_exist(file, dir) method, if the files not exist, the system will download it from internet.

Before the model training, the dataset will be separated into two set to avoid the trained model overfit the dataset, first set (90% of data) is the training set and the second (10% of data) is testing set, which is defined in speech_data.py.

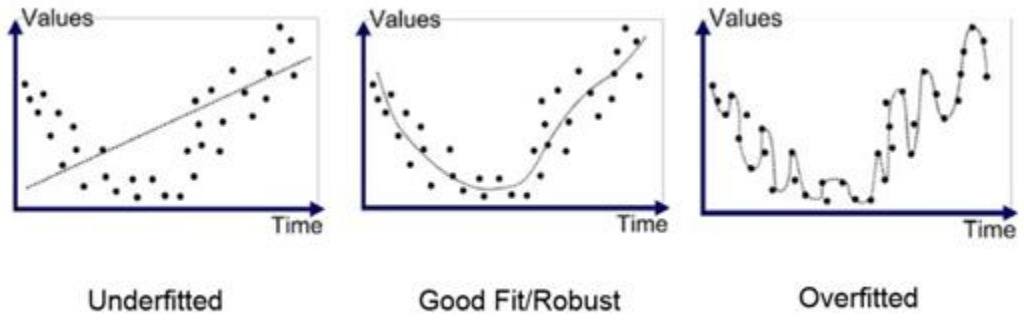


Figure 4.2.1 Underfitting, Good Fit and Overfitting

The training phase will let the model try to fit the dataset with minimizing the error rate. In the leftmost graph of Figure 4.2.1, the model has not any idea on the pattern of data, which call underfitting. The middle graph of the figure shows the model fit the training data with a just right pattern, that is what we want. In the third graph, the model found a perfect pattern match the training data, but it is just a kind of memorizing the dataset, the overfitting model can achieve a 100% accuracy on the training data, but it cannot perform better than the middle one in newer data (Dietterich 2002).

That is one of the reasons why the training data need to split into training set and test set. After training with 90% of the training data, the model will test on the fresh data (test set) so that it will reduce the overfitting problems. Besides, the performance of the model when it facing a new input also can be estimated.

Inside the training.py, learning rate of the model will be set, learning rate is a fixed constant that use to control the magnitude (how big) of the move. If the learning rate is large, it cannot reach the true local minima because update is too coarse, so a small learning rate will let the optimizer function reach the local minima. But if the learning rate is too small, it will consume a lot of time to reach the local minima.

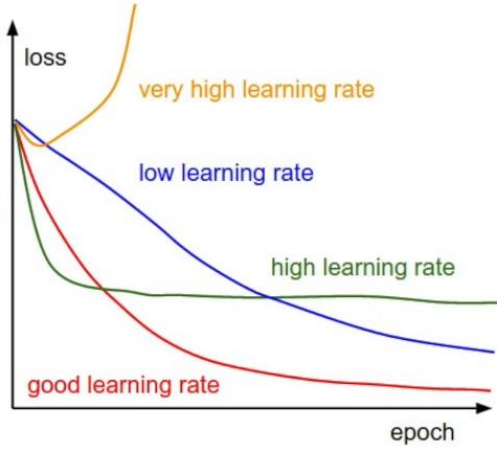


Figure 4.2.2 Relation between learning rate and the number of epochs needed to minimize the loss

The number of steps that need to train (training_iters = 300000) and the number of data need to train on every batch (batch_size = 64) are also defined in training.py. After that, a LSTM recurrent neural network with 128 neurons and dropout value of 0.8, too few neurons will lead to a bad prediction and too many will overfitting the training data, dropout helps prevent overfitting by randomly turning off some neurons during training, so data is forced to find new paths between layers to allowing for more generalized model.

```
19 # Network building
20 net = tflearn.input_data([None, width, height])
21 net = tflearn.lstm(net, 128, dropout=0.8)
22 net = tflearn.fully_connected(net, classes, activation='softmax')
23 net = tflearn.regression(net, optimizer='adam', learning_rate=learning_rate, loss='categorical_crossentropy')
```

Figure 4.2.3 Building the neural network for training

After that, the training loop will be initialized to fit the model to the training and testing data set for 10 epochs with the specified batch size (64). Last, save the trained model for later use.

```
model = tflearn.DNN(net, tensorboard_verbose=0)
while True: #training_iters
  model.fit(trainX, trainY, n_epoch=10, validation_set=(testX, testY), show_metric=True,
          batch_size=batch_size)
  _y=model.predict(X)
model.save("tflearn.lstm.model")
```

Figure 4.2.4 Training phase of Speech-to-Text model

4.2.2 Natural Language Processing

Cornell Movie Dialogs Corpus is selected to become the dataset to train the Smart Chat engine. Similar with the STT model, we need to separate the training set (80%) and testing set (20%) to avoid overfitting. After that, the learning rate, batch size, dropout value and other hyperparameters will defined in a file.

