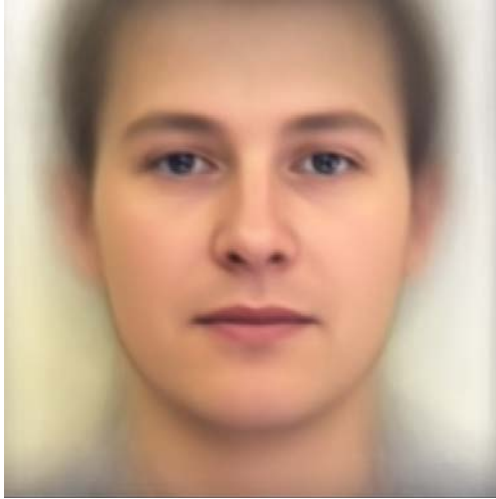


ML2017FALL HW6 Report

學號：B04902090 系級：資工三 姓名：施長元

1. PCA of colored faces

(.5%) 請畫出所有臉的平均。



(.5%) 畫出前四個 Eigenfaces，即對應前四大 Eigenvalues 的 Eigenvectors。



Eigenface 1



Eigenface 2









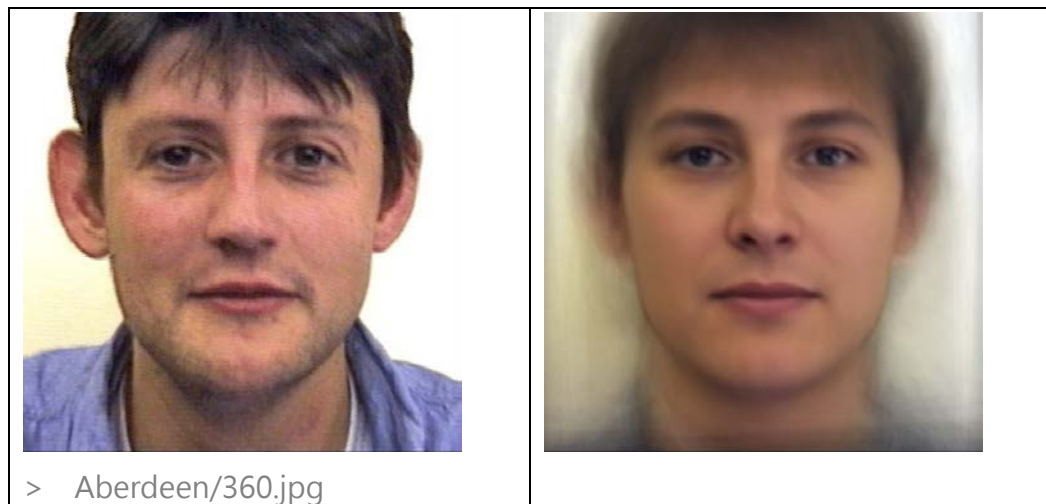
Eigenface 3



Eigenface 4

(.5%) 請從數據集中挑出任意四個圖片，並用前四大 Eigenfaces 進行 reconstruction，並畫出結果。(Collaborator: B04902012 劉瀚聲)

Original	Reconstructed
 > Aberdeen/22.jpg	
 > Aberdeen/77.jpg	
 > Aberdeen/197.jpg	



(.5%) 請寫出前四大 Eigenfaces 各自所佔的比重，請用百分比表示並四捨五入到小數點後一位。

Eigenfaces 1: 4.1%

Eigenfaces 1: 2.9%

Eigenfaces 1: 2.4%

Eigenfaces 1: 2.2%

2. Visualization of Chinese word embedding

(.5%) 請說明你用哪一個 word2vec 套件，並針對你有調整的參數說明那個參數的意義。(Collaborator: B04902089 林政豪)

