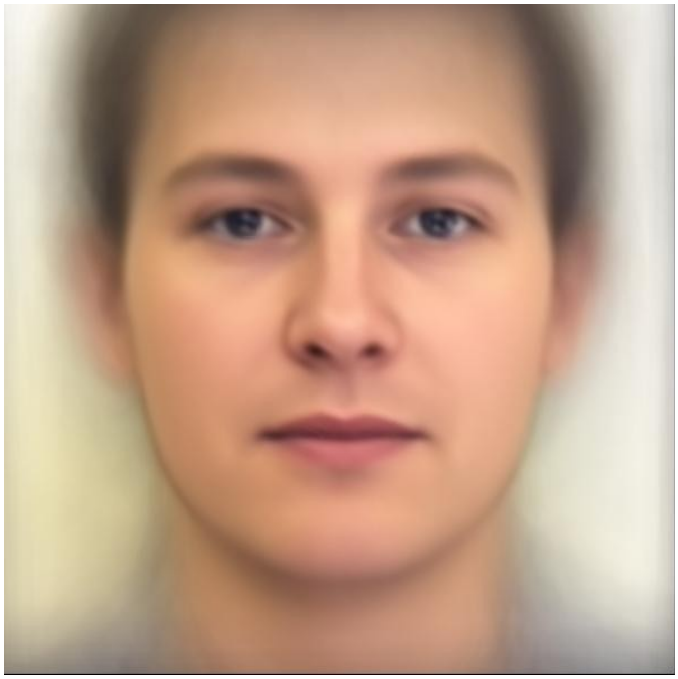
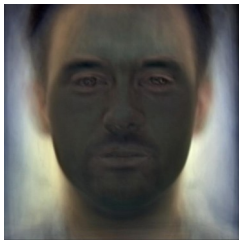
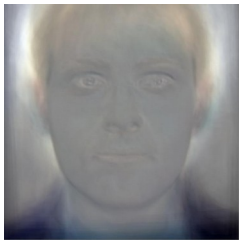
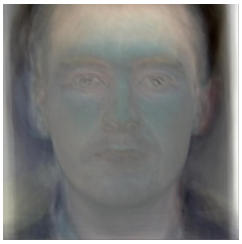
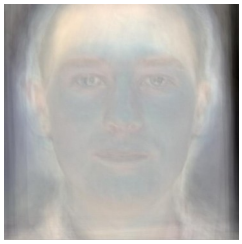
Machine Learning HW7 Report

學號：b05901170　系級：電機三　　姓名：陳柏志

1. PCA of color faces:
   a. 請畫出所有臉的平均。

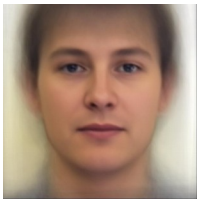
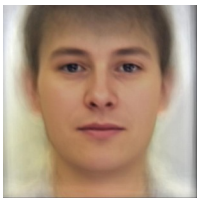
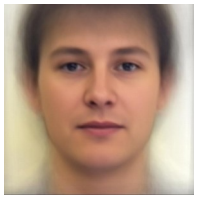
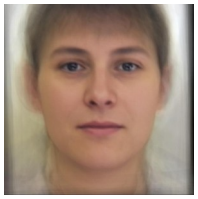
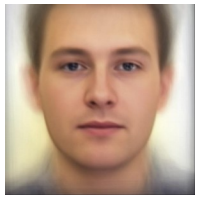


   b. 請畫出前五個 Eigenfaces，也就是對應到前五大 Eigenvalues 的 Eigenvectors。

				
Eigenfaces #1	Eigenfaces #2	Eigenfaces #3	Eigenfaces #4	Eigenfaces #5

   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。

Image name	30.jpg	50.jpg	99.jpg	137.jpg	272.jpg

Origin image					
Reconstructed image					

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。

Eigenfaces #1	Eigenfaces #2	Eigenfaces #3	Eigenfaces #4	Eigenfaces #5
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：

