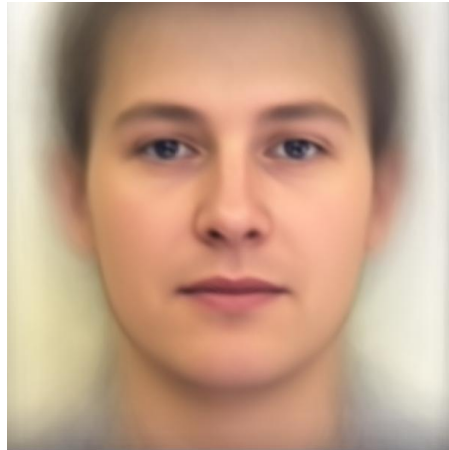


Machine Learning HW7 Report

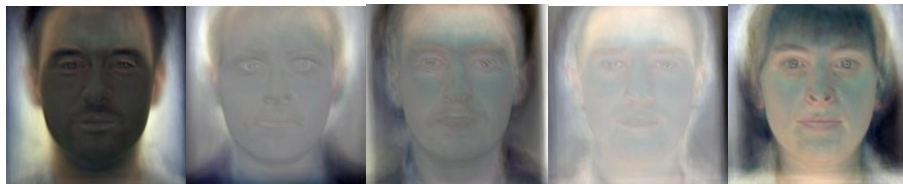
學號：B06507007 系級：材料二 姓名:王致雄

1. PCA of color faces:

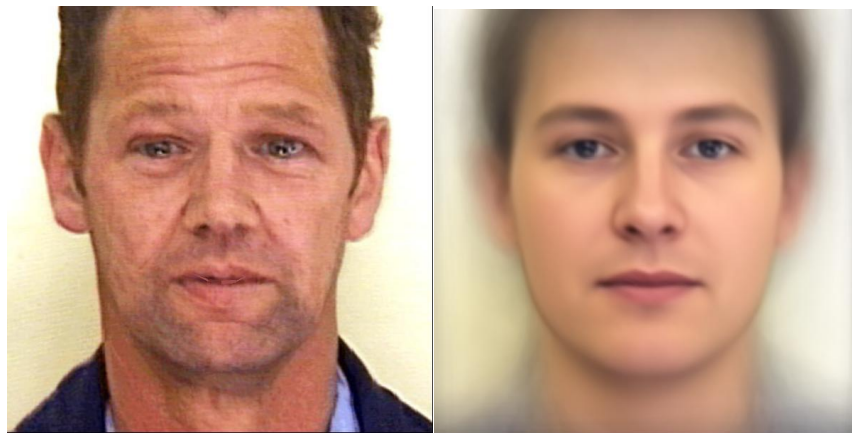
a. 請畫出所有臉的平均。

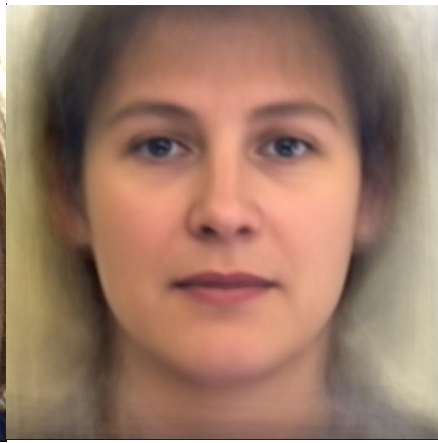
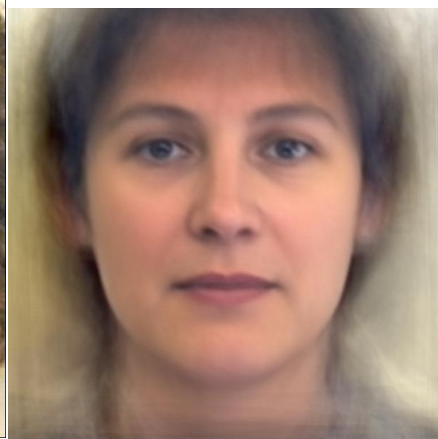


b. 請畫出前五個 Eigenfaces, 也就是對應到前五大 Eigenvalues 的 Eigenvectors。



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。(3,5,29,42,49)







