

嵌入式系統技術實驗

期末專題報告

人群數量估計與實聯制結合 LineBot 之應用

組員:0712233 蘇國智、0711070 黃旻濤

一、動機與目的

在疫情肆虐的現在，人們需要保持社交距離，盡量避免群聚，以達到抑制病毒傳播的目的。但是現階段並沒有任何方法可以讓人們在沒有到現場的情況下，了解某一場所的人群情況，往往要到了現場才發現人群擁擠，感染風險大增，這時再離開卻也已經暴露在染疫的風險下了。因此我們想到要透過物件辨識的方法來實時監控現場人數，並透過 Line Bot 來向使用者提供包含人數、現場照片等資訊。另外，我們也推出一個基於 Line Bot 更為安全方便的實聯制方案，來取代現行的簡訊實聯制。

二、作品介紹

整個系統大致可以分為三個部分，分別是機器學習-物件偵測、Line Bot Message 及實聯制系統。

機器學習-物件偵測使用 Nvidia 提供的 Jetson Inference 中的 Object Detection package，使用的 model 是 Nvidia 提供的訓練好的 Ped-100，該 model 是專門針對行人偵測訓練。

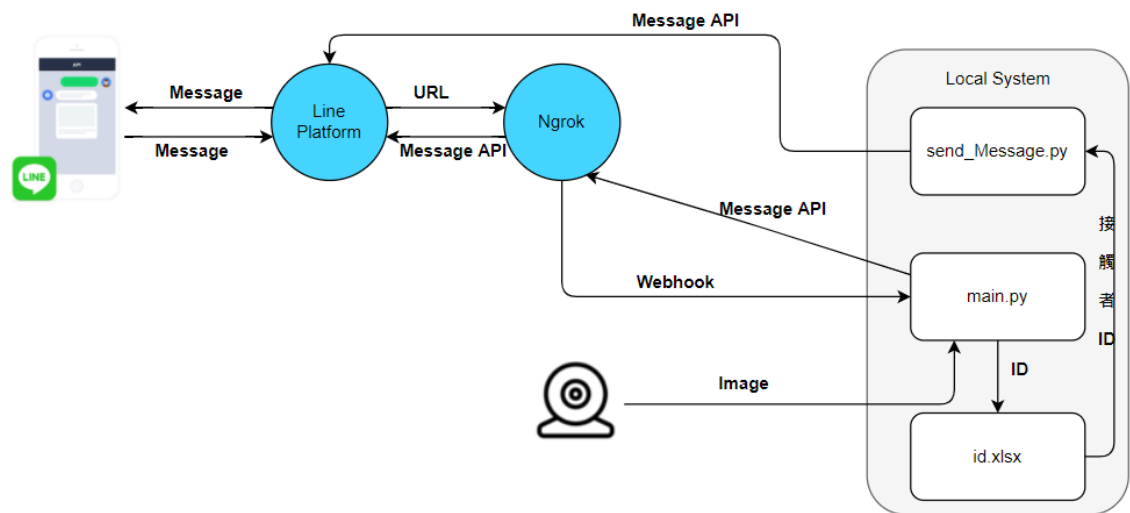
Line Bot Message 則依據 Line Developer 所提供的 Message API 進行開發，並利用 ngrok 作為伺服器。主要使用了 Message API 中的 Text Message、Image Message 及 Template Message 等。

實聯制的部分是基於 Line 所構建的透過記錄 Line 內部 ID 及使用者進出場時間進行登記。以下是 Line 實聯制與簡訊實聯制的比較：

	Line	簡訊
訊息傳遞方式	Wifi、行動數據	行動通訊
隱私	提供 Line 內部 ID	提供手機號碼
方便性	較常使用	較少使用
價格	4G 網路預付卡 1G=180 元 1G 約可傳送 1000 萬字 1 個字 0.000018 元	一則約 1 元 1 則 70 個字 1 個字 0.014 元

透過表格，可以知道 Line 實聯制相較於簡訊實聯制，在隱私方面更為安全，相較於 Line 內部 ID，手機號碼容易受到他人騷擾；在方便性方面，現代人以極少使用簡訊，更常使用諸如 Line 之類的通訊軟體；在價格方面 Line 則明顯優於簡訊。

三、系統架構圖



四、設計方法

首先，介紹本次期末專題使用的硬體設備，包含 Nvidia Jetson nano，Logitech Webcam，以及鍵盤、滑鼠與螢幕（如右下圖）。

Nvidia Jetson nano，是 Nvidia 為嵌入式系統設計的人工智慧開發板，適合在板子上跑一些深度學習與 AI 的套件模組，包括這次選用的人像辨識模組 ped-100。