A prepossessing on the Seq2Seq model is needed before the training start. A Seq2Seq data should similar to following:

```
input_seqs : ['how', 'are', 'you', '<PAD_ID'>]
decode_seqs : ['<START_ID>', 'I', 'am', 'fine', '<PAD_ID'>]
target_seqs : ['I', 'am', 'fine', '<END_ID>', '<PAD_ID'>]
target_mask : [1, 1, 1, 1, 0]
```

```
# Training Data Placeholders
encode_seqs = tf.placeholder(dtype=tf.int64, shape=[batch_size, None], name="encode_seqs")
decode_seqs = tf.placeholder(dtype=tf.int64, shape=[batch_size, None], name="decode_seqs")
target_seqs = tf.placeholder(dtype=tf.int64, shape=[batch_size, None], name="target_seqs")
target_mask = tf.placeholder(dtype=tf.int64, shape=[batch_size, None], name="target_mask")

net_out, _ = create_model(encode_seqs, decode_seqs, src_vocab_size, emb_dim, is_train=True, reuse=False)
```

Figure 4.2.5 Preprocessing on training data placeholders and create the Seq2Seq model

The create_model function in Figure 4.2.5 is defined to create a LSTM Seq2Seq model. After building the model, the training can be started.

```
Initializing training...
Shared Vocab size: 30291
Epochs: 500
Batch Size: 128
Epoch:   1/500, Batch:  100/1558, Stats for last 100 batches: (Training Loss: 24.289, Training Time: 74 seconds)
Epoch:   1/500, Batch:  200/1558, Stats for last 100 batches: (Training Loss: 15.931, Training Time: 73 seconds)
Epoch:   1/500, Batch:  300/1558, Stats for last 100 batches: (Training Loss:  7.390, Training Time: 74 seconds)
Epoch:   1/500, Batch:  400/1558, Stats for last 100 batches: (Training Loss:  6.224, Training Time: 79 seconds)
Epoch:   1/500, Batch:  500/1558, Stats for last 100 batches: (Training Loss:  6.040, Training Time: 80 seconds)
Epoch:   1/500, Batch:  600/1558, Stats for last 100 batches: (Training Loss:  5.931, Training Time: 80 seconds)
Epoch:   1/500, Batch:  700/1558, Stats for last 100 batches: (Training Loss:  5.890, Training Time: 81 seconds)
```

Figure 4.2.6 Model is training

TensorBoard can be used for visualizing the process while the Seq2Seq model is being trained. Inside the “projector” tab of TensorBoard, the grouping of words can be explored by viewing nearest neighbors.

