

A. PCA of colored faces

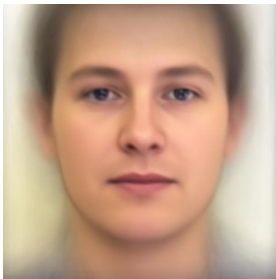
(.5%) 請畫出所有臉的平均。

A.1. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

A.2. (.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。

A.3. (.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。

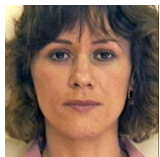


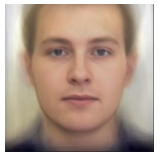
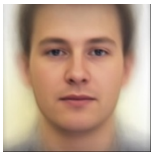
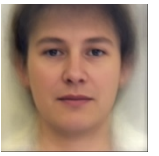
1.



2.



3.

原圖				
還原				

4. 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。
- 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。

1.

使用 gensim 做 word2vec

```
model = word2vec.Word2Vec(trainData, size=num_features, window=7, sg=1, hs=1, min_count=5, workers=12, iter=10)
```

size : embedding 維度

window : train 時往前看幾個字一起考慮

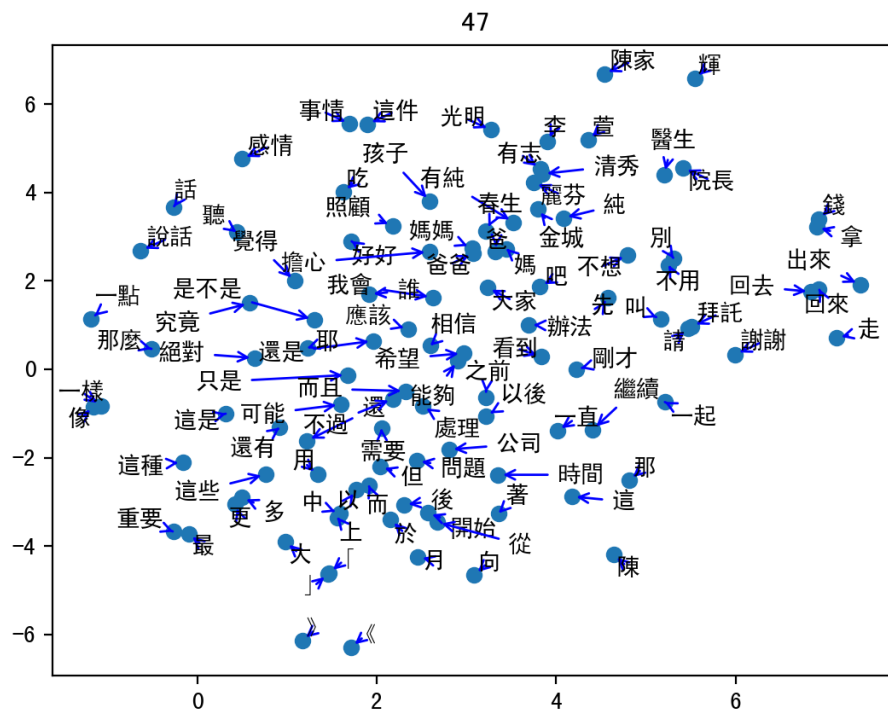
sg : train word vector 的方式

min_count：最少出現次數才考慮

workers：使用運算數量

iter : iteration

2.



3.

原則上來說，或許部分是根據真實的出現頻率 前後文關係等等，尤其以前後詞必相連的該些字詞觀察，他們皆會在附近，比如：人名

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

1.

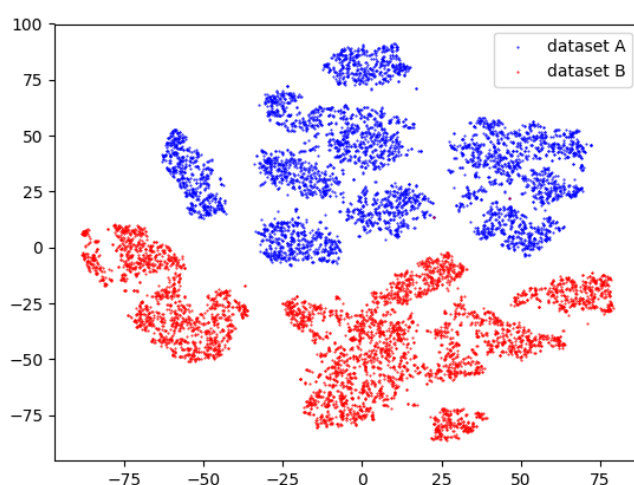
	private	public
法一	0.97904	0.98075
法二	0.98660	0.98820

法一：CNN 做 autoencoder 降維

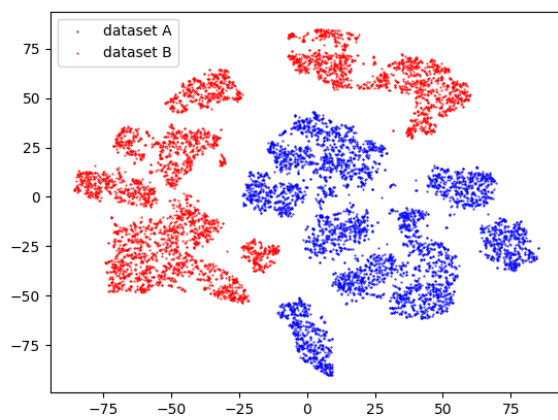
法二：DNN 做 autoencoder 降維

原則上來說 他們兩個具有差不多的能力能夠完成這個 task

2.



可見在藍色區塊有幾個紅點 表示有少數的錯誤發生
3.



tsne 分得很好