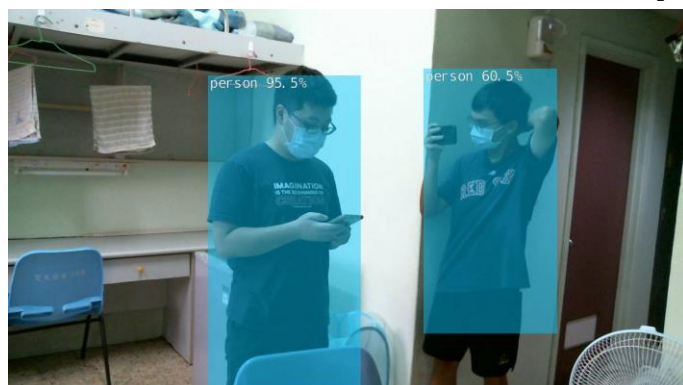
Logitech Webcam 作為人像辨識的攝像頭以及 Line Bot 拍照功能的鏡頭；最後搭配鍵盤、滑鼠與螢幕，完成系統的搭建。



接下來，提到軟體與程式的部分。在機器學習這方面我們使用的是 Nvidia 提供的 Jetson Inference 這個體驗套件，他支援 C++ 及 Python 兩種語言，我們選用的是 Python。在 Jetson Inference 中提供幾個已經訓練好的 Model 讓開發者可以直接使用(如右圖)。如圖所示，有相當多的 Model 供開發者使用，例如 SSD-Mobilenet-v2 是用來辨識日常所見的東西，涵蓋了 91 個類別包括人、電腦、食物等等；multiped-500 用來辨識行人與行李；DetectNet-COCO-Dog 則是用來辨識狗。而 ped-100 專門辨識行人，剛好符合我們的人像辨識需求，因此我們選用 ped-100 這個已經訓練好的 model 作為我們系統開發的基礎。

Network
SSD-Mobilenet-v1
SSD-Mobilenet-v2
SSD-Inception-v2
DetectNet-COCO-Dog
DetectNet-COCO-Bottle
DetectNet-COCO-Chair
DetectNet-COCO-Airplane
ped-100
multiped-500
facenet-120

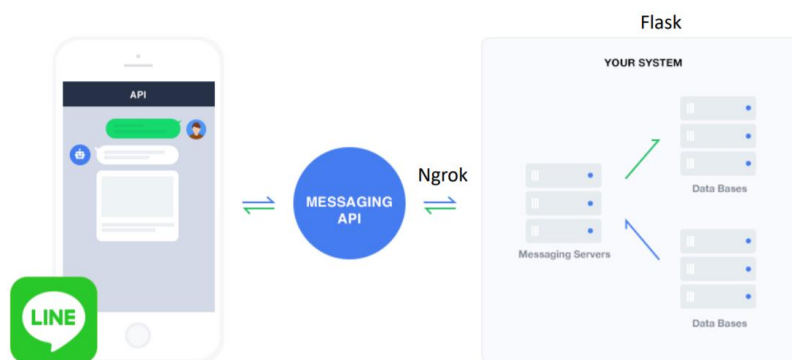
Ped-100 模組的運作模式是透過攝像頭錄影，辨識出畫面中的人像，並透過適當的渲染，使用藍框框將人框出並標示為 person(如下圖)。

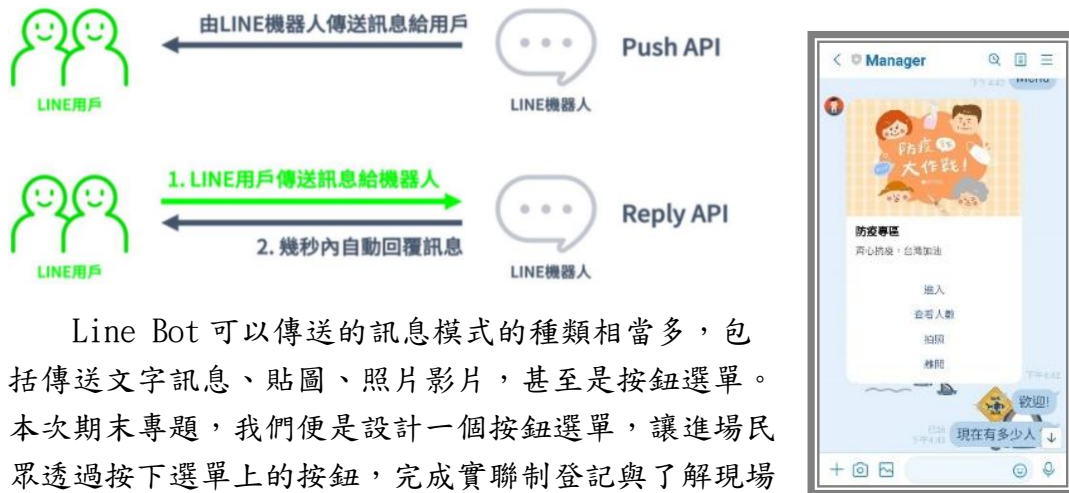


PedNet 在某些情況下，並不能正確的辨識行人，像是側身導致特徵過少，或者是部分軀幹被阻擋都導致無法辨識。同時，有些諸如衣服等物也有可能被辨識為行人，導致誤差。另外因為疫情關係，我們有合適場地進行距離測試，目前可以確定的是 10 公尺內都能正常辨識。

