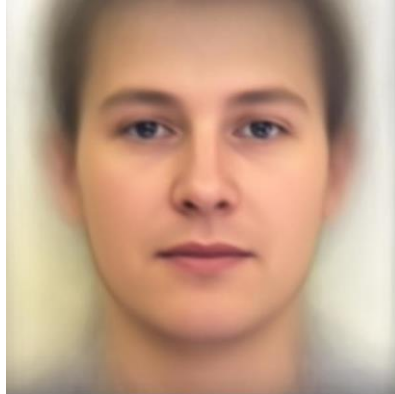


Machine Learning HW7 Report


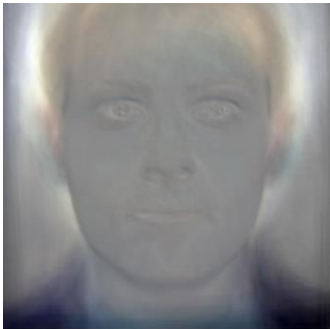



學號：R07943107 系級：電子所碩一 姓名：徐晨皓

1. PCA of color faces:



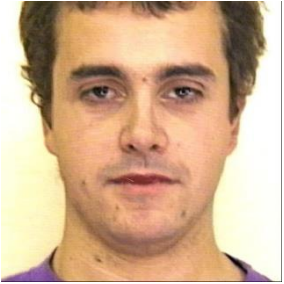


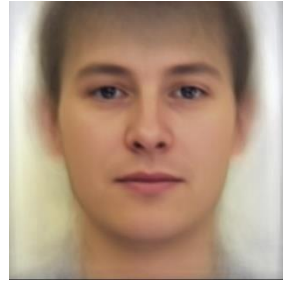

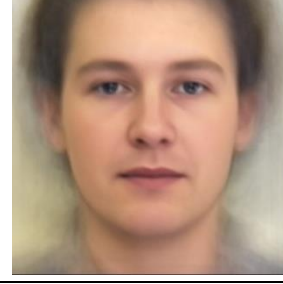


a. 請畫出所有臉的平均。



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

The 1st eigenface	The 2nd eigenface	The 3rd eigenface
		
The 4th eigenface	The 5th eigenface	
		

- 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。

Name	Original image		Reconstructed image	
1.jpg				
10.jpg				
22.jpg				
37.jpg				
72.jpg				

- 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。

1st eigenface	2nd eigenface	3rd eigenface	4th eigenface	5th eigenface
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)

- 方法一

先使用自己的 autoencoder (在問題 2c 有詳細介紹)降維，再使用 sklearn 的 PCA(whiten=True) 進一步降維，最後使用 sklearn 的 `KMeans(init='k-means++', n_clusters=2, max_iter=2000)`進行分群。

Autoencoder 的 reconstruction loss 為 0.00176。Private score 為 0.97935，public score 為 0.97954。

- 方法二

先使用自己的 autoencoder (在問題 2c 有詳細介紹)降維，再使用 sklearn 的 PCA(whiten=True) 進一步降維，最後使用 sklearn 的 `Birch(branching_factor=50, n_clusters=2)`進行分群。

Autoencoder 的 reconstruction loss 為 0.00176。Private score 為 0.85912，public score 為 0.85867。

- 結果比較

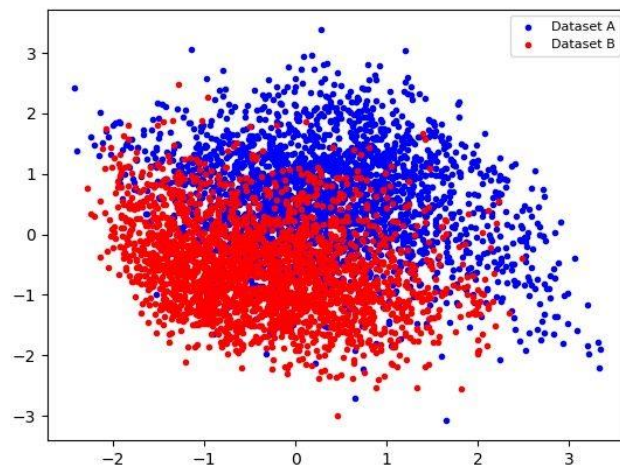
方法一使用 KMeans 的分群方法實驗上比方法二使用的 Birch 還要好。從實驗來看，branching_factor 的增加，能使方法二準確率上升，但 branching_factor 太大時，會造成記憶體不足的問題，故方法二只將 branching_factor 設為 50。

