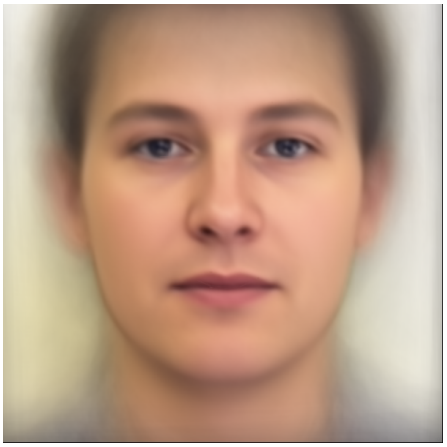
ML HW6 Report

學號：B04901138 系級： 電機三 姓名：張景程

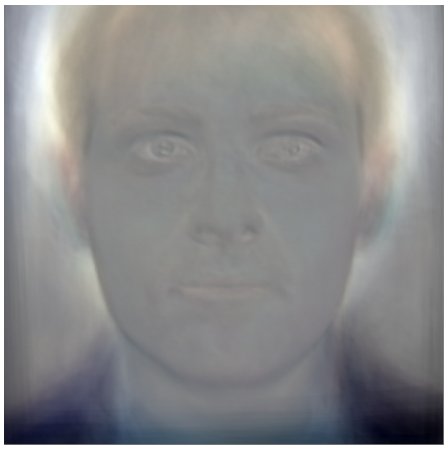
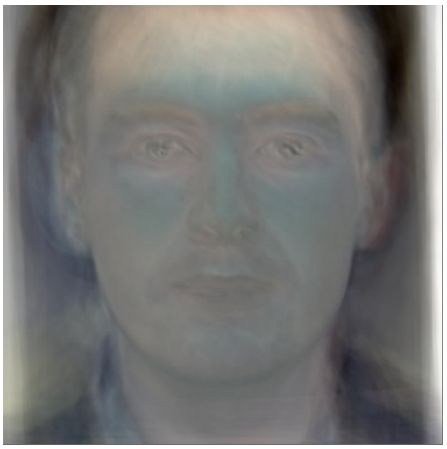
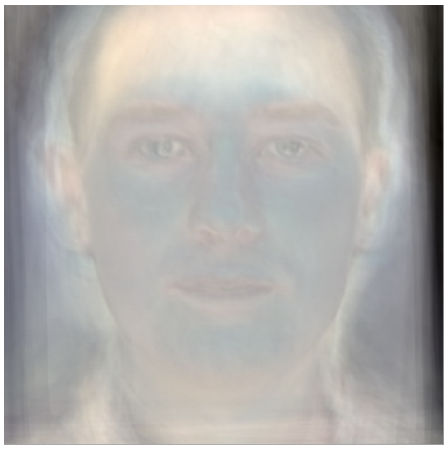
A. PCA of colored faces

(collaborator: B04901019 梁書哲)

