

# 多媒體作業——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: $\text{abs}(\text{double}(\text{TestData}) - \text{double}(\text{TrainData}(\text{CheckIndex},:)))$  )

(SSD 距離取法: $\text{abs}(\text{double}(\text{TestData}) .^2 - \text{double}(\text{TrainData}(\text{CheckIndex},:)) .^2)$  )

(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

- 2) 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
- 3) 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:

```

Command Window
New to MATLAB? See resources for Getting Started.
39-35:yaleB01_P00A-010E+10.pgm
train data is ready...
len=65
1-36:yaleB01_P00A-005E+10.pgm nearest index=1 dis=438651
1-37:yaleB01_P00A-005E-10.pgm nearest index=1 dis=482179
1-38:yaleB01_P00A-010E+00.pgm nearest index=1 dis=456621
1-39:yaleB01_P00A-010E-20.pgm nearest index=5 dis=392953
1-40:yaleB01_P00A-015E+20.pgm nearest index=2 dis=624150
1-41:yaleB01_P00A-020E+10.pgm nearest index=2 dis=918495
1-42:yaleB01_P00A-020E-10.pgm nearest index=5 dis=645148
1-43:yaleB01_P00A-020E-40.pgm nearest index=6 dis=610583
1-44:yaleB01_P00A-025E+00.pgm nearest index=1 dis=955657
1-45:yaleB01_P00A-035E+15.pgm nearest index=2 dis=1489411
1-46:yaleB01_P00A-035E+40.pgm nearest index=3 dis=1265374
1-47:yaleB01_P00A-035E+65.pgm nearest index=248 dis=1189419
1-48:yaleB01_P00A-035E-20.pgm nearest index=1265 dis=1326516
1-49:yaleB01_P00A-050E+00.pgm nearest index=420 dis=1862481
1-50:yaleB01_P00A-050E-40.pgm nearest index=111 dis=1410975
1-51:yaleB01_P00A-060E+20.pgm nearest index=738 dis=2192283
1-52:yaleB01_P00A-060E-20.pgm nearest index=1266 dis=1780919
1-53:yaleB01_P00A-070E+00.pgm nearest index=111 dis=2044885
1-54:yaleB01_P00A-070E+45.pgm nearest index=599 dis=1814398
1-55:yaleB01_P00A-070E-35.pgm nearest index=319 dis=1356974
1-56:yaleB01_P00A-085E+20.pgm nearest index=949 dis=1769362
1-57:yaleB01_P00A-085E-20.pgm nearest index=319 dis=1376337
1-58:yaleB01_P00A-095E+00.pgm nearest index=319 dis=1375191
1-59:yaleB01_P00A-110E+15.pgm nearest index=319 dis=896047
1-60:yaleB01_P00A-110E+40.pgm nearest index=984 dis=864066
1-61:yaleB01_P00A-110E+65.pgm nearest index=4 dis=472695
1-62:yaleB01_P00A-110E-20.pgm nearest index=879 dis=774899
1-63:yaleB01_P00A-120E+00.pgm nearest index=879 dis=497362
1-64:yaleB01_P00A-130E+20.pgm nearest index=1260 dis=294914
in 1, hit=12/29
len=65

```

SSD:

```
Command Window
New to MATLAB? See resources for Getting Started.
38-63:yaleB38_P00A-120E+00.pgm nearest index=879 dis=57540088
38-64:yaleB38_P00A-130E+20.pgm nearest index=336 dis=19834347
in 38, hit=14/29
len=65
39-36:yaleB39_P00A-005E+10.pgm nearest index=106 dis=103810790
39-37:yaleB39_P00A-005E-10.pgm nearest index=1300 dis=116554109
39-38:yaleB39_P00A-010E+00.pgm nearest index=1296 dis=124012057
39-39:yaleB39_P00A-010E-20.pgm nearest index=1300 dis=74372967
39-40:yaleB39_P00A-015E+20.pgm nearest index=1297 dis=116466517
39-41:yaleB39_P00A-020E+10.pgm nearest index=1297 dis=199163911
39-42:yaleB39_P00A-020E-10.pgm nearest index=1300 dis=117615255
39-43:yaleB39_P00A-020E-40.pgm nearest index=1301 dis=95209952
39-44:yaleB39_P00A-025E+00.pgm nearest index=1016 dis=183651092
39-45:yaleB39_P00A-035E+15.pgm nearest index=1017 dis=241833909
39-46:yaleB39_P00A-035E+40.pgm nearest index=1018 dis=192750407
39-47:yaleB39_P00A-035E+65.pgm nearest index=1018 dis=128197290
39-48:yaleB39_P00A-035E-20.pgm nearest index=1300 dis=237443574
39-49:yaleB39_P00A-050E+00.pgm nearest index=1016 dis=327021560
39-50:yaleB39_P00A-050E-40.pgm nearest index=1301 dis=175130893
39-51:yaleB39_P00A-060E+20.pgm nearest index=1018 dis=315609019
39-52:yaleB39_P00A-060E-20.pgm nearest index=356 dis=269969305
39-53:yaleB39_P00A-070E+00.pgm nearest index=739 dis=319501988
39-54:yaleB39_P00A-070E+45.pgm nearest index=1018 dis=252635637
39-55:yaleB39_P00A-070E-35.pgm nearest index=144 dis=175251385
39-56:yaleB39_P00A-085E+20.pgm nearest index=984 dis=315382061
39-57:yaleB39_P00A-085E-20.pgm nearest index=739 dis=195901824
39-58:yaleB39_P00A-095E+00.pgm nearest index=739 dis=263241350
39-59:yaleB39_P00A-110E+15.pgm nearest index=984 dis=170148126
39-60:yaleB39_P00A-110E+40.pgm nearest index=319 dis=167477653
39-61:yaleB39_P00A-110E+65.pgm nearest index=1019 dis=67085285
39-62:yaleB39_P00A-110E-20.pgm nearest index=879 dis=152393919
39-63:yaleB39_P00A-120E+00.pgm nearest index=879 dis=183397940
39-64:yaleB39_P00A-130E+20.pgm nearest index=879 dis=104418264
in 39, hit=9/29
```

結果:

Totally, by SAD, hit rate=489/1084=45.110701%

Totally, by SSD, hit rate=427/1084=39.391144%

SAD 正確率約 45%

SSD 正確率約 39%

#### 4) 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。

#### 5) 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**。