

大同大學 電腦視覺期末專題

用 CNN 辨識人臉實現居家門禁系統

GE1.27 陳煒凱

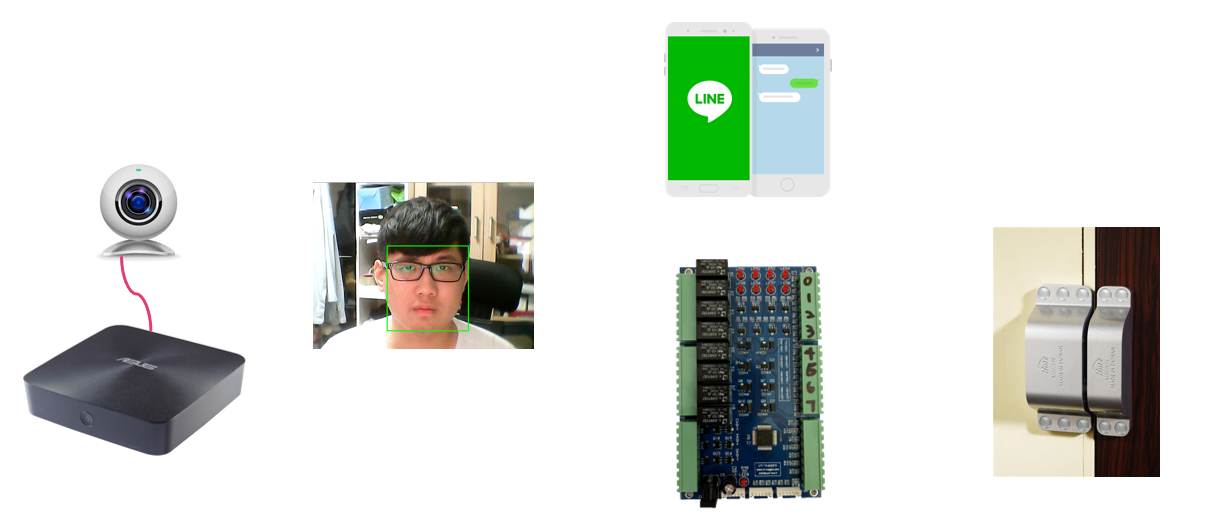
目錄

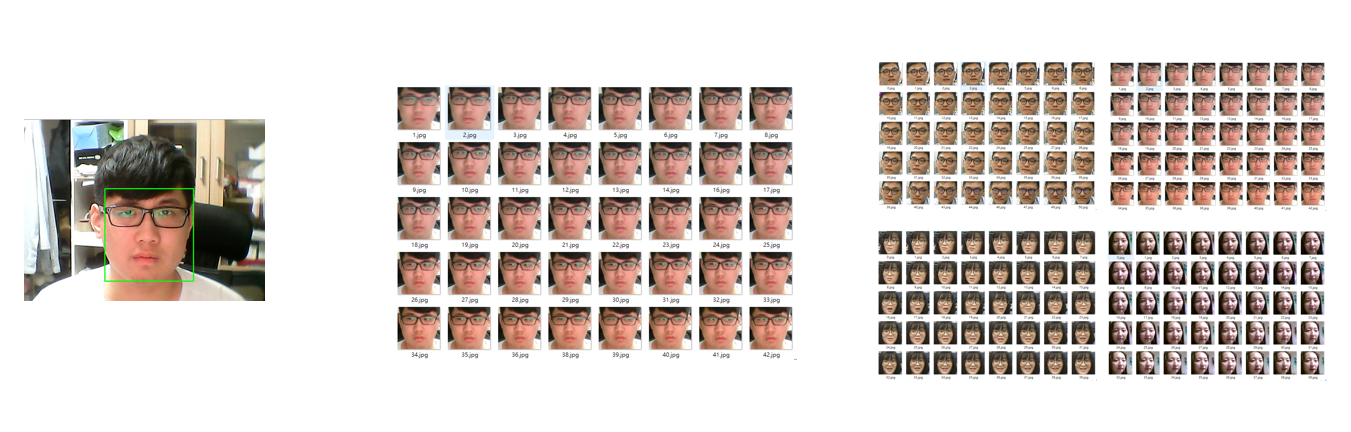
1. 背景與動機 ............................................................... 2
2. 研究設計與步驟 .................................................... 2
3. 程式說明與實現結果 ................................................ 4
4. 結論 ............................................................................ 5
5. 文獻參考 ................................................................... 6
6. **背景與動機**

