

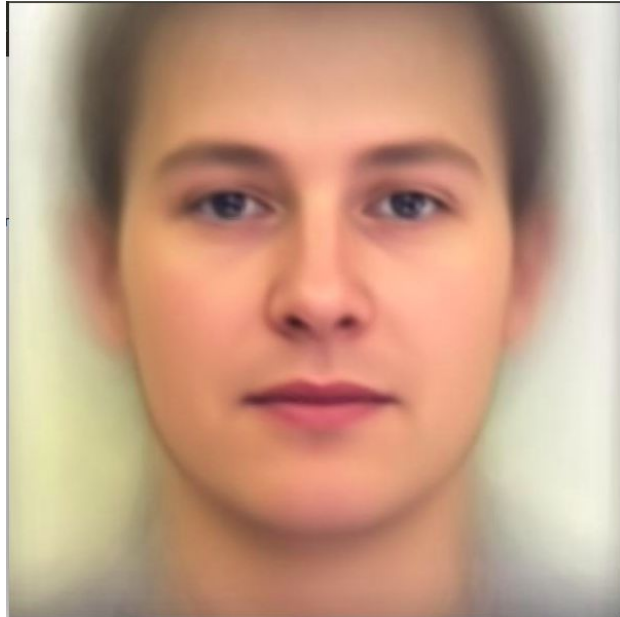
## Machine Learning HW7 Report

學號：b04507025 系級：電機四

姓名：韓秉勳

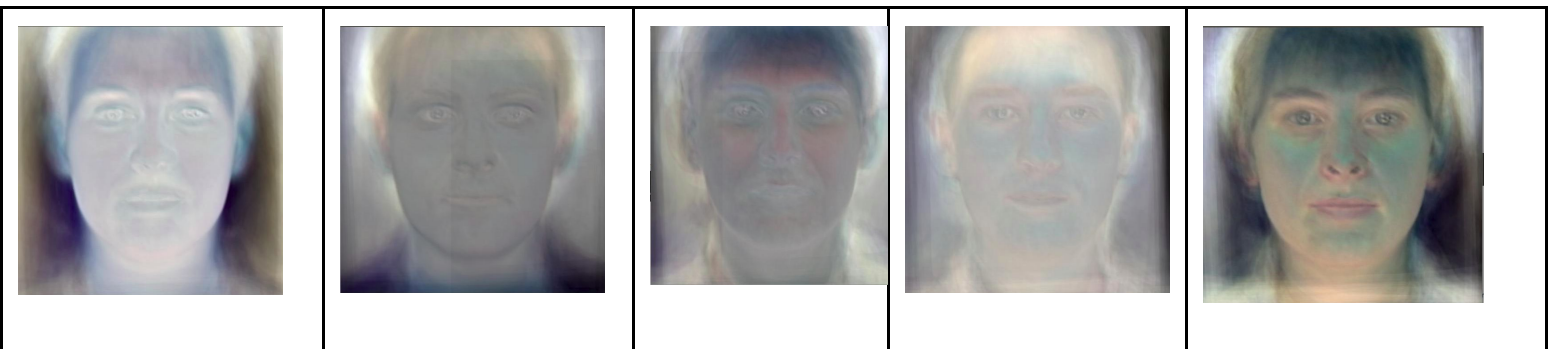
### 1. PCA of color faces:

a. 請畫出所有臉的平均。



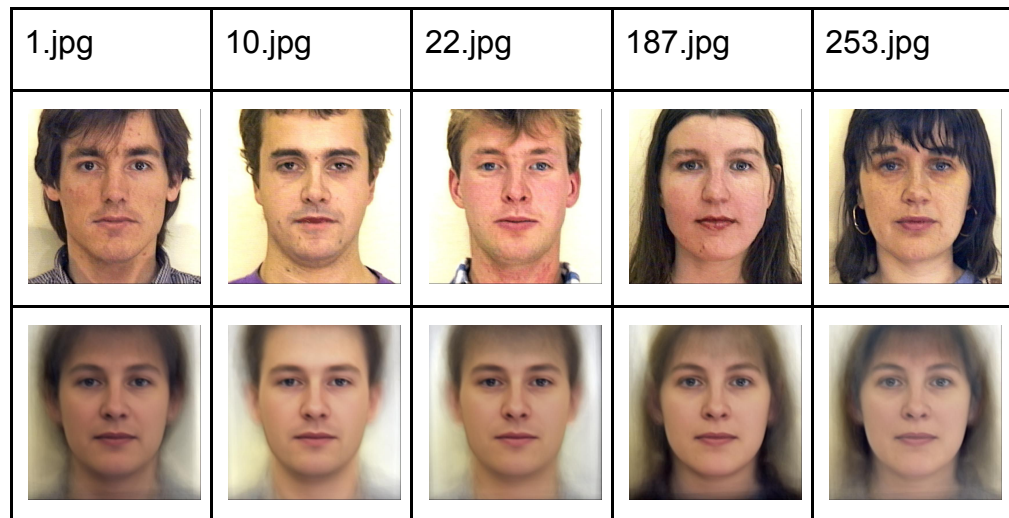
b. 請畫出前五個 Eigenfaces, 也就是對應到前五大 Eigenvalues 的 Eigenvectors。

左到右依大到小排列：



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。

上下兩列分別為原圖與reconstruct :



可看出前五維的eigenvector主要針對臉，頭髮等特徵，除非如長髮等明顯不同特徵才可能繪出

- 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。

eigenface 0 :4.1%,  
eigenface 1 : 2.9%,  
eigenface 2 : 2.4%,  
eigenface 3 : 2.2%,  
eigenface 4 : 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)

