|  |
| --- |
| 人工智慧導論期末專題書面報告 |
| 我們都是「好朋友」— 情緒辨識機器人實作 |
| 第25組  0819806\_劉潔昕、0819823\_陳子祈 |

一、專題概要

*我們都是好朋友，讓好朋友溫暖你的每一天！*

後疫情時代，我們經常要面臨獨自待在家的情況，因此開發能夠給予陪伴的機器人，成為現在的新興發展領域之一。**一台能讀懂人情緒並且給出回應的機器人相信可以成為大家生活當中的好夥伴**，以此作為我們的專題動機，我們想要實作出一台「好朋友」機器人，能夠辨識簡單的情緒並給出回應。

我們的專題是結合**DeepFace**模型和**Webduino**開發版來實現好朋友機器人。實作流程是透過在Colab中，藉由Webcam控制電腦鏡頭抓取照片，再將照片丟到DeepFace中進行辨識，最後將辨識結果更新到雲端的試算表中；而Webduino端接收到遙控器訊號，藉由WiFi讀取試算表的內容，依照讀取內容給出相對應的回應。

專題成果會分成兩個部分做技術簡介，分別是Colab中的程式以及Webduino機器人的線路和程式，在當中會進一步介紹DeepFace模型和Webduino開發版特色。

二、專題成果

(一) Colab程式

程式目標是要藉由Webcam控制電腦鏡頭抓取照片，再將照片丟到DeepFace中進行辨識，最後將辨識結果更新到雲端的試算表中。以下紀錄程式流程，並會在當中適時補充重要函式的功能。

* Webcam前置設定

|  |
| --- |
| 1. |
| 一張含有 文字 的圖片  自動產生的描述 |
| 由於我們這次專題需連網使用電腦攝影機，因此需匯入Javascript、cv2、PIL、io、html等模組。 |
| 2. |
| 一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述 |
| 接下來我們先定義Js\_to\_image、Bbox\_to\_bytes兩個之後會用到的函式。   * Js\_to\_image函式說明：  |  |  | | --- | --- | | 函式功能 | 我們先定義一個將Javascript轉opencv的圖片的函式，使之後可把網路攝影機拍到的照片轉成比較容易處理的圖片格式。 | | 輸入 | 包含攝影機拍到圖片的Javascript物件 | | 過程 | 將Javascript物件轉為numpy array，再將numpy array轉為 cv2物件 | | 輸出 | cv2物件 |  * Bbox\_to\_bytes 函式說明：  |  |  | | --- | --- | | 函式功能 | 將臉部偵測到的方形區域的邊界圖從cv2格式轉為base 64 bytes string，使之覆蓋在攝影機串流畫面之上 | | 輸入 | cv2格式的方形區域的邊界圖 | | 過程 | 將輸入的cv2圖轉為PIL圖，再將PIL圖轉為png圖存檔，再將存檔的圖檔轉為base 64 bytes string | | 輸出 | base 64 bytes string | |
| 3. |
|  |
| 再來我們將使用cv2已經訓練好的Haar Cascade face detection model來跑簡易的物件(臉部)偵測。 |
| 4. |
| 一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述 |
| 接下來我們使用取得photo資料，並使用之前定義過的js\_to\_image函式轉換成cv2圖檔。我們也使用Haar Cascade face detection model將臉部偵測到的邊界以cv2內建的長方形圖檔覆蓋原本的圖檔並存起來，最後回傳檔名。 |
| 5. |
| 一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述 |
| video\_stream這段副程式就是在幫助攝影機建立連線、設定開啟攝影機拍攝畫面，video\_frame這段副程式則是先將傳入參數傳到video\_stream的stream\_frame函式以顯示上一格畫面(包含上一格原始畫面、臉部偵測邊界、臉部表情字串結果)，並將下一格攝影機拍到的畫面傳到data變數並回傳此變數，以供之後做臉部偵測及表情辨識之用。這時還先不要呼叫這兩段副程式，還需先設定好gspread、deepface才可呼叫此段副程式。 |

