

1) 方法描述 - 演算法原理與實作方式描述

- (1)這次使用到的演算法分別是SAD和SSD，是nearest neighbor search中比較簡單的做法。原理是把每張圖片的每個像素分別相減後取絕對值的和(SAD)與取平方的和(SSD)。依照過往學習經驗可以大概知道，SSD的準確度相較於SAD應該是會比較高一些，因為差距越多的值在平方之後相當於被放大檢視了。而我們將每張testing photo分別跟全部的training photo做比較，求得的值中最小的值就代表兩張圖片最接近，接著我們就依此來判斷他們應該是同一個人。但並不是每個都剛好正確，我們便能算出這種演算法的正確度。
- (2)把全部資料夾依序分別打開，從中拿取是.pgm的檔案，每張照片拿取出來後都要展開成一個row並利用vstack存放到numpy array中。前面的35張將其存放到trainMatrix中，每個row都代表一張照片，依序存放，同時要用另一個trainBelong記錄他的資料夾來源。其餘的照片則放到testMatrix中，方法與前者相同。接著就拿兩個matrix使用兩個for迴圈(外迴圈是test、內迴圈是train)做SAD、SSD計算，比較出最接近的照片後再比對Belong是否一樣，最後即可計算出準確率。

2) 實驗結果

TrainCount: 1330

TestCount: 1085

correctSAD: 489

correctSSD: 499

SAD: 45.069124%

SSD: 45.990783%

用SAD執行的準確率大概是45.07%，用SSD執行的準確率大概是45.99%。

testing的資料總共有1085張，其中SAD的正確張數為489張、SSD的正確張數為499張。

3) 結果討論 - 對於實驗結果的一些解釋和討論

與預期的一樣SSD的準確率稍稍高於SAD，但其實也沒有多很多，或許是因為資料量還不夠大的原因。雖然準確率都沒有超過50%，但這個結果其實已經讓我有點驚訝了，畢竟很難想像有辦法光靠計算每個像素的差值就做到接近一半的辨識率。後來看了一下每張照片，發現其實每個人真的有些共通點，像是鼻子的部分會比較亮、嘴巴的部分比較暗，而每個人的五官都長得不太一樣，所以這部分不同人就會有所差別了，同一個人在同樣像素的部分自然會差比較小。不過還好這些資料每個人的臉部大小和表情之類的都差不多，同時也沒有其他太多雜訊。如果今天是像我們一般的照片，人臉的遠近都不一樣，而且還會有除了人臉之外的不同雜訊，大概用這個辨識方法就不適用了。

4) 問題討論 - 作業撰寫中遭遇的演算法問題與實作的困難

最困難的地方其實就是讀檔的部分了，雖然以前也有寫過python，但比較沒有試過讀取圖片這方面的。其他方面的問題相對來說少了很多，因為其實以前也用過MATLAB寫這個作業了該踩的地雷也都踩過了，所以這次主要還是熟悉python的語法，讓以後的作業比較能順利進行。