

# 2017-2 Embedded System

*Final Project -- 辨識頭盔*

B02901137 吳元魁 / B02901011 趙祐毅



Demo video: <https://youtu.be/40-1K9EXpRs>

github: <https://github.com/ywk991112/ESLab-final>

Date: 2017/6/16

## 1. 分工

- 吳元魁：影像辨識
- 趙祐毅：Raspberry Pi 基礎架設、硬體架設、語音辨識、文字轉語音

## 2. 構想

每個人小時候都有一個夢想，有人想當太空人，有人想當大老闆，有人想當咖啡廳老闆，有人想當掃地工，但在我小的時候，我一直有個夢想，就是想當個打擊犯罪的正義使者。隨著考試的壓力，同儕的眼光，這個夢想隨著年紀逐漸的掩埋在成堆的廢氣參考書下。

這天考試完，拖著疲憊的身體和心靈，我們決定看個復仇者聯盟好好犒賞自己，映入

眼簾的，是充滿正義感的，不畏惡勢力的鋼鐵人，把外星人一個一個解決掉，最後為了拯救市民，帶著導彈飛向太空。小時候的夢想突然如洪水一般，湧進了我的心頭，恨不得未來的我，也能像鋼鐵人一樣，拯救世界。

就在我還沉浸在我的幻想中的時候，我突然想到，如果它突然看到兩個人在對戰的時候，卻不知道誰才是自己的敵人的時候，該怎麼辦？

於是，我拿起了開發板，為了我兒時的夢想，為了我想要打敗惡勢力的偉大情操，製作了這個辨識頭盜，在相機照到人後，會從資料庫中抓出所有資料，再由麥克風的輸入想要知道的資料，由喇叭輸出從資料庫抓出來的資料。這樣，在打擊犯罪就可以如魚得水，無往不利。

### 3. 程式 Platform

- Raspberry pi 3
- Python 3
- Module: speech\_recognition, gtts

### 4. 系統具備哪些功能

- Input
  - a. 按下按鈕 A 後照相，並辨識此人的身份
  - b. 按下按鈕 B 後，語音回饋前一次照片中的人是不是“Enemy”
  - c. 按下按鈕 C 後透過麥克風說話，可以詢問系統此人是否為“Enemy”和“Weakness”為何
- Data saving
  - a. 接收到相片之後，存檔
- Data processing
  - a. 接收到相片之後，利用影像辨識，分辨出照片裡的人是誰
  - b. 從資料庫中找出這個人的資訊
  - c. 分析使用者說什麼指令，轉成文字，進行處理。並且把使用者要求的資訊找出
- Output
  - a. 將使用者要求的資訊，轉成語音回饋給使用者

