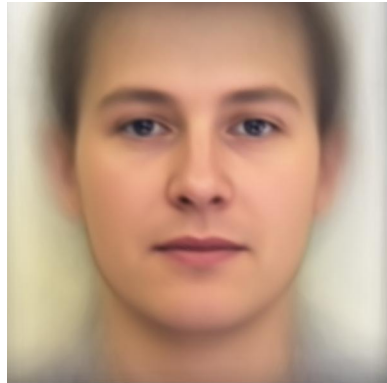


## Machine Learning HW7 Report

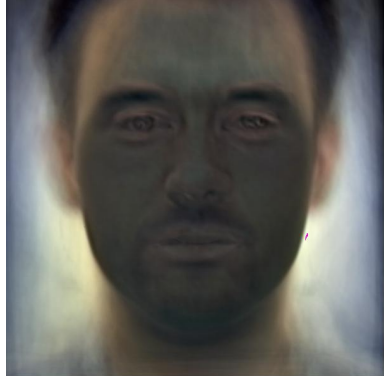
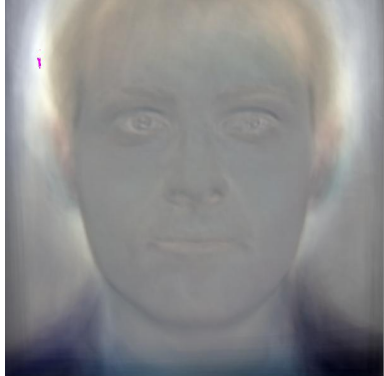
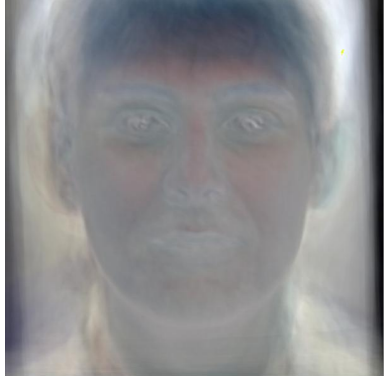
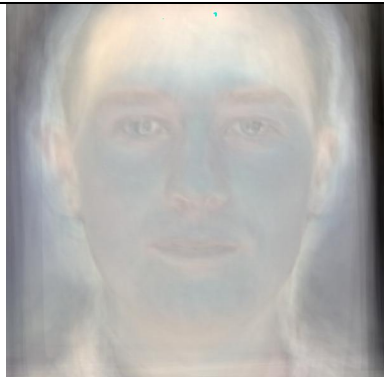
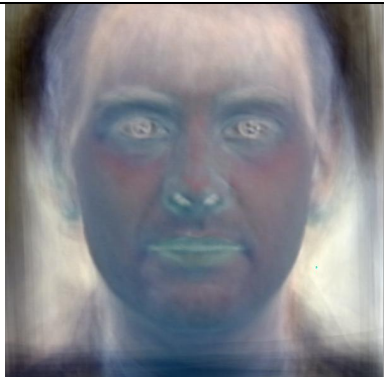
學號：R07942115 系級：電信碩一 姓名：謝硯澤

