

J.A.R.V.I.S –人臉識別保安AI智能助理

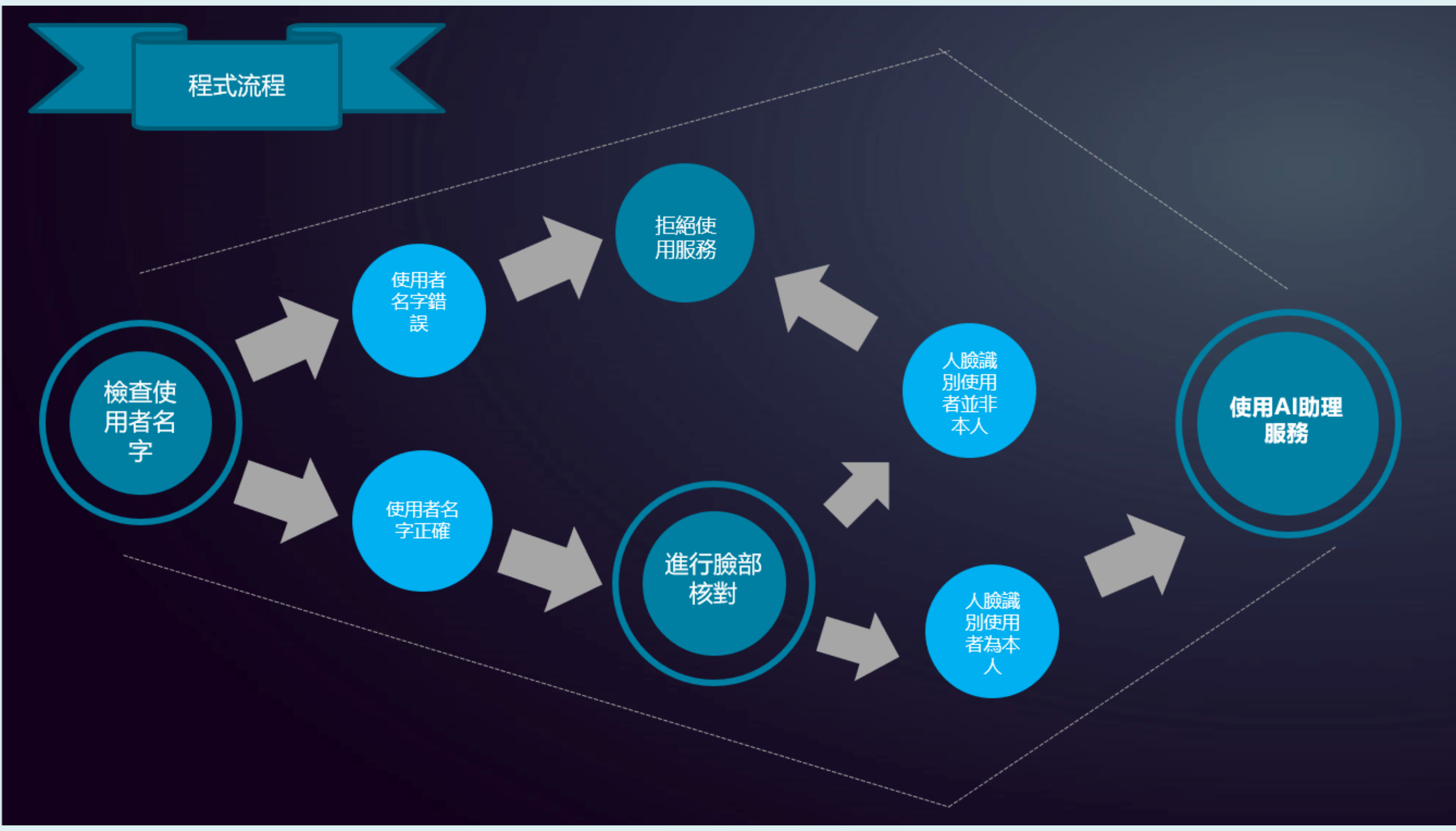
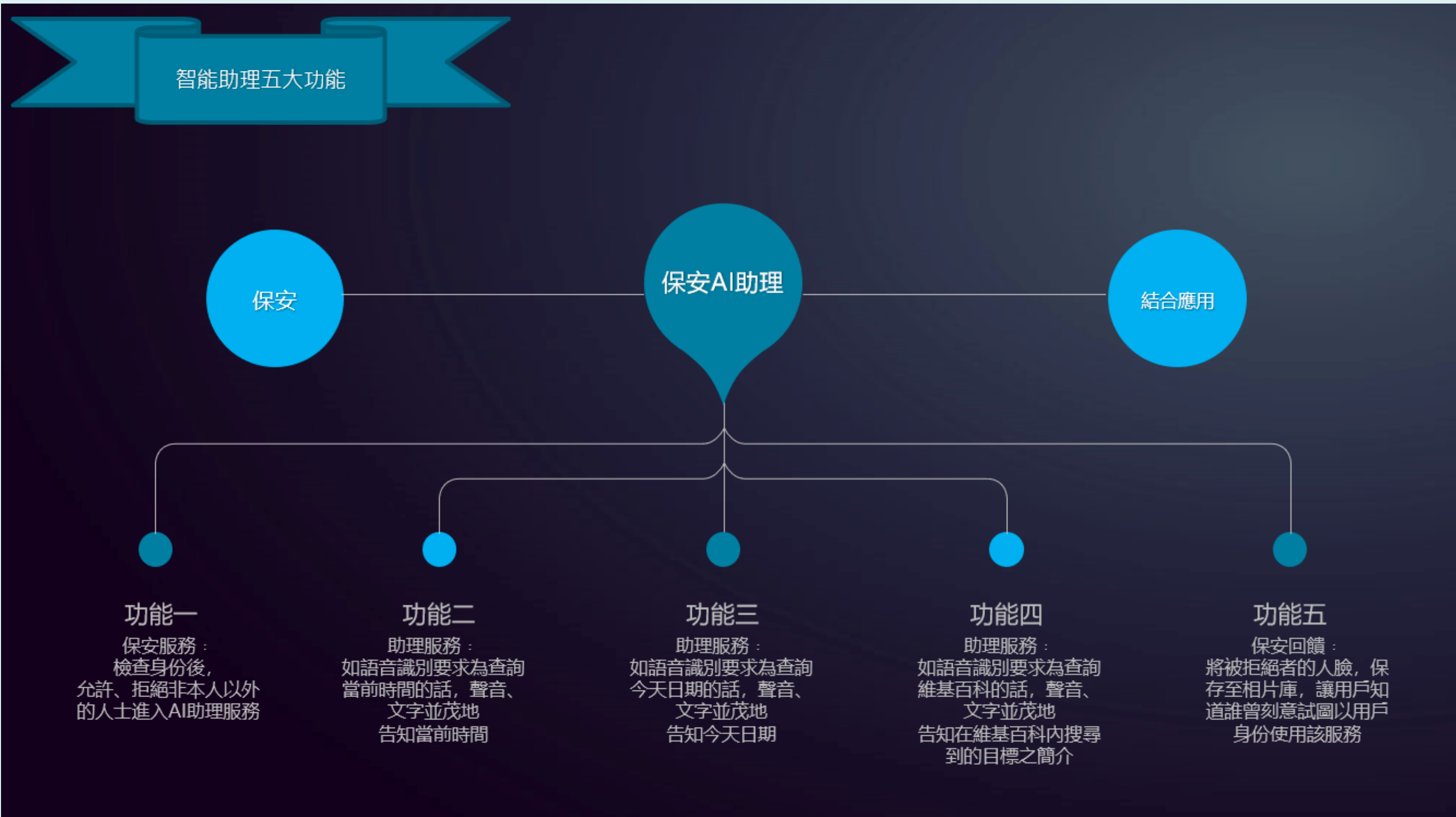
專題動機-應用與理解

