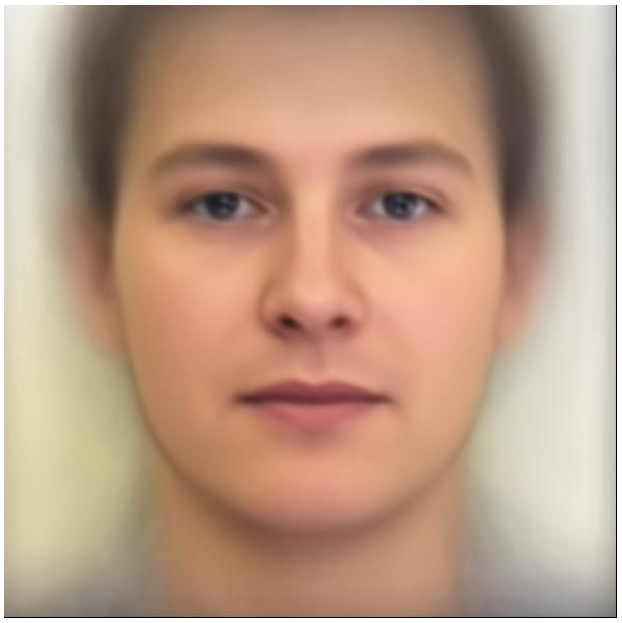
學號：B04901066 系級：電機三 姓名：洪國喨

A. PCA of colored faces

(collaborator: NULL)

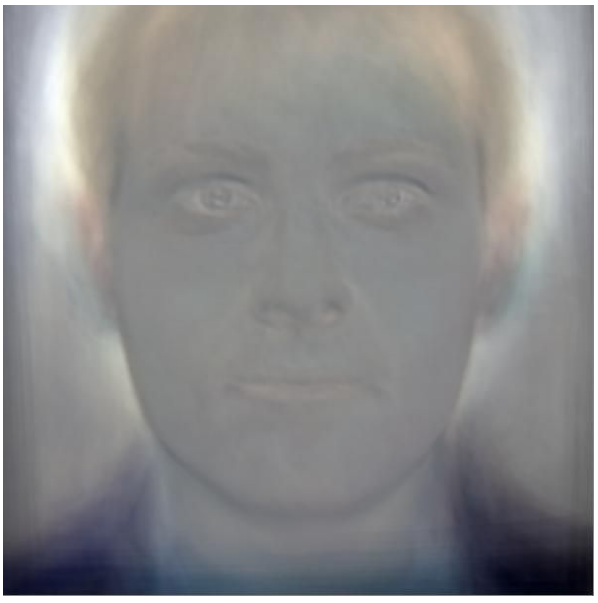
A.1. (.5%) 請畫出所有臉的平均。



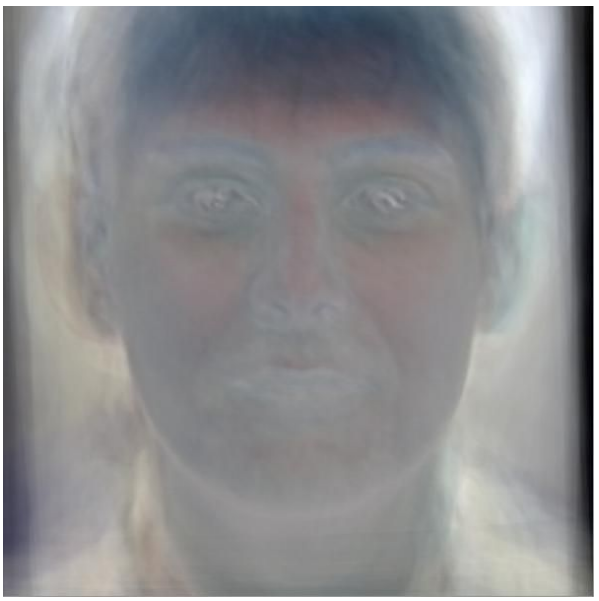
A.2. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