- 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。

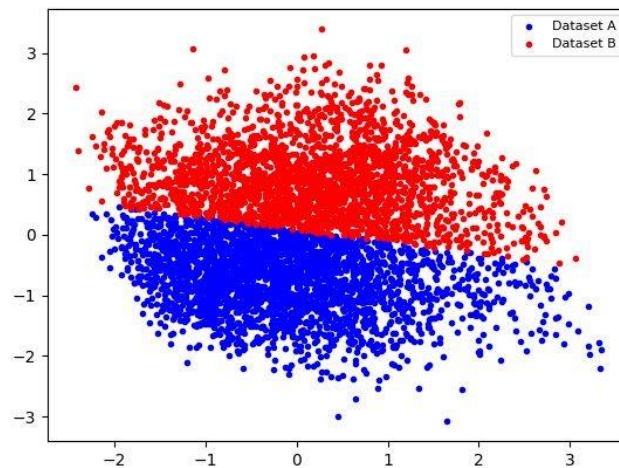
- 方法

直接使用 sklearn 的 PCA 將 data 降至二維，再使用 sklearn 的 KMeans 進行分群。由於直接降至二維因此 label 正確率只有 80%左右。

- 正確 label 的結果(圖一)



- 自己預測的 label 的結果(圖二)



- 觀察與討論

我們可以發現圖二有非常明顯的界線在兩群之間，這是因為 KMeans 會偏向於尋找 nearest neighbors，所以不會有混雜的情況。圖一為正確 labels 的結果，我們可以發現有很嚴重交雜的情況，這是因為我們將原 data 降至二維，因此損失很多資訊，導致 KMeans 無法很好的區分開這兩種 labels。

- 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。

● 模型架構

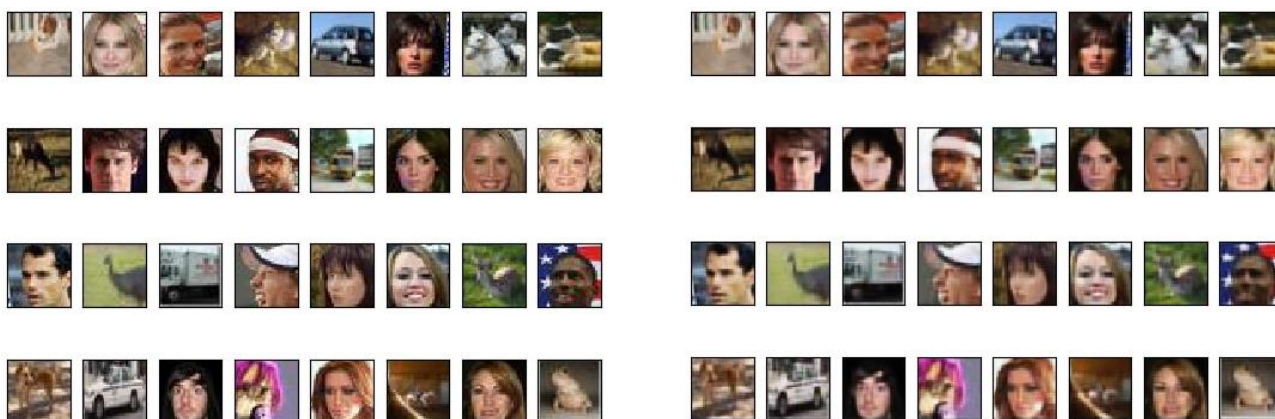
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	1792
max_pooling2d_1 (MaxPooling2)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 32)	18464
max_pooling2d_2 (MaxPooling2)	(None, 8, 8, 32)	0
conv2d_3 (Conv2D)	(None, 8, 8, 32)	9248
up_sampling2d_1 (UpSampling2)	(None, 16, 16, 32)	0
conv2d_4 (Conv2D)	(None, 16, 16, 64)	18496
up_sampling2d_2 (UpSampling2)	(None, 32, 32, 64)	0
conv2d_5 (Conv2D)	(None, 32, 32, 3)	1731
Total params: 49,731		
Trainable params: 49,731		
Non-trainable params: 0		

上圖為本次作業 autoencoder 的架構。

- I. 在 encoder 部分，使用兩層 convolutional layers 及兩層 maxpooling layers 交錯而成。
- II. 在 decoder 部分，使用三層 convolutional layers 及兩層 upsampling layers 交錯而成。
- III. Loss function 使用 mean squared error (mse)。
- IV. Optimizer 使用 adam。

以自己實作的 encoder 降維後，再以 sklearn 的 PCA(whiten=True)再進行降維。最後使用 sklearn 的 KMeans 將降維後的資料進行分群。

● 原圖與重建後結果



左圖為原圖，右圖為重建後圖片。可以發現右圖較左圖模糊一些。