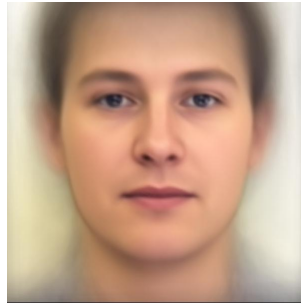


# Machine Learning HW7 Report

學號： B06902049 系級： 資工二 姓名： 林首志

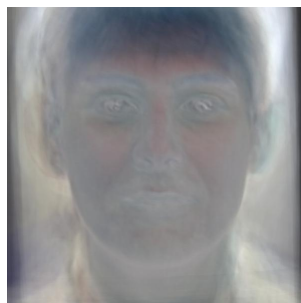
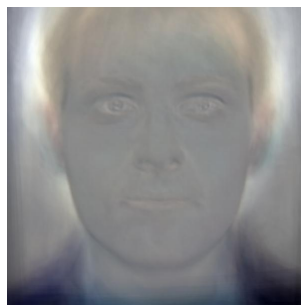
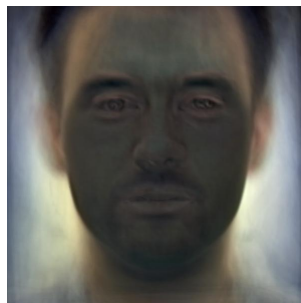
## 1. PCA of color faces:

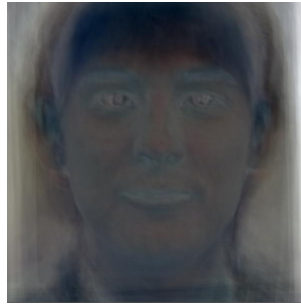
a. 請畫出所有臉的平均。



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

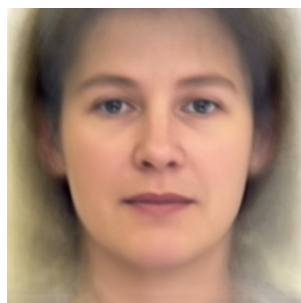
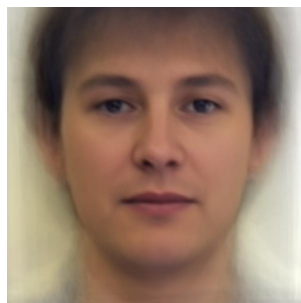
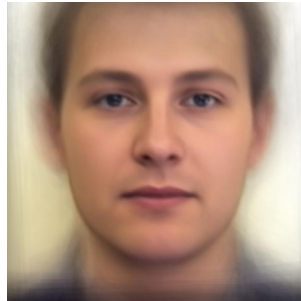
由大到小依序為：

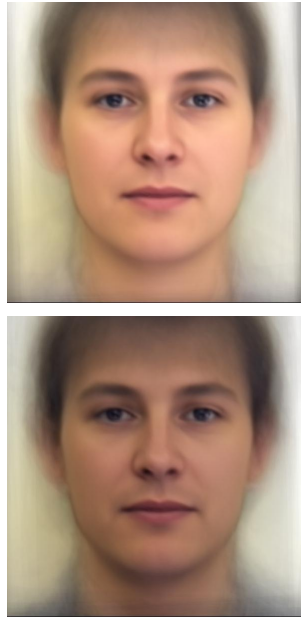




c. 請從數據集中挑出任意五張圖片，並用前五大 Eigenfaces 進行 reconstruction，並畫出結果。

依序為78.jpg、147.jpg、242.jpg、341.jpg、353.jpg的reconstruction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。

根據SVD算出來的 $\Sigma$  ( numpy回傳的s陣列 )，前五大的比重依序為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)

1. autoencoder + codes normalization(對encoder輸出的codes做標準化) + K-means :