以下皆以CNN autoencoder為例，但搭配pca或tsne的不同降維方法

PCA: 使用whiten與max\_iter=4000, n\_component = 700以增加cluster區別程度

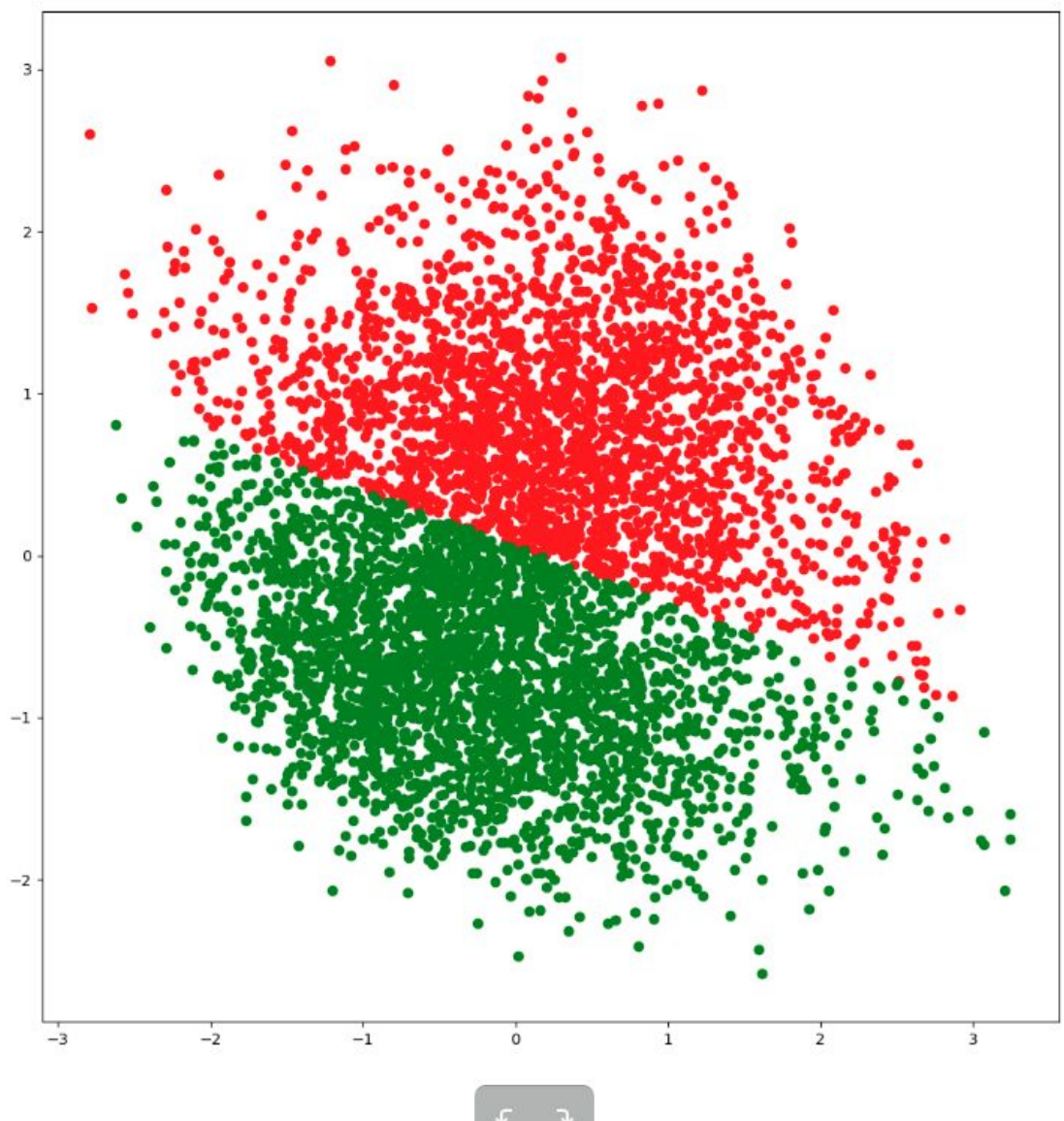
tsne: 使用max\_iter =4000, n\_component = 2

看結果可知tsne的準確率低很多，且訓練時間須半個小時，十分冗長。

	reconstruction loss(MSE)	kaggle accuracy
CNN auto encoder+PCA	0.0034	0.97493
CNN auto encoder+tsne	0.0034	0.89726

- 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。

以下為我kaggle上用的CNN autoencoder+PCA 2維的分類情況, (紅色為1綠色為0), 與正確label相比差了大概30%左右, 可能是因為我的encoder 出來 1024維降到pca2維時, 因維度太少導致許多feature並沒辦法拉開距離。由圖中可看出中間有明顯分界, 但因並沒有完全拉開, 在紅綠交界處可能會辨識錯誤。



- 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。

我使用了CNN 的autoencoder, 在encoder 前經過3層conv2d與2層maxpooling, channel 為3->32->32->16->16, 再經同樣的layer 進行decode。reconstruct loss 使用mse, 並可到0.0034。reconstruct image 如下圖所示, 可以見到照片稍微模糊, 但仍可以分辨出照片輪廓。

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 16)	4624
conv2d_3 (Conv2D)	(None, 16, 16, 16)	2320
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 16)	0
conv2d_4 (Conv2D)	(None, 8, 8, 16)	2320
conv2d_5 (Conv2D)	(None, 8, 8, 16)	2320
up_sampling2d_1 (UpSampling2D)	(None, 16, 16, 16)	0
conv2d_6 (Conv2D)	(None, 16, 16, 32)	4640
up_sampling2d_2 (UpSampling2D)	(None, 32, 32, 32)	0
conv2d_7 (Conv2D)	(None, 32, 32, 3)	867
Total params: 17,987		
Trainable params: 17,987		
Non-trainable params: 0		