我使用 Gensim 的 models.word2vec，並且使用 Jieba 進行分詞。

Jieba 的參數: 使用 dict.txt.big，用 Stop_words 將標點符號拿掉，

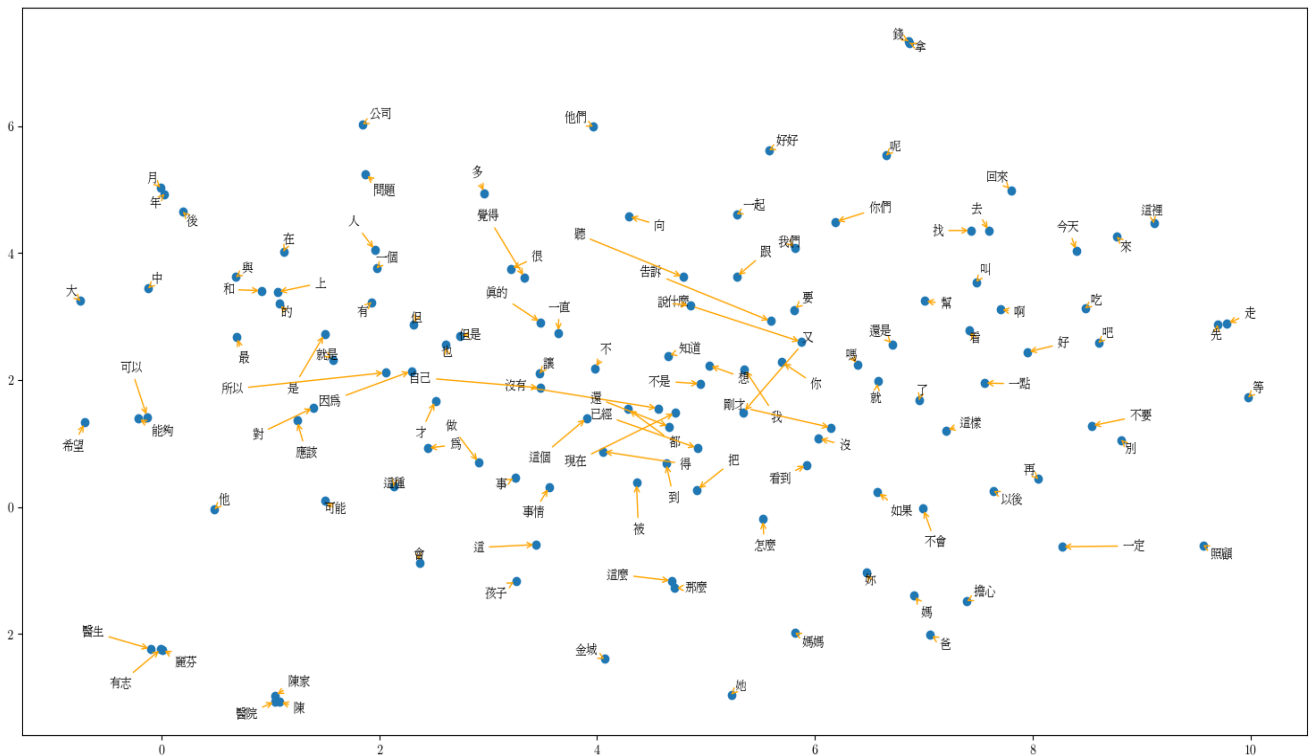
cut_all=False

Gensim 的參數: 使用 word2vec.Text8Corpus 讀入剛剛 Jieba 的結果；

而 word2vec 參數: 100 維，Windows=5，workers=4，min_count=3500

訓練出來之後畫圖，使用 T-sne 降維進行可視化；字體使用 cwTeXMing，因為必須是 TryeType 字體，找了好一陣子。一樣因為 T-sne 有一定的隨機性，所以進行了很多次想要找到某些字詞的規律

(.5%) 請在 Report 上放上你 visualization 的結果。



(.5%) 請討論你從 visualization 的結果觀察到什麼。

- 有相互關係的詞會出現在同一塊。Ex. (陳家,陳,醫院) · (醫生,有志,麗芬) · (可以,能夠,希望)...等等
- 因為是降維的圖，所以很難看到向量關係；這裡有找到(這,這麼,那麼,怎麼)，看得出來“怎麼”加上“這”，會接近“這麼”。其他的可能是不同維度才看得出來，其實蠻難找的
- 每次進行 T-sne 出來的結果都不同，應該有 Random 成分在裏頭(挑選維度進行降維)
- 金城多到出現在這張圖上了 xDDD

3. Image clustering

(.5%) 請比較至少兩種不同的 feature extraction 及其結果。(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)

(Collaborator: B04902089 林政豪)