reconstruction loss(MSE, pixel範圍[0, 1]):0.011566341435909271

public score:0.94907

private score:0.94902

2. DCEC(<https://xifengguo.github.io/papers/ICONIP17-DCEC.pdf>) (並非kaggle上的結果) :

reconstruction loss(MSE, pixel範圍[0, 1]):0.02234168998003006

public score:0.95934

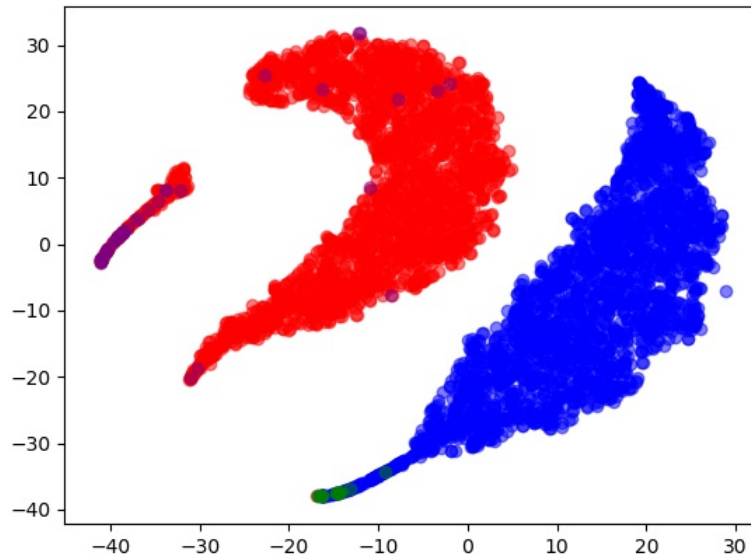
private score:0.95912

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。

我使用DCEC預測，t-SNE做視覺化。紅色的是A預測為A，藍色是B預測為B，綠色是A預測為B，紫色是B預測為A。每個點有alpha=0.5的透明度。



可以看出我的model非常厲害。A只有少數被預測為B，B被預測為A的也很少。

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。

我的model是DCEC(<https://xifengguo.github.io/papers/ICONIP17-DCEC.pdf>)，為autoencoder和cluster layer的組合。以下是我的架構(用類似pytorch的格式描述，大部分參數和原論文相同)：

Encoder:

```
Conv2d(3, 20, kernel_size=5, stride=2, padding=2) -> ReLU ->
Conv2d(20, 40, kernel_size=5, stride=2, padding=2) -> ReLU ->
Conv2d(40, 80, kernel_size=3, stride=2, padding=1) -> ReLU -> Flatten -> Linear(80 * 4 * 4, 24)
```

Decoder:

```
Linear(24, 80 * 4 * 4) -> ReLU -> Reshape(80, 4, 4) ->
ConvTranspose2d(80, 40, kernel_size=3, stride=2, padding=1, output_padding=1) -> ReLU ->
ConvTranspose2d(40, 20, kernel_size=5, stride=2, padding=2, output_padding=1) -> ReLU ->
ConvTranspose2d(20, 3, kernel_size=5, stride=2, padding=2, output_padding=1)
```

Cluster Layer: 同原論文，接在encoder後面，可訓練參數為2個cluster centers  $\mu_1, \mu_2$ ，

soft label定義為  $q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_j (1 + \|z_i - \mu_j\|^2)^{-1}}$ ，

cluster loss定義為  $L_c = KL(P||Q)$  (KL divergence)，其中P矩陣定義為  $p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})}$

當要預測時，取soft label值最大的index。

Loss function:

第一階段訓練：

$$L = L_r$$

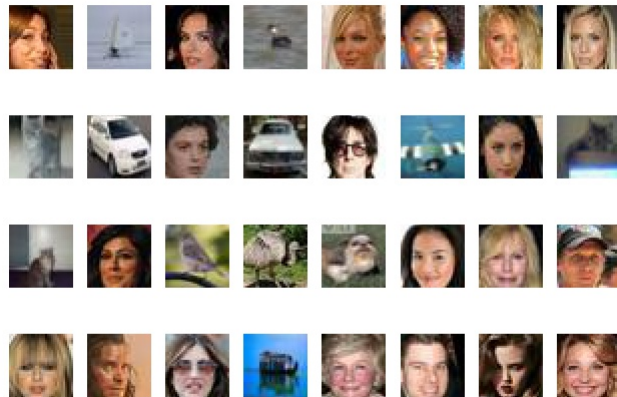
第二階段訓練：

$$L = L_r + 0.1L_c$$

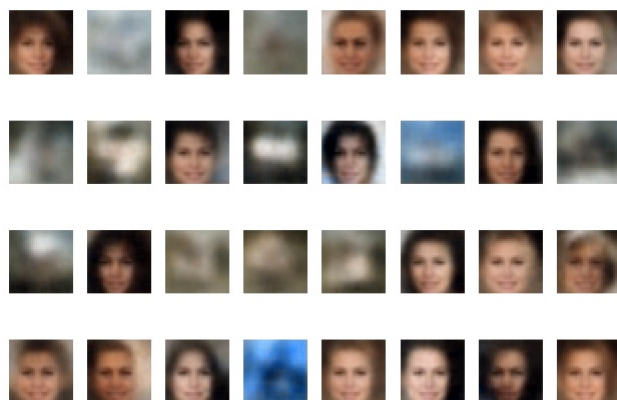
其中  $L_r = MSE(\text{original image, reconstructed image})$  ,  $L_c$  為 cluster layer 算出的 KL divergence 。

以下是其中32張圖片的比較。

原圖：



Reconstruction 結果：



可以看出重建出來的圖片品質沒有特別好，但是此model仍可以高正確率的將兩類圖片分群。這是因為這個model的autoencoder並不是非常強，而且在第二階段訓練時也會稍微的破壞autoencoder reconstruct的品質。然而reconstruct並不是我們的主要目標，讓autoencoder可以很好的separate兩類圖片才是重點。