

大樓管理系統

指導老師：徐偉智 教授

參賽組員：劉沐鑫、鍾秉榮、胡晉愷、陳冠名

摘要

本專題一套智慧大樓管理的系統方案，擁有的方案可以取代替管理員大部分的功能，住戶進出人的管理、大樓郵件包裹、大樓投票、語音對話系統，而進入大樓時會透過人臉辨識來辨識出進入者是否為住戶，如為住戶則可讓該住戶直接進入，否則會將記錄下來並進行通知等等功能，我們可以大幅減少一般大樓管理員的工作，甚至取而代之。擁有這些功能能夠減少管理員的人力資源，進行統一、系統化的管理大樓資訊，也可以管制外來人士的進入。

第一章、前言

1.1 研究動機

在現代化的社會中，高度密集的都市已成為常見的現況，大樓林立、人口高度的密集。然而這些大樓中往往都有管理員的存在負責管理大樓內部的資訊、處理一些事件，但聘請管理員需要一筆額外的開銷，為此我們希望利用大樓管理機器人來取代普通的管理員，提供住戶更便利的生活環境，及提供更經濟實惠的價格，以打造管理公司與住戶雙贏的局面。

1.2 研究目的

現今的社會都市化的趨勢越來越明顯，隨著大樓出現越來越多，我們決定提出一套可以解決每個大樓管理員的問題。此專題的目的在於減少管理員的負擔進而減少人力與金錢成本的同時確保了管理的效率與有效的維護社區安全與便利。

第二章、文獻與技術討論

2.1 Android 簡介

Android 作業系統的核心屬於 Linux 核心的一個分支，具有典型的 Linux 排程和功能，但並不是 GNU/Linux。Android 硬體抽象層 (HAL, Hardware Abstraction Layer) 把 Android framework 與 Linux kernel 隔開，這種中介層的方式使得 Android 能在行動裝置上獲得更高的執行效率。在 Android 應用程式開發中，通常通過在 Android SDK 中使用 Java 作為編程語言來開發應用程式。開發者亦可以通過在 Android NDK 中使用 C 語言或者 C++ 語言來作為編程語言開發應用程式。

2.2 Django 簡介

Django 是一個高級的 Python 網路框架，可以快速開發安全和可維護的網站。由經驗豐富的開發者構建，Django 負責處理網站開發中麻煩的部分，因此你可以專注於編寫應用程序，而無需重新開發。它是免費和開源的，有活躍繁榮的社區、豐富的文檔、以及很多免費和付費的解決方案。

2.3 OpenCV 與 Dlib

OpenCV 一個包含許多電腦視覺相關演算處理的開放原始碼 Library，並且 OpenCV 可以在 Android、ios 上開發，支援的程式語言也有 C/C++，Java，Python 用 C++ 語言編寫，它的主要介面也是 C++ 語言，但是依然保留了大量的 C 語言介面。該庫也有大量的 Python, Java and MATLAB/OCTAVE 的介面。這些語言的 API 介面函式可以透過線上文件取得。

Dlib 一套包含了機器學習、計算機視覺、圖像處理等的函式庫，包含廣泛的機器學習演算法，所有的設計都是高度模組化的，執行快速，並且通過一個乾淨而現代的 API，使用起來非常簡單。它用於各種應用，包括機器人技術、嵌入式裝置、手機和大型高效能運算環境。其中包含用於在 C++ 中建立複雜軟體以解決實際問題的機器學習演算法和工具。並且除了 C++ 之外還提供了 Python API。

第三章、系統目的與架構

3.1 系統目的

本專題的系統目的在於設計一個系統完善的自動智慧的大樓管理系統，會辨識進入者是否為住戶以維護住戶的安全杜絕外來人士的隨意進出，以及語音的助手聊天系統可以用語音查詢大樓和住戶相關事項、包裹的管理與查詢、大樓決議的投票。此系統操作簡易同時有淺顯易懂的介面可以使用，也可以使用語音的方式。而後台的管理介面可以隨著不同的使用裝置調適介面比例大小，管理者可以使用不同的裝置隨時做管理、設定。完整的後台管理系統能讓管理更加方便。

