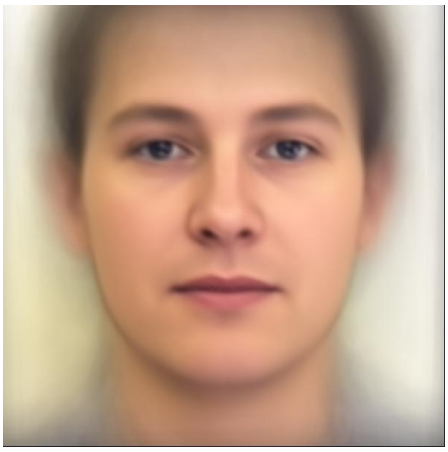
ML2017FALL-HW6-Report

學號：B03801039　系級： 電機四　姓名：楊福恩

A. PCA of colored faces

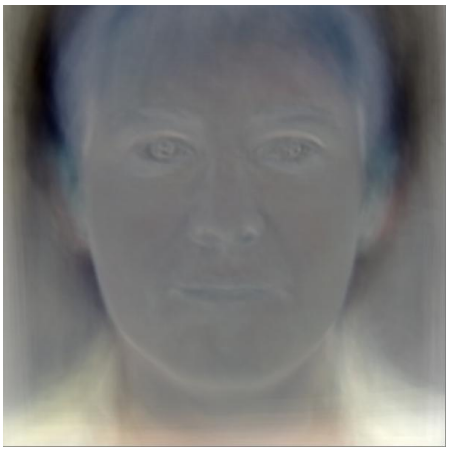
A.1.　(.5%) 請畫出所有臉的平均。



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

答:以下為前四個 Eigenfaces

第一個 Eigenface



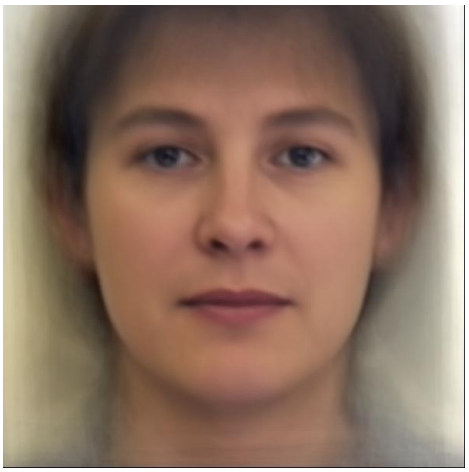
第二個 Eigenface

第三個 Eigenface

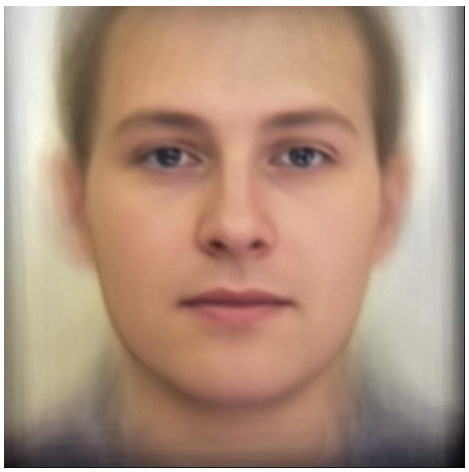
第四個 Eigenface

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。

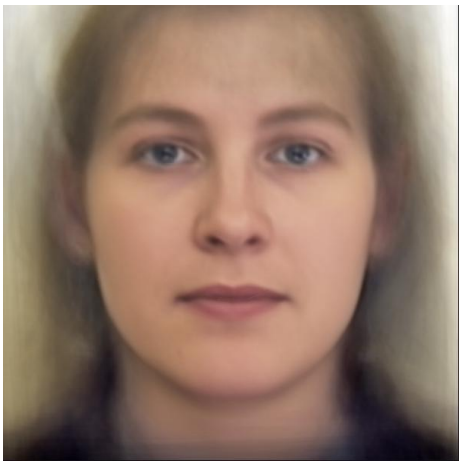
答:



0.jpg



214.jpg



215.jpg

227.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。

答:第一大 Eigenface 比重=41.4%

第二大 Eigenface 比重=29.5%

第三大 Eigenface 比重=23.9%

第四大 Eigenface 比重=22.1%

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。

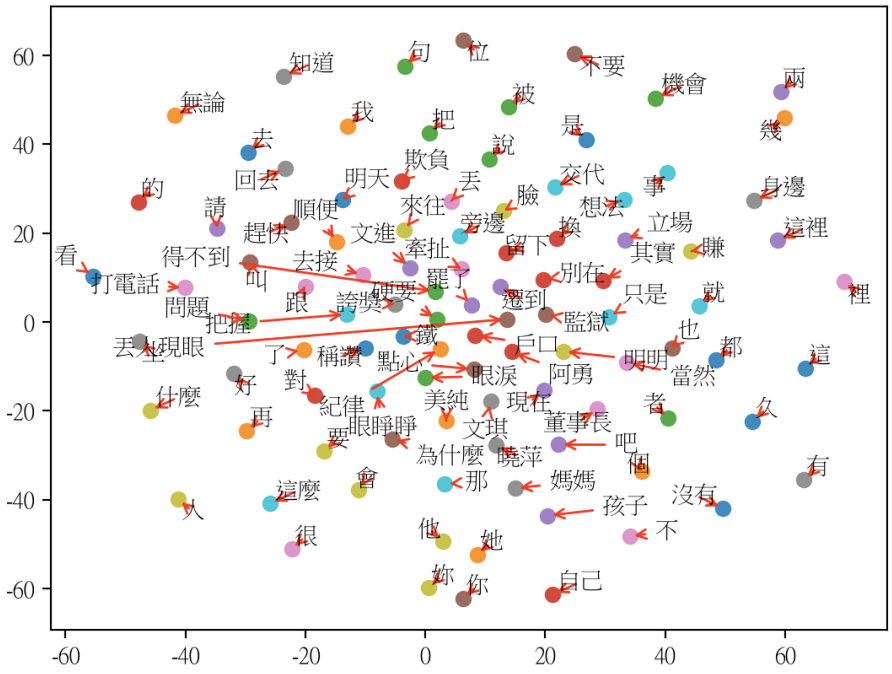
答:使用的套件為 gensim，有調整的參數為 size 和 min_count，

將 size 設為 120，min_count 設為 10，

其中，size 這個參數的意義為經由這個 word2vec 的模型訓練出來的詞

向量的維度;而 min_count 則是代表若在訓練的文本中，某個詞出現的次數小於 min_count，那麼這個詞就不會被視為訓練的對象。

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。

答:可以發現像是「你、妳、他、她」這幾個第二、第三人稱的代名詞，被聚集在一起，可推測原因應為這幾個代名詞都是在對話中指涉別人的，原本即為較為類似的代名詞，所以做了 word2vec 後也自然地被聚在一起。

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)

答:

第一種方法: Deep autoencoder + kmeans:

Deep autoencoder 的架構如下圖所示，將原本影像降維成 32 維的 code，再用 kmeans 做 clustering。

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 784)	0
dense_1 (Dense)	(None, 256)	200960
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 32)	2080

Total params: 244,192
Trainable params: 244,192
Non-trainable params: 0

結果:

Kaggle public score= 0.99945

Kaggle private score= 0.99914

可見結果非常好，幾乎有成功分辨出兩張圖片所屬的 dataset。

第二種方法: PCA + kmeans:

利用 PCA 將原影像降維成 32 維，再利用 kmeans 做 clustering

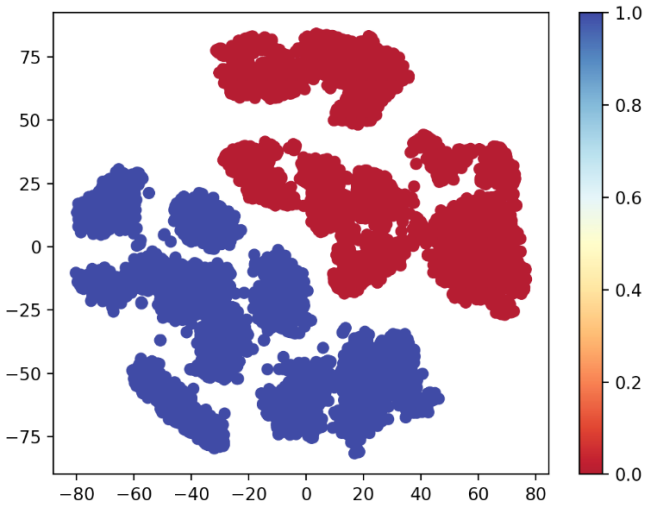
結果:

Kaggle public score= 0.03021

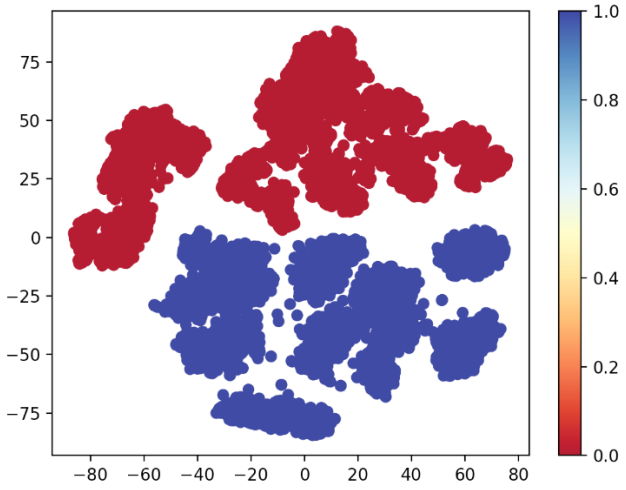
Kaggle private score= 0.03046

可見結果差非常多，可知在這個任務上，利用 PCA 做降維無法有效地取出有用的 feature 來進行下一步的 clustering。

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。



比較兩圖後可發現分布情況皆為一個 dataset 分布在圖中偏上方，另一個 dataset 分布在圖中偏下方，且在兩張圖上都可見兩個 dataset 有明顯的分隔，符合 Deep autoencoder + kmeans 這個架構在 kaggle 分數很高的這個結果，另外可觀察到兩圖在分布情況上看起來略有不同，推測應該是因為 TSNE 本身若初始化不同，則呈現結果也會不同，故導致可視化的分布情形略有差異。