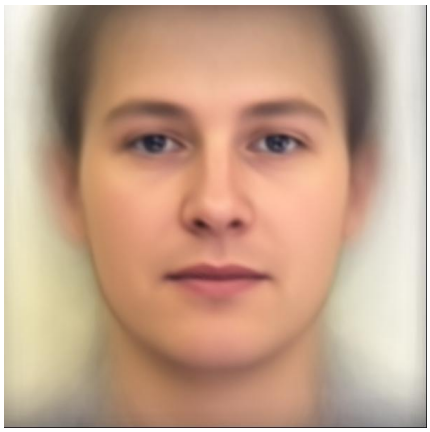
Machine Learning HW7 Report

學號：B06901020 系級：電機二 姓名：張恆瑞

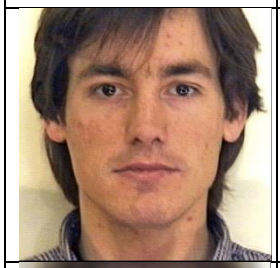
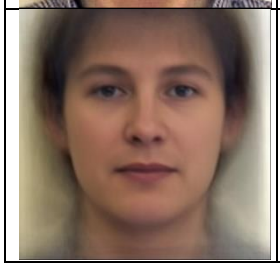
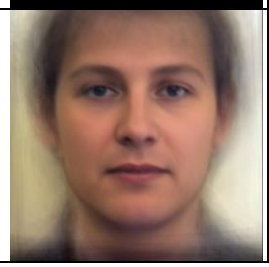
1. PCA of color faces:

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。

0	100	200	300	400
				
				

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。

4.1%	2.9%	2.4%	2.2%	2.1%

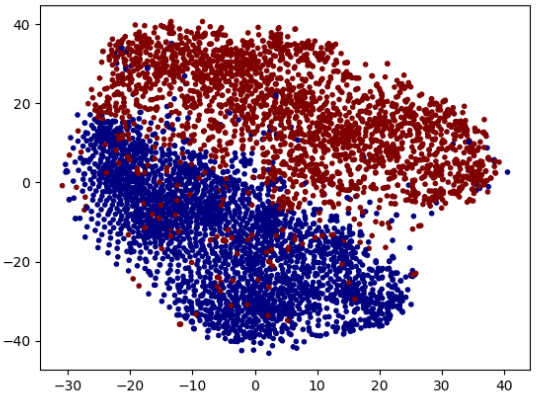
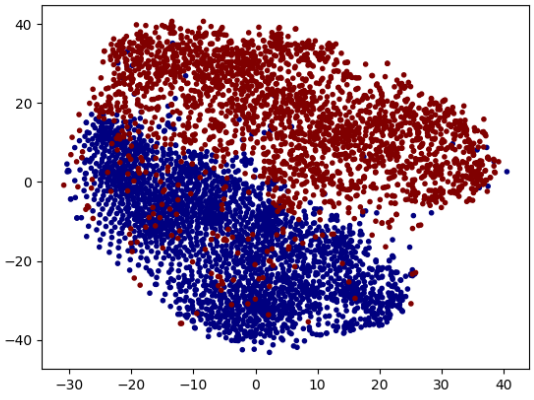
2. Image clustering:

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	reconstruction loss (Total MSE)	accuracy (avg)
autoencoder + PCA + K-means (dim : 3072→1024→1000)	0.021049	0.967835
autoencoder (encoder 加了一層 fully-connected) + PCA + K-means (dim : 3072→1024→1000)	0.029809	0.967840

我的這兩種方法間比較大的差異在於 recontruction loss，第二個方法的 reconstruction loss 幾乎是第一個的 1.5 倍，但效果還是相當，因此似乎在 encoder 中加入一層 fully-connected layer 會讓結果好一些。

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。

autoencoder + PCA + K-means 再用 t-SNE 畫圖	正確答案分布
	

準確率 97.32%

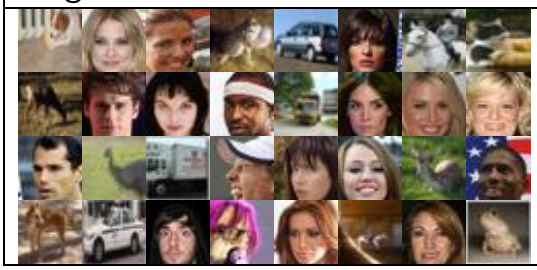
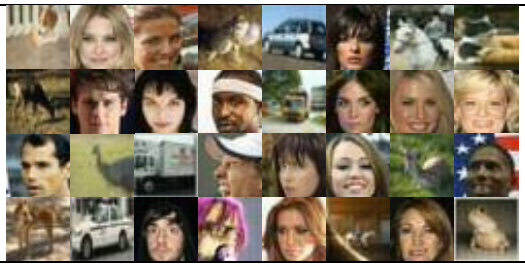
c. 請介紹你的 model 架構(encoder, decoder, loss function...)，並選出任意 32 張圖片，比較原圖片以及用 decoder reconstruct 的結果。

```
self.encoder = nn.Sequential(
  nn.Conv2d(3, 64, 4, 2, 1),
    # [64, 16, 16] = 16384
  nn.BatchNorm2d(64),
  nn.LeakyReLU(0.2),
  nn.Conv2d(64, 32, 3, 1, 1),
    # [32, 16, 16] = 8192
  nn.BatchNorm2d(32),
  nn.LeakyReLU(0.2),
  nn.Conv2d(32, 32, 4, 2, 1),
    # [32, 8, 8] = 2048
  nn.BatchNorm2d(32),
  nn.LeakyReLU(0.2),
  nn.Conv2d(32, 16, 3, 1, 1),
    # [16, 8, 8] = 1024
  nn.BatchNorm2d(16) )
```

```
self.decoder = nn.Sequential(
  nn.ConvTranspose2d(16, 16, 3, 1, 1),
    # [16, 8, 8] = 1024
  nn.BatchNorm2d(16),
  nn.LeakyReLU(0.2),
  nn.ConvTranspose2d(16, 32, 3, 1, 1),
    # [32, 8, 8] = 2048
  nn.BatchNorm2d(32),
  nn.LeakyReLU(0.2),
  nn.ConvTranspose2d(32, 32, 4, 2, 1),
    # [32, 16, 16] = 8192
  nn.BatchNorm2d(32),
  nn.LeakyReLU(0.2),
  nn.ConvTranspose2d(32, 64, 3, 1, 1),
    # [64, 16, 16] = 16384
  nn.BatchNorm2d(64),
  nn.LeakyReLU(0.2),
  nn.ConvTranspose2d(64, 3, 4, 2, 1),
    # [3, 32, 32] = 3072
  nn.Sigmoid() )
```

Loss：Mean Square Error

整體架構：Autoencoder + PCA 來 encode 成 1000 維，並且利用 K-means 來做 clustering

Original	Reconstructed
	

可以 reconstruct 得那麼像我也是嚇到了，應該是我的 latent space 是 1024 維，只是原本圖片的三分之一大小而已，所以大致都不失真，train 大約 200 個 epoch 就可以收斂得不錯了。