將圖片經過 autoencoder 降維後以 k-means 進行 cluster。

結果：Public Score: 0.54413　Private Score: 0.54544

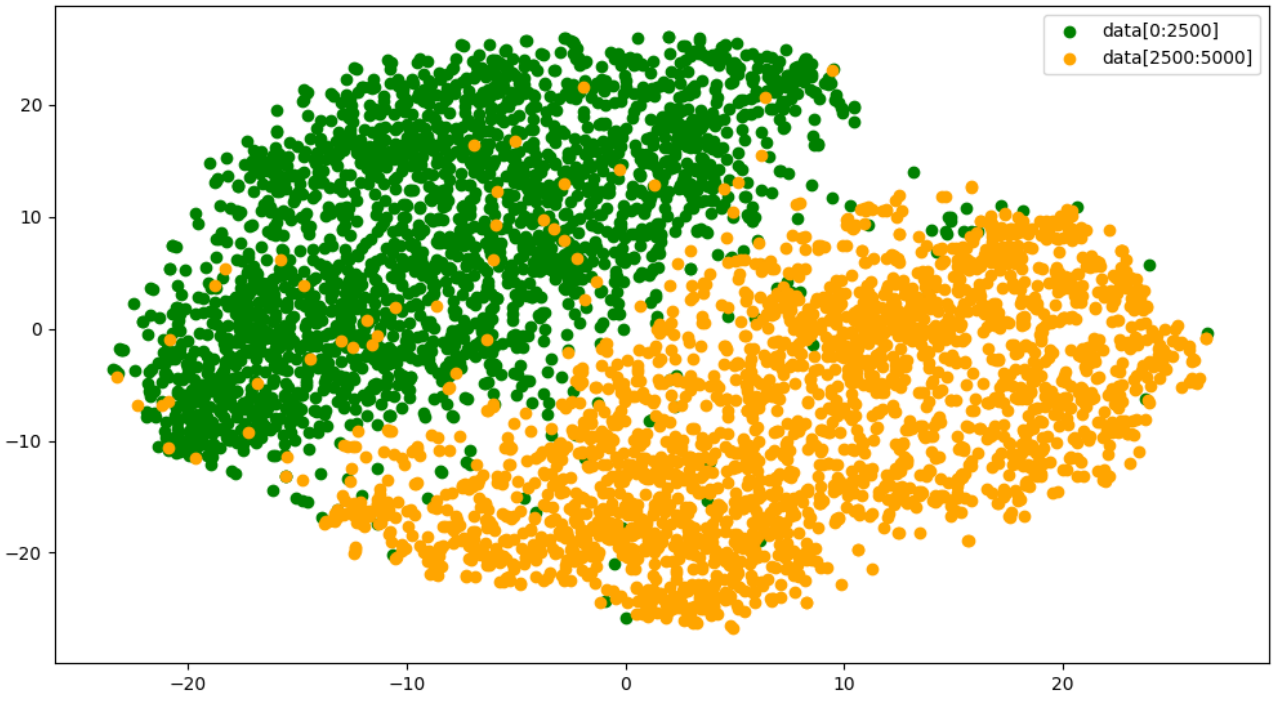
方法二：

將圖片經過 autoencoder 降維後再用 t-SNE 進行一次降維，再以 k-means 進行 cluster。

結果：Public Score: 0.96501　Private Score: 0.96500

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。



和自己預測的 label 比較： 在 5000 筆測資中有 4832 筆正確。

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。

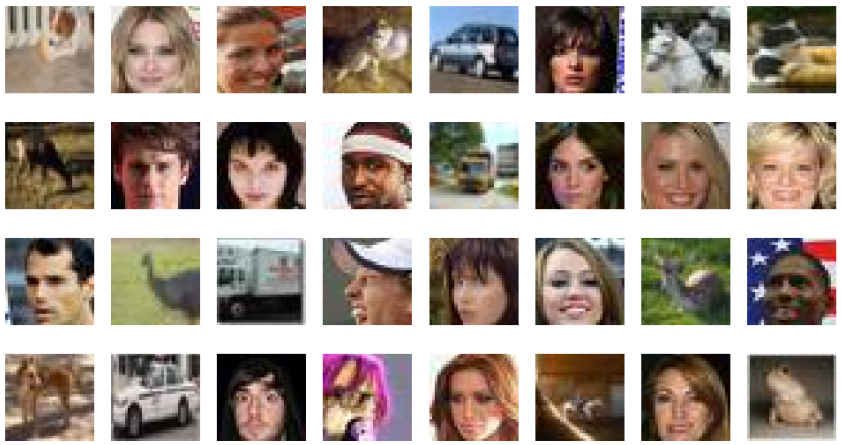
Encoder：

```
self.encoder = nn.Sequential(
        nn.Conv2d(3, 64, 3, stride=1, padding=1),
# b, 64, 32, 32
        nn.LeakyReLU(0.2),
        nn.Conv2d(64, 64, 3, stride=2, padding=1),
# b, 64, 16, 16
        nn.LeakyReLU(0.2),

        nn.Conv2d(64, 128, 3, stride=1,
padding=1),  # b, 128, 16, 16
        nn.LeakyReLU(0.2),
        nn.Conv2d(128, 256, 3, stride=2,
padding=1),  # b, 256, 8, 8
        nn.LeakyReLU(0.2)

    )
    self.encoder_fc = nn.Sequential(
        nn.Linear(256*8*8, 256),
        nn.LeakyReLU(0.2)
    )
```

Decoder:

```
self.decoder_fc = nn.Sequential(
        nn.Linear(256, 256*8*8),
        nn.LeakyReLU(0.2)
    )
    self.decoder = nn.Sequential(
        nn.ConvTranspose2d(256, 128, 3, stride=2,
padding=1, output_padding = 1),  # b, 128, 16, 16
        nn.LeakyReLU(0.2),
        nn.ConvTranspose2d(128, 64, 3, stride=2,
padding=1, output_padding = 1),  # b, 64, 32, 32
        nn.LeakyReLU(0.2),
        nn.ConvTranspose2d(64, 64, 3, stride=1,
padding=1),  # b, 64, 32, 32
        nn.LeakyReLU(0.2),
        nn.ConvTranspose2d(64, 3, 3, stride=1,
padding=1),  # b, 3, 32, 32
        nn.Tanh()
    )
```

Loss Function: *mean-square error*

Oring image [0:32]	
Reconstructed image [0:32]