A.1 (.5%) 請畫出所有臉的平均。

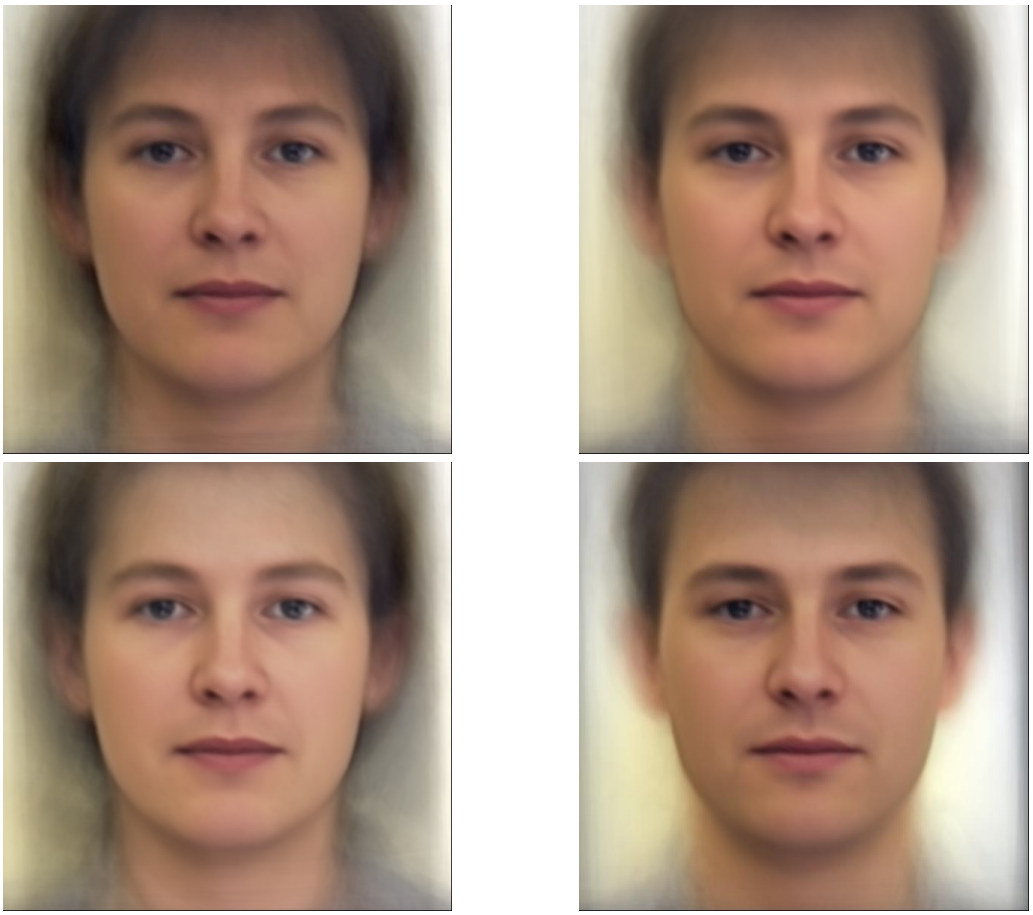


A.2 (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

前四個 eigenface 分別為左上、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。



我挑選了 0.jpg, 30.jpg, 40.jpg, 50.jpg 這四張 image，利用前四大 eigenfaces reconstruct 的結果分別為左上、右上、左下、右下。

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。

index	比重
1	4.1%
2	3.0%
3	2.4%
4	2.2%

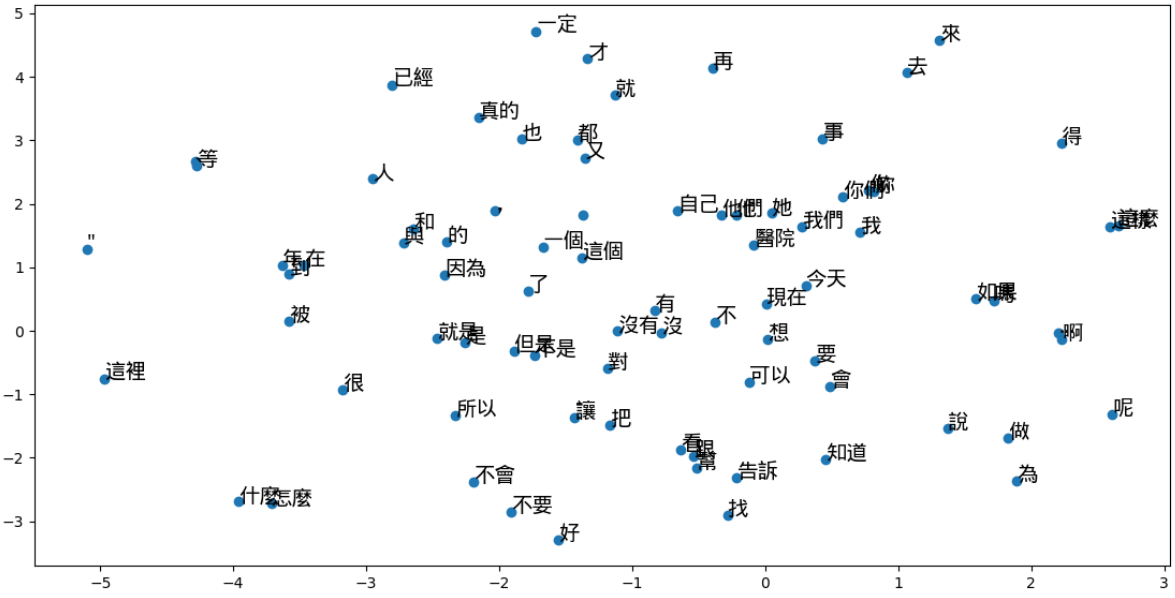
B. Visualization of Chinese word embedding

(collaborator:無)

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。

我使用了 gensim(3.2.0)的 Word2Vec，其參數 size 設為 64，min_count=5000。也就是將 label 和 nolabel 的句子中，出現超過五千次以上的單字才會保留起來做訓練，並將每個單字轉為長度 64 的 vector。

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。

圖中顯示的是 corpus 裡出現次數超過五千次的詞，所以都是日常生活中很常出現、使用的單字，從上圖可以觀察到：你/你們、我/我們、他/他們這三組有類似的關係，且你/妳、他/她這兩組也有類似的關聯性。

