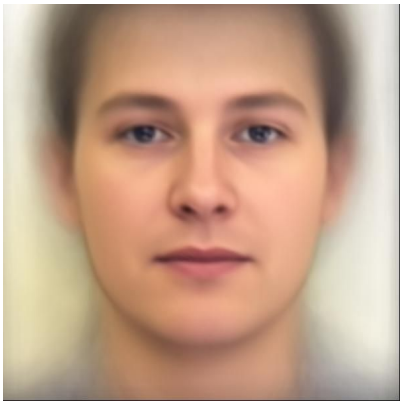
Machine Learning HW7 Report

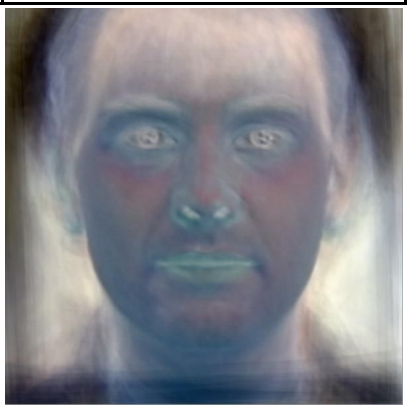
學號：R07922134　系級：資工碩一　　姓名：陳紘豪

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



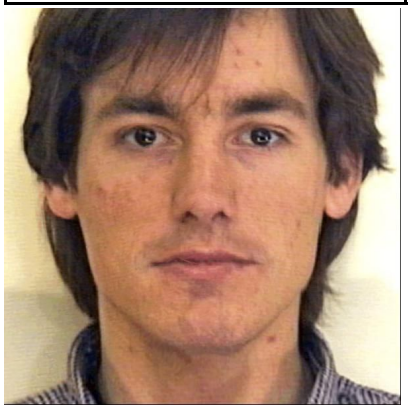
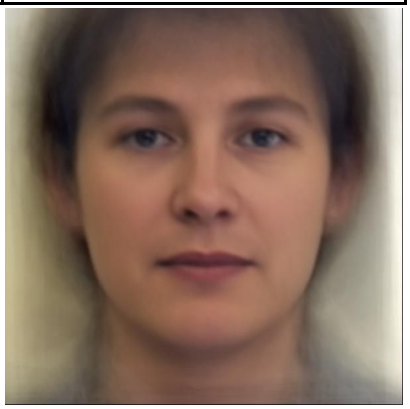
   b. 請畫出前五個 Eigenfaces，也就是對應到前五大 Eigenvalues 的 Eigenvectors。