現在人臉識別的技術，在業界其實已經相當成熟。在實際應用中，Face++、Lambda Labs等API人臉識別服務已經可以高達到99%準確度。在人臉識別模型中，TinaFace、RetinaFace、AlnooFace等已超過90%準確度。

人臉識別的應用很廣泛，遠至遊戲的真實人臉虛擬化、情緒偵測等醫學統計研究，但談到人臉識別，在保安上的應用人臉識別現得極為重要、且普及。在目前開發資源不足的情況下，比起建構模組，目前更適合進行應用導向的專題實作，把原理理解後，再純熟運用，變為自己的技術才是現階段最適合的方向，待開發資源以及知識面更深入後，日後再往更深入的方向發展，如建模。

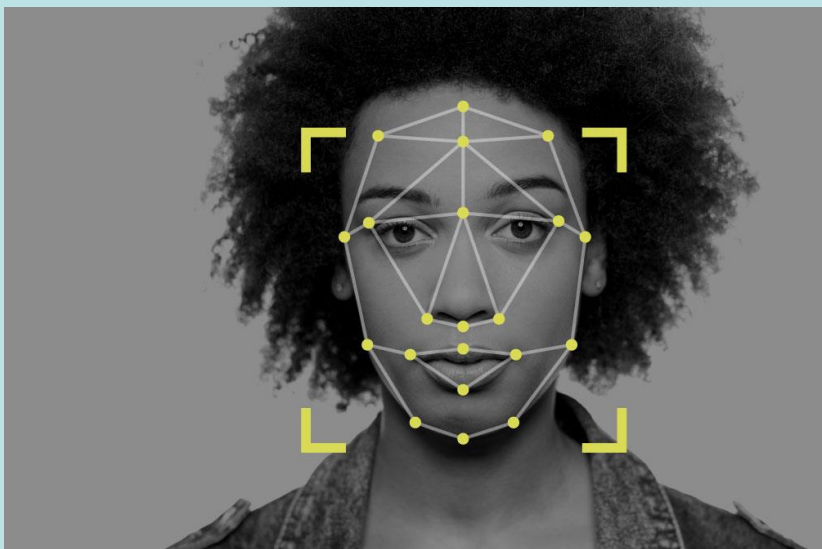
- 故此，此應用專題的三大重點為：
- 保安
 - 結合應用
 - 理解原理、純熟運用，將知識轉換為個人技術