3.2 系統架構

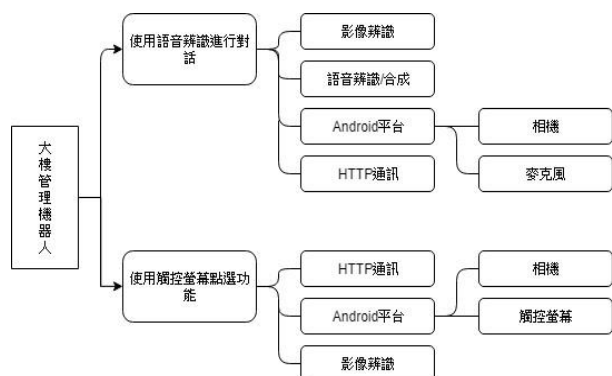


圖 3-1 系統功能

本系統平台主要分成 2 個部分，一個為使用者操作的平台，這裡我們採用 Android 7 (API level24) 的平台，暫時使用 Android 手機做 DEMO。另一個是以 Ubuntu 作為伺服器、資料庫的平台。

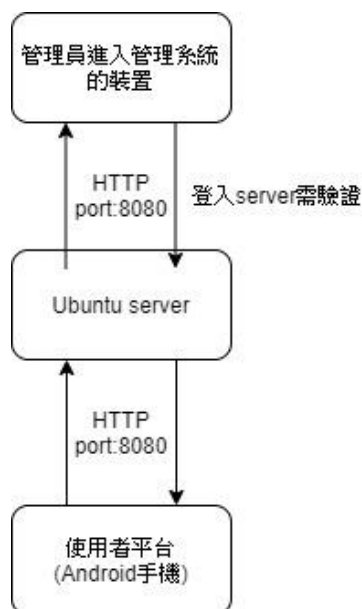


圖 3-2 系統結構

第四章、系統設計與實作

4.1 人臉辨識與圖片切割

4.1.1 人臉辨識

本人臉辨識採用 opencv 加上 dlib 函式庫內的人臉特徵點提取，而 dlib 使用的人臉偵測演算法是採用 HOG(Histogram of oriented gradient)的方法，並利用了 68 點臉部特徵值來進行比對，同時對人臉輪廓、眼睛、眉毛、鼻子、嘴巴進行比對。

需要的功能：

1. 讀取資料庫內的圖片以及名稱，將提取後的特徵點資料存成.pkl，以便辨識時快速使用。

2. 提取指定輸入的圖片內人臉特徵點資料，從.pkl檔內尋找資料並進行比對，將比對後的人名回傳，若辨識此人無在資料庫內則回傳此人不在資料庫內。

因為 dlib 官方有已經訓練完善的人臉特徵點偵測模型可以使用，因此需先下載 shape_predictor_68_face_landmarks.dat 以供使用。