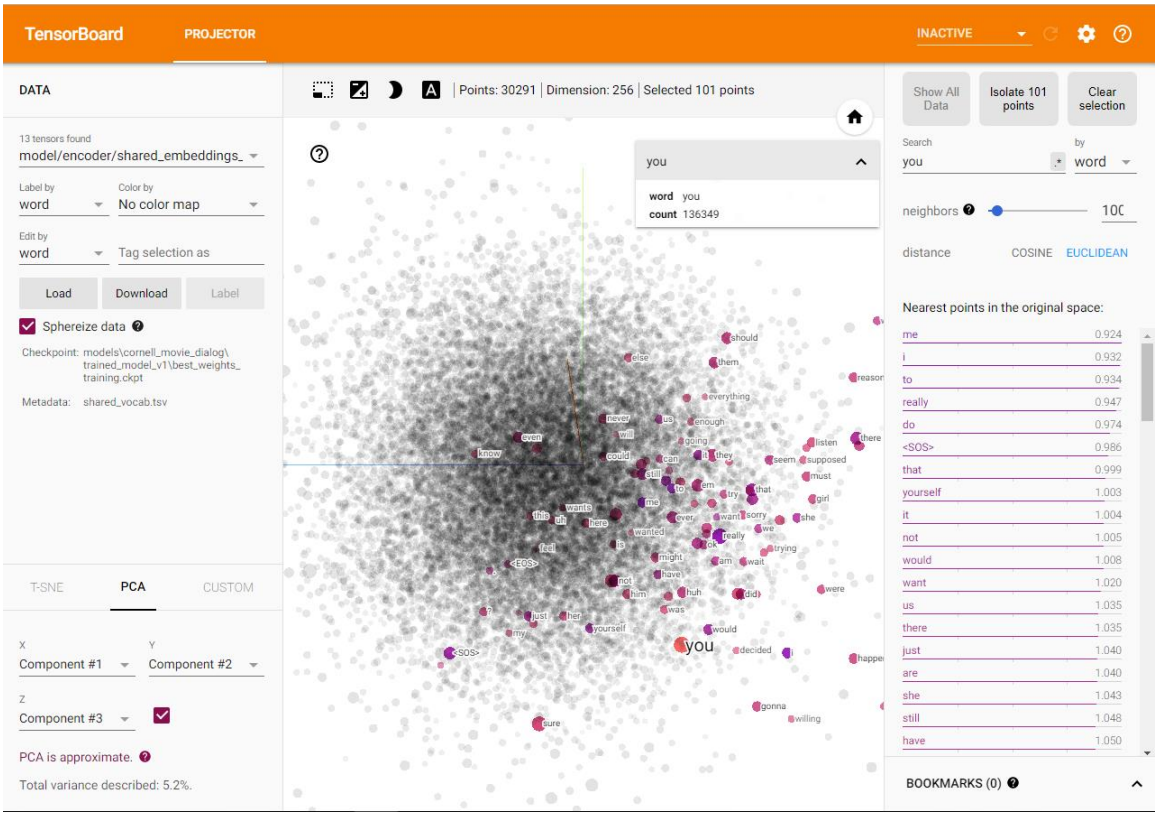


Figure 4.2.7 Projector tab in TensorBoard

4.2.3 Text-to-Speech

The Text-to-Speech (TTS) will be implemented using Google Cloud Text-to-Speech API. First, need to enable the API for the Smart Pet project on Google Cloud. After that, set up the authentication by create a service account key (in JSON file format) and save it into local PC. Next, install the Google Cloud TTS client library, then use the following code to send a "synthesize" requests to the cloud server to convert the output text to voice.

```
public static void Main(string[] args)
{
    // Instantiate a client
    TextToSpeechClient client = TextToSpeechClient.Create();

    // Set the text input to be synthesized.
    SynthesisInput input = new SynthesisInput
    {
        Text = smartChat.output()
    };

    // Build the voice request, select the language code ("en-US"),
    // and the SSML voice gender ("neutral").
    VoiceSelectionParams voice = new VoiceSelectionParams
    {
        LanguageCode = "en-US",
        SsmlGender = SsmlVoiceGender.Neutral
    };
```

Figure 4.2.8 Part of Cloud TTS request code

4.3 Face Recognition

The face recognition will be implemented using Baidu Cloud face recognition SDK. Before start, a Baidu Cloud account is needed to get the API key and secret key for calling the API.

The Baidu Cloud SDK support various type of programming language which included Java, PHP, C#, Python, Node.js etc. In Unity, C# is the main programming language, so C# version of the Baidu Cloud SDK which included API for develop face recognition and some third-party dependencies is needed to download.

After that, the downloaded SDK need to be import into Unity together with the third-party dependencies. Then, set the API Compatibility Level to .NET 2.0. Next, create a project on Baidu Cloud Platform and assign it to become a face recognition application to get the API key and secret key.

