深度學習手勢與語音居家操控

Deeplearinggesturesand voice home control

指導教授：李國川老師

學生：李啟弘、林益均、沈明毅、吳昇翰、呂安修

國立聯合大學資訊工程學系

苗栗市南勢里聯大 2 號

gclee@nuu.edu.tw

摘要

近年使用電視者的比例慢慢的高於使用手機者，因此我們決定將此套系統應用在電視、機上盒、與小米盒子中。將居家環境，加上 AndroidTV、Google ReSpeaker、語音辨識、手勢辨識、深度學習、網路爬蟲，形成深度學習手勢與語音居家操控。

以語音辨識與手勢辨識作為基礎，藉由連結 Raspberry Pi 3 透過 GoogleReSpeaker，將需求說出來，即可做出相對應舉動;然而有些家庭當中有瘖啞人士,我們運用深度學習與 WebCam 結合進行手語辨識，讓他們也能方便的使用此套系統，簡易的操作概括整個居家，提升生活的便利性及舒適度。

這套系統結合深度學習與網路大數據，將獲得的資料分別進行處理與分析，如：天氣、股票、房價、娛樂、新聞、交通等資訊。透過爬蟲將數據回報給使用者，藉此打造智慧語音管家與智能居家影音服務。

關鍵字：生活居家、大數據爬蟲、手勢辨識、語音辨識

一、前言

因年紀較大的人們會操作使用電視的比例高於使用手機者，因此我們決定將此套系統應用在電視、機上盒、與小米盒子中。簡單的操作，讓他們能夠更得心應手的使用，不必再去煩惱如何使用，因為只要使用聲音或者手勢便可簡單的操作。

不只瘖啞人士可利用深度學習手勢與語音居家操控，讓生活變得更方便，未來手勢操控將會成為一種趨勢，科技始終於人性，手勢辨識能為生活添增更多便利性。

現在電視通常配有一個機上盒，可是卻沒有隨著時代而有智慧化的功能，因此我們開發了此套系統應用於電視與小米盒子中，方便年紀較大、不會使用手機的長者操作；使用 Raspberry Pi 3 做為主機、4 麥克風陣列擴充板(Respeaker)，透過 GoogleReSpeaker，將需求說出來，即可做出相對應舉動；結合深度學習與網路大數據，如：天氣、股票、房價、娛樂、新聞、交通等資訊，透過爬蟲將數據回報給使用者，並將查詢的結果呈現在 Andriod TV 上。因為有些家庭當中有瘖啞人士，因此我們藉由深度學習與 WebCam 進行手語辨識，讓瘖啞人士也能即時且輕鬆的操作此套系統。

二、系統技術與架構

1.深度學習與手勢辨識：

每種手勢各 100 張圖片，共 1900 張做為訓練資料，接著為每張圖片框出 bounding box 並編列所屬之標籤，產生 XML 檔，將 1900 張圖片的 XML 檔轉換為 YOLOV3 的訓練格式後，切割訓練與測試集，建立 YOLOV3 之設定檔，進行深度學習建立模型。

透過 WebCam 串流拍攝，配合深度學習建立的模型進行手勢辨識，將辨識結果組織成語句，透過 dialogflow 進行語意分析，並根據分析結果進行網路爬蟲。

1-1 手勢辨識

透過深度學習技術使機器學習辨識特定手勢，將深度學習之輸出作為手勢特徵用於辨識系統之辨識規則以利於辨識系統的輸出結果。基於此研究目的，本研究將分為兩部分，地步分為透過深度學習實作手勢分辨，第二部分為實作辨識系統的特定手勢辨識。

每種手勢各 100 張圖片，共 1900 張做為訓練資料，接著為每張圖片框出 bounding box 並編列所屬之標籤，產生 XML 檔，將 1900 張圖片的 XML 檔轉換為 YOLOV3 的訓練格式後，切割訓練與測試集，建立 YOLOV3 之設定檔，