* 連結雲端試算表

|  |
| --- |
| 1. |
| 一張含有 文字 的圖片  自動產生的描述 |
| 想將後面使用deepface辨識出來的表情儲存在google sheet，就得使用gspread來更改Google sheet的內容。 我們使用pip下載gspread、認證，取得gspread存取google sheet的授權之後，就可以將表情內容的字串儲存在google sheet的工作表1。 |

* DeepFace

|  |
| --- |
| 1. |
| 一張含有 文字 的圖片  自動產生的描述 |
| 接下來要準備使用deepface中的表情辨識model，因此使用pip下載好deepface，並import Deepface進來。 |
| 2. |
| 一張含有 文字 的圖片  自動產生的描述 |
| Gspread、deepface都下載好了，接著就可以呼叫video\_stream副程式開啟攝影機拍攝畫面。將之後要存臉部偵測到的邊界圖檔初始化為空字串。 |
| 3. |
| (1)  一張含有 文字 的圖片  自動產生的描述  (2)  一張含有 文字 的圖片  自動產生的描述  (3)  一張含有 文字 的圖片  自動產生的描述  一張含有 文字 的圖片  自動產生的描述 |
| 全部程式碼最重要的就是這個while迴圈。此迴圈是一個無止盡的迴圈，除非操作者點擊畫面，不然拍攝畫面就會不斷進行下去。此迴圈有三個步驟：   1. 呼叫video\_frame函式，將上一格拍攝到的原始畫面加上臉部偵測長方形、deepface model臉部表情辨識結果作為輸入傳進video\_frame函式，將呼叫video\_stream拍攝到的這一格畫面的圖檔作為輸出傳回來並存在img。 2. 將此圖檔img傳到deepface analysis函式做人臉表情辨識，回傳結果到emo，使max\_emo為emo中權值最高的表情，並將max\_emo傳到google sheet的工作表1的A1格。   將圖檔從彩色轉為灰階，丟到Haar Cascade face detection model做臉部偵測，並將結果傳回到faces，根據faces中偵測到臉部長方形的四個點畫出cv2長方形的圖暫存到bbox\_array，也將人臉表情辨識最高權值的結果max\_emo這段文字也暫存到bbox\_array，再調整bbox\_array的size，之後呼叫先前定義過的bbox\_to\_bytes將bbox\_array轉為base 64 bytes string使之在下一次while迴圈作為輸入傳進video\_frame函式。   * Deepface.analyze補充說明：   # DeepFace.analyze(img\_path = "img.jpg", actions = ['age', 'gender', 'race', 'emotion'])  一張含有 文字 的圖片  自動產生的描述  此表情happy的權值最高   |  |  | | --- | --- | | 函式功能 | 如果輸入圖檔有被辨識到臉部則回傳此臉的年齡、性別、種族、表情的辨識結果，如果沒辨識到臉部則中斷執行程式。其中表情會回傳angry, fear, neutral, sad, disgust, happy and surprise七個表情特徵的權值。 | | 輸入 | 臉部圖檔、欲辨識動作 | | 過程(只列辨識表情過程) | Deepface.analysis建構過程如下:    使用fer2013.csv當作訓練的dataset。    訓練和測試的dataset可以儲存到特定變數。    Deepface使用tensorflow backend建構Keras的CNN模型。  此架構使用了三層的convolution layer及一層的fully connected neural networks。    損失函數將是採用[cross entropy](https://sefiks.com/2017/12/17/a-gentle-introduction-to-cross-entropy-loss-function/" \t "_blank)，因為這個個過程屬於多類分類。Fit\_generator是拿來訓練資料用。      可印出訓練及測試dataset的score，score[0]代表loss、score[1]代表accuracy。  但我們要做的是多類分類，上面印出的結果比較適合二元分類，所以我們應該建構出confusion matrix。    上圖為confusion matrix的範例。左行為實際上dataset的標記，上列代表程式辨識出來圖片的表情結果。例如程式辨識出來圖片中的表情是Angry，但實際上dataset的標記其實是Disgust有10個錯誤。    Sklearn有confuision matrix的函式可以用。    接下來就可以測試看看這個臉部辨識model了。    我們定義一個function可視化辨識結果。    結果如上。此圖片的happy權值最高。  參考資料: [Facial Expression Recognition with Keras](https://sefiks.com/2018/01/01/facial-expression-recognition-with-keras/) | | 輸出 | 此臉的年齡、性別、種族、表情的辨識結果。其中表情會回傳angry, fear, neutral, sad, disgust, happy and surprise七個表情特徵的權值。 | |