如圖 4-1 為將新的人臉資料寫進檔案的程式步驟。設定新的住戶成員時需要給資料庫此住戶的臉部的特徵點資訊。

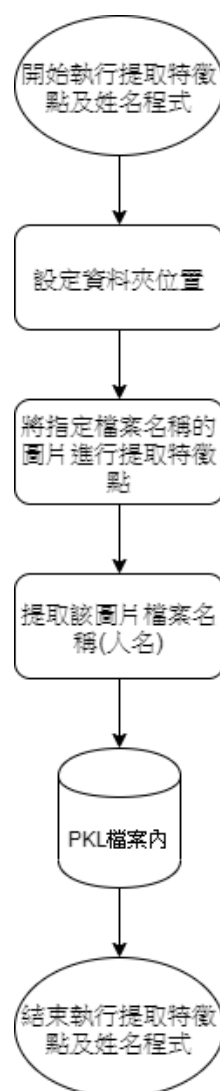


圖 4-1 將新的人臉資料寫進檔案的程式步驟。

人臉辨識判別的部分是使用歐式距離的方式找到與資料庫特徵點相似的住戶，如果找不到就會回傳 false 會判斷此人非住戶，如圖 4-2。

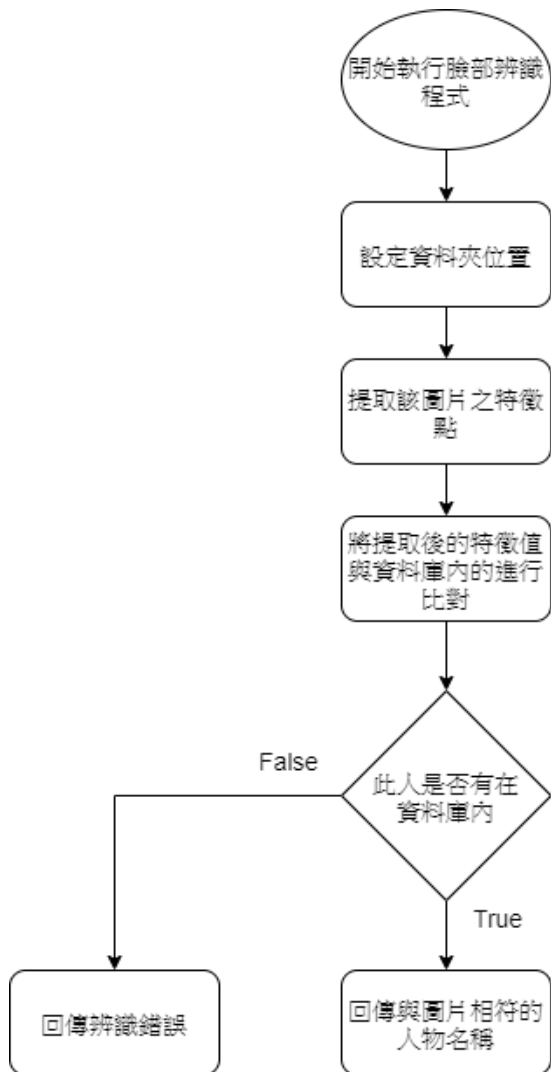


圖 4-2 將指定圖片與資料庫內進行比對的步驟。

4.1.2 人臉圖片切割

為了方便存取以及辨識，將一張擁有多人的入臉切割下來單獨處理與辨識。此部分使用 opencv 與 dlib 的人臉偵測給的位置計算出座標後將每個臉部切割成新的檔案。切割後會同時傳回偵測到的人臉數量，如圖 4-4。



圖 4-3 為欲切割的圖片

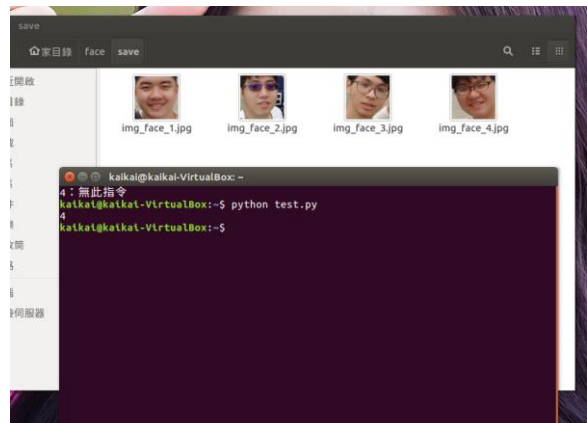


圖 4-4 切割後的圖片

4.2 語音辨識與合成

4.2.1 語音辨識

此語音辨識使用 Android 內建 SpeechRecognizerIntent 進行聲音採樣上傳至 Google 並回傳所有可能結果之文字。



圖 4-4 語音辨識流程



圖 4-5 語音辨識-辨識中



圖 4-6 語音辨識-回傳結果

4.2.2 發音

利用 Android 內建 Speech.tts.TextToSpeech 設定語言為中文，進行文字轉語音達到發音的效果。



圖 4-7 發音 TextToSpeech 流程

4.2.3 聊天機器人

使用 Jieba 分詞與 Numpy 處理向量運算以及 Tensorflow1.14 之 LSTM 訓練模組並預測，通過給予問題與相對應答案的監督式學習，讓其學習詞語詞之間的關聯性並預測下一個詞可能出現的選項。



