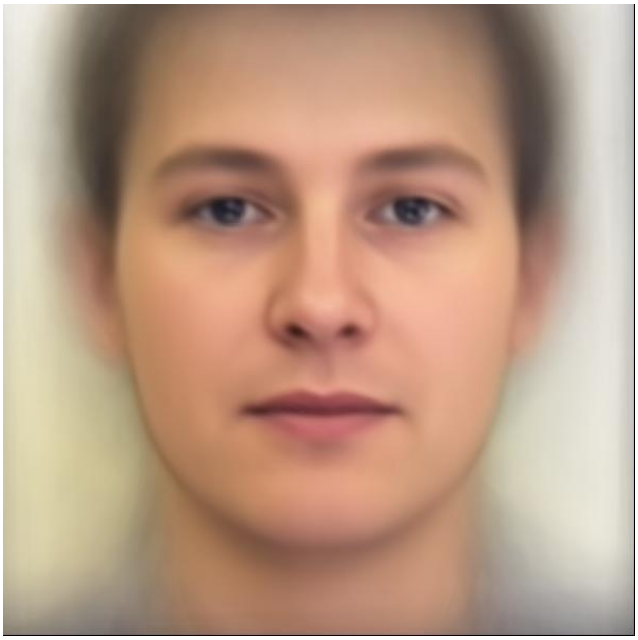
學號：R06943153 系級： 電子碩一 姓名：蘇旻彥
(Collaborators: R06943147 李明庭、R06943084 游彥勝)

A. PCA of colored faces

A.1. (.5%) 請畫出所有臉的平均。



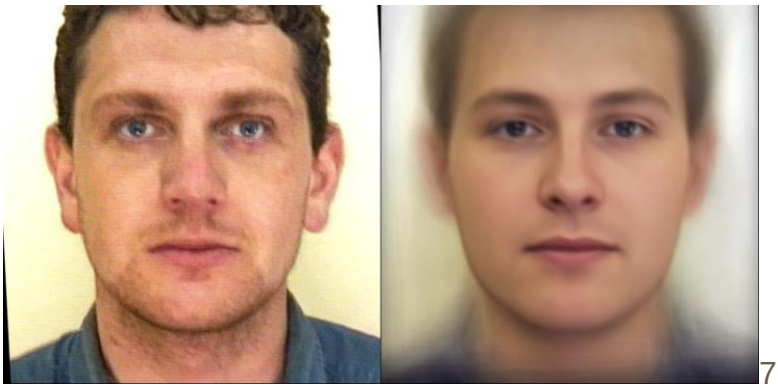
A.2. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

下圖依序為第一到第四個 Eigenface (左上、右上、左下、右下)。

A.3. (.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。

以下依照順序為第 3、76、119、216 張圖片的原圖(左)，以及重建後的結果(右)。可以觀察到若只用前四個 Eigenfaces 進行重建，男性照片的結果會很像平均臉，而女生的照片比較看得出差異。