現在市面已經有一些人臉辨識開門的技術，大多利用opencv或是其他電腦視覺辨識技術進行影像辨識，並且開門，但其中會遇到一個問題，就是辨識的圖像是平面的，代表有心人士可以利用拍照取得別人的臉進行辨識並且開門。所以這個專案會講這項系統進行改進，人臉辨識完之後，會傳Line訊息至家中的群組或是個人，告知使用者是誰在門口，訊息內容包含辨識的結果以及圖片，確認沒問題按下Line上的開門見即可完成開門動作，提高居家門禁的安全性。

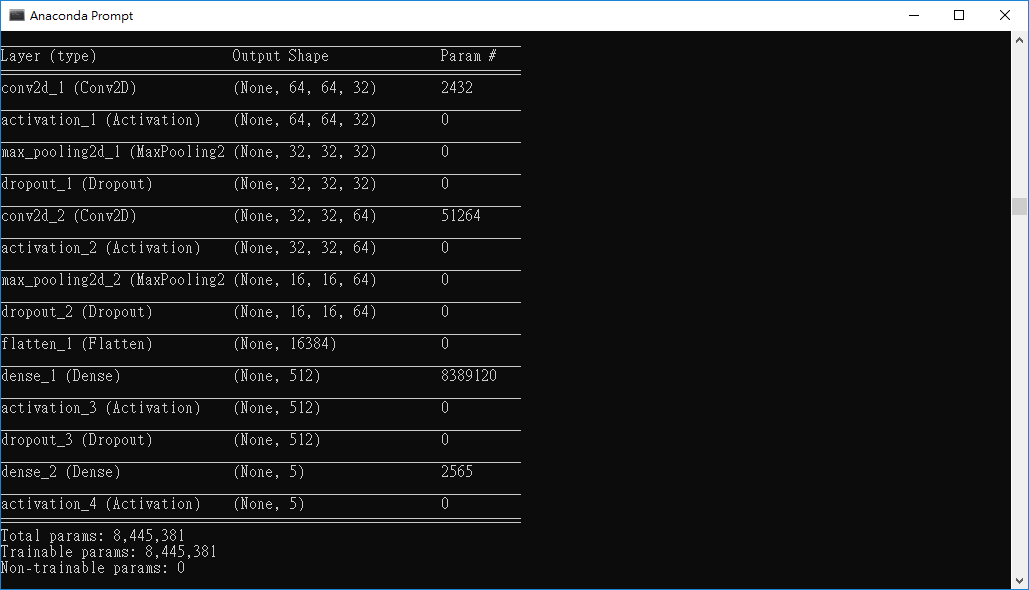
1. **研究設計與步驟**

首先，先取得使用者的樣本，並且編號，為了增加辨識的準確度，會取樣約500張樣本，使用者對著鏡頭，鏡頭只要偵測到人臉便會拍照，拍照期間，使用者的頭部可以轉動，以增加辨識的成功率，不同的人所擷取的圖檔，會存到不同資料夾中，都拍完之後，將拍攝出來的圖片進行標準化，全部轉為64\*64的圖片。之後便開始建立CNN模型並且訓練，我的CNN模型的順序是：1卷積層、1池化層、1卷積層、1池化層、1平坦層、2分類層，然後輸出結果，建完模之後將模型存檔。最後則是進行即時的人臉辨識，載入人臉分類器以及剛剛建立好的模型，開啟照相機之後，先用辨識器進行人臉偵測，把偵測到的圖片放入剛剛建立好的CNN模型進行辨識，最後在螢幕上輸出辨識結果，同時將辨識的結果透過LineBot傳到使用者的LINE上面，並問是否要開門，若使用者選擇開門，則會由RS485告訴單晶片控制板，並且打開陽極鎖開門。

 ↓**門禁系統的軟體及硬體架構**

 **↓人臉擷取示意圖**

**↓CNN架構圖**

****

1. **程式說明與實現結果**

因為程式碼非常長，故以上傳GitHub，並且在其中加入註解幫助閱讀：

GitHub 網址： <https://github.com/cheap122000/CV_FinalProject>

(因為API有些需要收費，故在程式碼中使用acc\_name、acc\_key等字取代API KEY)