第一張

第二張



第三張

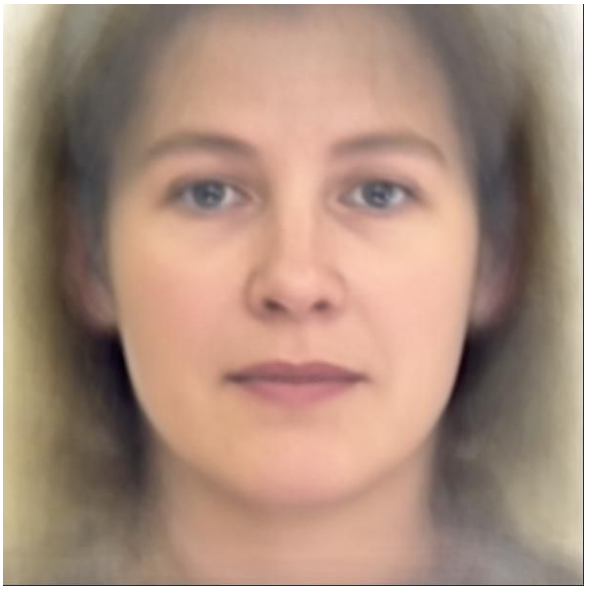


第四張

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。

輸入

重建結果



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。

第一	第二	第三	第四
4.1 %	2.9 %	2.4 %	2.2 %

B. Visualization of Chinese word embedding

(collaborator: NULL)

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。

gensim 內 models 提供的 word2vec 模型

model = word2vec.Word2Vec(sentences, size=128, window=5, min_count=5)

