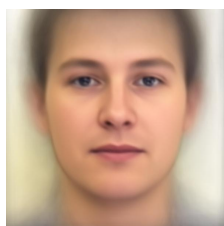


## Machine Learning HW7 Report

學號：B05705053 系級：資管三 姓名：蔡涵如

### 1. PCA of color faces:

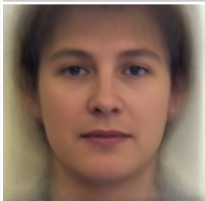
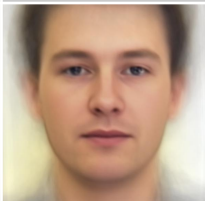
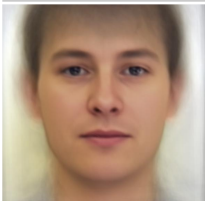
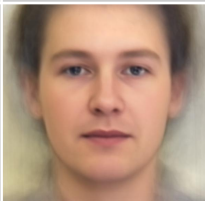
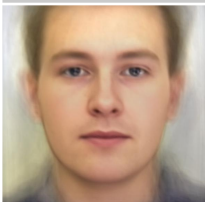
a. 請畫出所有臉的平均。



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

排名	1	2	3	4	5
Eigen Face					

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。

圖片編號	1	10	22	37	72
Eigen Face					

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。

排名	1	2	3	4	5
比重	4.1%	2.9%	2.4%	2.2%	2.1%

## 2. Image clustering:

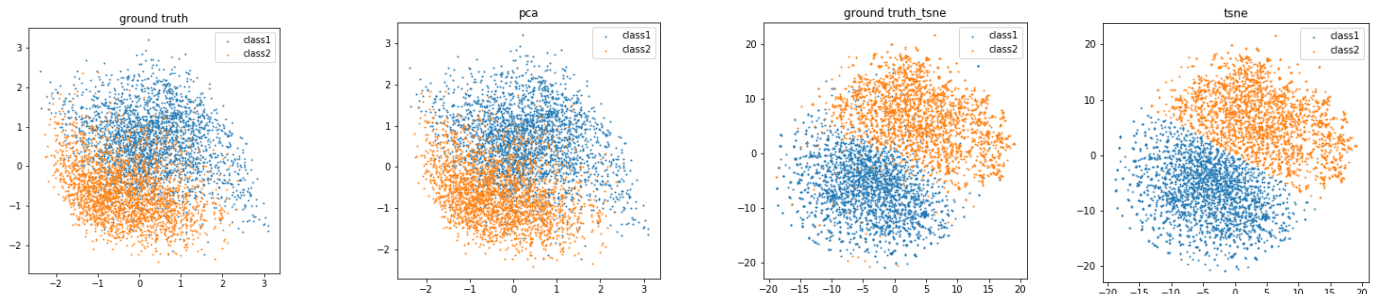
- a. 請實作兩種不同的方法，並比較其結果(reconstruction loss, accuracy)。  
(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)

實作兩種的方法皆是用 auto encoder 將code 降到 3,072 維 方式，再用 pca降到1100，嘗試過的許多模型最好的方法是用降到越多code越好，但是有一點疑問是在這個task裡面降到越多為會越好，相較起來gaussian mixture得方法比pca快而且 performace最好。

	Private leaderboard	Public leaderboard	Reconstruction loss(mse)
auto-encoder 3072 +pca + gaussian mixture	0.98743	0.98761	0.0049
auto-encoder 3072 +pca 1100 +k-means	0.98728	0.98754	0.0049

- 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，準確率達到0.9862，雖然tsne的視覺化效果比較好但是準確率只有0.96，相較之下蛤是PCA的降維方式比較適合kmeans的分類，但是覺化的部分tsne的表現較好。



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。

autoencoder 架構 將model降到3,072，而autoencoder，降維得效果可以發現到其實轉得效果還滿不錯得，基本上reconstruct的能力很好，會有一點模糊的效果：以下圖所示，loss function 採用MSE。

Layer (type)	Output Shape	Param #
input_layer (Conv2D)	(None, 32, 32, 256)	7168
max_pooling2d_3 (MaxPooling2D)	(None, 16, 16, 256)	0
batch_normalization_7 (Batch Normalization)	(None, 16, 16, 256)	1024
conv2d_7 (Conv2D)	(None, 16, 16, 48)	110640
code_layer (MaxPooling2D)	(None, 8, 8, 48)	0
batch_normalization_8 (Batch Normalization)	(None, 8, 8, 48)	192
conv2d_8 (Conv2D)	(None, 8, 8, 48)	20784
up_sampling2d_4 (UpSampling2D)	(None, 16, 16, 48)	0
batch_normalization_9 (Batch Normalization)	(None, 16, 16, 48)	192
conv2d_9 (Conv2D)	(None, 16, 16, 256)	110848
up_sampling2d_5 (UpSampling2D)	(None, 32, 32, 256)	0
batch_normalization_10 (Batch Normalization)	(None, 32, 32, 256)	1024
conv2d_10 (Conv2D)	(None, 32, 32, 3)	771
Total params: 252,643		
Trainable params: 251,427		
Non-trainable params: 1,216		