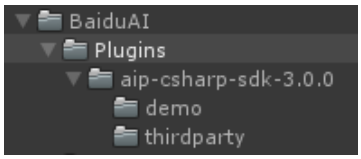


Figure 4.3.1 Import Baidu Cloud SDK into Unity

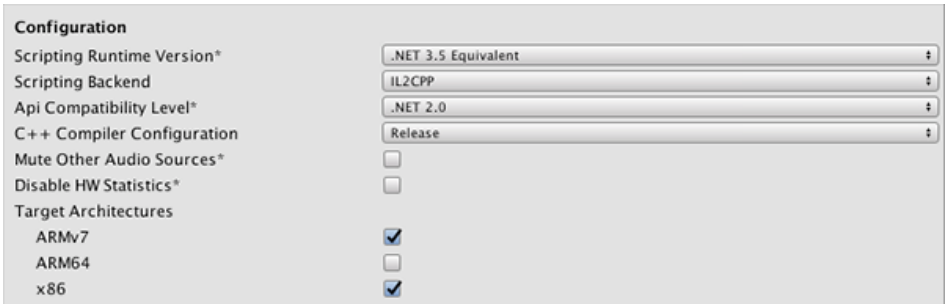


Figure 4.3.2 Set API Compatibility Level form .NET 2.0 Subset to .NET 2.0

```
12  public class FaceDetect : MonoBehaviour
13  {
14      public string API_KEY = "fafBsCdmntyA7uCmgHjtiV0b";
15      public string SECRET_KEY = "b9OqjBGeUOMocqI0xnCb9EMQvcNaF9B3";
16      Face client;
17
18      void Start()
19      {
20          client = new Face(API_KEY, SECRET_KEY);
21      }
22  }
```

Figure 4.3.3 Connect the Cloud with API key and Secret key

The implementation of face recognition module can start after the environment setup. First, a function is needed to call the phone camera to capture user's picture to created the training data set. After that, convert the captured image into BASE64 string and upload it to the Baidu Cloud face library.

```
// Register a face
   public void SignUpFace(string image, string imageType, string groupId, string userId)
   {
       var options = new Dictionary<string, object>{
       {"user_info", "PetOwner"},
       {"quality_control", "NORMAL"},
       {"liveness_control", "LOW"}
       };
       // Upload to library
       var result = client.UserAdd(image, imageType, groupId, userId, options);

   }
```

Figure 4.3.4 Register a face inside Baidu Cloud face library

When user using the Smart Pet App, user's face will be recorded in real-time and compare with the face library to indicate wheatear the user is the owner of the pet or not. In the following code, a video with resolution of 1920*1080 and 20 of the frame rates will be capture from the camera:

```
IEnumerator IECallCamera()
{
    yield return Application.RequestUserAuthorization(UserAuthorization.WebCam);
    if (Application.HasUserAuthorization(UserAuthorization.WebCam))
    {
        WebCamDevice[] devices = WebCamTexture.devices;
        deviceName = devices[0].name;

        webTex = new WebCamTexture(deviceName, 1920, 1080, 20);
        webTex.Play();
        rawImage.texture = webTex;

    }
}
```

Figure 4.3.5 Capture user's face from camera

Chapter 5: System Evaluation

5.1 Model Testing

The current model of the Smart Chat is not very good, the improvement process will keep continue, here is the testing on the conversation with the Smart Chat model.

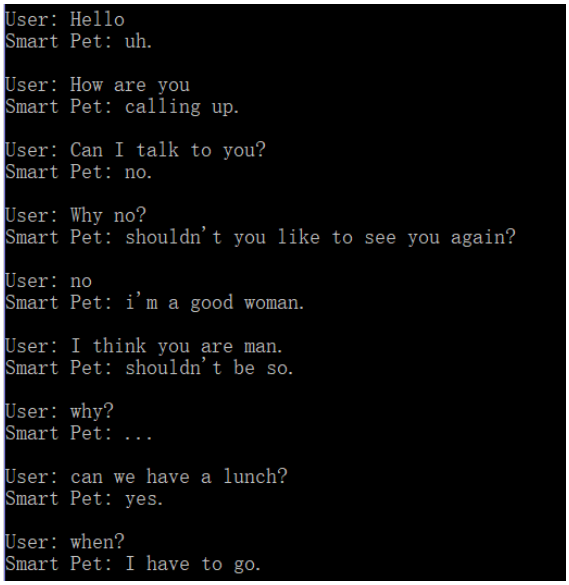


Figure 5.1.1 Testing on the conversation with the Smart Chat Model

5.2 Objective Evaluation

As we look back to the objectives of this system, all of the following objectives has achieved as described below:

- To develop a virtual pet application that let everyone can have a pet.
- To implement dual communication between the owner and the virtual pet using voice recognition.
- To implement face recognition feature on the virtual pet.

In conclusion, the system has achieved all of the 3 objectives.

Chapter 6: Conclusion

6.1 Project Review

The aim of the proposed app is to let lonely people to get a companion stay beside them which can let those people communicate with the pet and speak out their mind. Besides that, everyone can take care of virtual pet but not everyone effort take care of a real pet. Furthermore, to let the virtual pet smarter, AI technique which is voice and image recognition is implemented into it.