### 1. PCA of color faces:

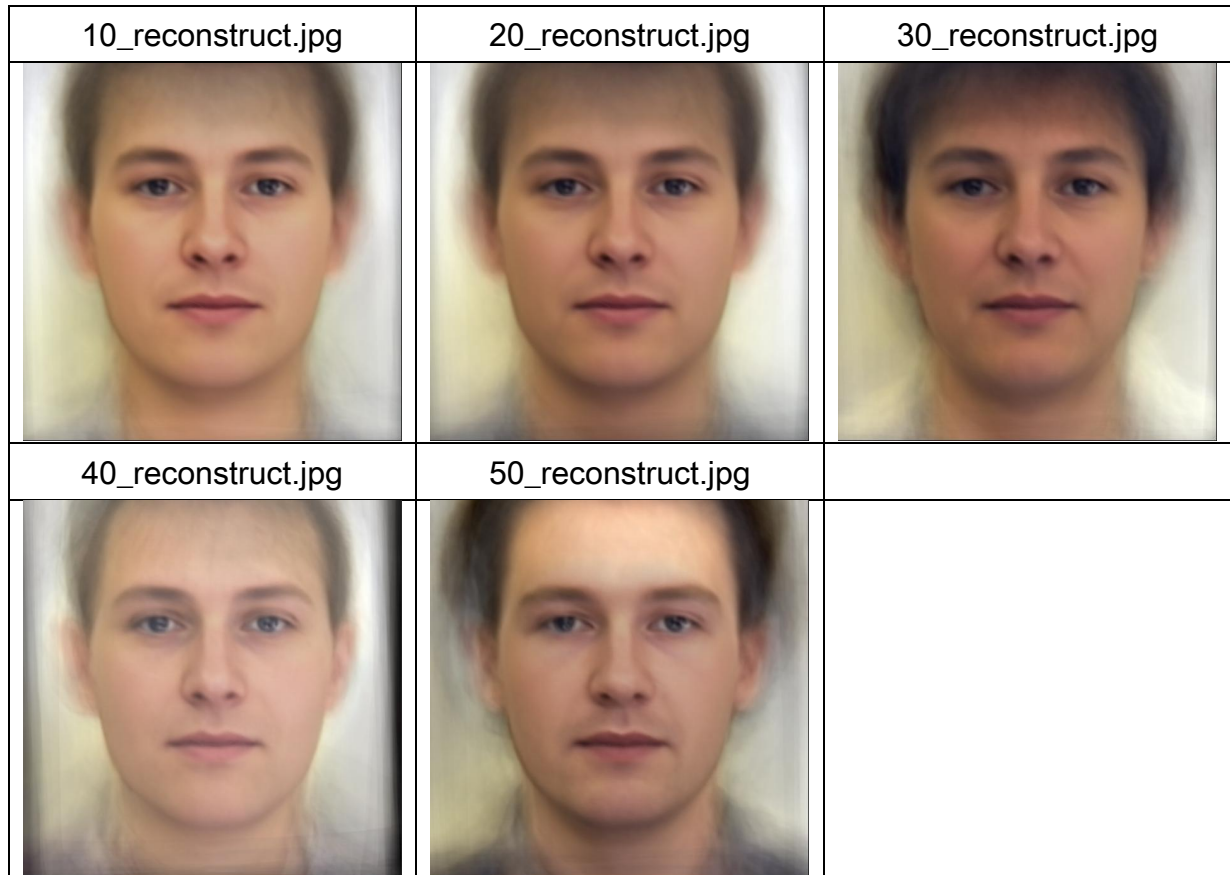
a. 請畫出所有臉的平均。



b. 請畫出前五個 Eigenfaces，也就是對應到前五大 Eigenvalues 的 Eigenvectors。

Eigenfaces1	Eigenfaces2	Eigenfaces3
		
Eigenfaces4	Eigenfaces5	
		

- c. 請從數據集中挑出任意五張圖片，並用前五大 Eigenfaces 進行 reconstruction，並畫出結果。



- d. 請寫出前五大 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。

Eigenface1	4.1%
Eigenface2	2.9%
Eigenface3	2.4%
Eigenface4	2.2%
Eigenface5	2.1%

## 2. Image clustering:

- a. 請實作兩種不同的方法，並比較其結果(reconstruction loss, accuracy)。

(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)

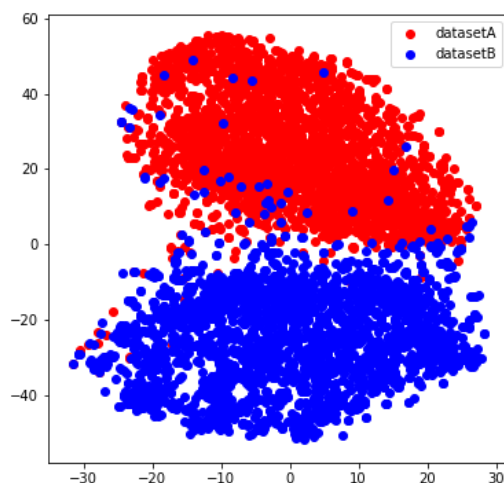
Reconstruction loss (per image)計算方法：(1)求出原本 data 與經過 autoencoder 後的 reconstruct data 之間的誤差(2)再取平方(3)全部相加(4)最後除以 data 量。

	Autoencoder + TSNE + Kmeans	Autoencoder + PCA + Kmeans
Reconstruction loss	10.715	10.715
Kaggle public score	0.95894	0.94372
Kaggle private score	0.95910	0.94353

- b. 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。

(用 PCA, t-SNE 等工具把你抽出來的 feature 投影到二維，或簡單的取前兩維 2 的 feature)

其中 visualization.npy 中前 2500 個 images 來自 dataset A，後 2500 個 images 來自 dataset B，比較和自己預測的 label 之間有何不同。



跟我預測的結果蠻接近的，我的作法就是先透過 **encoder** 將維到 32 維，再用 **TSNE** 降維到 2 維，最後做 **Kmeans** 分群。

c. 請介紹你的 **model 架構**(encoder, decoder, loss function...)，並選出任意 32 張圖片，比較原圖片以及用 decoder reconstruct 的結果。

Model Architecture :

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 1024)	0
dense_1 (Dense)	(None, 384)	393600
dense_2 (Dense)	(None, 32)	12320
dense_3 (Dense)	(None, 384)	12672
dense_4 (Dense)	(None, 1024)	394240
Total params: 812,832		
Trainable params: 812,832		
Non-trainable params: 0		

Encoder

Decoder

Loss function	mse
Optimizer	adam
Epoch	1000
Batch size	256
TSNE	n_components=2, n_iter=3000, perplexity=50
K-Means	n_clusters=2

我的 **model 架構**如上表所述，先經過 **Encoder** 後，再用 **TSNE** 降到 2 維，最後使用 **kmeans** 分成兩群。

32 張原圖



32 張還原圖



我那時候在寫作業時，讀取影像的時候是以灰階影像做讀取，所以上圖的比較圖也是用灰階影像來呈現。從上圖的比較結果可以發現整體其實還原的沒有很好，但人臉的影像相對還原的好很多，可以大致看得清楚臉部及五官的輪廓，代表 **encoder** 可能只有做到抽取出人臉的資訊？！這可能也是為什麼最後仍有辦法在經過 **encoder** 後區分兩個 **dataset** 的原因。