Eigenface 1	Eigenface 2	Eigenface 3
		

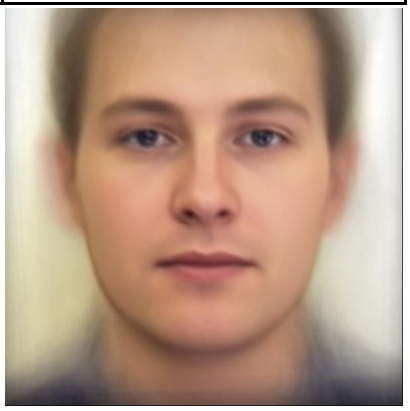
Eigenface 4	Eigenface 5
	

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。

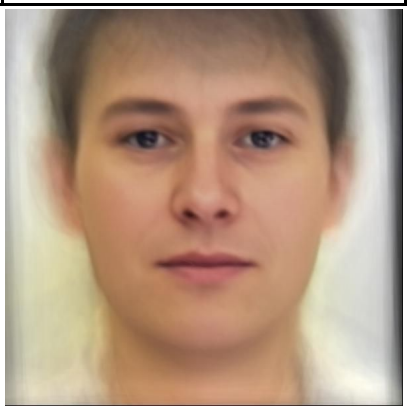
0.jpg

Original image	Reconstructed image
	

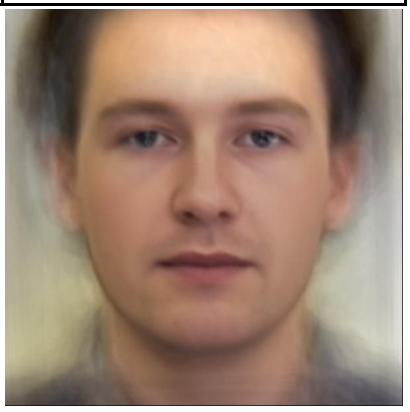
69.jpg

Original image	Reconstructed image
	

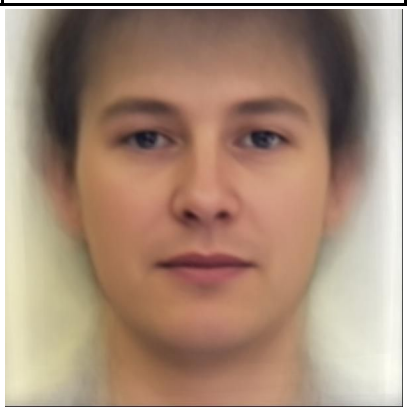
87.jpg

Original image	Reconstructed image
	

111.jpg

Original image	Reconstructed image
	

414.jpg

Original 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。

Eigenface 1	Eigenface 2	Eigenface 3	Eigenface 4	Eigenface 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)

註：Reconstruction loss 統一都用 MSE 去計算，pixel 值範圍是在 [-1, 1] 之間

方法一：Autoencoder 降到 64 維，再用 t-SNE 降到 2 維，最後用 K-means 分 2 群

Reconstruction loss	Public Accuracy	Private Accuracy
0.03070498320754546	0.96974	0.96935

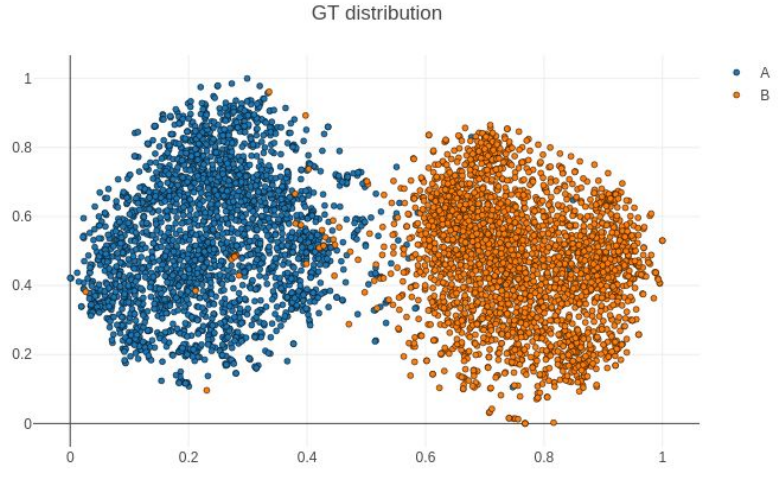
(方法一的成績是 kaggle 上較好的成績，但考慮到執行時間，在 cluster.sh 中我 reproduce 的結果是另外一個 public 為0.96766 的分數)

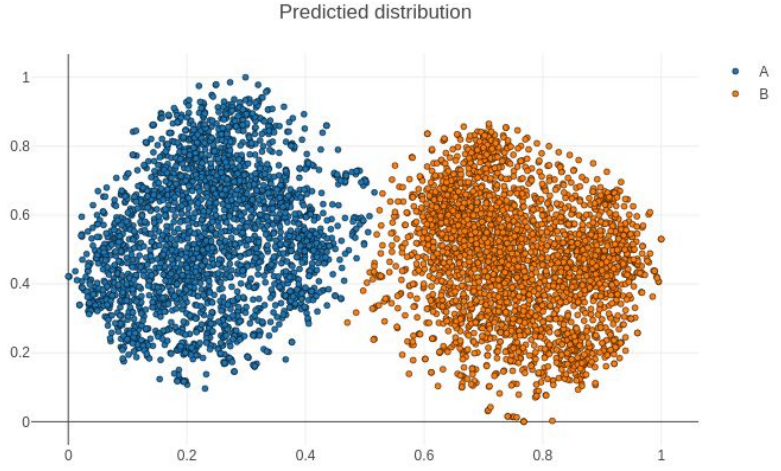
<u>方法二</u>：PCA 降到 10 維，再用 K-means 分 2 群

Reconstruction loss	Public Accuracy	Private Accuracy
0.08372610062360764	0.50096	0.50019