In this project, although the Smart Chat engine is developed but it still not smart enough. Sometime, it may answer some illogical reply for some question. Also, the Speech-to-Text model is not working perfectly, sometime it will be misunderstanding what user say.

The most difficult part of the project is to integrate the trained model with the Smart Pet App especially the face recognition part. In the beginning of the development phase, the face recognition is implemented in a PC and we decided to integrate the trained model with the Smart Pet App which develop by Unity. But Unity does not allow training a TensorFlow model in Android smartphone even using the TensorFlow Lite, so the face recognition model can only be pre-trained on a PC and integrate it with Unity, it will cause the model can only recognize the people during the training phase and cannot add the pet's owner using smart phone. Due to this issue, we use Baidu Cloud face recognition SDK to replace the TensorFlow Object Detection API to perform the face recognition part. Using the Baidu Cloud SDK will also causing another issue that have not solve yet, which is all the data will process in the cloud, so when we perform real-time face recognition, the internet connection needs to fast and stable.

6.2 Future Work

Although the Smart Pet can talk with people, but when user need to talk with it, they need to press the “Mic” button on the game GUI to trigger the Smart Chat feature. In future work, we can make a voice command such as “Hi pet” to awaken the pet to talk to user. Also, AIML can be implement for the Smart Chat to predefine the reply of some common input to let the Smart Pet look like more intelligence.

In the face recognition part, we still need to show the camera screen when perform real-time face recognition, it will interrupt the interaction of user with the Smart Pet. In future work, the real-time face recognition part of the Smart Pet can be implemented that does not need to show the camera frame on the screen, so the user will not aware that when the Smart Pet will try to recognize the owner. Also, some technique also can be implemented to let the pet able to recognize the owner more accurate, such as when the owner play with the pet, the front-camera of user’s phone will continue collect the user’s picture to enrich the face dataset.

BIBLIOGRAPHY

Abadi, M. et al., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Available from: <http://arxiv.org/abs/1603.04467>. [1 August 2018].

AISB, 2018, *AISB - The Society for the Study of Artificial Intelligence and Simulation of Behaviour - Loebner Prize* [Online]. Available from: <http://www.aisb.org.uk/events/loebner-prize>. [1 August 2018].

Angus, 2016, *Time series prediction using ARIMA vs LSTM* [Online]. Available from: <https://datascience.stackexchange.com/questions/12721/time-series-prediction-using-arima-vs-lstm>. [5 April 2019].

Apple, 2018a, *iOS - Feature Availability - Apple* [Online]. Available from: <https://www.apple.com/ios/feature-availability/#siri>. [1 August 2018].

Apple, 2018b, *iOS - Siri - Apple* [Online]. Available from: <https://www.apple.com/ios/siri/>. [1 August 2018].

Bostic, K., 2013. Nuance confirms its voice technology is behind Apple's Siri. *AppleInsider*. Available from: <http://appleinsider.com/articles/13/05/30/nuance-confirms-its-technology-is-behind-apples-siri>. [1 August 2018].

Chaib, F., 2017, *"Depression: let's talk" says WHO, as depression tops list of causes of ill health* [Online]. Available from: <http://www.who.int/news-room/detail/30-03-2017--depression-let-s-talk-says-who-as-depression-tops-list-of-causes-of-ill-health>. [1 August 2018].

Cho, K. et al., 2014. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*,

Deshpande, A., 2017, *How I Used Deep Learning To Train A Chatbot To Talk Like Me (Sorta) – Adit Deshpande – CS Undergrad at UCLA ('19)* [Online]. Available from: <https://adeshpande3.github.io/How-I-Used-Deep-Learning-to-Train-a-Chatbot-to-Talk-Like-Me>. [1 August 2018].

Dietterich, T., 2002. Overfitting and undercomputing in machine learning. *ACM Computing Surveys*.

Dreams, A., 2013, *Steve Worswick Interview - Loebner 2013 winner* [Online]. Available from: <https://aidreams.co.uk/forum/index.php?page=Steve_Worswick_Interview_-_Loebner_2013_winner#.W_coHGgzZPZ>. [1 August 2018].

Dysthymia, M.D., 2014, *Depression* [Online]. Available from: <https://www.psychologytoday.com/us/basics/depression>. [1 August 2018].

Emmanuel, S., 2015, *4 Ways To Make Owning A Pet Cheaper | iMoney* [Online]. Available from: <https://www.imoney.my/articles/4-ways-to-make-owning-a-pet-cheaper>. [15 November 2018].