(二) Webduino機器人

這次選擇使用的開發板是Webduino Fly + Arduino Uno板，其是慶奇科技開發的套組，主要技術是雲端更新韌體程式，用簡單的積木程式就能夠讓Arduino開發板連上WiFi。

而機器人主要的功能是通過WiFi連線，若遙控器按下▶，則會讀取上述的雲端試算表，分別做出以下動作：

(1) Happy: LCD顯示開心的表情，並開始繞圈圈。

(2) Sad： LCD顯示難過的表情，並開始像是無理取鬧般的前後移動。

(3) Angry： LCD顯示難過的表情，並開始直線後退。

Webduino開發板使用的程式語言是Java和Html，而不是原本Arduino的C/C++或是Python，這是為了要達成雲端更新韌體程式的功能，所以要用和網頁編譯相同的語言。因此它不需要用線連接電腦才能完成程式燒錄，僅需要在第一次聯網時進行初始化，就可以無限的上傳程式。

雖然它操作方便、簡單，但是它能達到的功能確有侷限性。最大的差別是在等待和迴圈的功能，傳統編譯時，工作流程會是同步的，意思是不同流程工作前進的步伐一致，為了要協調彼此的步伐，會有「等待其他工作完成」的情況發生；然而，網頁的語法卻不同，其是採用「非同步」的方式，各個程式流程間並不會互相等待，因此在實作的過程會有傳統程式邏輯可以通過，但是轉換到網頁上卻會出現語法錯誤的情況發生。

|  |
| --- |
| 1.硬體接線圖 |
|  |
| 2. 積木程式 |
|  |
| 3. 輸出結果 |
| (1)開心  表情顯示：    Demo影片：<https://drive.google.com/file/d/1Kyxj6xRIu0Y7zdolSFAq0UXppTEL1uNw/view?usp=sharing>  (2)難過  表情顯示：    Demo影片： |
| <https://drive.google.com/file/d/166KORnpT-gc_lx9y6RQwVLyEFMXJWc6G/view?usp=sharing>  (3)生氣  表情顯示：    Demo影片：  <https://drive.google.com/file/d/1OA4J11fLgMtCmtLNonn4ZyS_P-ivWdVg/view?usp=sharing> |

三、專題心得

(一) 劉潔昕

這次的專題真的收穫良多，不管是實作的過程，還是聽大家分享自己的作品，可以從不同的面向去瞭解課堂中的理論被實踐的實務經驗。

這次的專題選題，我覺得是比較活潑的題目，但是我覺得這次遇到的其中一個難題，是在於軟硬體的整合。原本有考慮要用Tensorflow Lite簡化機器學習的模型，進而實現機器學習模型燒錄到開發板上，但是Arduino Uno仍然無法去運行Tensorflow Lite機器學習的模型，因此還是要找到其它的開發板。後來找到Webduino開發板，實現了Arduino Uno板連網的功能，雖然它的程式運作彈性仍然有可以加強的空間，但是他仍不失為一個表現優異、功能強大的開發板。

這學期的課程也讓我有很多學習，雖然課程進度很快，但是課程當中介紹到的機器學習和機器人相關的內容，並搭配每周的程式作業，使這個學期的課程能對人工智慧和機器人領域有進一步的了解和學習。

最讓我印象深刻的是MasterRCNN的Demo，看到不同的端機連線到同一個虛擬環境中，並控制當中的機器人移動及作圖象辨識，這真的是一個很新奇的體驗，如果未來還有機會，真的很想有更多的實驗經驗。