### 5. 系統架構

- a. 系統包含哪些子元件（軟體 & 硬體）？

- Raspberry pi 3



- Camera module



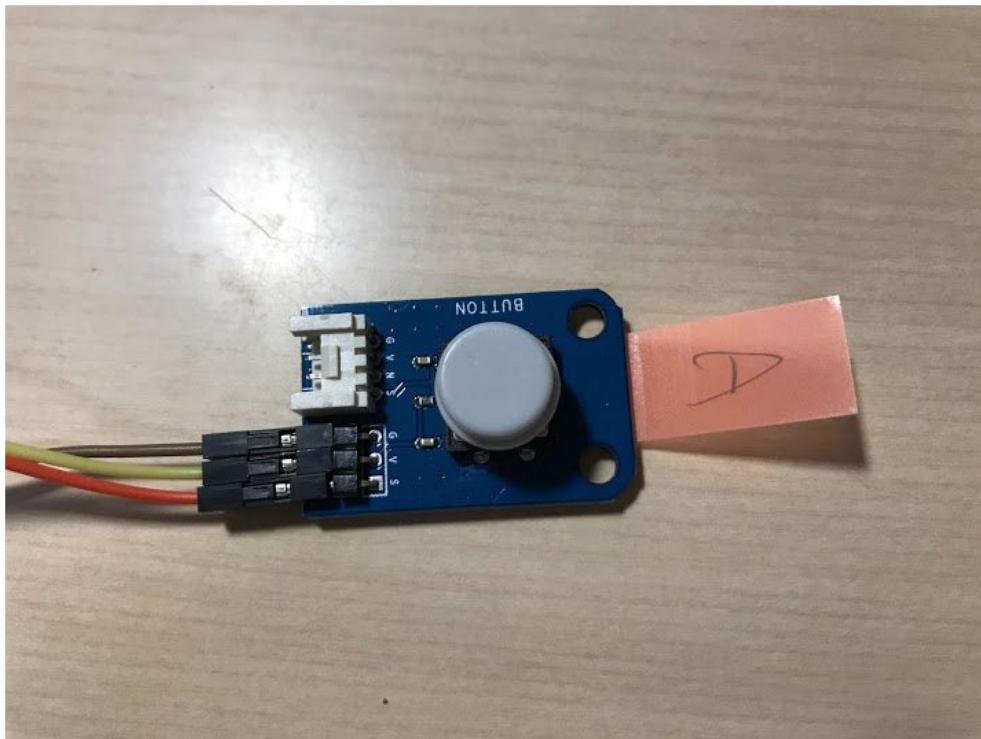
- Microphone



- Speaker



- Button



### b. 每個元件如何運作？

- Raspberry pi 3

Raspberry pi 基本上就是一台有作業系統的電腦，但能夠處理的運算不能太高，我們的 R pi 的作業系統是：[Ubuntu 16.04 mate](#)，也有完整的 UI 介面、File system。我們主要使用 Python 3.5 來跑程式，以及與硬體互動。

- Camera module

採用的相機模組是與 R pi Camera module interface 搭配，可以支援的解析度為：3280 x 2464，30 fps，我們實驗中使用的解析度為 1920 x 1080。

使用 Python 的 API 相當容易：

```
##### code #####
import picamera
camera = picamera.PiCamera()
camera.capture( path + str(file_name) )
##### end #####
```

這樣即可完成拍照。

- Microphone

因為在嘗試使用 python module 錄音做辨識時，一直遇到沒辦法讀取到正確麥克風的問題，因此我們使用 OS 內建的錄音功能，在 python 裡面跑 Command line:

```
arecord --format=S16_LE --duration=3 --rate=16k record.wav
```

如此即可錄製 3 秒鐘的音訊檔，存成 ‘record.wav’，後面再讀檔做辨識。

- 影像辨識：

按下A 按鈕即可拍照並進行影像辨識，找出這個人是誰，以下解釋影像辨識的步驟：

- a. create group: 首先，先建立一個群組
- b. add person: 把要辨識的人加到群組裡
- c. add face: 把每個人的照片存進去當training data
- d. train group: 將training data做training
- e. identify: 辨識出照片裡的人

- 語音辨識

在這裡我們想要達成語音轉文字的功能。回讀取剛錄下來的音檔，進行辨識。

使用 python module: speech\_recognition

這個 module 有提供多種不同辨識系統的 API，包含：Sphinx, Google speech recognition, Microsoft Bing Voice recognition, IBM system 等等。

我們嘗試 Sphinx, google speech recognition 後，選用後者，因為表現比較穩定。

- 讀取指令並回覆

我們這裡並沒有用機器學習訓練，因為需要的指令不多，因此直接使用 Rule based，直接分析我們的指令有沒有包含一些關鍵字：Enemy, Weakness 等。

讀到關鍵字之後，會去資料庫抓取目前針對的人的資料：他是不是Enemy, 他的Weakness 有什麼。

- 文字轉語音

我們把資訊讀取到後，要用語音回饋給使用者。

使用 python module: gtts。

gtts 可以把文字轉成音檔之後儲存，（我們使用英文）。後續我們要把音檔讀進來，再播放，這部分也遇到一些python module dependency 的問題，後來是使用OS 內建的播放器，從 python call command line:

```
omxplayer -p -o local reponse.wav
```

