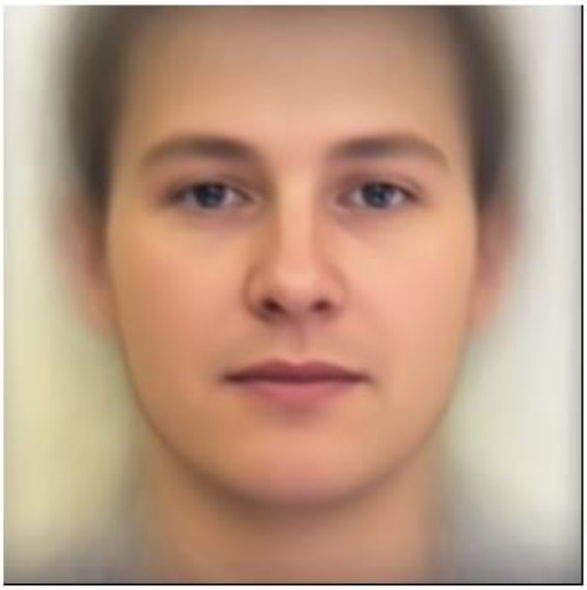
Machine Learning HW6: Unsupervised Learning & Dimension Reduction

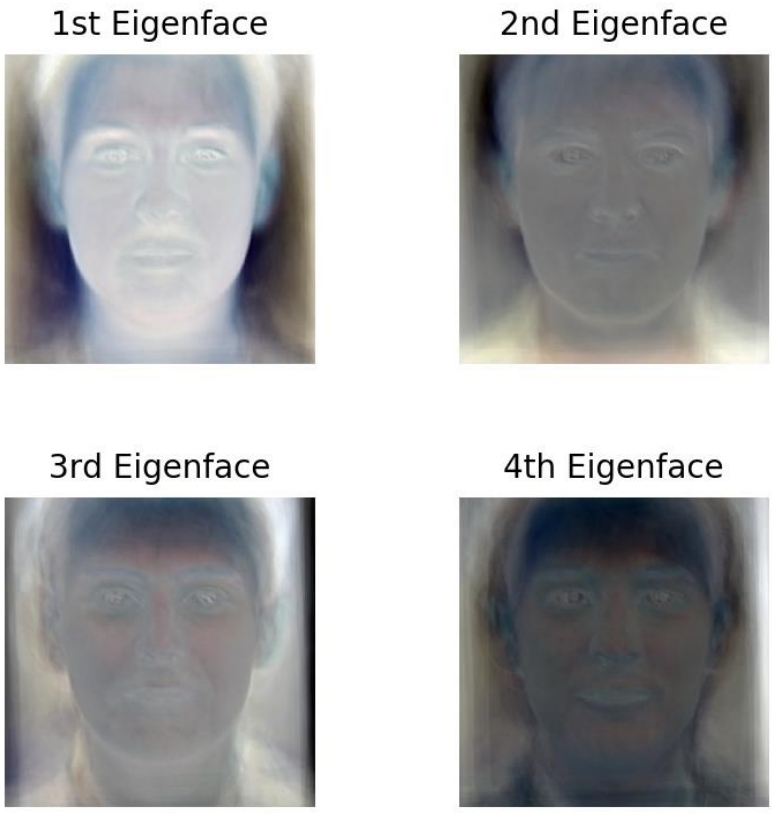
學號：R06922030　系級：資工碩一　　姓名：傅敏桓

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。



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。

1.jpg

3.jpg

5.jpg

7.jpg

A.4. (.5%) 請寫出前四大 Eigenfaces 各自所佔的比重(explained variance ratio)，請四捨五入到小數點後一位。

取 SVD 解出之前 4 大奇異值，計算各自所佔比重為 [4.1%, 2.9%, 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。