Eric Arthur, 2016, *Using Rule-Based AI in Our SMS Chatbot* [Online]. Available from: <https://botpublication.com/using-rule-based-ai-in-our-sms-chatbot-23cb493b3096>. [5 April 2019].

Eugine Kang, 2017, *Long Short-Term Memory (LSTM): Concept* [Online]. Available from: <https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359>. [5 April 2019].

Guzzoni, D., 2008. Active: a unified platform for building intelligent applications. Available from: <https://infoscience.epfl.ch/record/114758>. [1 August 2018].

Harris, P., 2015. Pearson: Student Mobile Device Survey. *Pearsoned*, p.53. Available from: <http://www.pearsoned.com/wp-content/uploads/2015-Pearson-Student-Mobile-Device-Survey-College.pdf>. [1 August 2018].

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735–1780. Available from: <http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735>. [20 November 2018].

HuiWen, H., 2017, *小愛同學融入小米家族 可控制8500萬IoT設備 - 中時電子報* [Online]. Available from: <https://www.chinatimes.com/realtimenews/20171130000001-260412>. [1 August 2018].

IObit, 2018, *IObit Applock* [Online]. Available from: <https://www.iobit.com/ko/applock.php#googtrans(ko)>. [21 November 2018].

Krunal, V., 2018, *Everything You Need to Know About Voice Recognition Technology - DZone AI* [Online]. Available from: <https://dzone.com/articles/everything-you-need-to-know-about-voice-recognitio>. [15 November 2018].

Kumparak, G., 2011. The Original Siri App Gets Pulled From The App Store, Servers To Be Killed. *TechCrunch*. Available from: <https://techcrunch.com/2011/10/04/the-original-siri-app-gets-pulled-from-the-app-store-servers-killed/>. [20 November 2018].

Levy, S., 2016, *An Exclusive Look at How AI and Machine Learning Work at Apple | WIRED* [Online]. Available from: <https://www.wired.com/2016/08/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple/>. [1 August 2018].

Lowe, R., 2018, *Ubuntu Dialogue Corpus v2.0* [Online]. Available from: <https://github.com/rkadlec/ubuntu-ranking-dataset-creator>. [1 August 2018].

Merola, I., Prato-Previde, E. and Marshall-Pescini, S., 2012. Dogs' Social Referencing towards Owners and Strangers Kalueff, A. V., (ed.). *PLoS ONE*, 7(10), p.e47653. Available from: <http://dx.plos.org/10.1371/journal.pone.0047653>. [16 November 2018].

Mizil, D., 2018, *Cornell Movie-Dialogs Corpus* [Online]. Available from: <https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html>. [1 August 2018].

Oord, A. van den et al., 2016. WaveNet: A Generative Model for Raw Audio Based on PixelCNN Architecture. *arXiv*.

Robin，Reni, 2018, *Realtime Object and Face Detection in Android using Tensorflow Object Detection API | Skcript* [Online]. Available from: <https://www.skcript.com/svr/realtime-object-and-face-detection-in-android-using-tensorflow-object-detection-api/>. [1 August 2018].

Robinson, K.M., 2017, *How Pets Help Manage Depression* [Online]. Available from: <https://www.webmd.com/depression/features/pets-depression>. [1 August 2018].

Ross, M., 2015, *Severe Loneliness Leads To Depression* [Online]. Available from: <https://www.webpsychology.com/news/2015/12/14/severe-loneliness-leads-depression-248116>. [1 August 2018].

Rousseau, A., Deléglise, P. and Estève, Y., 2012. TED-LIUM: an Automatic Speech Recognition dedicated corpus. *LREC*, (March 2015), pp.125–129. Available from: <http://www.lrec-conf.org/proceedings/lrec2012/pdf/698_Paper.pdf>. .

Scoble, R., 2010. BREAKING NEWS: Siri bought by Apple. *Scobleizer*. Available from: <https://scobleizer.blog/2010/04/28/breaking-news-siri-bought-by-apple/>. [1 August 2018].

SelPhi, 2018, *SelPhi – FacePhi es líder global en Reconocimiento Facial para el sector bancario* [Online]. Available from: <https://selphi.es/?lang=en>. [21 November 2018].

Sheets, C.A., 2012. What is Simsimi? All About the iPhone and Android App That Has the World Laughing Out Loud. *International Business Times*. Available from: <https://www.ibtimes.com/what-simsimi-all-about-iphone-android-app-has-world-laughing-out-loud-399156>. [1 August 2018].