人臉識別保安智能助理五大功能與使用流程



程式原理

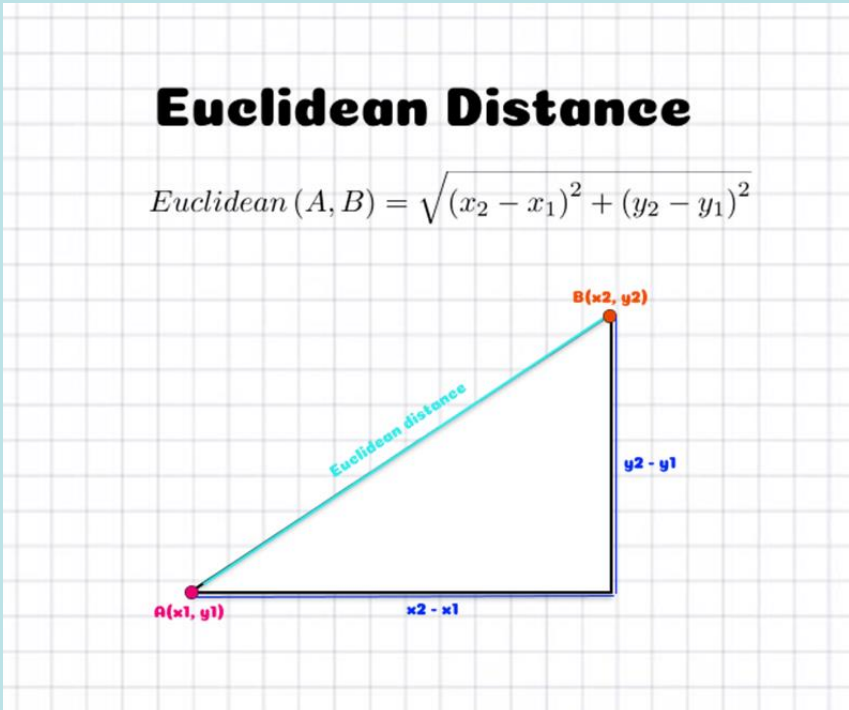
使用haarcascade_frontalface_alt.xml階層式分類器，得到一個人臉編碼的列表 β ，將它們與已知的人臉編碼列表 α 進行比較，並為每個比較的人臉構成一個臉網格(Face Mesh)，面網格由128條線連接起來組成，並用歐幾里得距離來計算這些臉網格線的長度。



人臉網格

這些臉網格線的長度代表著這些面孔的相似程度，在比照方面，一般採用的 $0.6 \leq \epsilon$ ，適用於檢查長相不相似的兩人身上，卻不適用於識別面部相似的兩人，如親生兄弟、親生姊妹，有鑑於一些雙胞胎兄弟、姊妹可高達到 $4.8 \times 10^{-4} \leq \epsilon$ 。

在計算出這128條臉網格線的長度後，以最短長度的那一條線k作為比照準則，比較人臉 α 與人臉 β 並得出比照結果。若 $\text{face_distance}_{\alpha,k} - \text{face_distance}_{\beta,k} \leq \epsilon$ ，則判定比照的人臉為同一人。



