多媒體作業一－Face Recognition\_NN

404410039 李維哲

1. 方法描述 - 演算法原理與實作方式描述

原理:

將資料庫裡的前35筆資料作為training data

剩餘資料當作test data

利用SAD與SSD來取出與test data最相近的圖片

並比對最相近的圖片是否與其來源是同一人

最後計算正確率

實作:

先製作training data

抓出各資料夾中的圖檔

(沒有第14號資料夾 所以要跳過)

(會自動判掉.pgm.bad檔與ambient(最後一個檔案))

(使用dir(filename)抓出檔案 再使用imread抓出圖片資料)

reshape成單行(192\*168->1\*32256)

(使用reshape)

全部放入TrainData的矩陣裡 所以最後矩陣大小會是1330\*32256

(使用cat連接矩陣內容)

利用SAD與SSD方式找出最相近圖:

取出test data

(使用dir(filename)抓出檔案 再使用imread抓出圖片資料)

reshape成單行

(使用reshape)

與所有TrainData裡的資料相比較

依據SAD與SSD找出最相近的圖的index

(需轉成double 原本格式為uint8 範圍只有0~255 負數會變成0)

(SAD距離取法:abs(double(TestData)-double(TrainData(CheckIndex,:)) )

(SSD距離取法:abs(double(TestData).^2-double(TrainData(CheckIndex,:).^2) )

(特別的實作法:剪枝

若在迴圈加總距離時已經比之前發生的最小距離還大時

可以不用繼續作 因為必定不會小於之前的最小距離

可以省一些計算成本)

因為index為圖片的編號而非對應資料夾

->需除以35取整數(使用floor),若整除須在加1

->沒有14號資料夾 需微調

統計正確對應的圖片數 算出正確率

1. 執行方式 - 執行的函數名稱、參數設定等

在有測試圖片的情況下 輸入檔名”hw1\_404410039.m”即可

1. 實驗結果 - 每一個階段的圖片、數據結果

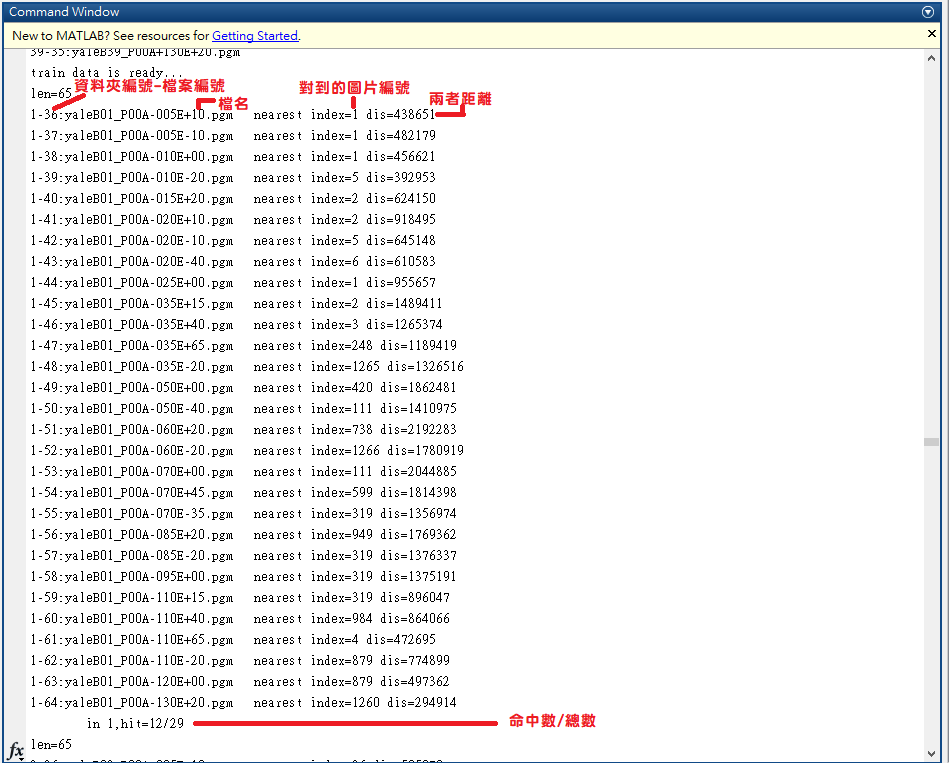
以兩次比對作為例子

各行格式為:

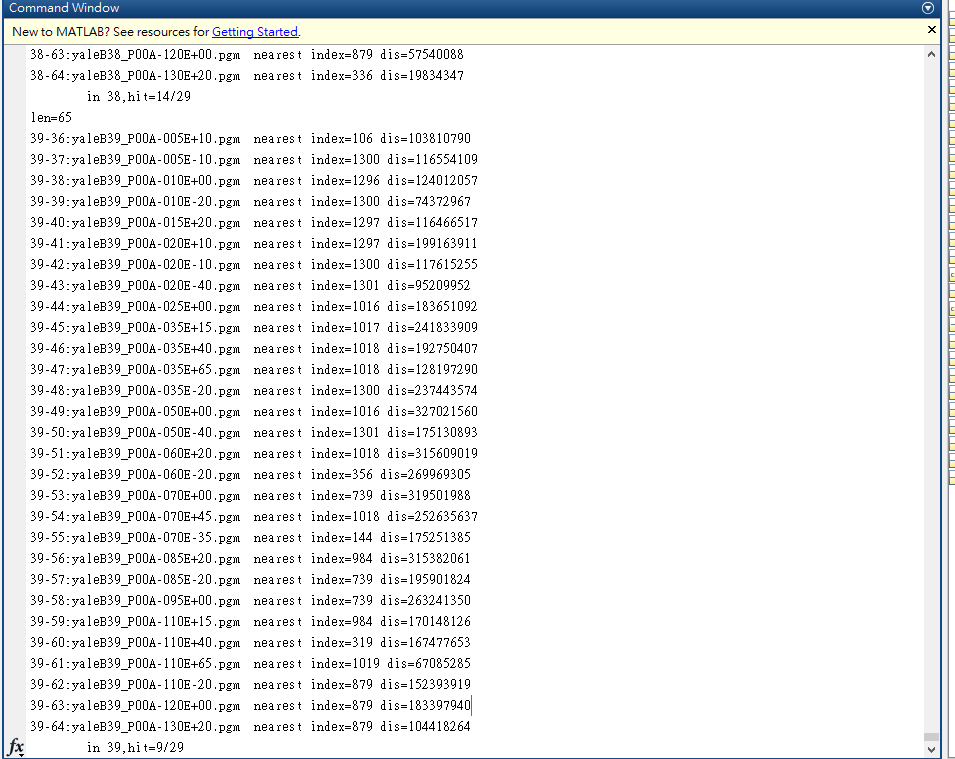
[資料夾]-[檔案編號]:[檔名] nearest index=[最近的檔案號碼] dis=[兩者的差異度(距離)]

每個資料夾做完會有該資料夾內的命中數與總數

SAD:



SSD:



結果:



SAD正確率約45%

SSD正確率約39%

1. 結果討論 - 對於實驗結果的一些解釋和討論

因為只有考慮圖片裡各點的距離差，如果是同一個人但圖片平移可能造成很大的差距，所以準確率並不是太高，不過令我比較好奇的是SSD比較接近真實的距離定義，但準確率卻相對較低，讓我蠻疑惑的。

1. 問題討論 - 作業撰寫中遭遇的演算法問題與實作的困難

開檔部分:

因為資料庫裡有蠻多例外狀況的，像是沒有14號資料夾，有些.bad檔，所以踩了蠻多雷的

比對過程:

原本想直接使用函式作距離的加總，不過整體速度真的有點慢，所以才想到能用剪枝的方式稍微加快速度。

在做距離相減時，一開始沒注意到檔案的格式是uint8，所以準確率低到誇張，後來在一個一個檔案檢查的過程中發現當下加總的距離卡在255而加不上去，解決方式是只要先轉成double在相減即可

實驗結果:

雖然最後是做出來了，不過跟同學的結果比對後發現各點得到的最小距離有些不同，所以正確率雖然非常近但有些許不同，會有點懷疑自己的程式有沒有BUG。