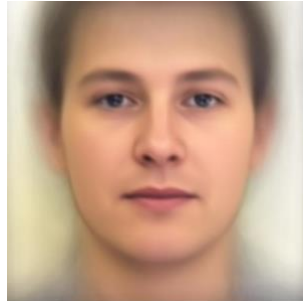


Machine Learning HW7 Report

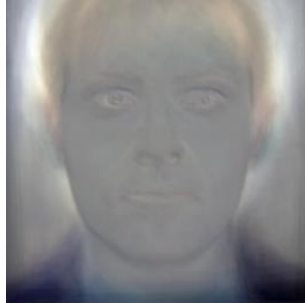
學號：B05901003 系級：電機二 姓名：徐敏倩

1. PCA of color faces:

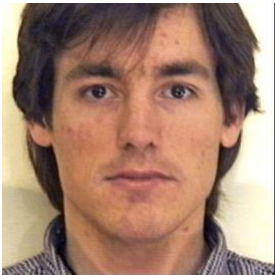







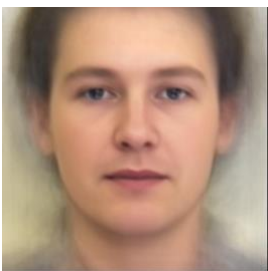

a. 請畫出所有臉的平均。



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

1st Eigenface	2nd Eigenface	3rd Eigenface
		
4th Eigenface	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。

	1.jpg	10.jpg	22.jpg
原圖			
reconstruction			
	37.jpg	72.jpg	
原圖			
reconstruction			

- 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。

# of eigenfaces	1 st	2 nd	3 rd	4 th	5 th
proportion (%)	4.1	2.9	2.4	2.2	2.1

2. Image clustering:

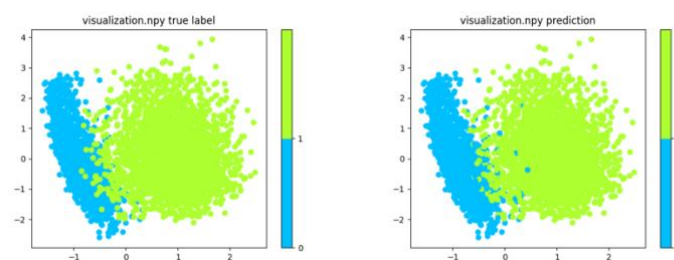
- a. 請實作兩種不同的方法，並比較其結果(reconstruction loss, accuracy)。(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)
- i. 降維方法：實作 autoencoder 將 32x32x3 的圖片降為 4x4x128
cluster 分法：以 KMeans 的方式，設 random_state 為 1346，將資料分為兩類
- ii. 降維方法：以 autoencoder 實作先將 32x32x3 的圖片降為 4x4x128，並以 flatten 的方式將圖轉為 2048 維，再用 PCA 的方式將其降為 1442 維，且設 random_state 為 1346。
cluster 方法：以 KMeans 的方式，設 random_state 為 1346，將資料分為兩類

	reconstruction loss	public score	private score
i.	0.5845	0.91430	0.91375
ii.	0.5845	0.98547	0.98552

(reconstruction loss 為利用 autoencoder 將圖片 encode 後 decode 的圖片和原圖比較後的 loss，以 binary cross entropy 的方式計算)

上述兩種方法 autoencoder 的模型和 cluster 的方法相同，唯一的差異在於是否再加一層的 PCA 後才進行 cluster，由結果可知，相較於搭配兩種方式降維，只使用 autoencoder 的 accuracy 較低（同樣的若只使用 PCA 降維，也只能得到約 0.95 的準確率）。因此，結合兩種方式降維能得到最好的效果。

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



(圖片的數據先經過 autoencoder 降維後再經過 PCA，x 軸為 PCA 所抽出的第一個 feature，y 軸為 PCA 所抽出的第二個 feature)

比較上面兩張圖得知，對於 `visualization.npy` 的圖檔而言，兩個 `feature` 確實能將圖片大略得分為兩個部分（左邊的棒狀和右邊的圓形），但同時也發現其實中間有重疊的部分，而該區域的分類準確率較低。因此，僅靠其中的兩維是無法將圖片準確的分類。

- 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。

`Encoder` 的部分，總共三層的 `Conv2D + LeakyReLU + MaxPooling2D`，`filter` 數分別為 32, 64, 128，`kernel_size` 皆用 (3, 3)，並以 `alpha` 為 0.02 的 `LeakyReLU` 作為其 `activation function`，和 `pool_size` 為 (2, 2) 的 `MaxPooling2D`。因此，再經過 `encoder` 後，每一張 32x32x3 的圖片都會被轉為 4x4x128 的資料。

`Decoder` 的部分，總共三層的 `Conv2D + LeakyReLU + UpSampling2D`，再加上最後一層 `Conv2D`。前三層 `Conv2D` 的 `filter` 數分別為 128, 64, 32，`kernel_size` 為 (3, 3)，並以 `alpha` 為 0.02 的 `LeakyReLU` 作為其 `activation function`，和 `size` 為 (2, 2) 的 `UpSampling2D`。而最後一層的 `Conv2D`，則是使用 `filter = 3`，`kernel_size = (3, 3)`，並以 'sigmoid' 作為其 `activation function`。

`Loss function` 使用 'binary_crossentropy'，`optimizer` 使用 'adam'，`data` 的部分有作 `data normalization` 和 `data augmentation`，總共 train 50 個 `epochs`。最後的 `reconstruction loss` 為 0.5845。

原圖	
Decoder reconstruct	