- 個人 Kaggle Best: 使用 Auto-encoder 降到 32 維，再進行 Kmean 進行分群即得到答案。

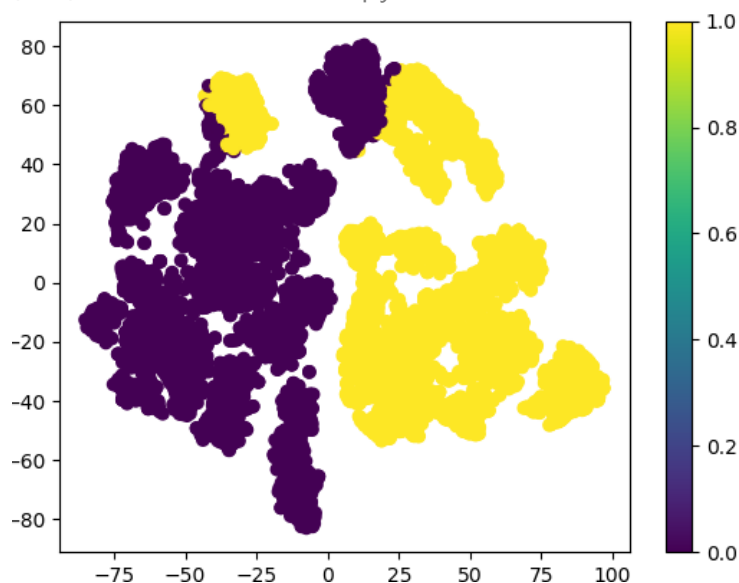
> Result: Kaggle **Public 0.99940, Private 0.99934**

- 後來實作方法: 使用 Auto-encoder 之後降到 32 維，然後使用 T-sne 降到 2 維，而後進行 Kmean 進行分群得到答案

> Result: Kaggle **Public 1.00000, Private 1.00000**

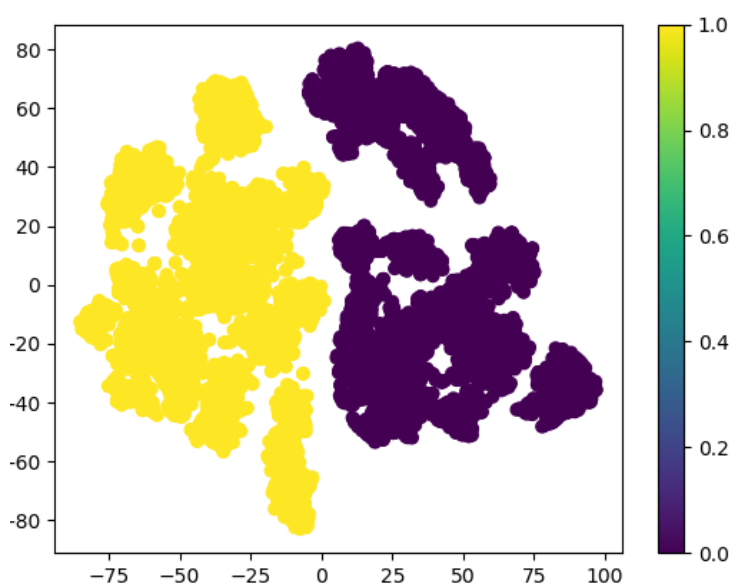
III. 無聊實作方法:使用 Sklearn PCA 降到 32 維之後，使用 Kmean 進行分群。
> Result: Kaggle Public 0.03024, Private 0.03051

(.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 分佈



Model 是重新針對新給的 npy 檔案進行 Train，訓練是 200 個 Epoch，使用的是 Autoencoder 將資料降成 32 維，而後直接使用 Kmean 進行分群，之後用 T-sne 降維使之可視化，再利用 Kmean 之後的結果進行標註顏色。T-sne 有他的隨機性，有時候可以很正常的分成兩堆(中間沒有隔開)，但這張很顯然有一堆黃色是跑到紫色那邊去了，證明我的 Model 並沒有完全分好這兩堆。

(.5%) visualization.npy 中前 5000 個 images 跟後 5000 個 images 來自不同 dataset。請根據這個資訊，在二維平面上視覺化 label 的分佈，接著比較和自己預測的 label 之間有何不同。



我刻意使用相同次的 T-sne 進行可視化，可以見到兩者就分得非常開，完整呈現分堆。