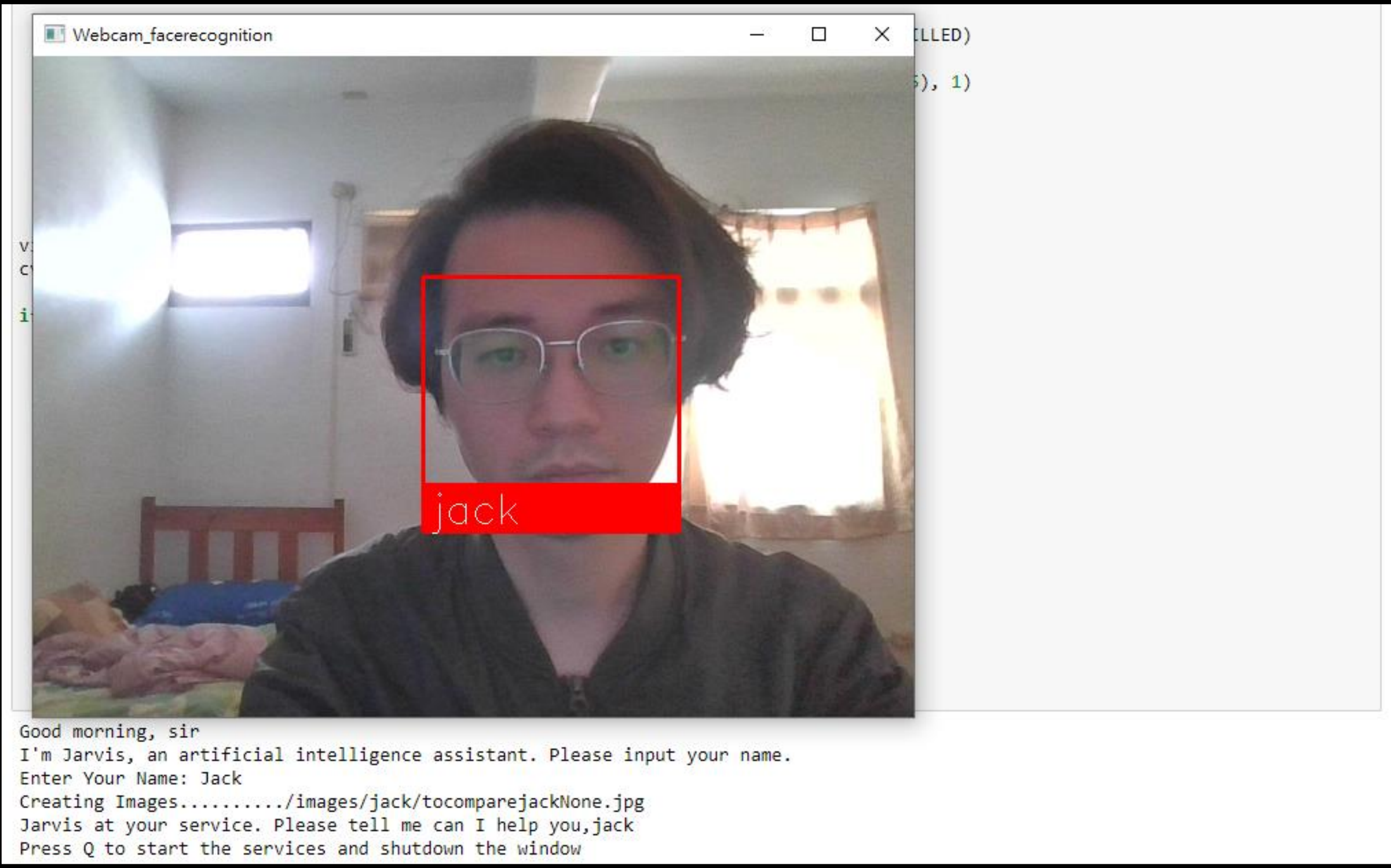
歐幾里得距離公底與原理

專題構思-三大特點



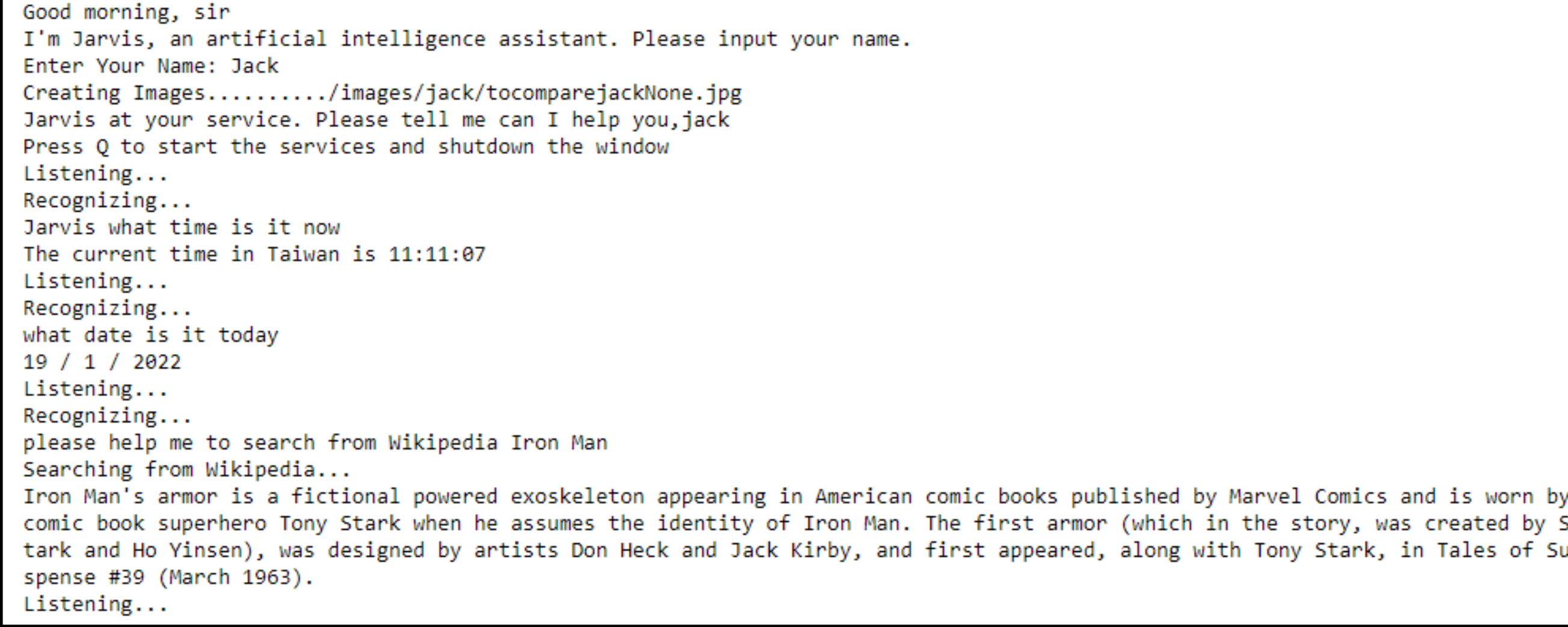
成果展示

進行臉部核對：



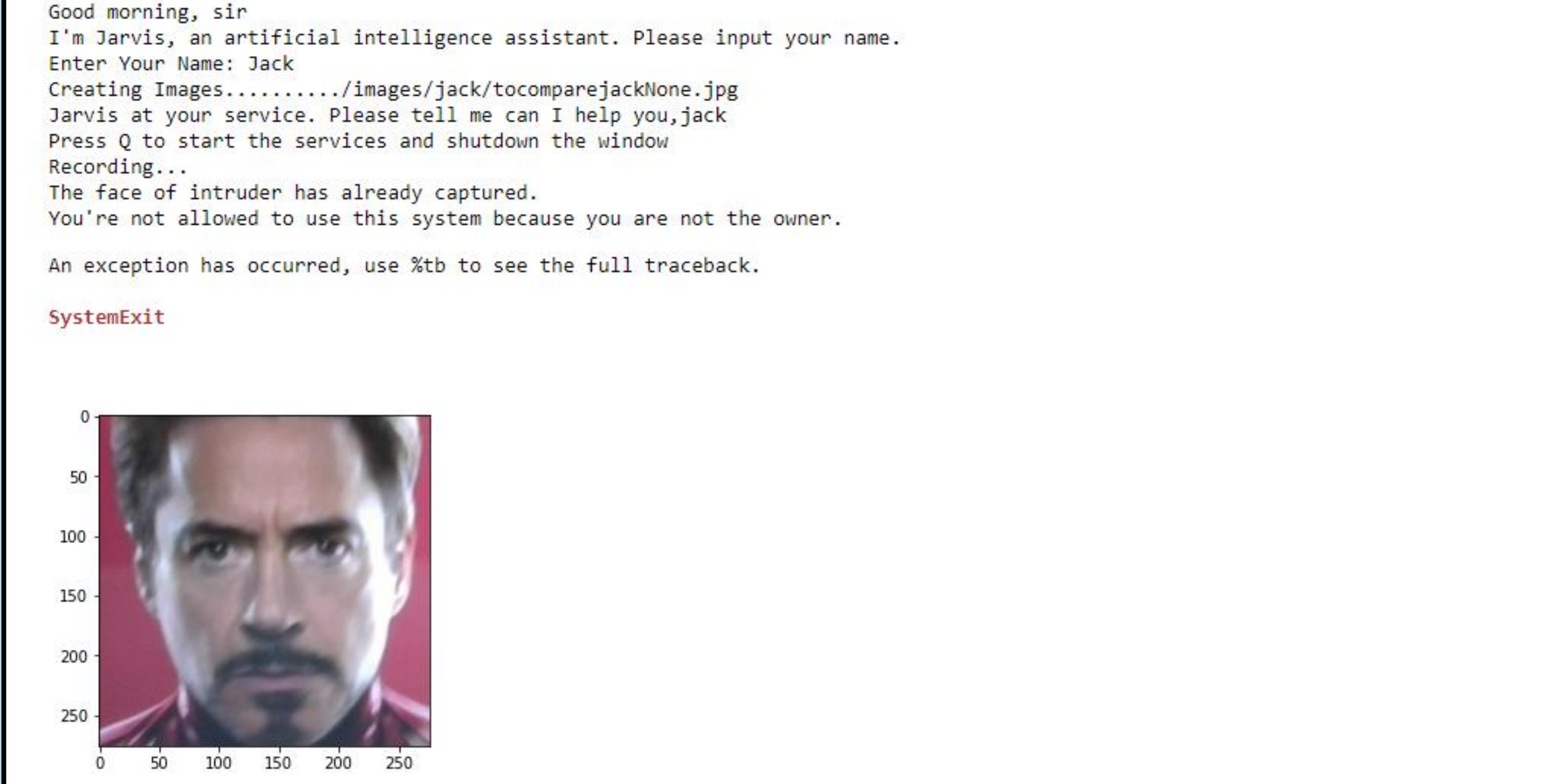
當使用者輸入名字後，會檢查使用者的名字是否包含在數據庫內，若是，則攝影機會偵測畫面內是否僅有一人，以鎖定進行臉部識別之人士，並進行臉部識別以核實身份。

允許使用智能助理服務：



核實身份成功後，就可以按照指示啟用AI智能助理服務，並透過語音辨別使用系統內各種智能助理服務。在上圖，展示了查詢當前時間、日期、搜尋維基百科等功能。

拒絕使用智能助理服務：



在上圖，展示了一個拒絕使用智能助理服務的實例。我們可以看到入侵者輸入智能助理擁有者名字，正試圖冒充智能助理擁有者本人，而進行臉部識別後判斷身份為惡意入侵者，拒絕進入智能助理服務，並退出系統，以及停止一切操作。最後，系統會把惡意入侵者的人臉儲存到入侵者的數據庫內，讓智能助理擁有者本人查詢有哪些人惡意冒充身份來使用智能助理系統。

