

HW1 REPORT

Author: 404410030 資工三 鄭光宇

Method description

簡單來說，就是拿測試集中，電腦沒看過的新人臉，去看看訓練集中哪張臉跟它最像，就以這張「最像」的人臉所對應的類別，作為測試集中這張人臉對應的類別。

兩張人臉之間的相似度以平方距離和 (SSD)，或是以絕對值距離和 (SAD) 為基準，誤差愈小代表兩者愈接近。

這次作業評估方法效果的方式是使用 accuracy，在測試集上計算。

在作業程式碼中，每種人臉取前 35 張做為訓練資料，其餘做為測試資料。

為了使得程式碼更簡潔、有效率，我在程式碼中用了特別多類似 Functional Programming 的方式去實作 SSD, SAD 與 Nearest Neighbor，並且在計算 Nearest Neighbor 時，使用 python 中的 `threading` 模組平行化、加速計算 Nearest Neighbor。多虧這次一時興起想練習這種寫法，讓我可以更加了解如何寫出更簡潔、有效率的 python 程式碼。

Experimental results - accuracy

照作業題目要求，將圖片先以檔名遞增排序後，每種臉取前 35 張作為訓練資料，其餘做為測試資料，跑 Nearest Neighbor 在訓練集 / 測試集上的 accuracy 如下：

```
Train SSD acc: 100.00%
Train SAD acc: 100.00%
Test SSD acc: 45.99%
Test SAD acc: 45.07%
```

Discussion of difficulty or problem encountered

因為作業要求分割訓練集與測試集的時候，必須把人臉圖片以檔名排序後的前 35 張當訓練集，剩下的當作測試集；再加上排序後的前 35 張與剩下的圖片，光線的明暗差異非常大，所以圖片以檔名排序後跑 SSD, SAD 的準確

率，相較將人臉圖片順序先洗牌後，再分割訓練集與測試集跑 SSD, SAD 的結果還要差。例如，如果多了將人臉順序洗牌的步驟，再去計算 SSD, SAD，可能會得到如下 accuracy:

```
Train SSD acc: 100.00%  
Train SAD acc: 100.00%  
Test SSD acc: 68.20%  
Test SAD acc: 63.87%
```

因為有隨機的成分在程式碼內，所以不一定能重現這組 accuracy，但是應該會很接近。有隨機洗牌的程式碼在 `NN_random_shuffle.ipynb`。

另外，剛開始做這個作業時，忘了把 `np.uint8` 轉成 `np.float32`，導致計算時 overflow，過了一段時間看程式碼才發現這件事。