進行深度學習建立模型。

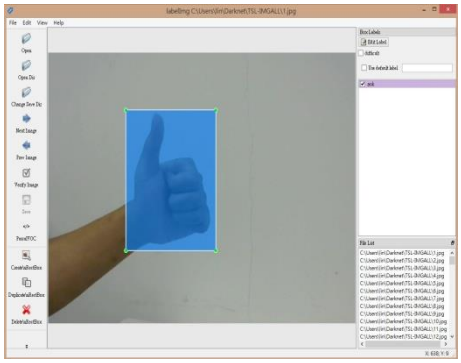

圖 1. bounding box 圖

```
<annotation>
    <folder>all</folder>
    <filename>1.jpg</filename>
    <path>C:\Users\User\Desktop\TSL\photo\all\1.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>640</width>
        <height>480</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>ask</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>162</xmin>
            <ymin>102</ymin>
            <xmax>349</xmax>
            <ymax>351</ymax>
        </bndbox>
    </object>
</annotation>
```

圖 2. XML 檔圖

YoloV3 使用 resnet 網路(Residual Network) 新的基底網路為 Darknet-53，有 53 層的卷積層和池化層，採用了一般類神經網路加深時常用的 ResNet 結構來解決梯度問題，以及引進 Faster RCNN 的 anchor 設計所以最後一層是卷積層輸出。另外 YoloV3 使用 FPN 網路(Feature Pyramid Networks)提升小物體偵測能力，從(縮小 1/32、縮小 1/16 和縮小 1/8)三個尺度分別去做偵測，每個尺度各帶入 3 個 anchor。使用 FPN 的架構可以讓低層較佳的目標位置和高層較佳的語義特徵融合，並且在不同特徵層獨立進行預測，使得小物體檢測改善效果十分明顯。

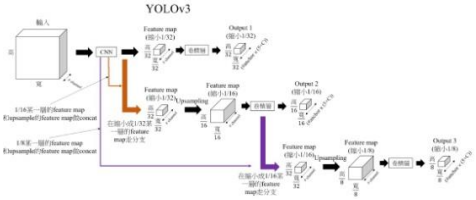

圖 3. YOLOv3 網路圖

本專題採用 YoloV3 開始訓練後，每一百次儲存一個權重檔，當產生權重檔時即可停止，但為求較高的準確率，因此我們觀察平均 loss 值開始產生震盪時(如圖 6)，停止訓練並取最後一次的權重檔。

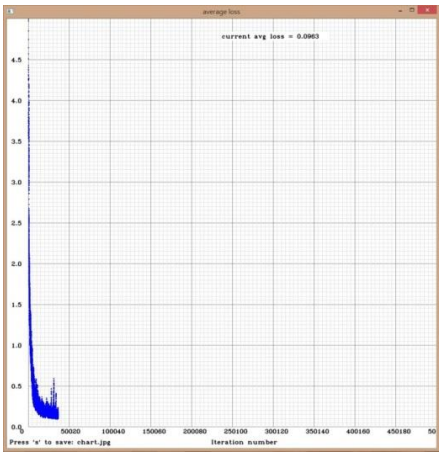

圖 4. 平均 loss 圖

透過 WebCam 串流拍攝，配合深度學習建立的模型進行手勢辨識，將辨識結果組織成語句，透過 dialogflow 進行語意分析，根據分析結果進行相對應之爬蟲程式，爬蟲結果存入資料庫，再利用文字轉語音，將結果透過 Raspberry Pi 3 告知使用者。

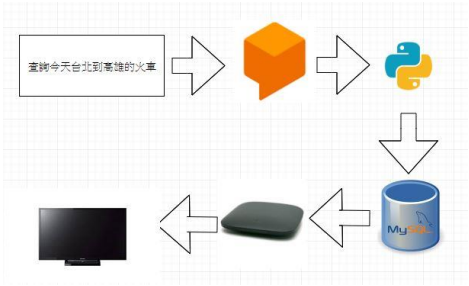

圖 5. 手勢流程圖

2. 語音辨識：

語音辨識使用 Google 所提供之語音套件「Google Text To Speak（簡稱 GTTS）」結合 Dialogflow 語意分析平台進行實作，設定 Entity（實例），切割語句的關鍵字：事件、時間、地點等；再依照關鍵字的組合來進行 Intent(意圖）分類，在各 Intent 上形成訓練短句，對個別短句進行回應句和參數設定，最後將分析好的語句透過 Google Api 封裝成 JSON 檔回傳到程式。

例如：「查詢今天苗栗到台北的火車時刻」及「查詢今天苗栗到台南的高鐵時刻」，皆為詢問時刻相關的語意，經由訓練後的模組，區隔出語句詢問方式、城市地點和時刻表種類的不同，以達成語意分析。