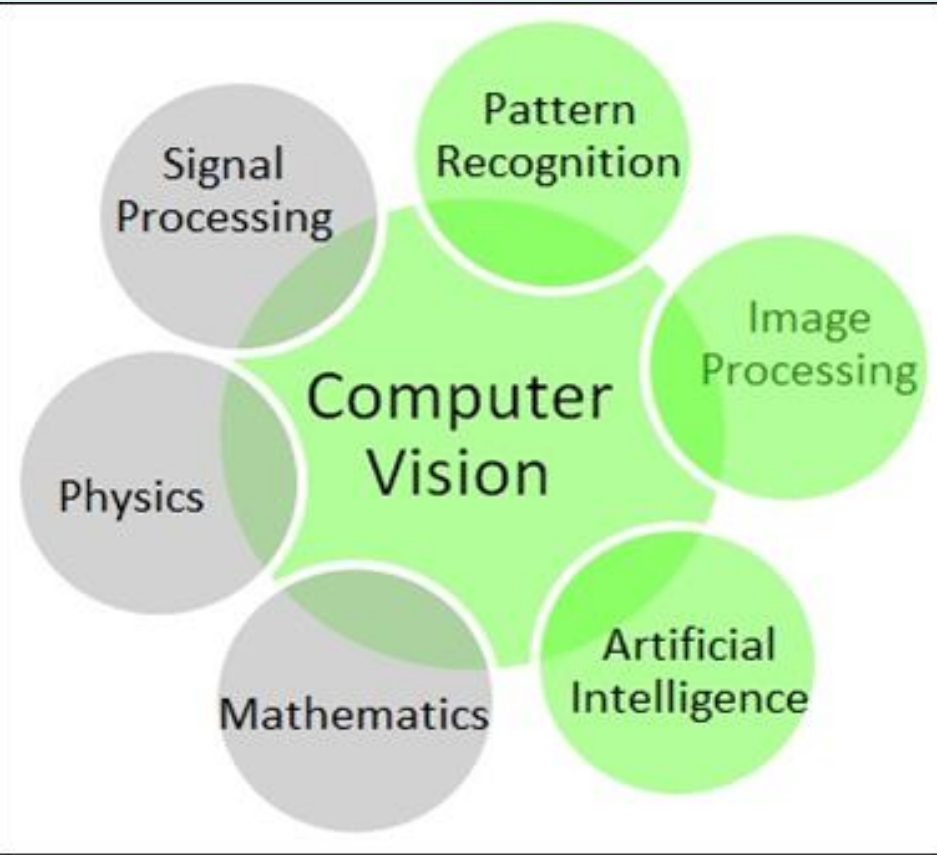
人臉辨識

研究目的與動機

我們生活在二十一世紀的智慧城市中，我們的生活處處都是人工智慧的應用，如手機螢幕解鎖、上班打卡、付款結帳等等，這些其實都是人工智慧中的其中之一人臉辨識的用途，這些對我們來說稀鬆平常，甚至是一個舉手轉瞬間就能完成的事是要經過多少努力以及嘗試才能完成的。我們想使用的是我們平常上課習慣使用的Python去嘗試運用我們現在所學到的以及在教授的幫助下我們能否完成，在我們認知中的人臉辨識，我們希望最終的結果能夠像我們在臉書上看到的照片標出方框並且辨識出是誰，更希望不只是匯入的照片而是我們打開鏡頭資料庫中的人我們都能夠即時辨識。

資料介紹

OpenCV全名是Open Source Computer Vision Library，是當今最知名、也被最廣泛採用的影像處理函式庫。OpenCV在影像處理方面應用廣泛，可用在以下部分：



- Computer Vision 電腦視覺
- Pattern Recognition 圖型識別
- Image Processing 影像處理
- Artificial Intelligence 人工智慧
- Mathematics 數學
- Physics 物理
- Signal Processing 訊號處理

Face_recognition是基於Dlib發展出的人臉辨識，並將程式簡化，回傳True/False判斷是否為同一人。有高達99.38%的準確率。世界上最簡潔的人臉識別庫，可以使用Python和命令行工具提取、識別、操作人臉。在 Face Recognition (人臉辨識)的問題上，通常會再進一步分成兩個種類：

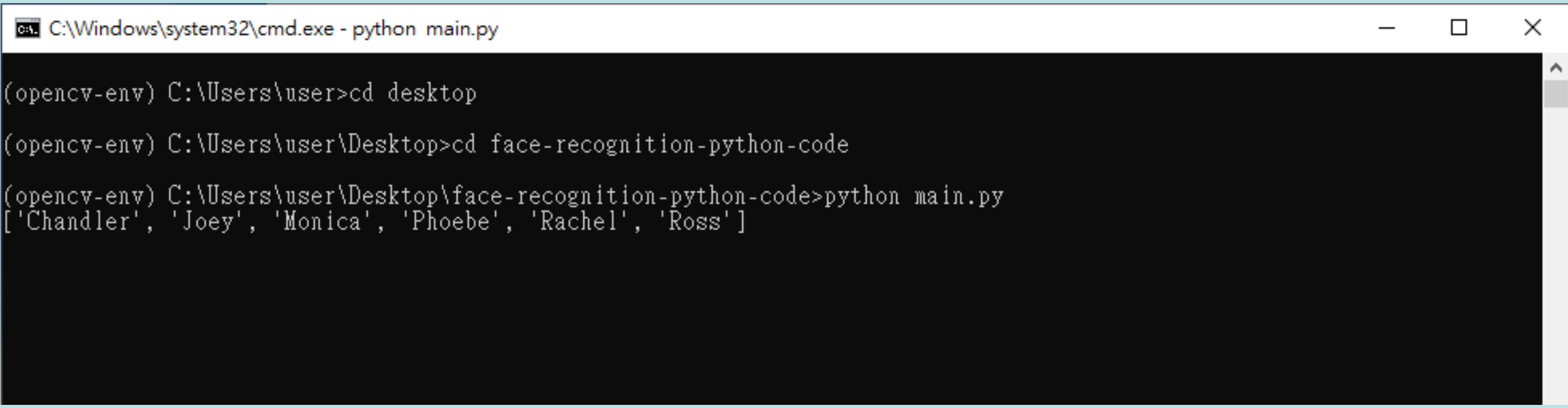
Face Verification (人臉驗證)	Face Recognition (人臉辨識)
給予輸入 image, name/ID	擁有 K 個人物的Database
輸出是否為此人	給予 Input image
視為 1:1 matching problem	輸出 ID, if (image 為 K 個人物中的其中一個)
e.g. 手機的人臉解鎖	視為 1:K matching problem
	e.g. 使用的人臉辨識的員工通行開門