Siew May, L., 2016, *The cost of having a pet | Focus Malaysia* [Online]. Available from: <http://www.focusmalaysia.my/Income/the-cost-of-having-a-pet>. [15 November 2018].

SimSimi, 2013, *SimSimi API - Pricing* [Online]. Available from: <http://developer.simsimi.com/pricing>. [1 August 2018].

Sordoni, A., 2016, *Microsoft Research Social Media Conversation Corpus* [Online]. Available from: <https://www.microsoft.com/en-us/download/details.aspx?id=52375>. [1 August 2018].

Srirupa, 2017, *Why Face Recognition Technology is More Important than You Think |* [Online]. Available from: <https://acadgild.com/blog/why-face-recognition-technology-is-more-important-than-you-think>. [15 November 2018].

TechBubble, 2016, *Mitsuku Chatbot Wins Most Humanlike AI In Loebner Prize For A Second Time.* [Online]. Available from: <https://www.techbubble.info/blog/artificial-intelligence/chatbots/entry/mitsuku-chatbot>. [1 August 2018].

Thanaki, J., 2018. *Machine Learning Solutions : Expert techniques to tackle complex machine learning problems using Python.*, Packt Publishing.

Thorben, 2018, *Price drop: Xiaomi Mi 8 - 6/128 GB flagship for $389.99* [Online]. Available from: <https://china-gadgets.com/xiaomi-mi-8-smartphone-review/>. [1 August 2018].

Twenge, J.M., 2014, *Why So Many People Are Stressed and Depressed* [Online]. Available from: <https://www.psychologytoday.com/us/blog/our-changing-culture/201410/why-so-many-people-are-stressed-and-depressed>. [1 August 2018].

Wood, L. et al., 2015. The Pet Factor - Companion Animals as a Conduit for Getting to Know People, Friendship Formation and Social Support Uchino, B.N., (ed.). *PLOS ONE*, 10(4), p.e0122085. Available from: <https://dx.plos.org/10.1371/journal.pone.0122085>. [22 November 2018].

Xiaomi, 2018a, *How to use Xiaomi Flash Tool - Official Tool* [Online]. Available from: <https://xiaomiflashtool.com/tutorial/use-xiaomi-flash-tool>. [20 November 2018].

Xiaomi, 2018b, *小爱开放平台-为开发者提供一站式人工智能服务* [Online]. Available from: <https://xiaoai.mi.com/>. [1 August 2018].

POSTER

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

Universiti Tunku Abdul Rahman

Faculty of Information Communication Technology

Voice and Image Recognition for Smart Pet App

Introduction

This project is going to develop a smart pet game android application using Augmented Reality (AR) and Artificial Intelligence (AI) technology.

Objective

- To develop a virtual pet application that let everyone can have a pet.
- To implement dual communication between the owner and the virtual pet using voice recognition.
- To implement face recognition feature on the virtual pet.

Methodology

<Smart Chat>

i. Voice → Input Text (STT)
ii. Input Text → Output Text (Seq2Seq Model)
iii. Output Text → Voice (Google Cloud TTS)

<Face Recognition>

i. Capture Image from Camera
ii. Stored into Baidu Cloud Face Library.
iii. When user use it, capture user's face and compare with Baidu Cloud Face Library.

Conclusion

Results:

- The pet can talk to user.
- The pet can recognize owner.