圖 4-8 訓練模組流程

```

File Edit View Search Terminal Help
2020-05-23 22:01:44.648827: W tensorflow/core/framework/allocator.cc:107] Allocation of 13559200 exceeds 10% of system memory.
step= 100 loss= 3.7421439 learning_rate= 0.5
step= 200 loss= 3.1991627 learning_rate= 0.5
step= 300 loss= 2.8429132 learning_rate= 0.5
step= 400 loss= 2.4882252 learning_rate= 0.5
step= 500 loss= 2.0899408 learning_rate= 0.5
step= 600 loss= 1.7104905 learning_rate= 0.5
step= 700 loss= 1.375257 learning_rate= 0.5
step= 800 loss= 1.1622787 learning_rate= 0.5
step= 900 loss= 0.9443064 learning_rate= 0.5
step= 1000 loss= 0.73640996 learning_rate= 0.5
step= 1100 loss= 0.5860894 learning_rate= 0.5
step= 1200 loss= 0.49400533 learning_rate= 0.5
step= 1300 loss= 0.6912577 learning_rate= 0.5
step= 1400 loss= 0.42429456 learning_rate= 0.5
step= 1500 loss= 0.3081825 learning_rate= 0.5
step= 1600 loss= 0.28111923 learning_rate= 0.5
step= 1700 loss= 0.20738548 learning_rate= 0.5
step= 1800 loss= 0.18443233 learning_rate= 0.5
step= 1900 loss= 0.1361905 learning_rate= 0.5
  
```

圖 4-9 訓練中

```
jason@ubuntu:~/Desktop/twblogs$ python3 accessdata.py 你好
jason@ubuntu:~/Desktop/twblogs$ python3 accessdata.py 你今天過得如何
jason@ubuntu:~/Desktop/twblogs$ python3 accessdata.py 你看過電影嗎
```

圖 4-10 問題

```
chatbot>> 你好
{'answer': '你好'}
chatbot>> 還不錯 你呢?
{'answer': '還不錯 你呢?'}
chatbot>> 我看過 魔鬼 終結者
{'answer': '我看過 魔鬼 終結者'}
```

圖 4-11 預測答案



圖 4-13 語音辨識按鈕及語音對話顯示

4.3 使用者操作平台

4.3.1 使用者介面

給住戶使用的介面（User Interface）是以 Android 作業系統為架構，利用 Android Studio 開發的 Android App，主要使用 Java 語言編寫。

使用者介面上有以下特點：

- 圖形操作介面，藉由圖像增加使用者的理解程度，減低操作難易度。
- 透過人臉即時檢測(Real Time Face Detection)功能搭配相機控制(Camera 2 api)，使用者可以省去手動拍攝照片的步驟。
- 透過人臉辨識功能，讓使用者不須透過帳號即可快速登入使用。
- 語音辨識功能，使用者可以透過聲控執行各種操作。



