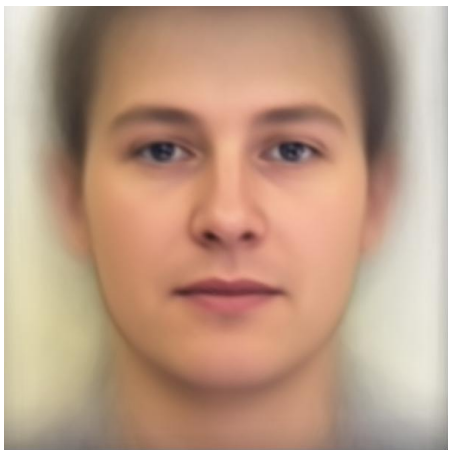
學號：B04705018 系級： 資管三 姓名：楊之郡

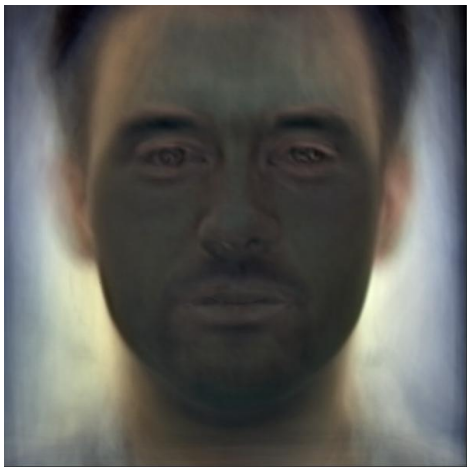
Collaborators: 盧慶原，林雨新

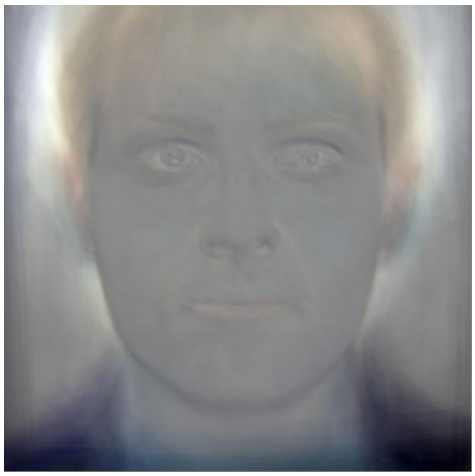
1. PCA of colored faces

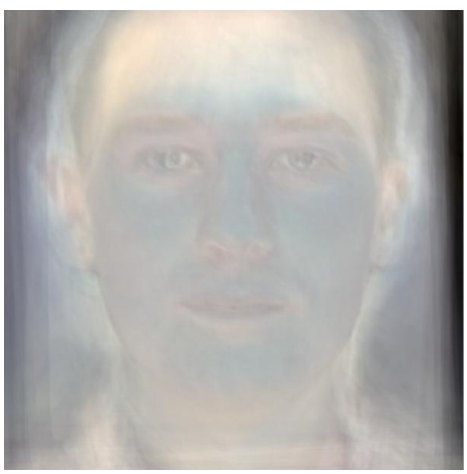
1. (.5%) 請畫出所有臉的平均。



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

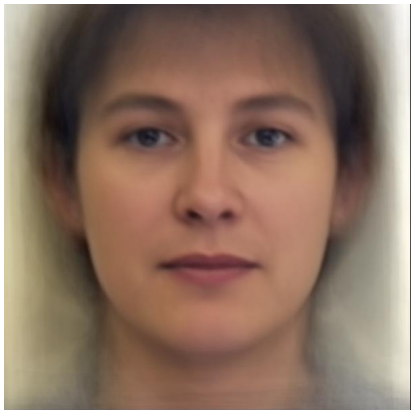




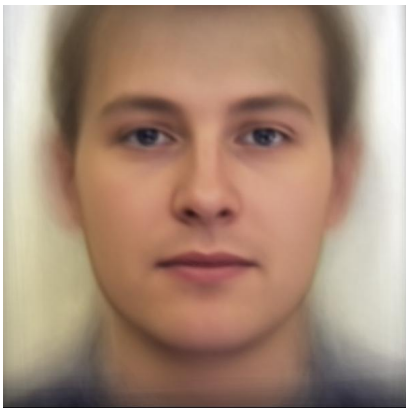


3. (.5%) 請從數據集中挑出任意四個圖片，並用前四大 Eigenfaces 進行 reconstruction，並畫出結果。

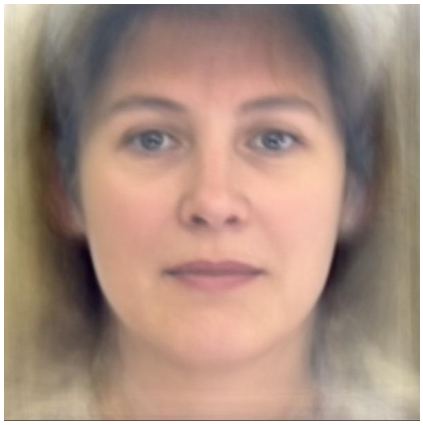
0.jpg:



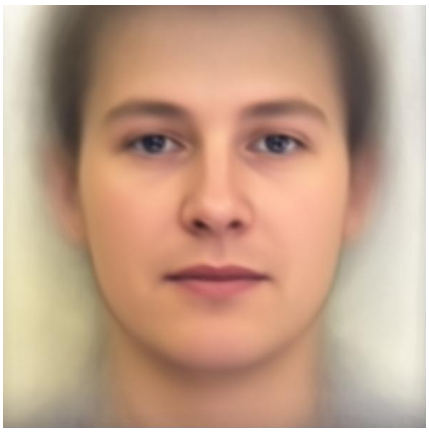
3.jpg



4.jpg



6.jpg



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。

w1 = 4.1,　　w2 = 3,　　w3 = 2.4,　　w4 = 2.2

2. Visualization of Chinese word embedding

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。

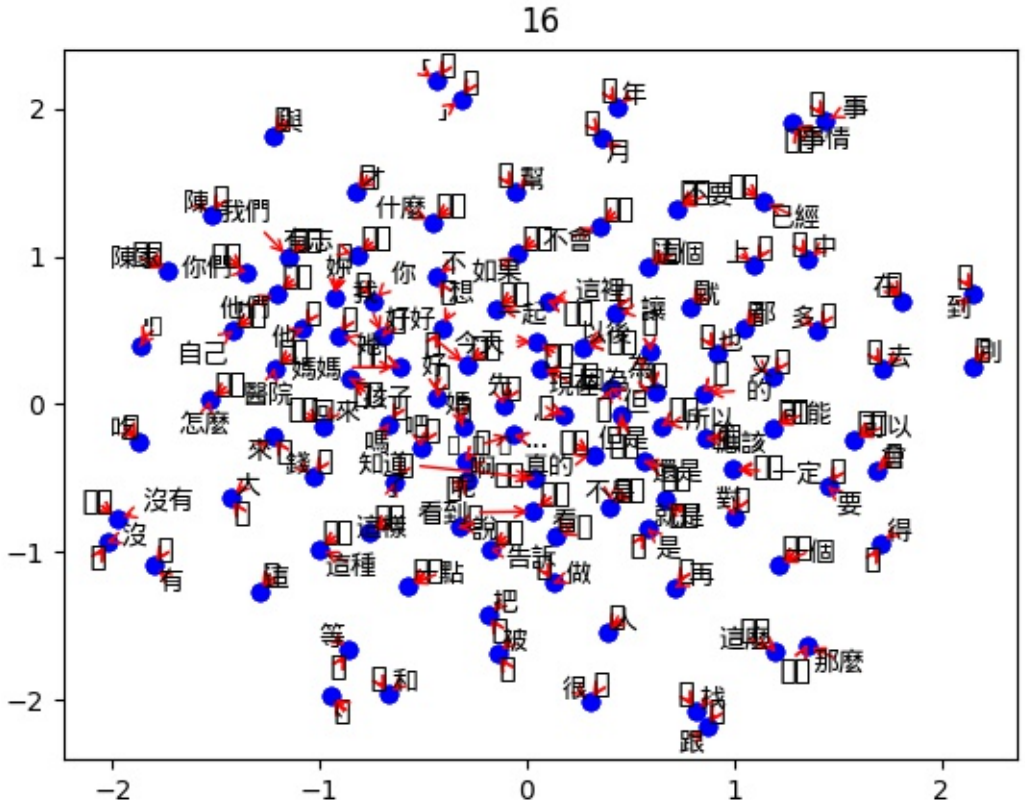
使用 gensim 的 Word2Vec

sentences: 放入使用 jieba 切好的句子

size:　最後訓練出來的 word vector 維度

min_count:　捨棄出現次數小於 min_count 的 word

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



data 有使用 sklearn 的 scale 做正規劃，讓 x 軸和 y 軸的 range 縮小，方便畫圖

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

語意比較相近的 word 距離比較接近

像是”有”跟”沒有”，”事””事情”

3.Image clustering

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)

	private	public
Convolutional autoencoder:	0.99790	0.99790

```
rdim = 64

#autoencoder model
input_img = Input(shape=(28,28,1,))
encoded = Conv2D(cnn1_nfilter,cnn1_size,padding='same')(input_img)
encoded = LeakyReLU()(encoded)
encoded = MaxPooling2D(pool_size=2,padding='same')(encoded)