C. Image clustering (collaborator:無)

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)

1. 利用 deep autoencoder 來做降維：

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, 128)	8320
dense_5 (Dense)	(None, 256)	33024
dense_6 (Dense)	(None, 784)	201488

Total params: 484,944
Trainable params: 484,944
Non-trainable params: 0

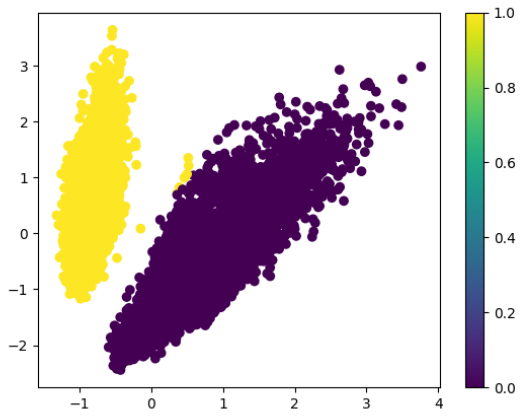
autoencoder model 架構如左圖，將 784 維的 input image 通過三層 unit 分別為 256,128,64 的 dense 後，encode 成維度 64 的 vector。
訓練時再將 encode 得到的 vector 通過 decoder 回原來 784 維度，並和原圖取 mse，所使用的 optimizer 為 Adam，訓練 60 個 epoch 後在 kaggle 上 public set 和 private set 得到的 F1 score 皆為 1.0000。

2. 直接做 PCA 降維 (此方法 collaborator : B04901153 劉維凱)
利用 sklearn 內建的 PCA，直接將原本 784 維度的 vector 降成 400，然後再做 Kmeans。

```
PCA(n_components=400, whiten=True, svd_solver = 'arpack',
        random_state=424).fit_transform(train)
```

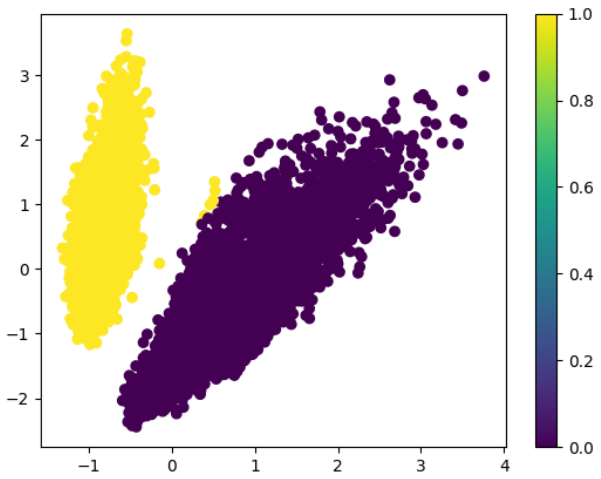
比較特別的是將參數 whiten 從 default=False 改設為 True，此方法可以大幅增加準確率，在 kaggle 上 public set 和 private set 得到的 F1 score 也為 1.0000。

C.2 (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。
這是根據上述 autoencoder 的 model predict 出來後再經過 PCA(n_components=2) 畫出來的圖：

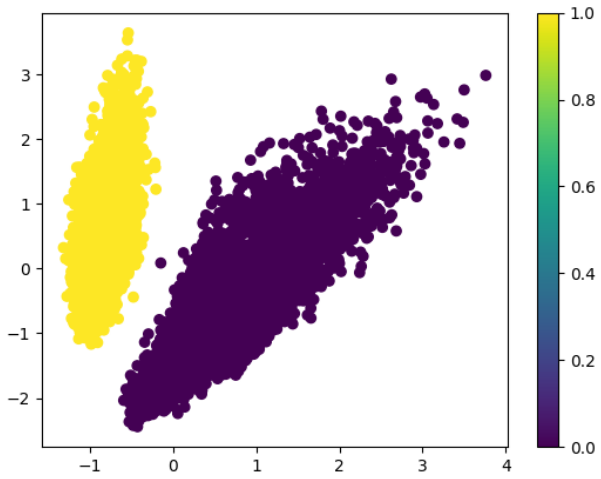


C.3 (.5%) visualization.npy 中前 5000 個 images 跟後 5000 個 images 來自不同 dataset。請根據這個資訊，在二維平面上視覺化 label 的分佈，接著比較和自己預測的 label 之間有何不同。

自己預測：



實際 label:



由兩張圖比較出來可發現預測正確率算是蠻高的，只有在邊界的少數幾個點 predict 錯誤而已，算是滿意的結果。