圖 4-14 辨識後獲得使用者 ID 與姓名



圖 4-12 主畫面



圖 4-15



圖 4-16



圖 4-17

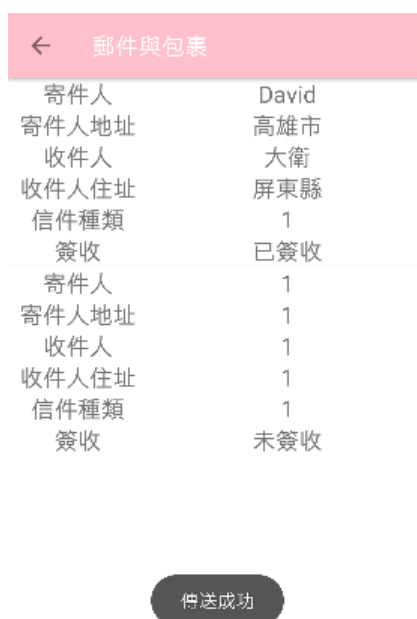


圖 4-18

← 帳單	
繳費類型	電費
月份	10
金額	10
繳費期限	2020-01-01T00:00:00
繳費日期	請盡速繳費

傳送成功

圖 4-19

圖 4-15~4-19 為使用者畫面 X5 張

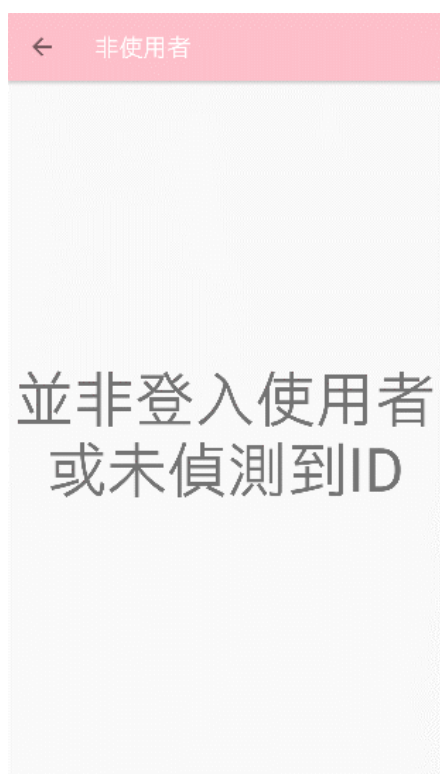


圖 4-20 非住戶使用畫面

4.3.2 人臉即時檢測 Real Time Face Detection

人臉檢測是透過 Android 系統內建的人臉識別 api- FaceDetector 完成的，此 API 可以透過少量代碼完成人臉檢測，可以檢測出 Bitmap 圖像中的人臉及數量。

在專題中，我們將透過裝置內建的相機獲得即時影像，再透過 FaceDetector API 進行識別，當 FaceDetector 識別出人臉時，再透過 Camera2 API 控制相機的拍照功能，進行拍照擷取使用者照片。

用戶端與伺服器端之請求 http request。利用第三方資源庫 Retrofit 2 完成與伺服器的請求(request)，使用 HTTP method 中的 GET 方法獲取伺服器(response)資料，顯示在用戶端，使用 POST 方法上傳使用者照片到伺服器進行人臉辨識。

Retrofit2 是 Android 一種 HTTP 請求(request)的框架，是以 OkHttp 為基礎設計，並且使用 REST API 設計，是遵循 RESTful 的 Http 框架進行封裝，在進行請求時 Header、Url 等資料是介由 Retrofit 2 封裝，後續進行網路請求的動作主要還是 OkHttp 來進行，如下圖 4-21。

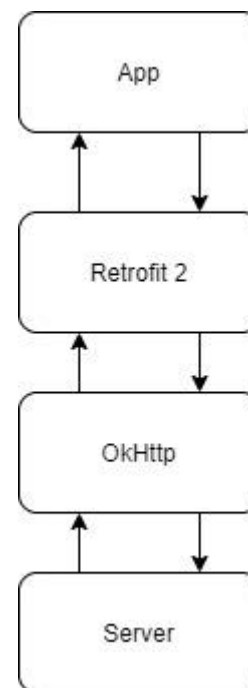


圖 4-21

4.4 大樓後臺管理系統(含 app 後台)

4.4.1 系統使用技術

作業系統:Ubuntu

伺服器軟體：uWSGI, nginx

系統實作語言：python2.7(Django), MySQL, HTML, JavaScript, CSS, shell script

使用套件：

臉部辨識：dlib, numpy, opencv, skimage, imutils

聊天機器人：Python3.6, jieba(中文分詞系統), tensorflow

4.4.2 系統功能

(1)登入登出功能

(2)主頁面

(3)住戶清單

- 1.新增住戶
- 2.刪除住戶
- 3.修改住戶資料
- 4.建立人臉辨識

(4)費用

- 1.新增費用
- 2.刪除費用
- 3.修改費用

(5)郵件與包裹

- 1.新增郵件與包裹
- 2.刪除郵件與包裹
- 3.修改郵件與包裹

(6)大樓公告

- 1.新增大樓公告
- 2.刪除大樓公告
- 3.修改大樓公告

(7)投票清單

- 1.新增投票
- 2.刪除投票
- 3.修改投票

(8)建立臉部辨識

(9)姓名查詢 ID

(10)大樓資訊

- 1.社區資訊
- 2.設施介紹
- 3.廠商資料

4.4.3 系統介面

電腦版介面:



圖 4-22

手機板:

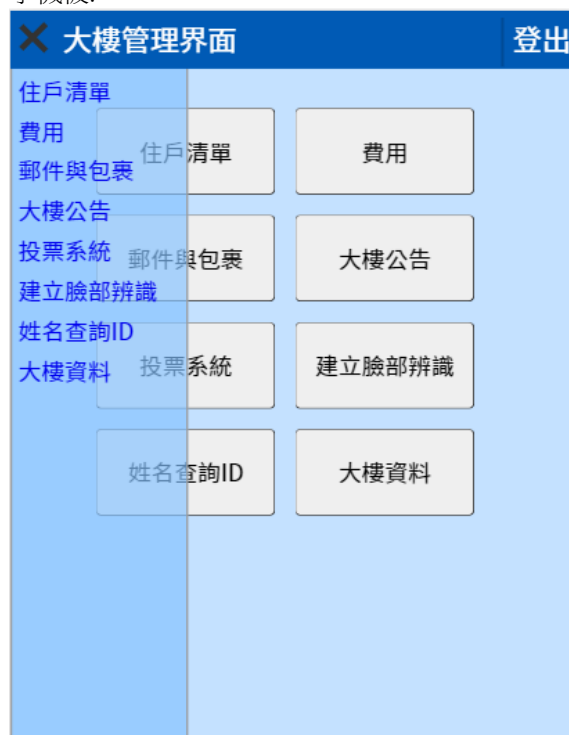


圖 4-23

4.4.4 機器人後臺 API

為了提供前台機器人資料，後端必須建立 api 介面，以提供機器人查詢資料庫內容，以及調用聊天機器人、人臉辨識功能，在此列出所有功能列表。

(1) user_data->用戶資料查詢

(2) all_user_data->全資料輸出

(3) fee->繳費查詢

(4) mail_packet->信件與包裹查詢

(5) notice->公告提醒

(6) vote->投票查詢

(7) new_vote->投票

(8) face_check->人臉辨識

(9) remind->提示訊息

(10) Information->大樓資訊

(11) Facilities_Introduction->設施介紹

(12) Manufacturer_Information->廠商資訊

第五章、結論

本專題完成了一套大樓管理系統，可以使管理員減少負擔，而且更有效率的管理大樓各種大小事項，水電、包裹、投票……等等都可以在我們的平台上操作查詢。在未來大樓運用中，可以設置在大樓門口，如果非住戶且沒有住戶同意讓此人上樓大門是無法開啟的。而當住戶進入時，會自動開啟大門以及電梯到該住戶的樓層，即使雙手提滿物品也可以輕鬆上樓回家。目前的系統只能做到減少管理員的負擔，在未來系統完善後就可以獨立運行，就無需管理員，且擁有更加便利與安全的大樓管理系統。

誌謝

感謝我們的指導老師徐偉智教授在期間教導我們應用的層面，讓我們能夠了解了現今常常不再於技術上的問題，反而是要怎麼解決生活上的問題才是最困難的。也特別感謝王懷毅經理的幫助，給予我們較能貼切社會需求的解決方案、探討面向。

參考文獻

- [1] python+OpenCv+dlib 實現人臉 68 個關鍵點檢測並標註
網址：
<https://www.itread01.com/content/1547447235.html>
- [2] Developing a Face Recognition System for Indoor Security
網址：
https://www.yucelozdemir.com/docs/Report_of_Face_Recognition_System_for_Indoor_Security.pdf
- [3] 中文聊天機器人的實現
網址：
https://blog.csdn.net/zzZ_CMing/article/details/81316033
- [4] 在 Django 使用 MySQL 資料庫
網址：<https://jerrynest.io/django-mysql-database/>
- [5] Django 教學 2: 創建一個骨架網站
網址：
https://developer.mozilla.org/zh-TW/docs/Learn/Server-side/Django/skeleton_website
- [6] 鄧文淵，Python 架站特訓班：Django 最強實戰，基峰出版社(2017)