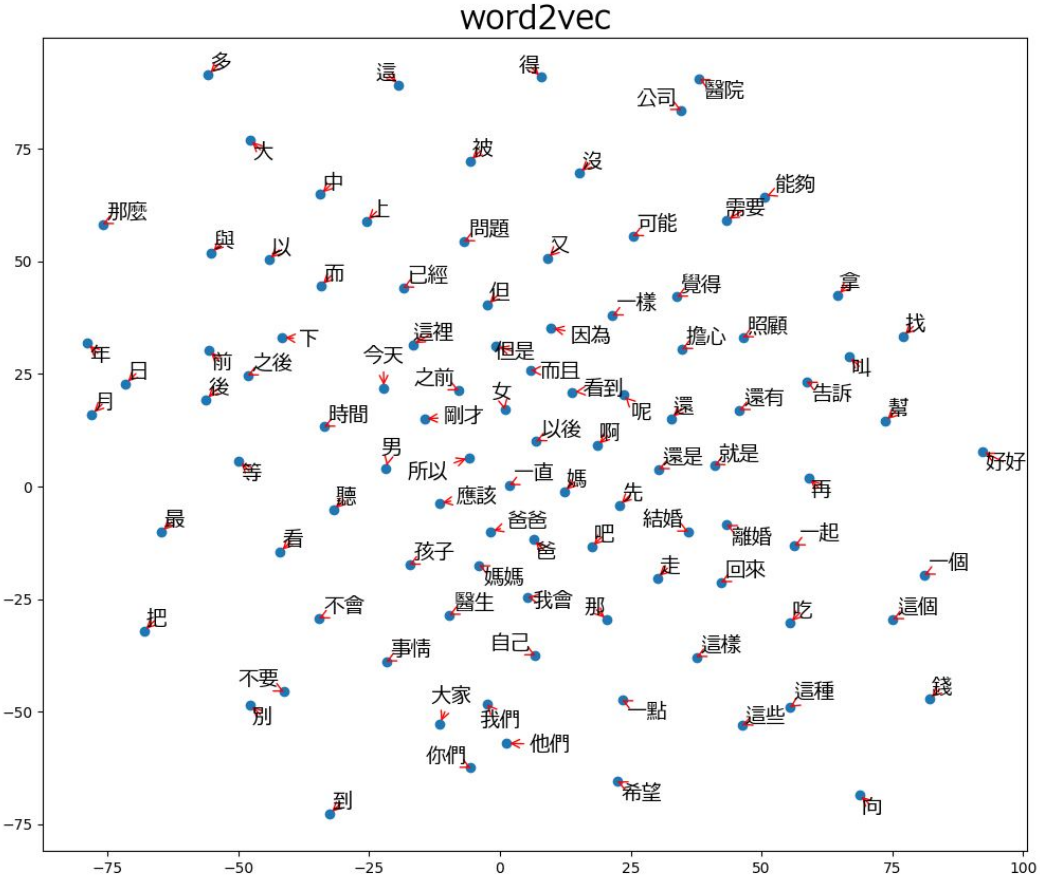
sentences：輸入的句子

size：輸出向量的維度

window：當前單詞 與 預測單詞 之間的最大距離。因為這次資料庫句子大都蠻短的，所以這個值改動好像沒啥差別(除非設成 1 或 2)。

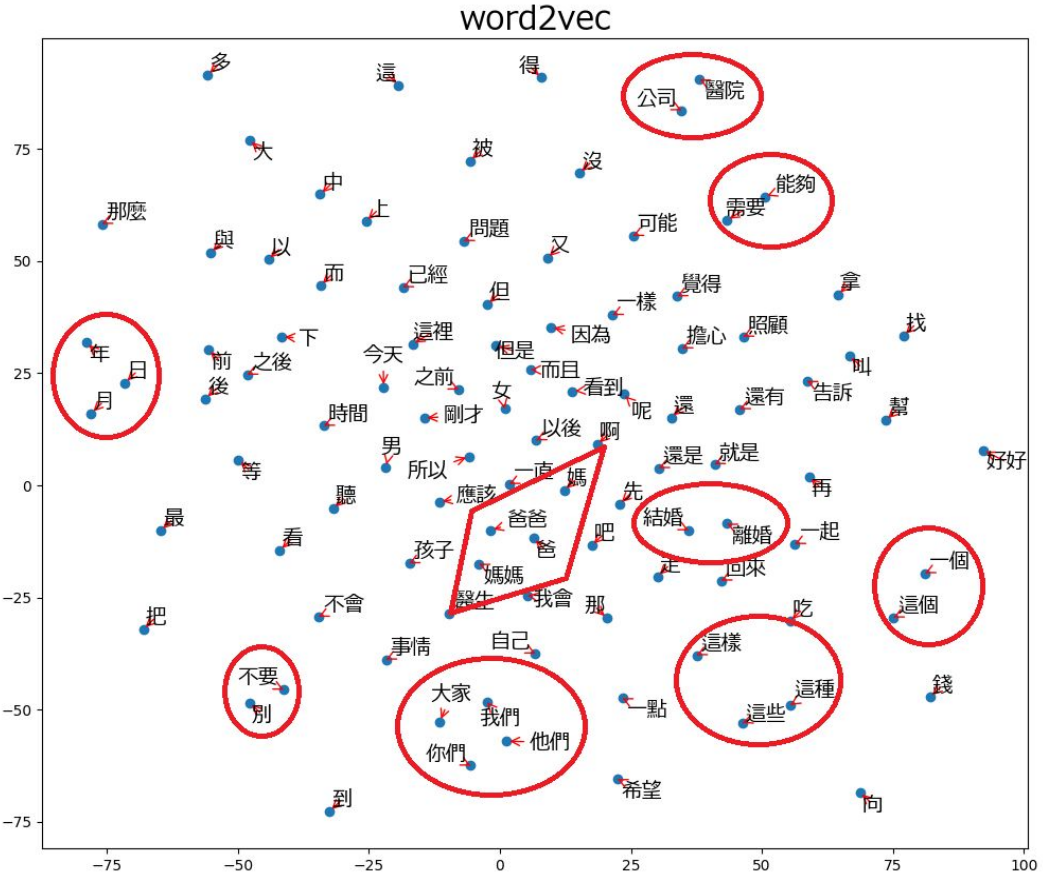
min_count：忽略總頻率低於這個值的所有單詞。

B.2. (.5%) 請在 Report 上放上你 visualization 的結果。 註：實作方式為先挑出要 visualization 單詞(出現3000次以上 再人工挑一些字) 再 t-SNE，所以分布較為均勻。



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

發現紅色框框內的詞都具有高度相關性(像是 '不要' 幾乎就等於 '別')，可能是因為 t-SNE 不是線性降維，所以無法觀察到很明顯的向量線性關係(像是 爸爸 = 媽媽 + 男 - 女)，此外還發現詞性相同者，分布也可能較接近。



C. Image clustering

(collaborator: NULL)

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)

方法一：

Step 1. 使用 sklearn.decomposition 的 PCA 降維至 300維

(PCA 內的參數 whiten=True，即所有 samples 間的同個 '點' 做標準化處理)

Step 2. 使用 sklearn.cluster 的 KMeans 分為兩群即完成。

Kaggle分數：Private Score：1.00000　Public Score：1.00000

討論：結果為全對，可見單純的 PCA 在降維上就有很好的效果。沒有 whiten 的話結果會直接爛掉，推測是因為每個像素點的變化程度都不一樣，像是邊邊的點可能一直都是 0 ；圖片中間的點可能變異就很大，因此標準化後就可以投影到比較好的空間上。