執行步驟

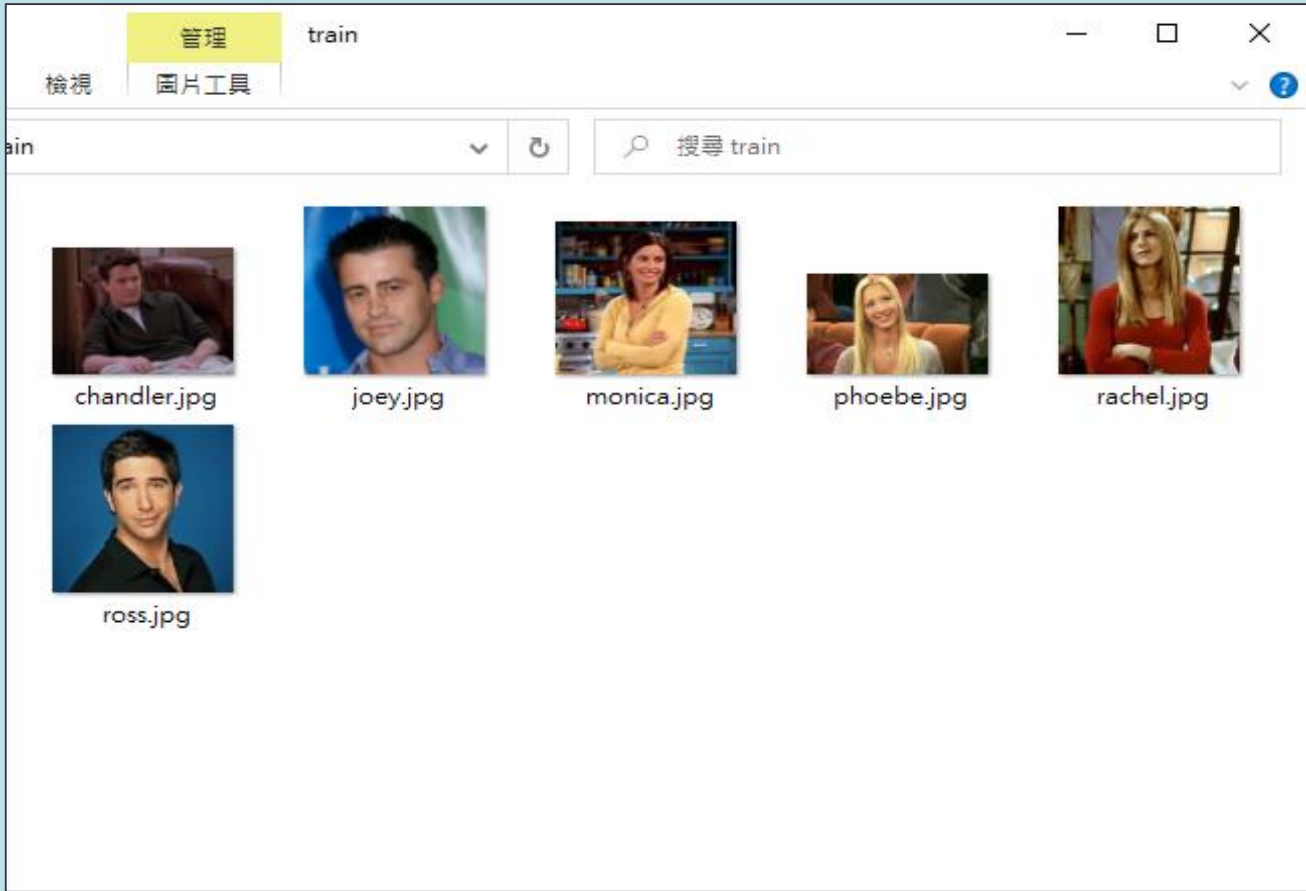
- STEP 1:使用 Anaconda建立虛擬環境將Python版本轉換為 3.6.8
- STEP 2:進入虛擬環境 Activate the env
- STEP 3:安裝套件 Install OpenCV and Other Packages
- STEP 4:安裝套件 Install cmake
- STEP 5:安裝套件 Install dlib
- STEP 6:安裝套件 Install face_recognition
- STEP 7:開啟程式執行程式碼