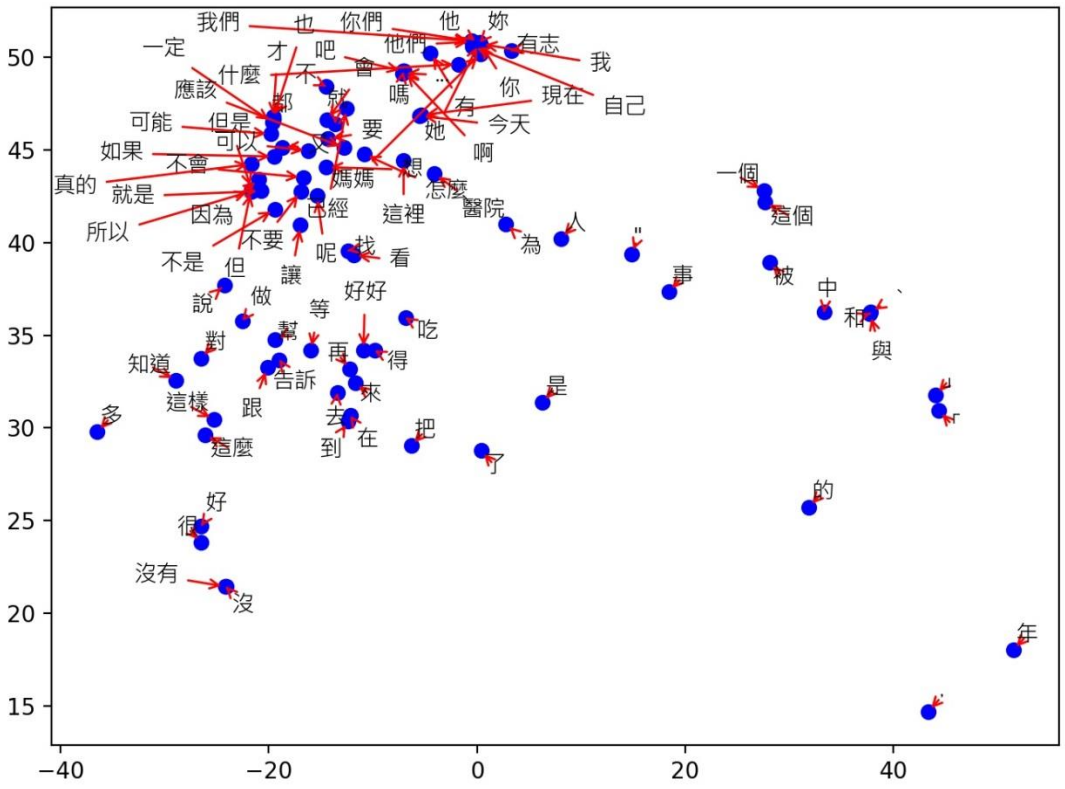
這次用 gensim 的 Word2Vec 來實作詞嵌入，預設方法是 CBOW。以下是實驗的參數設定，以不同 window 大小進行實驗，未列出的部分使用預設值。

size	100	詞嵌入向量的維度大小
window	{3,5}	考慮的前後文總長度
min_count	5	忽略出現總數在 min_count 以下的詞
workers	4	訓練時的平行線程數

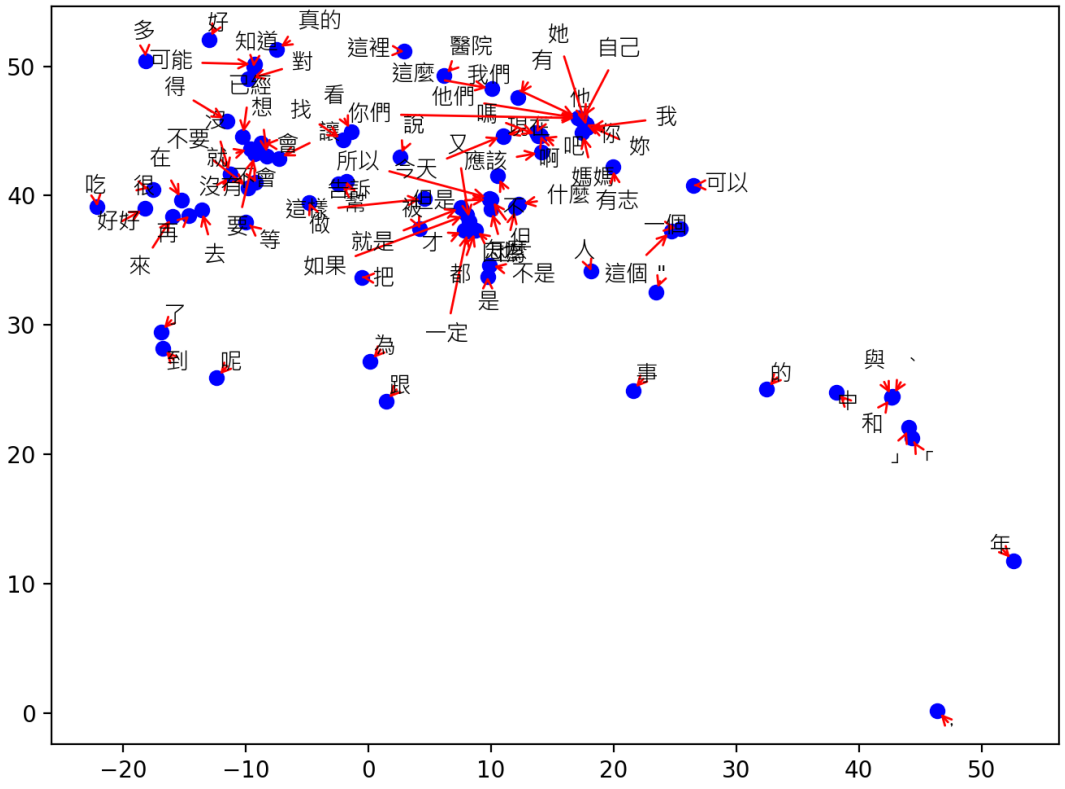
最後算出共 31429 個詞之嵌入向量，透過 t-SNE 降到 2 維，挑選出現次數超過 5000 的詞做視覺化，結果如 B.2.所示。