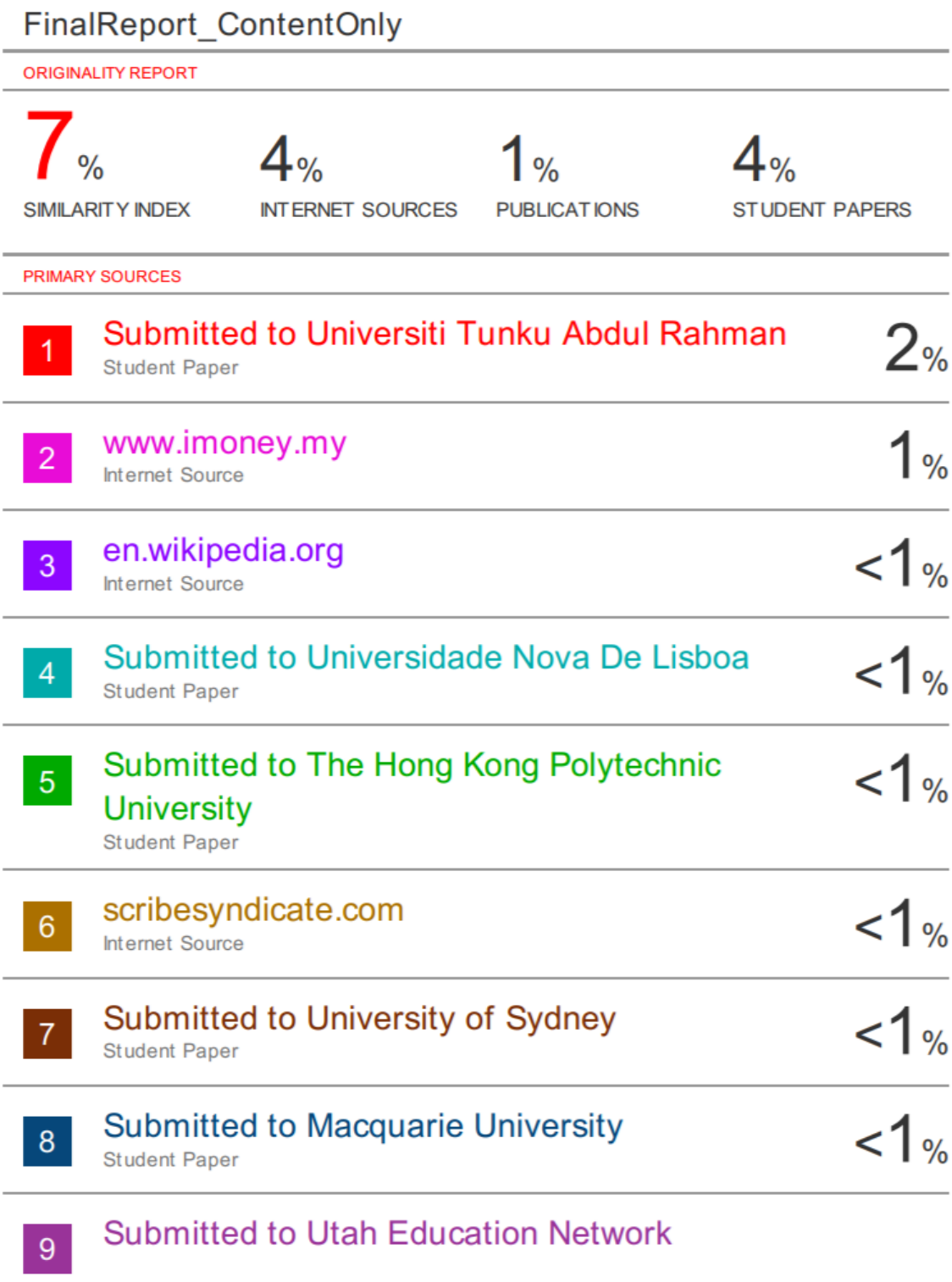
Future Works:

- Implement voice command to trigger the Smart Chat.
- Learn user's face when user play with the pet without showing a camera frame.

Project Developer : Lee Yen Long
Project Supervisor: Ts Saw Seow Hui

TensorFlow

PLAGIARISM CHECK RESULT



FinalReport_ContentOnly

ORIGINALITY REPORT

7%	4%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	2%
2	www.imoney.my Internet Source	1%
3	en.wikipedia.org Internet Source	<1%
4	Submitted to Universidade Nova De Lisboa Student Paper	<1%
5	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
6	scribesyndicate.com Internet Source	<1%
7	Submitted to University of Sydney Student Paper	<1%
8	Submitted to Macquarie University Student Paper	<1%
9	Submitted to Utah Education Network	

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lee Yen Long
ID Number(s)	1603235
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Voice and Image Recognition for Smart Pet App

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

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

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

_____ _____

Signature of Supervisor Signature of Co-Supervisor

Name: _____ Name: _____

Date: _____ Date: _____

Appendices B – Plagiarism Check Result

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lee Yen Long
ID Number(s)	1603235
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Voice and Image Recognition for Smart Pet App

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

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

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

_____ _____

Signature of Supervisor Signature of Co-Supervisor

Name: _____ Name: _____

Date: _____ Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	1603235
Student Name	Lee Yen Long
Supervisor Name	Ts Saw Seow Hui

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

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

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 marks (1 grade) reduction. _____ (Signature of Supervisor) Date: