

第 12 組 智慧電梯人數查詢系統

Background, Motivation, and Goals

Background

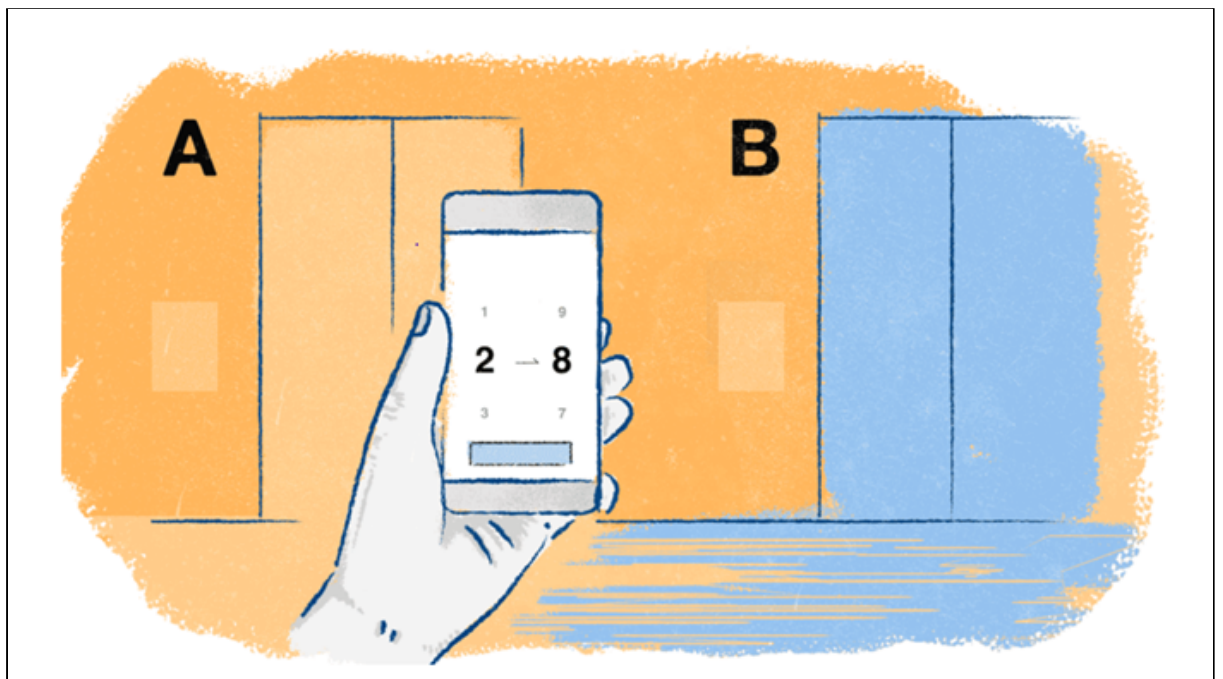
過去電梯的滿載限制是透過重量偵測，但在這樣的衡量標準之下，電梯內若是人數已滿、或是放置了大型的貨件，即便重量尚未達標，也已經容納不下更多的使用者，造成等待電梯的人會需要在等待下一班電梯，可能會耗費更多時間在等待電梯，因此希望能以智慧化的方式減少電梯等待時間，並且可以順利搭乘到電梯，不需再等下一班次。

Motivation

這份 project 的出發點是想針對在尖峰時刻搭乘中央大學女14舍宿舍電梯時，可以避免搭乘的使用者花費更多的等待時間，改善搭乘電梯的擁塞情況，此專案的動機是希望能夠結合 AIoT，透過物聯網來讓電梯更具有智慧化。

Goals

為了提升搭乘電梯的便利性，因此希望透過 camera 遠程拍照，分析對電梯內的滿載程度進行人數偵測，將宿舍電梯運作效率最佳化，所以我們目標是讓有需要搭乘電梯的使用者可以透過 Line Bot 查詢目前電梯附載的情況。



Contribution of each member

Question	Answer	Member1: 丁于晏 110522061	Member2: 葉庭 110522070	Member3: 楊惠隆 108502004
1. What physical scenario and what physical object (or phenomenon)?	elevator image(people)	10%	10%	0%
2. What sensor and platform?	Web:AI 開發板套件, Webduino 平台	20%	10%	0%
3. What protocol to connect to Internet?	Wi-Fi	15%	5%	20%
4. What web site and database?	Webduino 開發平台, Google sheet	30%	10%	0%
5. What the visualizer to visualize data?	python opencv	5%	10%	0%
6. What AI technology to analyze data?	python opencv	5%	10%	50%
7. What insight of your analysis?	透過圖片判斷電梯人數是否達額滿	10%	10%	30%
8. What public dataset or information used?	No	0%	0%	0%
9. What applications of the insight?	判斷人數負載若達上限	5%	35%	0%
Subtotal	\$ 2615	100%	100%	100%

Solutions

這份project主要分為三個部分，分別為使用Web: AI 開發板進行拍攝圖片、將拍攝出來的圖片辨識出有個人以及將人數以及將拍攝出來的圖片回傳到Line Bot中。

1. 使用Web: AI 開發板進行拍攝圖

本專案希望透過遠端拍攝電梯內的負載情況，但由於牽涉到隱私和資安問題，目前是以能正確拍攝到人並判斷出正確人數為主，藉此來模擬我們期望達到的效果。

Web: AI 是一個具有鏡頭且具有 LCD 螢幕的開發板，可以輔助我們本次系統來實作遠端拍攝影像。



首先先將Web:AI 開發板做初始化以及設定 Wi-Fi, 並且更新韌體, 最後透過usb 連接板子到電腦上, 確認 Web:AI 開發板進入 play 模式後即可燒錄程式;本次使用 Webduino 積木程式平台來完成自動化的遠端拍攝, 每隔5秒將會更新一次 Web:AI 所拍攝到的照片。

每次所拍攝到的影像會更新到 google sheet 上, 並且透過 requests 下載圖片, 並將此圖片串到「辨識圖片人數」的程式上。

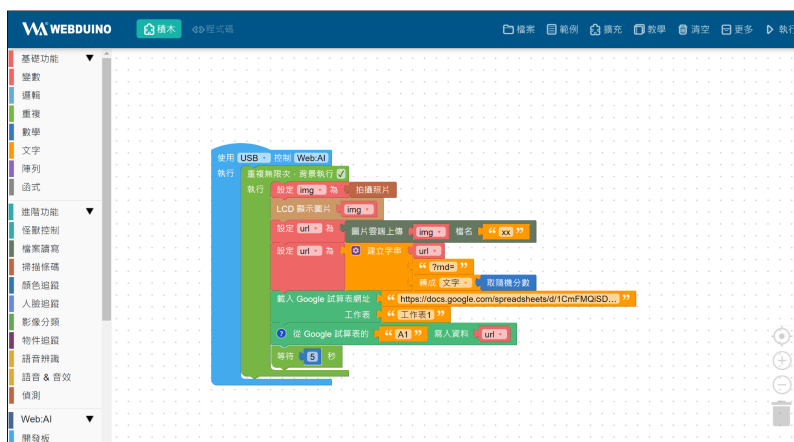


圖 1, Webduino 積木平台以及遠端拍攝程式

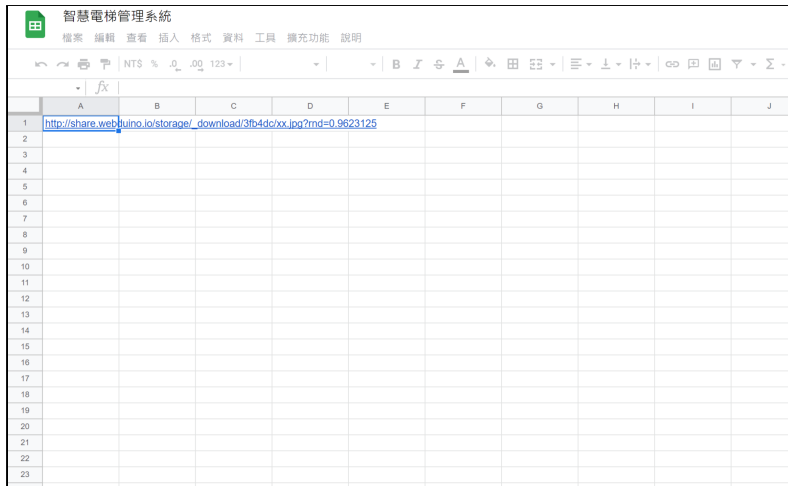


圖 2, 遠端拍攝後儲存在 google sheet

2. 拍攝出來的圖片辨識出有個人

使用google 才colaboratory實現靜態照片的人臉檢測。使用colab的原因是平台擁有完整的第三方函式庫, 並且能使用GPU加速計算。在colab上進行人臉檢測需要使用dlib深度學習庫的face-recognition以及OpenCV。

通過load_image_file將下載的的圖片文件轉變為數組數據, face_locations模塊會對圖像數據進行處理, 定位人臉位置(四值->兩坐標: y1,x1,y2,x2)。根據處理數據的結果數量(list的長度)就能得到人臉數量。

本項目的人臉識別是基於C++開源庫dlib中的深度學習模型, 用的是Labeled Faces in the Wild人臉數據進行訓練, 這個庫的人臉數據幾乎都是外國成人, 因此此人臉檢測模型對亞洲人和小孩的識別準確度有待提升。

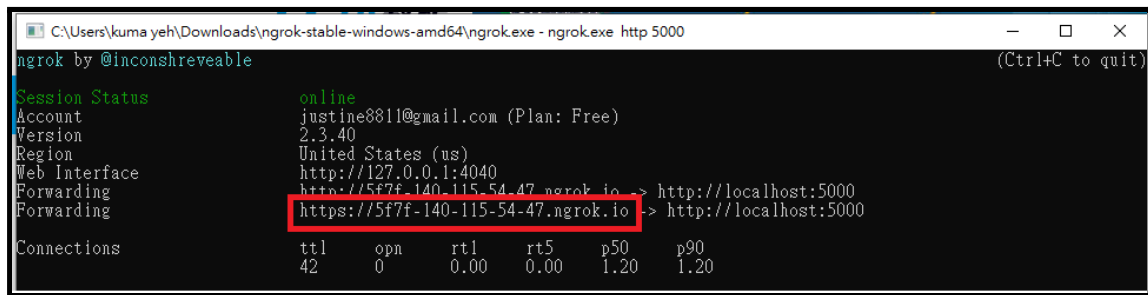
3. 將拍攝出來的圖片回傳到Line Bot中

首先要到 Line developer 官網[1]註冊一個帳號, 取得 channel access token 以及 channel_secret, 並將這兩項資訊加入 config.ini。

接著創建 app.py 檔[2], 創建 Line Bot 基本的設定主要是參考 [3], 裡面主要就是將 Web: AI 開發板拍攝出來的圖片寫入, 以及將辨識出來的人數回傳。由於拍攝出來的圖片以URL 的方式存入 Google sheet 中, 所以需要引入 Google API 的相關套件, 這樣才可以存取 Google sheet。

比較特別需要注意的是在 Line developer 中設定 Webhook URL, 主要是希望Line Bot 可以將訊息發佈到我們的伺服器上面。因為一般來說我們在本機開發時, 會使用開瀏覽器連到 http://localhost 或 http://127.0.0.1 的本機端伺服器預覽成果, 但 Line Bot 會希望可以由外網連線到自己本地的 server, 而 ngrok 是可以讓內網伺服器與外界溝通的一個服務, 因此我們使用這個服務將我們的內網伺服器網

址轉換成外網伺服器網址。主要代換的 URL 如圖 1 框起來的部分。



```
C:\Users\kuma yeh\Downloads\ngrok-stable-windows-amd64\ngrok.exe - ngrok.exe http 5000
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             justine8811@gmail.com (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding            http://5f7f-140-115-54-47.ngrok.io -> http://localhost:5000
Forwarding            https://5f7f-140-115-54-47.ngrok.io -> http://localhost:5000
Connections
  ttl    opn    rt1    rt5    p50    p90
    42     0     0.00   0.00   1.20   1.20
```