再來，我們設計了一款名叫 Manager 的聊天機器人，提供進入室內場所的民眾一些現場的資訊，例如現場的人數、現場拍照的畫面，以及作為實聯制登記的平台。當民眾一來到場所的門口時，透過掃描 QRcode，與聊天機器人成為好友，就可以取得現場資訊與完成實聯制登記。當現場人數超過一定的數量時，Line Bot 會發出通知給所有在場的民眾(點擊進入，尚未點擊離開之民眾)，建議民眾離開現場，避免人潮過度群聚。

Line Bot 發送訊息主要是透過 Messaging API(如下圖)。其中的 Reply API，讓用戶發送訊息，幾秒後機器人回應訊息。當用戶發送一個訊息給聊天機器人時，Line Platform 將一個 webhook 事件傳送至 bot server 的 webhook URL。之後 Bot server 將依據 webhook event，透過 Line Platform 使用 Message API 回應用戶。而另一種訊息發送模式是藉由 Push API，使聊天機器人主動傳送訊息給用戶；實作中發送人數過多的警示訊息就是透過此方法傳送。





Line Bot 可以傳送的訊息模式的種類相當多，包括傳送文字訊息、貼圖、照片影片，甚至是按鈕選單。本次期末專題，我們便是設計一個按鈕選單，讓進場民眾透過按下選單上的按鈕，完成實聯制登記與了解現場狀況(如右圖)。

至於實聯制登記的部分，當使用者於 Line Bot 畫面點擊「進入」後，使用者的 Line ID 與進場時間會透過我們幕後執行的程式，寫入一個 excel 檔案中。當使用者離場時，點擊畫面中的「離開」後，其離場時間就會被記錄下來，寫在檔案中。

最後，我們設計了另一支程式，完善實聯制登記的功能。當曾經進入本場所的民眾是 COVID-19 的確診者或接觸者時，我們可以透過實聯制登記的檔案，找出當時在現場的其他民眾，並透過機器人主動發出訊息的傳送模式，通知相關人員留意身體狀況與聯繫疫調單位。

在開發系統的過程當中，我們遇到兩個問題：

第一，人像辨識系統與 Line Bot 會爭奪有限的資源而無法同時執行。因此我們做出調整：採用多執行緒的設計。透過 Python 內建的 threading 模組，將人像辨識系統設定為子執行緒，主執行緒為 Line Bot 系統。

第二，持續的進行影像辨識會佔據大量運算資源，從而使的其他功能無法順利運行。因此，我們在影像辨識的迴圈中加入 sleep(1)使程式停滯 1 秒再繼續運行，達到輕量化程式的目的。如此一來就能順利運行所有功能。

五、成果

- YouTube 影片 <https://www.youtube.com/watch?v=RE-xpL6Vj6U>
- Demo 影片 <https://www.youtube.com/watch?v=tvWMhzVsulY>

六、結論

這次專題我們整合了多個不同的系統，花了大量時間調適，最後成功達成一開始設立的預期目標，甚至猶有超過，新增了不少功能。整個系統的完成剛好遇上疫情爆發，正是需要這套系統的時候。未來我們希望能透過自己訓練 model 改善 Ped-100 辨識問題，以及改善 Line Bot 介面等方式改進整個系統，並希望有機會可以將整個系統付諸實用。

七、工作分配

工作項目	負責人	說明
程式編寫	蘇國智	人像辨識與系統整合
程式編寫	黃旻濤	Line Bot 設計與實名制
影片錄製與編輯	黃旻濤	
影片解說	蘇國智	
投影片製作與報告	蘇國智	
結報撰寫	黃旻濤、蘇國智	

八、參考資料

[1]Dustin Franklin. GitHub jetson-inference

Retrieved from <https://github.com/dusty-nv/jetson-inference>

[2]Line Developer, Message API overview

Retrieved from <https://developers.line.biz/en/docs/messaging-api/overview/>

[3]LineBot+Python，輕鬆建立聊天機器人

Retrieved from <https://blackmaple.me/line-bot-tutorial/>

[4]Line. GitHub line-bot-sdk-python

Retrieved from <https://github.com/line/line-bot-sdk-python>