2-1 語意分析

透過 Google ReSpeaker 抓取使用者的語

音，使用 Google Speech to Text 將語音轉換成文字，Dialogflow 根據 Intent (意圖) 與 Entity (實例) 對文字進行辨識，了解使用者的語意。

語意分析的事前設定有：1.設定 Entity (實例)，讓機器人知道有那些關鍵字。比如要查詢火車時刻、還是房價，同義字也可指向相同關鍵字，Dialogflow 內部也有提供 System Entities，例如；City(台北、桃園…) 、時間……等。
2.Intent(意圖)分類，依照關鍵字來進行分類，在各意圖上形成訓練短句、根據語序不同也能導向相同結果，以及對個別短句進行回應句和參數設定、以及重點參數的部分做出提示。

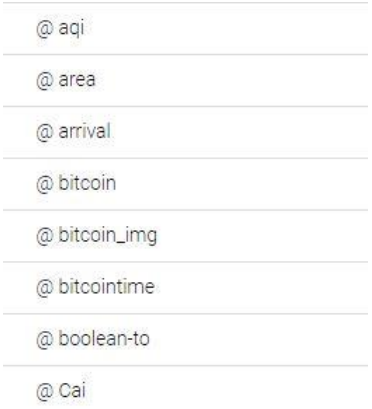

圖 6.設定實例

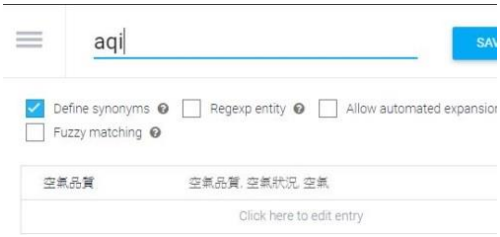

圖 7.新增實例

圖 8.意圖分類

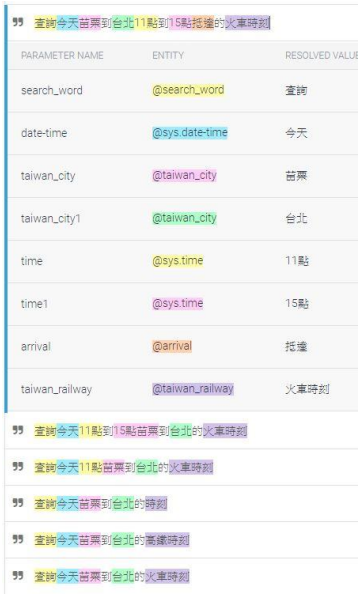

圖 9.形成訓練短句

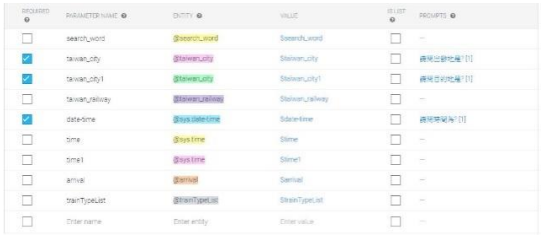

圖 10.參數設定、以及回應

圖 11.驗證結果

Dialogflow 會將所有使用者語句的意圖類別記錄下來，當發生錯誤時，只需針對錯誤的部分作修正，進行後續 Training，降低誤判機率。

圖 12.使用者語句與意圖類別記錄

3. 大數據網路爬蟲：

使用 Python 撰寫網路爬蟲，至台鐵網站抓取時刻表資料，查詢各個網頁標籤，找出底下的時刻資料，或者至高鐵網站抓取時刻之 JSON 檔，查詢各開頭陣列名稱，找到底下的時刻資料，進行彙整，體現簡潔的資料型態供使用者觀看，並進一步將結果透過語音回報給使用者。

4. 語音接收與查詢：

透過四麥克風陣列擴充版(ReSpeaker)接受查詢指令，指令透過語音進入 GoogleGTTS，轉換為文字並進行切割後，將關鍵字放入爬蟲程式當中，將抓取到的 JSON 欄位、元素代號位置進行篩選、比對，將日期、時間、站名、乘車時間等欄位資訊，進行整理後，傳入資料庫端，透過 Android TV 體現出整理好的資訊，並經由 Google Text To Speak，轉換為結果，透過 Raspberry Pi 3 的語音輸出呈現。