須先建立train資料夾與test資料夾
將要訓練的個人照檔名以人名存入train資料夾
將需受辨識的團體照放入test資料夾

成果展示



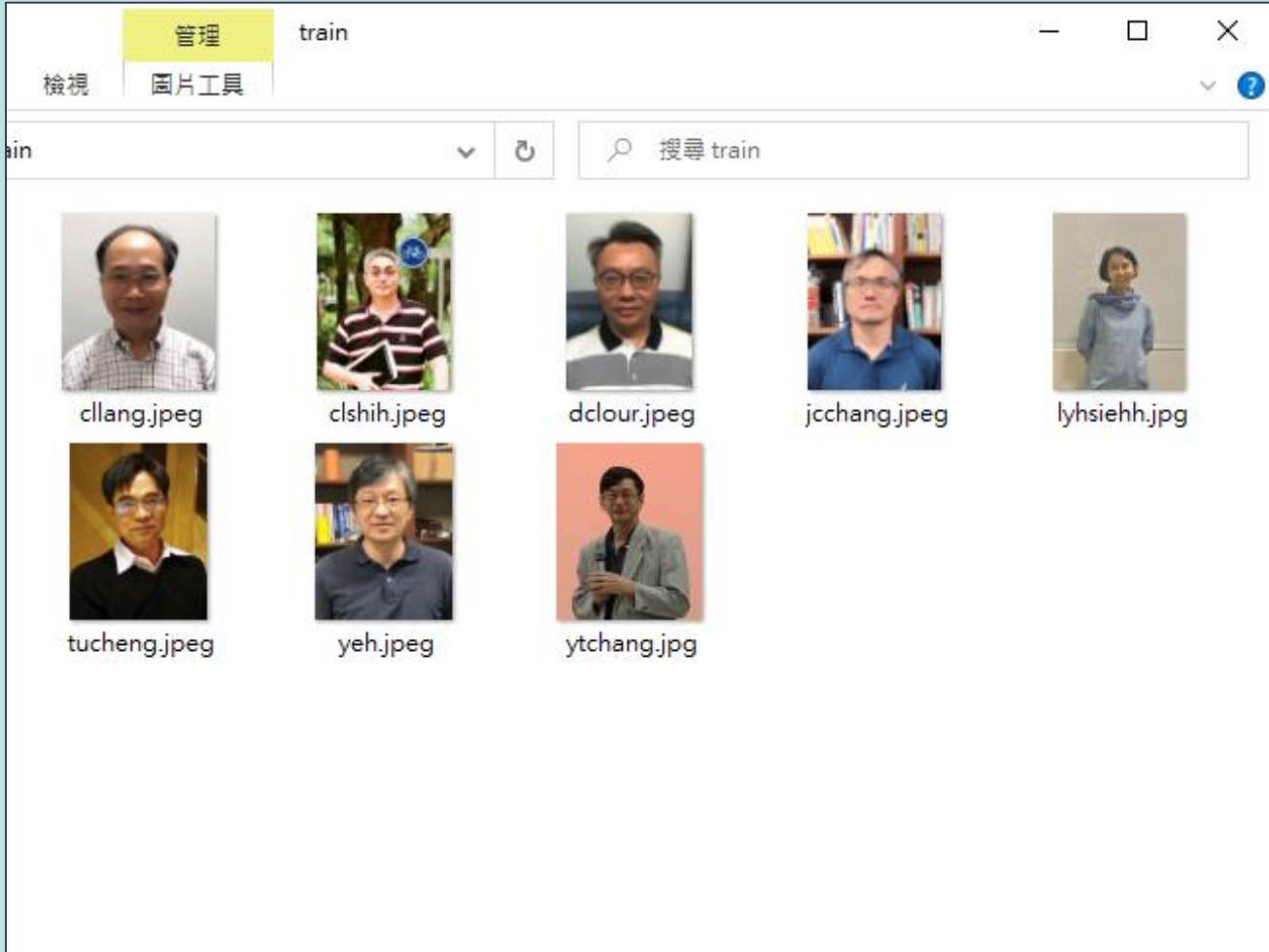
程式碼執行後會先顯示train資料夾中的人名



train資料夾



匯出標記出人名團體照



資料量多寡-臉部辨識系統的資料庫大小會影響辨識的精準度，舉例來說，如果系統的基礎資料大多是西方白人，缺少亞洲人的臉部資料，在進行亞洲臉孔辨識時，錯誤率就會比較高。
素材解析度-進行臉部辨識時，臉部的拍攝角度、光源等都會影響系統獲取臉部資訊的效率。此外，如果是使用影像資料建立資料庫，素材的解析度也會影響辨識的精準度。
臉部配件及臉部特徵變化-配戴口罩、墨鏡或是帽子等，都可能影響臉部辨識的精準度，而人隨著年齡增長，臉部特徵也會產生變化，資料庫如果沒有定期更新，也會導致臉部辨識的判斷失準。

結論

人臉辨識為目前公認最準確且可用度最高的人工智慧生物辨識技術。人臉辨識技術除了常見的臉部偵測及身分比對外，亦包含了特徵偵測(如：性別、年齡、情緒)、口罩偵測等多種功能。企業可透過此技術進行辦公場所的安控來保障員工安全、零售商可在商店中使用此技術強化客戶體驗、製造業可使用此技術來強化進入限制區域的人員控管，而金融科技產業則可使用此技術引進更強大、安全的認證機制。在可期的未來，人臉辨識在許多應用領域極具潛力。

然而伴之而來的，廠商也須謹慎運用此技術，並透過適當的監管規範、使用者教育來消弭對人臉辨識相關的疑慮。在不妨礙創新的前提之下，讓消費者可以體驗、接受人臉辨識的優點，進而採用。

我們使用相較於常見人臉辨識系統簡單的程式碼執行，因此可能會出現資料量不足、素材解析度不夠、臉部配件及臉部特徵變化不同導致臉部辨識的判斷失準的情況出現，希望未來經過學習有機會能夠將程式在寫得更完善。