encoded = Conv2D(cnn2_nfilter,cnn2_size,padding='same')(encoded)
encoded = LeakyReLU()(encoded)
encoded = MaxPooling2D(pool_size=2,padding='same')(encoded)
```

```
dse = Flatten()(encoded)

dse = Dense(units=1024)(dse)
dse = BatchNormalization()(dse)
dse = LeakyReLU()(dse)

dse = Dense(units=512)(dse)
dse = BatchNormalization()(dse)
dse = LeakyReLU()(dse)

#bottle neck
btn = Dense(units=rdim)(dse)

dse2 = Dense(units=cnn2_nfilter*7*7)(btn)
dse2 = BatchNormalization()(dse2)
dse2 = LeakyReLU()(dse2)

rs = Reshape((7,7,cnn2_nfilter))(dse2)
```

```
decoded = Conv2D(cnn2_nfilter,cnn2_size,padding='same')(rs)
decoded = LeakyReLU()(decoded)
# decoded = Activation('relu')(decoded)

decoded = UpSampling2D(size=(2,2))(decoded)

decoded = Conv2D(cnn1_nfilter,cnn1_size,padding='same')(decoded)
decoded = LeakyReLU()(decoded)
# decoded = Activation('relu')(decoded)

decoded = UpSampling2D(size=(2,2))(decoded)

decoded = Conv2D(1,2,padding='same')(decoded)
```