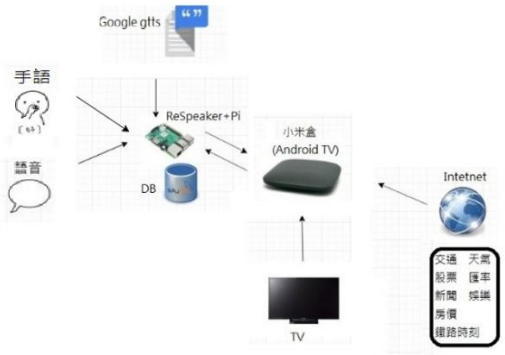

圖 13.系統架構圖

三、專題實作部分

(1)智慧網路資訊查詢

1. 台鐵高鐵網站:

藉由 dialogflow 辨識語意，獲得結果參數，將結果參數做預先處理，台鐵部份先將站名轉換成編號-站名的形式、依據查詢出發或抵達給定 true or false、根據查詢對號、非對號或全部給定 RESERVED_TRAIN or NON_RESERVED、ALL。將這些資料傳送至台鐵網站抓取時刻表資料，查詢各個網頁標籤，找出底下的時刻資料，擷取最近五筆資料，並傳送至資料庫，再利用文字轉語音，將近一筆資料透過 Raspberry Pi 3 告知使用者班次、出發時間、到站時間、旅途時間。高鐵部份則是將站名轉換為高鐵之編碼，將資料傳送至高鐵網站抓取時刻之 JSON 檔，查詢各開頭陣列名稱，找到底下的時刻資料，擷取最近五筆資料，傳送至資料庫，再利用文字轉語音，將近一筆資料透過 Raspberry Pi 3 告知使用者班次、出發時間、到站時間、旅途時間，使用者也可以透過 AndroidTV 查看查詢結果。

圖 14.時刻表畫面圖

2. 比特幣匯率圖:

藉由 dialogflow 辨識語意，獲得結果參數，將結果參數做預先處理，透過參數判定查詢的幣值及時間，根據以上兩種參數抓取 CoinGecko 網站之 JSON 檔，再將資料透過 Python 內建函數 matplotlib 繪製成圖表並將圖表儲存成 JPG 檔，最後利用 Python 內建函數 FTP 將圖檔上傳至 FTP 主機之資料庫，使用者再透過操作 AndroidTV 查看查詢結果。

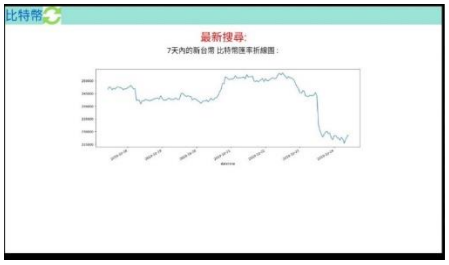

圖 15.比特幣畫面圖

3. 民調&新聞查詢

使用者語音或手勢輸入後，轉成的字串藉由 Dialogflow 進行語意辨識，將獲得的結果參數預先處理後判斷執行那些總統候選人的民調查詢方式或者是執行新聞的查詢。執行查詢民

調的話，從資料庫中抓相關的民調資料跟以及圖表資料，透過 AndroidTV 可以看到呈現的結果，配合著 JavaScript 的使用，使用者可以動態的瀏覽各種時間或是各種民意調查單位的民調指數。如果資料庫中沒有最新的民調資料的話，則會根據「維基百科-2020 年中華民國總統選舉民意調查」網站中的資料，使用 Python 中 BeautifulSoup 套件進行爬蟲後傳至資料庫，並用 Matplotlib 套件將所得資料製成圓餅圖，圖表再傳至 FTP 主機。執行新聞查詢的話，會根據輸入的關鍵字，一樣使用 BeautifulSoup 套件爬取 Google 和 Youtube 的資料並存進資料庫，最後可以透過資料庫中的資料在 AndroidTV 瀏覽。