方法二：

Step 1. 使用 Convolutional Autoencoder 編碼至 256維

keras model 如下(dense_1 之輸出即為編碼結果)：

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 28, 28, 1)	0
conv2d_1 (Conv2D)	(None, 28, 28, 64)	640
conv2d_2 (Conv2D)	(None, 28, 28, 32)	18464
max_pooling2d_1 (MaxPooling2	(None, 14, 14, 32)	0
flatten_1 (Flatten)	(None, 6272)	0
dense_1 (Dense)	(None, 256)	1605888
reshape_1 (Reshape)	(None, 2, 2, 64)	0
up_sampling2d_1 (UpSampling2	(None, 4, 4, 64)	0
conv2d_3 (Conv2D)	(None, 4, 4, 32)	18464
up_sampling2d_2 (UpSampling2	(None, 8, 8, 32)	0
conv2d_4 (Conv2D)	(None, 8, 8, 64)	18496
up_sampling2d_3 (UpSampling2	(None, 16, 16, 64)	0
conv2d_5 (Conv2D)	(None, 14, 14, 128)	73856
up_sampling2d_4 (UpSampling2	(None, 28, 28, 128)	0
conv2d_6 (Conv2D)	(None, 28, 28, 1)	1153

Total params: 1,736,961
Trainable params: 1,736,961
Non-trainable params: 0

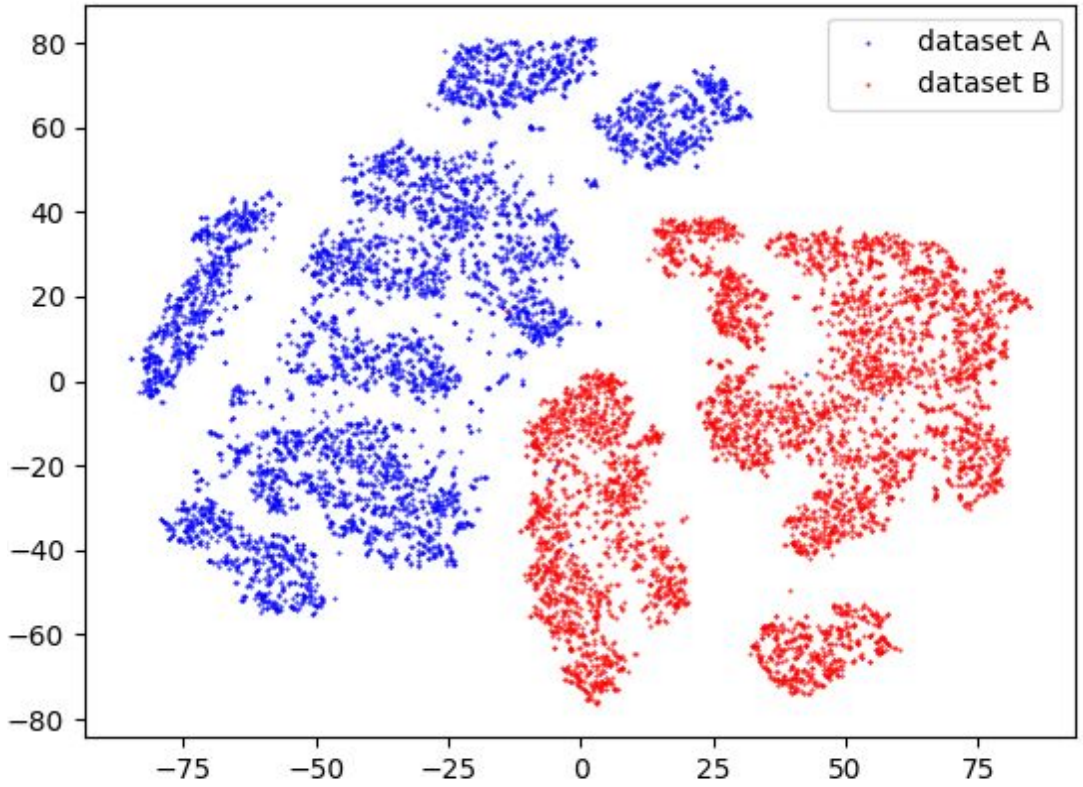
Step 2. 使用 sklearn.cluster 的 KMeans 分為兩群即完成。

Kaggle分數：Private Score：0.64328　Public Score：0.64403

討論：結果算略差，猜想是因為 CNN 會掉太多資訊，而本次的 dataset 之維度本來就不算高(以圖片來說)，資訊可能集中在某些像素點上，因此編碼後要再分群就變得較為困難。

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

使用C.1.的方法一，預測全對。下圖是先 PCA 再 t-SNE。



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。

下圖是原始的 784維 data 直接做 t-SNE，C.2.和 C.3.的共同特徵有： dataset B 又分成了兩個區塊、 dataset A 雖然較緊密，但是可發現其實有很明顯的 9~10 個小區塊、兩者都很難找到一條簡單的曲線，就將兩個 dataset 分開；自己預測的 label dataset A 切出來的子小塊較清楚，可見降維有其效果。