其中requirements.txt是本次專案中所有會用到的python套件

datacollect2.py 是拍照建立樣本的程式碼

load\_face\_dataset.py 是讀取樣本並建立標籤的程式碼

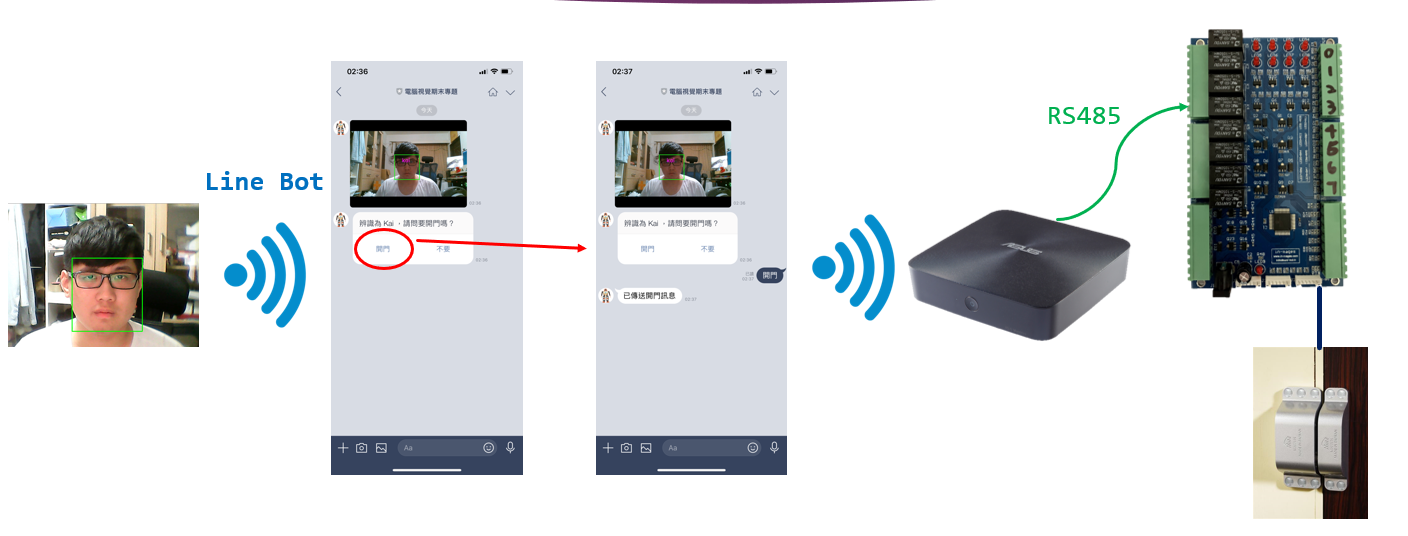
face\_train\_use\_keras\_gpu3.py 是用GPU顯示卡去訓練CNN模型的程式碼

face\_predict\_use\_keras.py 是拍照並且讀取CNN模型進行人臉預測的程式碼

opendoor.py 是透過RS485開門的程式碼

DIO.c 是單晶片控制板接收RS485並且導通繼電器開門的程式碼

line\_bot.py 是放在Heroku伺服器上的Line API程式碼

**↓執行結果示意圖**

1. **結論**

在訓練模型的時候，我有使用CPU及GPU去做模型的訓練，發現GPU的訓練速度是快上CPU非常多的，因為AI的訓練中，許多東西都只是單純的數學計算而GPU雖然單核心的運算沒有比CPU快，但是GPU的核心數量卻是CPU的幾百倍、幾千倍，所以這也是為什麼大家都說GPU比CPU更適合拿來訓練AI模型。

在人臉辨識中，發現只要換一個環境，會影響最終辨識結果的準確定，可能是因為我的辨識是採用RGB彩色辨識，在不同的環境、不同的光源下，可能會讓自己的膚色變得很像其他人的膚色導致辨識出錯；還有就是因為人的臉不是方的，所以可能會擷取到部分的背景，發現有時候背景也會影響到辨識的準確度。

最後，如果未來要應用在商業上的化，首先，需要建立一個強大，且用GPU運算的雲端伺服器，將使用者的圖片上傳之後訓練完將模型存檔，再由使用者下載訓練好的模型回去，開始辨識；除此之外，影像的標準化還要多下些功夫，除了camera的位置要固定以外，也要將人臉以外的部分去掉，用白色取代，如此便可大幅增加辨識的成功率。

1. **文獻參考**

* 人脸检测及识别python实现系列（5）——利用keras库训练人脸识别模型： <https://www.cnblogs.com/neo-T/p/6477378.html>
* <<TensorFlow + Keras深度學習人工智慧實務應用>>： <https://www.books.com.tw/products/0010754327>
* PySerial: <https://pythonhosted.org/pyserial/>
* Line Messaging API: <https://developers.line.biz/en/docs/messaging-api/overview/>