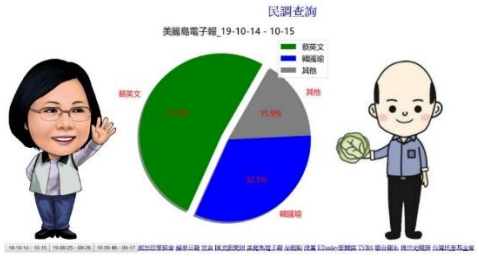

圖 16.民調畫面圖

4. 天氣、交通、及空氣品質

使用者語音或手勢輸入後，轉成的字串藉由 Dialogflow 進行語意辨識，將獲得的結果參數預先處理，判斷關鍵語句後決定前往所需的網站(天氣→中央氣象局、空氣品質→環保署)，抓取所需地點的 JSON 檔，分析此 JSON 檔的資料並加以處理，將處理完的數據上傳至資料庫，再利用文字轉語音，將結果透過 Raspberry Pi 3 告知使用者，最後可以透過資料庫中的資料在 AndroidTV 瀏覽。

圖 17.空氣品質畫面圖

5. 房價查詢

使用者語音或手勢輸入後，例如:查詢台北房價，轉成的字串藉由 Dialogflow 進行語意辨識，透過參數判定查詢地區，指令將會傳至 Rasperry Pi 3 做處理，透過爬蟲程式，爬取內政部近 8 年房地產交易資料，將抓取到的資料進行整合與分析，並用 Matplotlib 套件將所得資料製成折線圖，圖表再傳至 FTP 主機。呈現出該地區近八年房價的折線圖與平均房價的折線圖。再透過小米機上盒的 APP，將結果顯示在 Android TV 上。

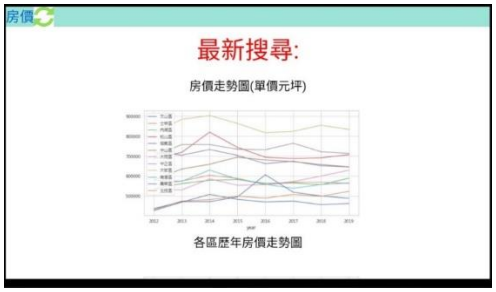

圖 18.房價畫面圖

6. 手勢辨識

目前我們可辨識的手勢有 18 種(如圖 8)，例如我們要查詢從台北到高雄的火車時刻表，依序比出查詢、台北、高雄、火車、結束(如圖 9)，將辨識結果組成字串，並透過 dialogflow 辨識語意為查詢火車，接著將回傳的資料透過火車的爬蟲程式，先進行資料的預先處理，再將整理好的資料傳送至台鐵網站，找出時刻資料，擷取最近五筆資料，並傳送至資料庫，再利用文字轉語音，將近一筆資料透過 Raspberry Pi 3 告知使用者班次、出發時間、到站時間、旅途時間。

圖 19. 手勢語意圖

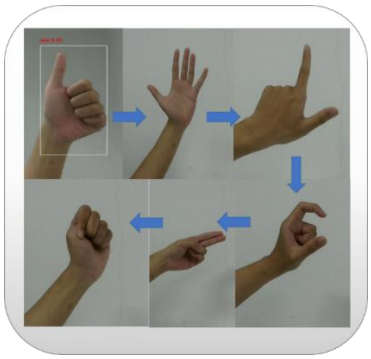

圖 20. 手勢組合語句圖

四、結論

本次的專題製作，使用很多網路上的大數據，透過資料的分析、並利用深度學習套件，應用在：天氣、股票、房價、娛樂、新聞、交通…等資訊，讓智慧居家系統越來越聰明；並且透過語音與手勢辨識這些簡單的操作，讓使用者能夠更輕鬆地去使用。

這次專題做的系統是較為龐大、整合型的系統；再製作專題的過程中，學到了很多之前不曾接觸過的技術，在這過程中，碰到了許許多多的挫折，例如:程式的撰寫與除錯，花了許多的時間在網路上查詢如何解決這些問題。在這碰撞及挫折過程中，讓我們思考如何去運用大學這四年間所學的種種，並加以融會貫通。

五、參考文獻

參考網站

【1】python 爬蟲

https://buzzorange.com/techorange/2017/08/04/python-scraping/

【2】Google gTTS 文字轉語音

http://yhhuang1966.blogspot.com/2017/08/google-gtts-api.html

【3】YoloV3cfg 檔解讀(一)

https://medium.com/@chih.sheng.huang821/深度學習-物件偵測 yolov1-yolov2 和 yolov3-cfg-檔解讀-75793cd61a01

【4】YoloV3cfg 檔解讀(二)

https://medium.com/@chih.sheng.huang821/深度學習-物件偵測 yolov1-yolov2 和 yolov3-cfg-檔解讀-二-f5c2347bea68