

ML Final Project: Apparent Age & Gender Classification

電資 0310010 吳奕樂、0310025 洪子修

- Outline:
 1. Introduction
 2. Local Binary Pattern (LBP)
 3. Face Detector
 4. Structure & Feature Extraction
 5. Classification Model & Comparison
 6. Conclusion
 7. Reference

1. Introduction

這次作業的目標，主要是要進行人臉辨識，並按照性別、年齡層等分成八大類，如下表所示：

0: male child	1: male young	2: male adult	3: male elder
4: female child	5: female young	6: female adult	7: female elder

每個 class 的原始資料量如下：

	child	young	adult	elder
male	325	415	794	439
female	387	501	605	226

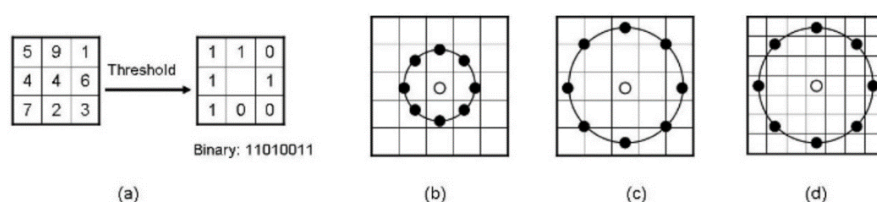
可以發現到，這樣的資料量對於訓練人臉辨識 model 其實相當不足，但因為本次作業並不允許我們用額外的人臉，因此我們需要想辦法做 Data Augmentation，把資料量提高；此外，也需注意 Data Balance 的問題。最後，我們利用一些方式，例如左右相反、在原始人臉上面增加一些 random noise 作為新圖像。

至於 Classification 的部分，我們在這次作業中嘗試了三種 model (SVM, Random forest, Adaboost)，以及一種 feature extraction 的方式(LBP)，並觀察 feature extraction 對於 model 的 performance 的影響。最後我們發現 LBP + SVM 的效果最佳，達到 57% 的正確率。

2. Local Binary Pattern (LBP)

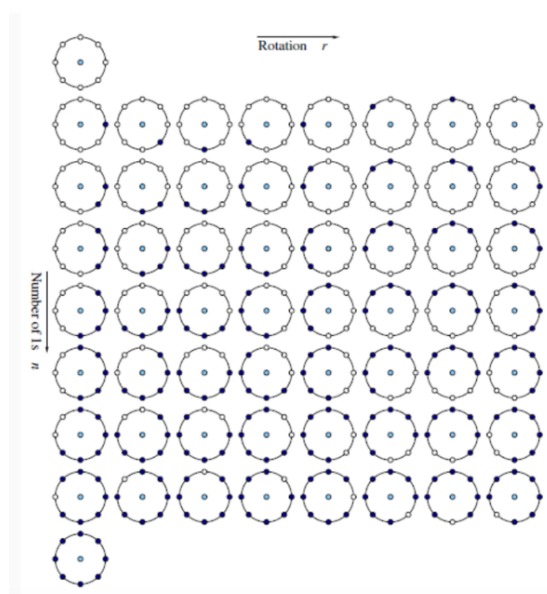
在進入實驗之前，我們要先介紹 LBP。LBP 是一種用來描述圖像局部紋理的運算子；從紋理分析的角度，圖像上某個像素點的紋理特徵，大多情況下指這個點與周圍像素點的關係。從哪個角度對這種關係提取特徵，便形成不同種類的特徵。有了特徵，就能根據紋理，進行分類。LBP 構造了一種衡量一個像素點和他周圍像素點的關係。

原始的 LBP，定義在 3×3 的窗口，以窗口中心像素為 threshold，將相鄰的 8 個像素的灰階值與其進行比較，若周圍像素大於中心像素值，則該像素的點標記為 1，否則為 0。這樣 3×3 的窗口便產生 8 個 bit 的表達，即得到該窗口的 LBP。(參考下圖(a))



但這樣的 LBP，只能抓到 3×3 的 window，無法抓到更大區域，所以有人提出改進，變成像上圖(b), (c), (d)這種圓環的 neighbor，再利用 bilinear interpolation 的方式計算 pixel 的灰階值，我們便能任意調整半徑、neighbor 數量，得到 LBP。那我們這次用的 uniform LBP 就是對上面的 LBP 再做些改進。

接著，介紹何謂 uniform patterns：我們定義它為記錄 0-1 變化少於或等於兩次的二進位編碼形成的 LBP，ex: 00000000, 00011110, 和 10000011 便是 uniform patterns。研究發現，在 texture image，我們計算出來的 LBP 大部分都落在 uniform patterns 內。以 (8, 1) 的 LBP，也就是 neighbors = 8 與 radius = 1 (圖(b))為例進行計算，有約九成的 pattern 都是 uniform pattern。如果 neighbor = 8，我們就有 58 個 uniform patterns:



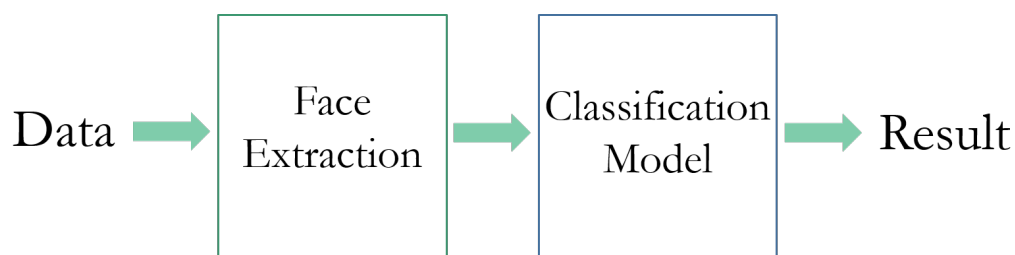
其餘不屬於 uniform patterns 的，我們另歸為一類，因此 neighbor = 8，我們總共有 59 類，有效從原本的 256 維 (2^8) 降到 59 維。

3. Face Detector

我們使用的是 dlib 的 face detector。他是利用方向梯度直方圖（Histogram of Oriented Gradients）的方法。首先，我們將圖像從彩色轉換成黑白。然後，對於每個像素，我們查看它周圍的其他像素，找出並比較當前像素與直接圍繞它的像素的深度，找出 gradients，也就是找出圖像上從明亮到黑暗的流動變化。

事實上，不用把每個 pixel 的 gradient 都存起來，我們只需要將圖像分割成很多小方格，算出 dominate gradient 來代替那個區域的 gradient 就好。接著我們用 sliding window 的方式，搭配一個 linear classifier 來分辨出哪一部分是人臉。

4. Structure & Feature Extraction



上圖是我們進行 Classification 的大致架構。一開始先將 Data 傳進 Face Extraction，利用 dlib 的 face detector 將人臉抓出來（有時候會碰到抓不到的情形，就先將原本的圖片 resize 成 2 倍大再去抓，通常可以抓到；如果還是抓不到的話，就直接 crop 中間的部分）。把臉從原始的圖中抓出來以後，做 LBP（詳細步驟下方會提到），得到 Feature Vector 以後，送到之前已經訓練好之 Model 進行辨識，最後再將結果輸出。

如同前面介紹 LBP 時提到的，LBP 可以提取出一些紋理特徵，對於我們做辨識或許會有較佳的結果。不過在做 LBP 之前，我們要先做一些 pre-processing：

首先，做 Face alignment。我們實作的方法是先定義兩個眼睛要在的水平位置，接著從臉的影像抓出眼睛所在的位置，再進行旋轉(affine transform)，使眼睛在這條水平線上。接著我們 crop image，使整張 image 主要是臉，而我們定義這個 cropped image 的 size 是 120*120。之後我們進行 histogram equalization，最後再做 LBP。而我們做 LBP 的方式，跟一般不太一樣。因為我們發現直接把 120*120 的 image 壓縮到 59 個 dimension，可能是

因為資料損失太多了，classification 的效果很差，大約 20% 左右，因此我們改成將 image 切成 12×12 等分，再對每一個 10×10 的 window 做 LBP (如下圖，不過下圖的 window size 為 5×5)，最後將這 12×12 個 LBP concatenate 成一個 $12 \times 12 \times 59 = 8496$ 維的 feature vector，送入 classifier 進行 classification。



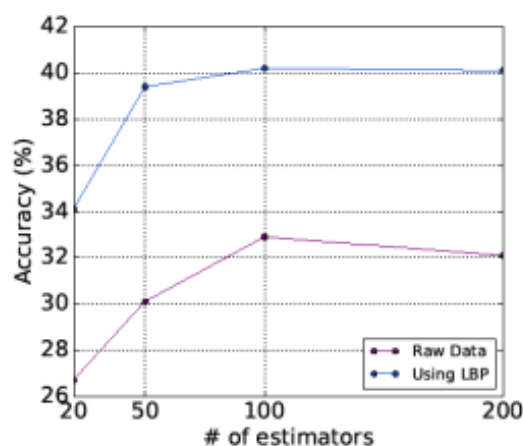
總結一下，在送入 Classification Model 之前，步驟如下：

- Step 1. 先利用 dlib library 中人臉辨識功能，找出圖片中的人臉。
- Step 2. 做 Face Alignment (讓人臉的眼睛是平行線)。
- Step 3. 把抓出來的人臉進行 resize，resize 成 120×120 。
- Step 4. 做 Histogram Equalization，移除光線之影響。