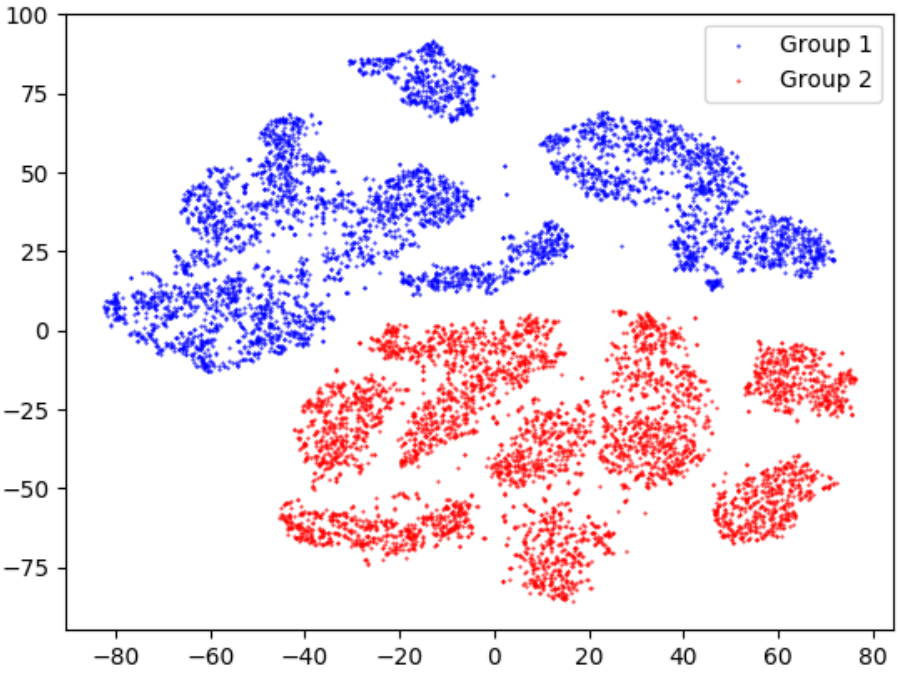
降至 64 維（rdim = 64）

Deep autoencoder:	0.74974	0.75323

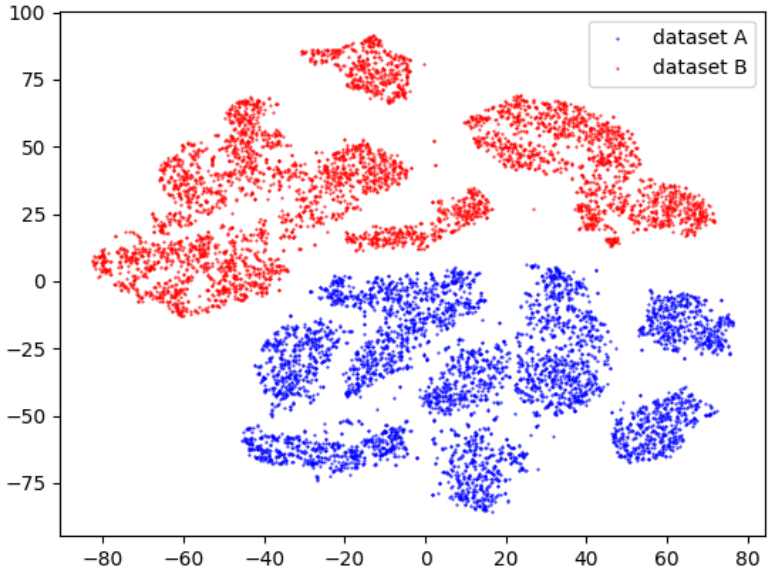
```

input_img = Input((784,))
dse = Dense(units=392,activation='relu')(input_img)

dse = Dense(units=196,activation='relu')(dse)

#bottle neck
btn = Dense(units=rdim)(dse)

ddse = Dense(units=196,activation='relu')(btn)

ddse = Dense(units=392,activation='relu')(ddse)

decoded = Dense(units=784)(ddse)
```

我做出來的結果 CAE 比 DNN 好，可能是因為 CNN 有擷取圖片特徵的特性，因此比較能分辨兩種不同的 cluster，但是和同學討論後發現他們用同樣的架構去 train DNN，反而 DNN 效果比較好，甚至可以做到 1，不過我並沒有很仔細去調 DNN 的參數，可能是因為這樣所以我的 DNN 效果不太好

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



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。



兩者的分類看起來非常的接近，只有在邊界的部份稍微一點點不一樣，應該是因為我的 encoder 有做到 0.99，算是蠻強大的 model，所以做出來的分類效果不錯