3


76


119


216

A.4. (.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。

前四大 Eigenfaces 的比重依序為：4.1%、3.0%、2.4%、2.2%。

B. Visualization of Chinese word embedding

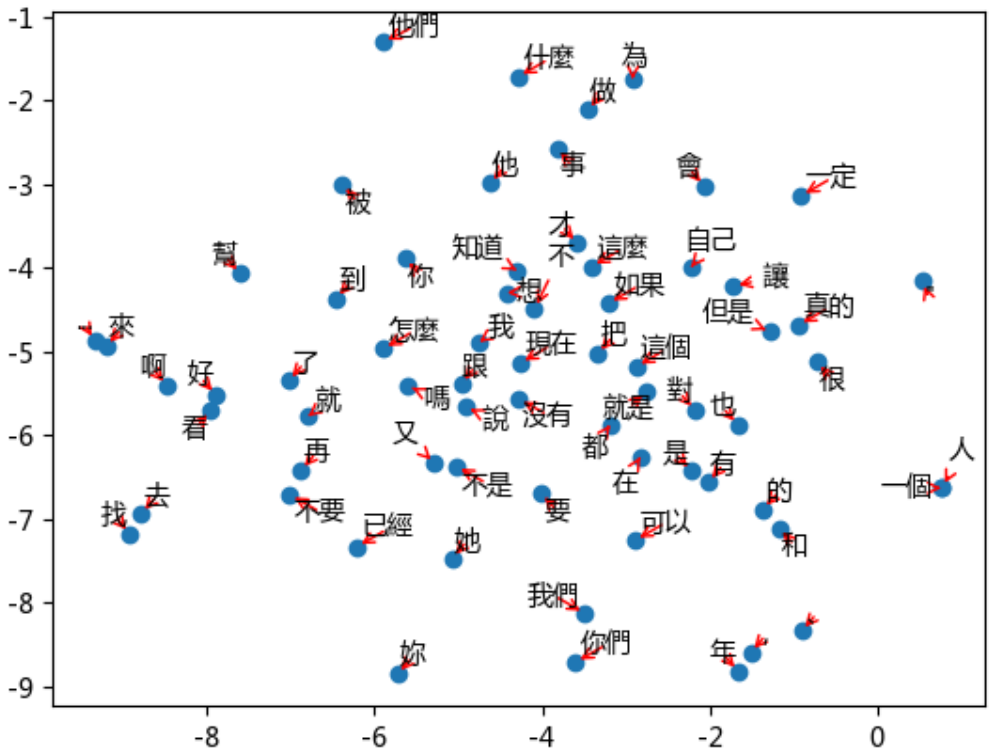
B.1. (.5%) 請說明你用哪一個 word2vec 套件，並針對你有調整的參數說明那個參數的意義。

我使用的是 gensim.models 當中的 word2vec，當中除了 default 的參數以外，有自己改動的參數如下：

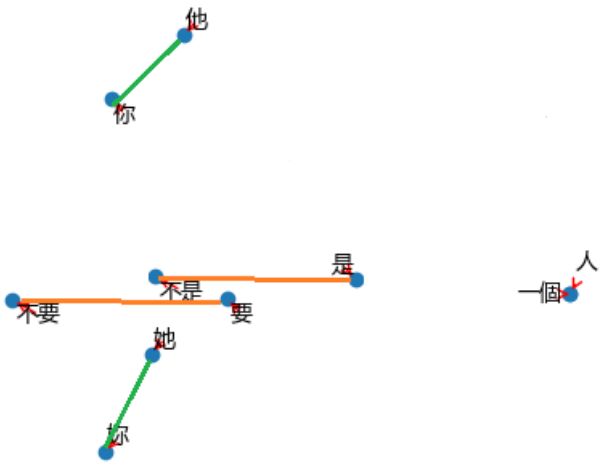
(1) size=128，這個 size 其實就是 embedding dimension，感覺用 2 的指數比較好。

(2) mincount=7000，是代表出現超過 7000 次的單字才計算，若設定太少會出現很多專有名詞，比較難分析。

B.2. (.5%) 請在 Report 上放上你 visualization 的結果。



B.3. (.5%) 請討論你從 visualization 的結果觀察到什麼。



把幾個觀察到的比較有意義的結果取出來看：

(1) [你,他] 和 [妳,她] 的方向是相同的，對應了不同性別的相同代名詞。

(2) [是,不是] 以及[要,不要] 的方向相同，對應了肯定以及否定的用詞。

(3) [一個,人] 兩個點完全重和，代表我終究只能一個人，嗚嗚……。

C. Image clustering

C.1. (.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)