5. Classification Model & Comparison

我們認為，如果使用 Convolution Neural Network，需要大量的資料，才能達到較高的正確率，因此我們在實作的時候，就沒有去使用 CNN 了。

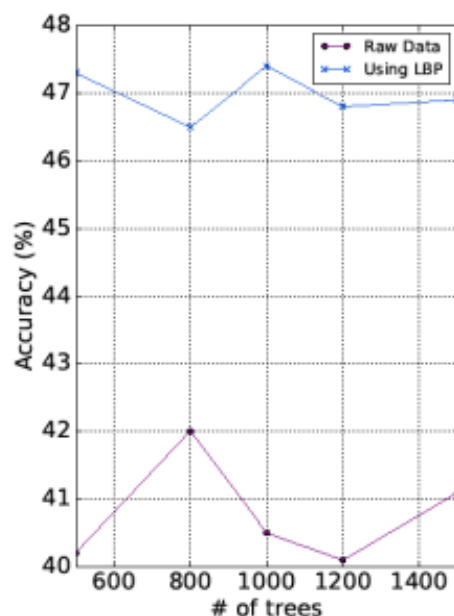
我們總共用了三種 Model，Adaboost、Random Forest 以及 SVM。下圖為使用 Adaboost 所得到之測試結果：



從上圖，我們可以觀察到兩件事：第一，如果我們有對 Data 進行如 LBP 的特殊處理，可以有較高的 Accuracy (相較於 Raw Data，做完 LBP 以後得到的 feature 數量較少，某種程度上，也算是達到了降維的目標)；第二，隨著 estimators 數量的增加，Accuracy 一開始會有顯著的進步，但是當

estimators 數量達到一定數目時，準確率就會趨於飽和，不再有顯著的進步，會在特定範圍之間上下浮動。

接著，試試 Random Forest，得到下圖：



同樣的，透過 LBP 處理，我們有效地將辨識準確率提升了起來，另外，Tree 的數量和準確率間，感覺並沒有絕對的關係，不過這還需要進一步地往下測試，我們時間有限，就沒有繼續做這部份了。

最後一個 model，是 SVM；我們使用的是 Nu SVM，因此能夠調整的參數就只有 Nu 值（定義：“The parameter Nu is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training examples.”），在經過測試以後，我們發現 SVM 的效果遠比上面兩種方法來得好，因此我們試著調整參數（包括 LBP 的 window size、neighbor 取法、Nu 值），希望可以有效提升 classification model 的 accuracy，測試結果如下表（註：0.1~0.5 為 Nu 值，Method 中的 LBP(8, 2)，代表的意思是取距離中心點 2 個 pixel 的 8 個 neighbor）：

Method	LBP Window Size	0.1	0.2	0.3	0.4	0.5
Raw Data	X	0.453	0.462	0.481	0.473	0.471
LBP (8, 2)	10 * 10	0.528	0.559	0.574	0.572	0.527
LBP (8, 2)	12 * 12	0.511	0.549	0.564	0.549	0.526
LBP (8, 2)	15 * 15	0.513	0.532	0.569	0.555	0.542

從上表，可以看得出來，不同的 window size，對於結果都會有所影響；我們原本以為，當 window size 越變越大，代表得到的 feature vector 之 dimension 會變小，Model performance 可能會越變越不好，不過最後結果出

乎我們意料之外：當 window size 為 15*15 時，model performance 整體看起來，甚至還比 12*12 來得好！

另外，在前面我們也有提到，我們有對人臉使用了 Histogram Equalization，為了證實 Histogram Equalization 對於人臉辨識有較好的結果，我們也做了一些測試，得到了下表之結果（註：使用 random forest model、以及 raw data）：

# of tree	500	800	1000	1200	1500
No equalization	35.1 %	34.9 %	34.3 %	33.9 %	35.7 %
Equalization	40.2 %	42.0 %	40.5 %	40.1 %	41.1 %

至於為什麼 Histogram Equalization 對於結果有所幫助，我想是因為在有正確擷取到人臉的前提下，他將膚色對於辨識結果的影響移除了；期末專題的目標是要可以準確地判斷年齡、性別，若能將資料先做處理，把特定的因素給排除，肯定有機會能因此得到更好的分類結（ex:黑人的人臉可能偏暗，白人的人臉可能偏亮，在經過 Histogram Equalization 以後，可以讓像素分布地更為平均，排除掉膚色的影響）。

在做完各種測試以後，我們挑出各種 Classification Model 的最佳數據，進行比較：

Correct Rate	SVM	Ransom Forest	Adaboost
Raw Data	48.0 %	42.0 %	32.9 %
LBP	56.9 %	47.3 %	40.2 %

很明顯地，使用 LBP + SVM，將能得到最佳結果。

6. Conclusion

在使用了 LBP，並以 SVM 作為 classification model 的情況下，我們可以有較高的準確率。我認為 LBP 之所以會有較好的結果，是因為他可以將紋理特徵提取出來，不同的年齡層、性別，或許有些細節是我們肉眼看不出來的，但是透過 LBP，可以將這些細節放大，當我們利用這些細節作為我們的 training data 時，或許就能讓辨識度有效地提升起來。

在實際 demo 時，測試出來的準確率，和我們實作時得到的 accuracy 相差不到 2%，正確率都落在 56~58%之間；這代表我們的 model 還滿穩定的，並沒有發生 overfitting 的問題。

7. Reference

- A. Quora Forum
- B. Stack Overflow Forum
- C. Scikit-learn
- D. Multi-view Gender Classification Using Local Binary Patterns and Support Vector Machines (Author: Hui-Cheng Lian, Bao-Liang Lu)