- 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。

Face1	Face2	Face3	Face4	Face5
21.6%	10.9%	7.2%	6.1%	5.4%

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)



我實作使用 DNN 降維(256 維)以及使用 convolution layer 降維。並都使用 Sklearn 裡的 Kmeans 函數分類。(以下 Reconsturction loss 使用最後 epoch 的 loss function 數值平均，為相對而言，loss function 均為 MSE loss。)

結果 DNN 的準確度為 Public:0.94267 Privite:0.94225，Reconsturction loss=0.0095
CNN 的準確度為 Public:0.75233 Privite:0.75128，Reconsturction loss=0.0002

DNN reconstrucion 物像:  (原圖)  (reconstruct 後)

DNN reconstrucion 人像:  (原圖)  (reconstruct 後)

CNN reconstrucion 物像:  (原圖)  (reconstruct 後)

CNN reconstrucion 人像:  (原圖)  (reconstruct 後)

可以看到 CNN 的 loss 與 DNN 相比低非常多，而原圖也能非常清晰的 reconstruct 出來。DNN 則是對人像效果尚可，物像就糊成一片了。然而最終 Kmeans 結果 DNN 卻勝於 CNN，不知道是有哪裡出了什麼問題。

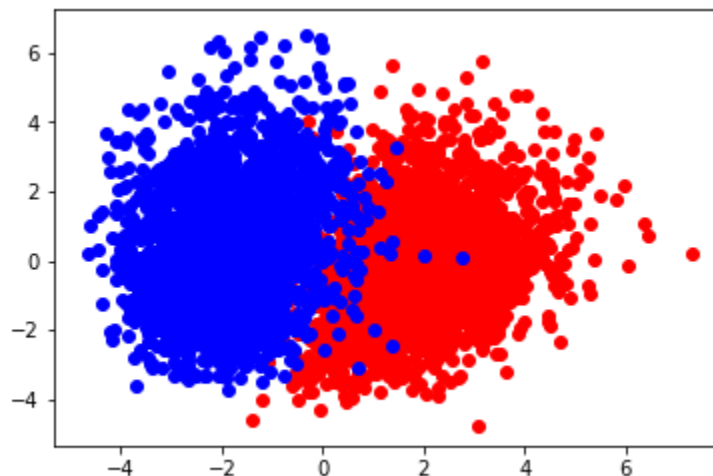
而兩者參數量則是 DNN 遠遠大於 CNN(model 大小 DNN:559MB，CNN:209KB)

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。

使用 PCA 降到二維並作圖得:



其中兩類的數量分別是 2476 跟 2524，我認為已經有相當的準確度。

- 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.Linear(32*32*3, 8192),  
    nn.ReLU(),  
    nn.Linear(8192, 4096),  
    nn.ReLU(),  
    nn.Linear(4096, 2048),  
    nn.ReLU(),  
    nn.Linear(2048, 1024),  
    nn.ReLU(),  
    nn.Linear(1024, 512),  
    nn.ReLU(),  
    nn.Linear(512, 256),  
)
```

Decoder:

```
self.decoder = nn.Sequential(  
    nn.Linear(256, 512),  
    nn.ReLU(),  
    nn.Linear(512, 1024),  
    nn.ReLU(),  
    nn.Linear(1024, 2048),  
    nn.ReLU(),  
    nn.Linear(2048, 4096),  
    nn.ReLU(),  
    nn.Linear(4096, 8192),  
    nn.ReLU(),  
    nn.Linear(8192, 32*32*3),  
    nn.Sigmoid(),  
)
```

參考老師上課投影片的架構, 先放大至 8192 維再縮小, 最後壓縮至 256 維, loss function 使用 MSE loss, optimizer 用 Adam。參數量 139757824(好多)。

32 張圖片如下, 如上所說, 我認為結果並非特別好, 且人像 reconstruct 結果好於物像。