這學期的導論課，是一個收穫滿行囊的旅程，在課堂的最後還能完成一個自己的專題，真的是很有成就感。非常感謝老師和同組的同學，讓這堂課會是大學階段讓我最印象深刻的一堂課程。

(二) 陳子祈

這個人工智慧導論專題可以做的題目很多，我們這組選擇網路資源最多的圖形識別主題的專題。我負責的部分就是打好google colab的程式碼，將程式碼跑起來，使用電腦的攝影機將串流的每一幀畫面以圖檔的方式傳到臉部表情辨識的model跑出臉部表情辨識結果，並將結果傳到google sheet以供Webduino存取表情辨識結果用。我也順便將表情辨識結果顯示出來，另外我也把臉部偵測邊界也顯示在網頁上。

雖然網路資源很多，但技術門檻仍然很高，對於初學者來說，要自己訓練人工智慧model太困難了，除了沒有架構CNN model的經驗，我也沒有找到合適的開源dataset讓我訓練model，免費的dataset圖片太小畫質也不夠好，我又不想花錢買品質比較好的dataset，所以我只好選擇已經訓練好的臉部表情辨識model，也就是使用deepface model來做臉部表情辨識。Deepface真的是一個非常強大的套件，其中我們這組使用的臉部表情辨識model只是其中一個model，還有臉部比對、尋找等強大功能。

另外如果使用電腦的攝影機來拍攝臉部表情，並將拍攝到的畫面輸入到model，那在輸入到model之前就要先處理圖片資料，讓資料可以轉成model能接受的檔案格式，若還想將結果顯示出來，那還要把model跑出來的結果以cv2畫出來再轉為網頁顯示圖片的檔案格式，這些操作太複雜了，我對Javascript、html沒有很熟，所以我參考網路上的資源把Webcam做出來。

後來我還想到可以結合gspread將結果輸出到google sheet，以供Webduino存取臉部辨識的結果用，這樣就可以命令Webduino根據臉部辨識的結果作相對應的反應。我想到這個方法非常有成就感，因為我上網爬文爬了好久才終於想到這個方法，只要Arduino上還要加裝Wifi擴充板就可以使用Webduino操控Arduino板。

最後我也非常感謝組員讓我有發揮空間，讓我學到很多，希望以後可以自己做圖像辨識有關的題目，更深入研究圖像辨識，並結合自動控制做出自駕車。這門課就這樣告一段落了，也非常感謝老師這學期的教導，雖然到後期很多概念都好難，但都很有趣值得我花時間去探索。

四、未來展望

這次的專題因為進行時間較短，所以這次的成果，我覺得仍屬比較粗糙。我覺得可以從兩個方向改善。第一是表情辨識模型，因為這次的模型是使用已經train好的Deepface去進行實作，但是沒有機會去嘗試其它模型，甚至是可以加入自己臉部表情的data去做模型的更新及訓練，可以讓表情的辨識準確度有效提升。

第二，是現在的鏡頭是使用筆電的鏡頭，和機器人的機體是分開的，因此若還有機會可以把鏡頭和機器人整合在一起。我們有看到一個相機模組ESP32-CAM，其本身有連網的功能，因此之後機器人的優化可以試著將ESP32-CAM和Webduino整合。

五、參考資料

1.[How to Use Webcam In Google Colab for Images and Video (FACE DETECTION)](https://www.youtube.com/watch?v=YjWh7QvVH60)

2.[deepface](https://pypi.org/project/deepface/)

3.[Facial Attribute Analysis with Deep Learning in Python: Emotion, Age, Gender, Race](https://www.youtube.com/watch?v=GT2UeN85BdA)

4.[Google Colab: 用gspread處理Google Spreadsheet的資料](https://cyublog.com/articles/python-zh/colab-tutorial-google-spreadsheet/)

5. <https://tutorials.webduino.io/zh-tw/docs/basic/index.html>