在進行降維前，我都有把圖片 normalize 到 [-1, 1] 之間，所以算出來的 MSE loss 才會是小於 1 的小數，由這兩個方法的結果看來，單純只用 PCA 降維的 accuracy 只有 0.5，跟亂猜沒什麼兩樣，但是若是使用 autoencoder + t-SNE 可以達到相當高的 accuracy，相較於傳統的 PCA，autoencoder 學出來的 latent features 比較能完整的表達出兩個 dataset 的相異性

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。





這兩張圖的結果，是先經由 Autoencoder 降到 64 維再用 TSNE 降到 2 維後可視化的結果，其中上圖是利用 Ground-truth labels 所畫出的圖，下圖是利用 kmeans 分 2 群後所預測出 labels 所構成的圖。
可以觀察到 kmeans 做分群會根據點彼此的距離分成兩個 clusters，但是如果遇到像 ground-truth 中的情形，也就是兩個不同 label 的點距離很近，那 kmeans 可能就無法分得很完美，如果我 autoencoder 的 features 取的更完美些，那麼可能就可再得到更高的準確率。

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。

Loss function：MSE loss

Encoder architecture：
前面是 4 大層的 Conv layer 所組成，經過降維後再經過 Linear 最終降成 64 維的 latent vector

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 16, 16, 16]             784
         LeakyReLU-2          [-1, 16, 16, 16]               0
            Conv2d-3            [-1, 32, 8, 8]           8,224
       BatchNorm2d-4            [-1, 32, 8, 8]              64
         LeakyReLU-5            [-1, 32, 8, 8]               0
            Conv2d-6            [-1, 64, 4, 4]          32,832
       BatchNorm2d-7            [-1, 64, 4, 4]             128
         LeakyReLU-8            [-1, 64, 4, 4]               0
            Conv2d-9           [-1, 128, 2, 2]         131,200
      BatchNorm2d-10           [-1, 128, 2, 2]             256
        LeakyReLU-11           [-1, 128, 2, 2]               0
           Linear-12                 [-1, 128]          65,664
      BatchNorm1d-13                 [-1, 128]             256
        LeakyReLU-14                 [-1, 128]               0
           Linear-15                  [-1, 64]           8,256
================================================================
Total params: 247,664
Trainable params: 247,664
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 0.15
Params size (MB): 0.94
Estimated Total Size (MB): 1.10
----------------------------------------------------------------
```

Decoder architecture：
採用和 Encoder 完全相反且對稱的架構，最後一層會過 Tanh 使得 output image 的範圍是 [-1, 1]，和當初 input image 的範圍是一樣的。

Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	8,320
ReLU-2	[-1, 128]	0
Linear-3	[-1, 512]	66,048
BatchNorm1d-4	[-1, 512]	1,024
ReLU-5	[-1, 512]	0
ConvTranspose2d-6	[-1, 64, 4, 4]	131,136
BatchNorm2d-7	[-1, 64, 4, 4]	128
ReLU-8	[-1, 64, 4, 4]	0
ConvTranspose2d-9	[-1, 32, 8, 8]	32,800
BatchNorm2d-10	[-1, 32, 8, 8]	64
ReLU-11	[-1, 32, 8, 8]	0
ConvTranspose2d-12	[-1, 16, 16, 16]	8,208
BatchNorm2d-13	[-1, 16, 16, 16]	32
ReLU-14	[-1, 16, 16, 16]	0
ConvTranspose2d-15	[-1, 3, 32, 32]	771
Tanh-16	[-1, 3, 32, 32]	0

Total params: 248,531
Trainable params: 248,531
Non-trainable params: 0

Input size (MB): 0.00
Forward/backward pass size (MB): 0.22
Params size (MB): 0.95
Estimated Total Size (MB): 1.17

Reconstruct：

Original image：

Reconstructed image：