以下使用三種不同的方法：

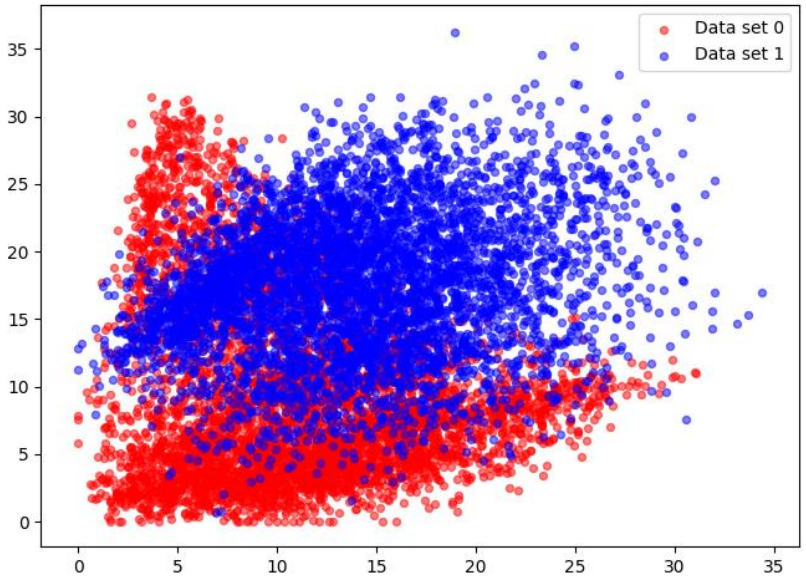
	降維方法	分群方法	Kaggle Public Score
1	autoencoder(DNN)	Kmeans	0.99148
2	autoencoder(CNN)	Kmeans	0.91400
3	autoencoder(CNN)	Cosine Similarity	0.35164

在降維方法的部分，使用 DNN 及 CNN 兩種不同的 model 做 auto-encoder，層數都是 encode，decode 各三層，壓縮到 32bit；分群方法方面，Kmeans 為 sklearn.cluster 中的功能，而 Cosine Similarity 則是直接將兩張圖片降維過後比較他們的餘弦相似度，若高於 0.99 則判斷維相同。下圖為 DNN 的 model summary：

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 784)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_4 (Dense)              (None, 64)                2112
_________________________________________________________________
dense_5 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_6 (Dense)              (None, 784)               101136
=================================================================
```

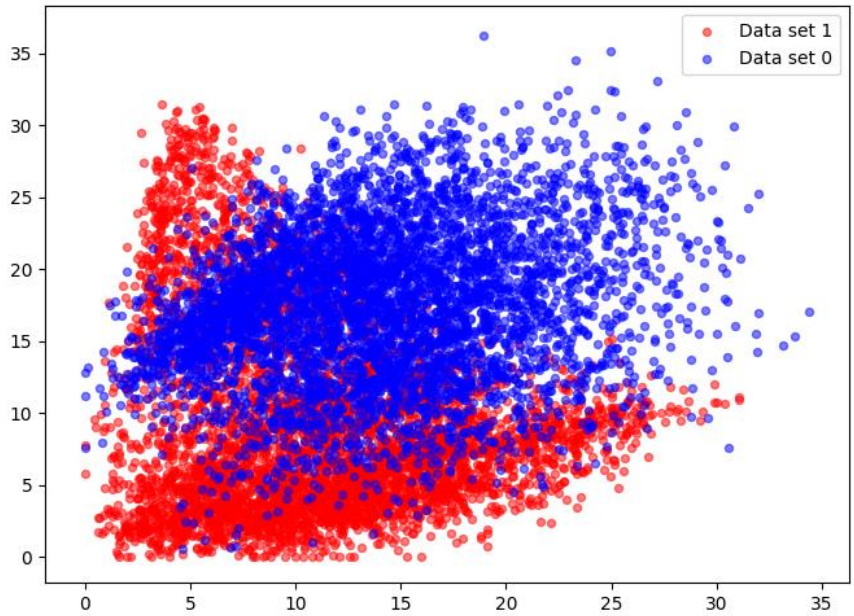
從結果可以觀察到 DNN 比 CNN 的結果好上許多，而 Kmeans 為套件中的功能，效果自然很強，比直接比較餘弦相似度分數高上很多。

C.2. (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。



上圖為 encode 到 32 維後取前 2 個維度的視覺化結果，可以看到雖然有明顯分群，但仍然有重疊的部分。

C.3. (.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。



可以看到預測出的結果跟解答所繪出的結果幾乎相同。

(我很努力地找出不同點→座標(0,7)的地方有些許不同)