## 6. 遇到的問題

- 影像辨識：原本傳送照片，想用傳送url的方式，所以要用imgur的api，先上傳照片後，再用url來進行影像辨識。但後來覺得太麻煩了，於是研究怎麼將照好的照片，用octet-stream的格式上傳，解決方式用python的open，在mode為'rb'的方式讀入後，再傳上去。
- OS 不同的問題：一開始都在電腦上（使用 Mac ）嘗試，因為覺得 python 在 R pi 上運作不會差太多，嘗試語音辨識、文字轉語音，在電腦上成功之後，再灌到 R pi 上，沒想到居然出現很多問題，硬體上包含：麥克風要用外接，所以有些設定要更改，軟體也不一定能夠抓取到正確的來源。軟體上包含：音訊播放的 Module “playsound” 不能 work，後來採用 “pygame”，居然出現某些音訊檔無法正確讀取的情形，且明明都是 wav 檔。最後才採用 OS 內建的程式，進行錄音、播放等功能。
- 系統太笨重：目前各部分獨立設計，組裝起來相當笨重，最重的是行動電源，再來是音響、R pi，麥克風也有些佔位子，無法收起來。對於一個行動裝置來說，務必要做到輕便舒適，我們在試戴的時候，相當不舒服，還有很大改進空間。
- 辨識太慢：我們在實作的時候，需要用到網路，將資料上傳，照片上傳：進行影像辨識，音訊上傳：進行語音辨識。整體而言辨識速度不足，大概會 Delay 幾秒鐘，應當改進到可以 Real-time 才有足夠實用性。

## 7. 未來展望

- 系統速度加快：可以考慮使用更 Powerful 的開發版，使用離線辨識
- 結合其他功能：目前系統只支援特定功能，應當可以合併更多指令
- 結合 Google assistant：我們不太可能自己做一台聊天機器人，但是可以使用現有的聊天機器人，請他幫忙做一些事情，包括問天氣、時間等等。讓頭盔功能更加多元。
- 資料庫：資料庫的建置我們沒有深入探討，但是現有資料相當豐富，若能融合 Facebook, google contact 等等，在資料上會更為齊全。
- 鋼鐵人是我們的夢想



## 8. 結論與心得

- 在討論 Final 題目時，我們絞盡腦汁，想要想一些特別的題目，曾經想過智慧魚缸，可以自動監測環境並調控。後來決定這個題目，希望可以用我們所學，搭配套件，做一個好的運用，這樣的應用不止可以在偵查犯罪中應用，也許未來也會延伸到日常生活中，幫忙提醒使用者一些人的名字、學校等。
- R pi 是個相當方便的系統，尤其到第三代的效能已經相當好，但是還有一些與其他 Linux kernel 不同之處，在開發時需要特別注意。
- 看了看其他組的成果，有人做自動餵食、有人做多輛車的溝通等，覺得相當有趣，也許之後發想題目時能夠更加 “Think outside the box”，也可以多嘗試在硬體設備的設計、組裝。把一些不會想過的功能合併在一起，也許就會是個新的突破。

## 9. 參考資料

- [Google assistant SDK](#)
  - [instructions](#)
  - [instructions2](#)
  - [API.AI instructions](#)
  - [API.AI example](#)
  - [API.AI Python documentation](#)
  - [API.AI example: profile bot](#)
  - [How to Build Your Own AI Assistant Using Api.ai](#)
- [python speech recognition](#)
  - [pyaudio 'portaudio.h' file not found](#)
- [text to speech — gTTS](#)
- [R pi Button: GPIO](#)
- [R pi3 GRIO description](#)
- [Button sample code](#)

~End~