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

以下依序是 window = 3 和 5 的結果：



window=3



window=5

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

推測出現超過 5000 次的詞大多是在台詞中出現的，主要可以看到代名詞和一些概念較簡單的動詞。首先可以觀察到代名詞有聚集的現象，由於 word2vec 是透過 CBOW 的方法實作，前後文相似的詞相似度應該會較高，而這些代名詞在句子中的位置和人或人名差不多所以也都聚在一起。標點符號觀察到到除了上下引號相似度很高外，其餘標點之間並沒辦法找出相似關係。另外可以觀察到像是「沒有、沒」、「這個、一個」、「和、與」、「很、好」等概念相近的詞，確實也有很高的相似度。

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)

圖片分別來自手寫數字和服飾的資料集，觀察兩個資料集的資料特性：

- 手寫數字的圖片強度是 0 的部分通常佔大部分；服飾除了鞋子以外非 0 的點較多。
- 手寫數字強度接近 255 的點較多，服飾分布得平均一些。

根據上述的特性差異，可以提出一個 naïve 的特徵抽取方法：對每一筆資料，計算強度為 0 的資料點總數作為圖片的特徵，另外加上強度大於 0.95 的資料點總數，可以稍微將鞋子和數字再區分開來。假設兩個資料集提供差不多的圖片，直接以大於中位數與否進行聚類，嘗試三種預測方式：①兩邊都大於中位數②兩邊都小於中位數③兩邊都大於中位數或都小於中位數，最後是②在 Kaggle 得到較高的 Private 分數 0.42921。

另外也使用 sklearn 的 PCA 進行降維後再以 sklearn 的 KMeans 進行聚類。以 PCA 和 KMeans 可以很有效的解決這個問題，但使用的參數需慎選。使用 randomized SVD solver 會讓每次結果有些微差異，所以實驗時有多試幾次。最後 PCA 和 Kmeans 的參數設定如下：

PCA		**KMeans**	
n_components	280	init	k-means++
whiten	True	n_clusters	2
svd_solver	randomized	n_init	10

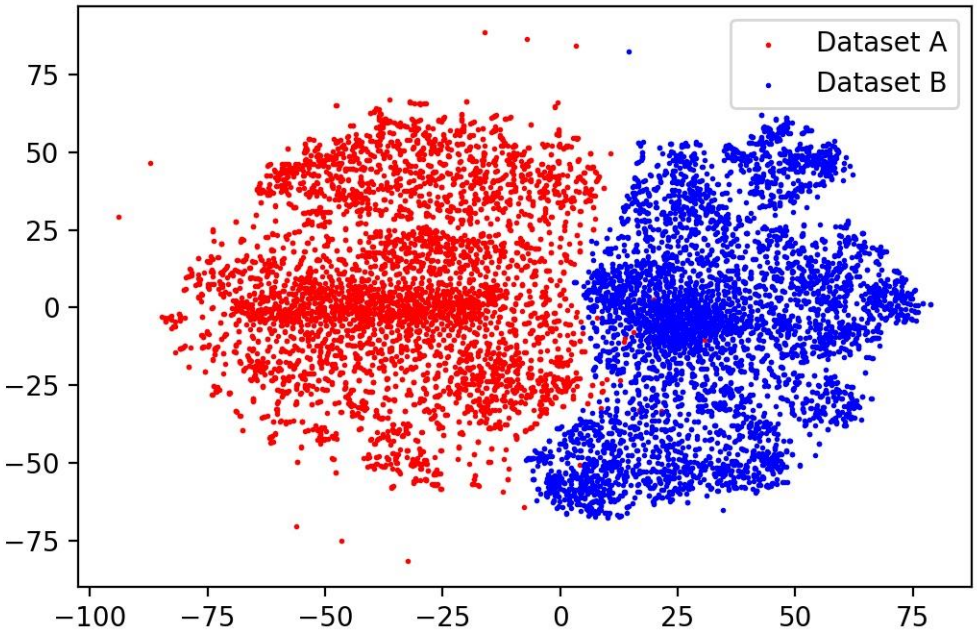
得到的結果非常理想，得到的 F1-score 是 1.0，且切開的兩類分別都是 70000 筆資料，應該是切得很完美。

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

以 fit 在全部資料 (image.npy) 上的 PCA 對部分資料 (visualization.npy) 進行降維後再以 KMeans 進行聚類 (n=2 的情況下用 fit 在全部資料和 fit 在部分資料得到的聚類結果相同)。視覺化時以 t-SNE 再將資料降到二維空間：

```
TSNE(n_components=2, init='pca').fit_transform()
```

作圖得到的結果如下，觀察不同標記的資料之分布，除了少數資料點外大致切成完整的兩塊，顯示 PCA 降維取特徵的效果還不錯，聚類後即便投影到二維平面上也還看得出明顯分界。



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。

依照 C.2.之方法進行聚類的結果與實際情況相符，前 5000 個和後 5000 個影像來自兩個不同的資料集，作圖的結果也和上圖相同。