圖3：Webhook URL 填入的 URL

最後提供測試出來的樣子。輸入「人數」會回傳目前電梯的人數以及拍攝到的圖片(圖4)，若是輸入跟人數無關的字，就會回傳「請輸入含有'人數'的句子!!!」。(圖5)目前設定電梯已額滿的人數為三人，若人數大於等於3人，會回傳目前電梯人數，圖片，以及電梯已額滿的警訊(圖6)。

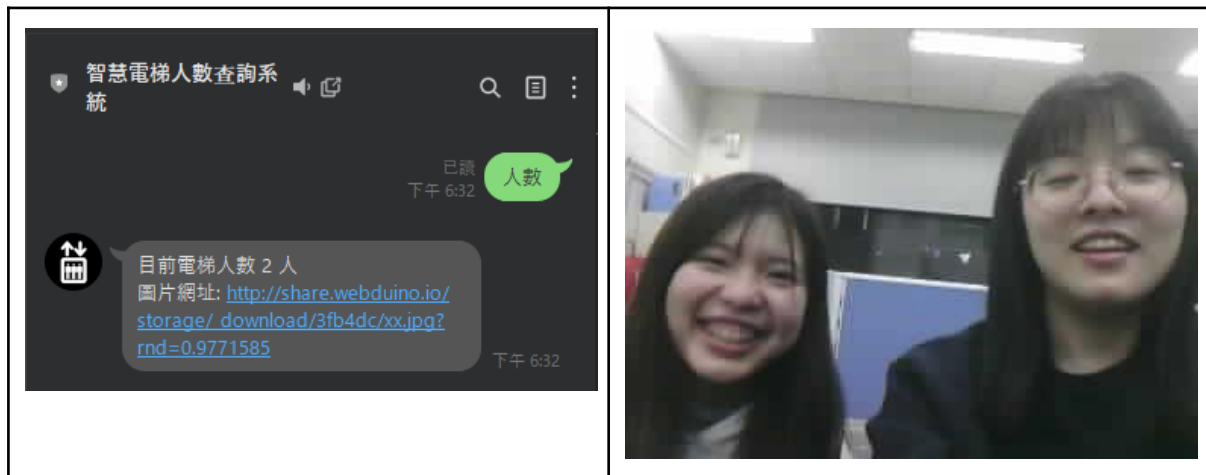


圖4：左半部分為測試圖片 右半部分為下載下來的圖片

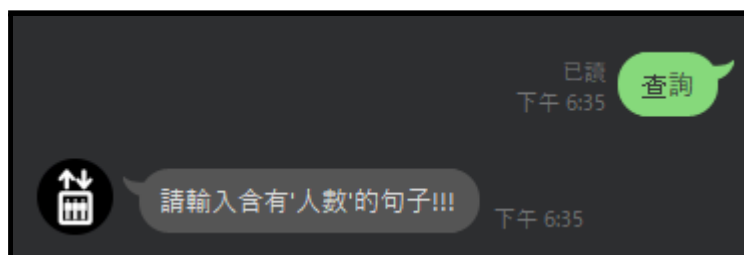


圖5



圖 6 電梯已額滿警訊

Concluding Remarks

這次的IoT報告，我們透過 Web:AI 設計出遠端拍照的功能，以dlib 深度學習庫中 face-recognition 以及 OpenCV 辨識人數，並透過擴充套件 google sheet 以及 LineBot 的結合開發出整套智慧電梯人數查詢系統，未來希望能將系統實際運用在電梯中，並且設立更多台系統在不同的電梯中應用。

Reference

- [1] [LINE Developers](#)
- [2] [Github link](#)
(https://github.com/justine1999/IoT_linebot_Smart_elevator_number_query_system)
- [3] [\[Python Linebot\]教你如何使用Python成功串接Linebot\(2020版\)](#)
- [4] [ngrok 不求人：自己搭一個窮人版的 ngrok 服務](#)
- [5] [CoLab - 图像人脸检测 框出人脸](#)
- [6] [Web:AI 教學大綱](#)
- [7] [Web:AI 遠端拍照](#)