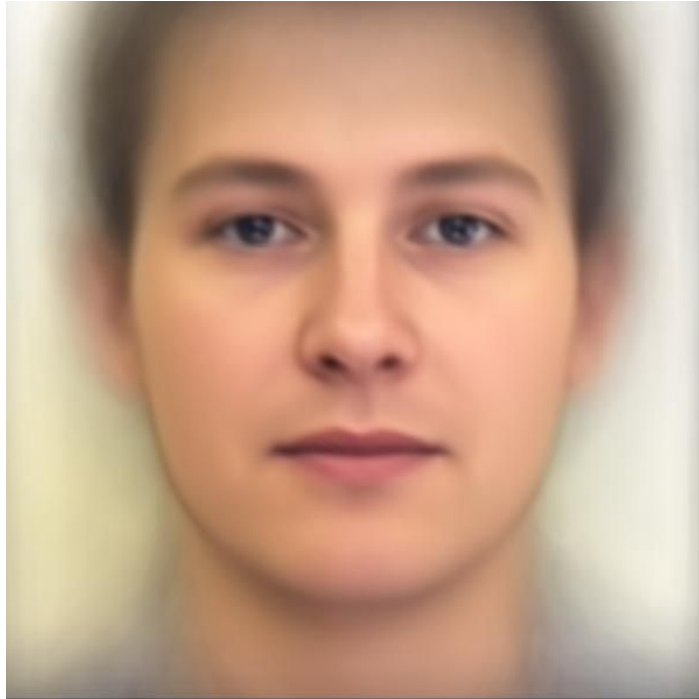


Machine Learning HW7 Report

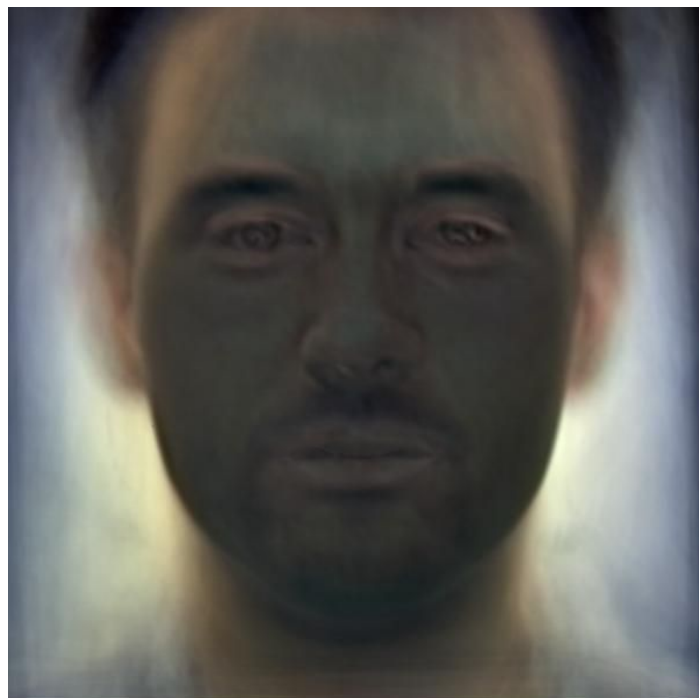
學號：B04505028 系級：工科四 姓名：林秀銓

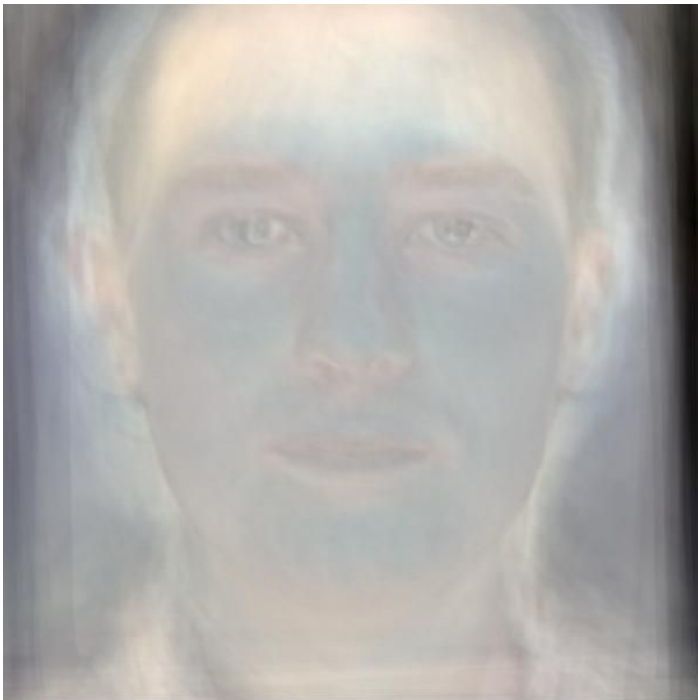
1. PCA of color faces:

- a. 請畫出所有臉的平均。



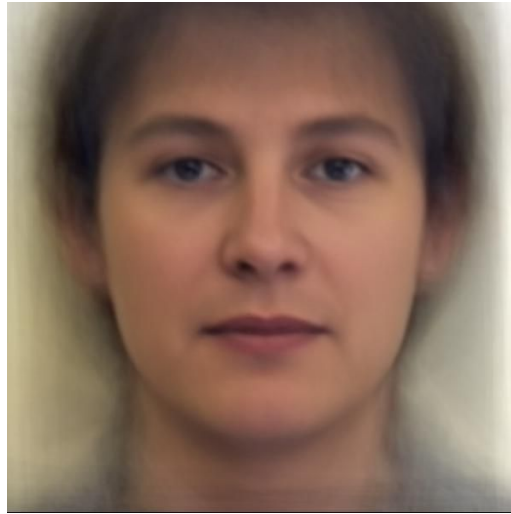
- b. 請畫出前五個 Eigenfaces, 也就是對應到前五大 Eigenvalues 的 Eigenvectors。



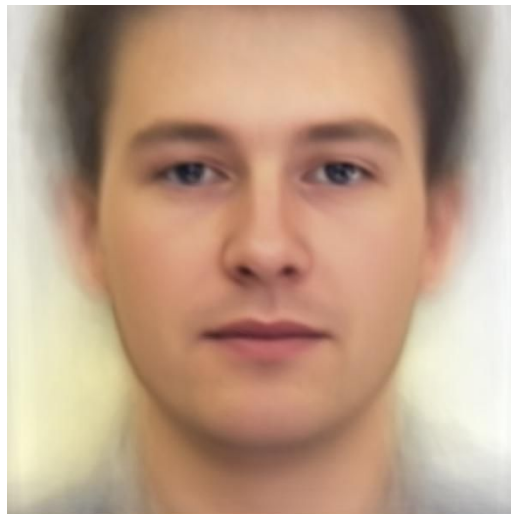


- c. 請從數據集中挑出任意五張圖片，並用前五大 Eigenfaces 進行 reconstruction，並畫出結果。

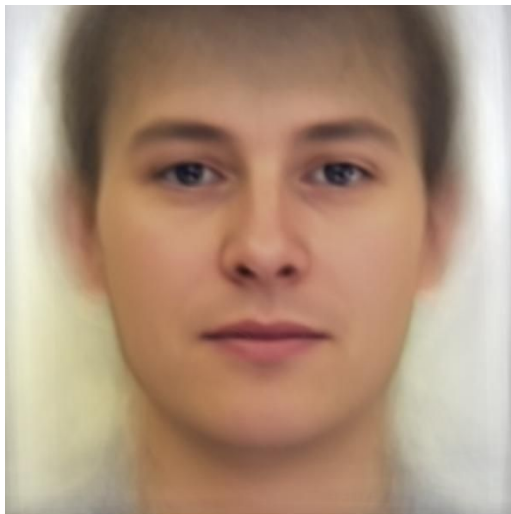
1.jpg



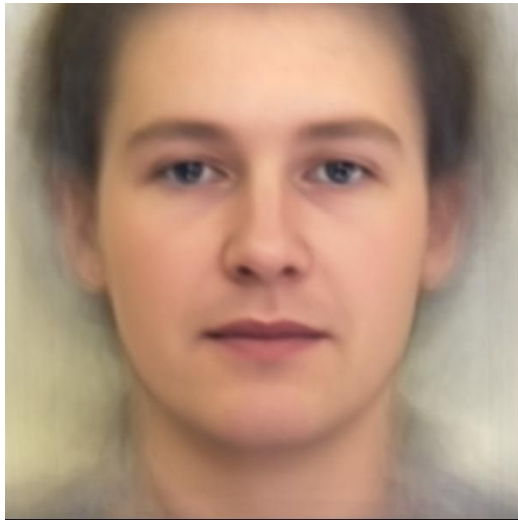
10.jpg



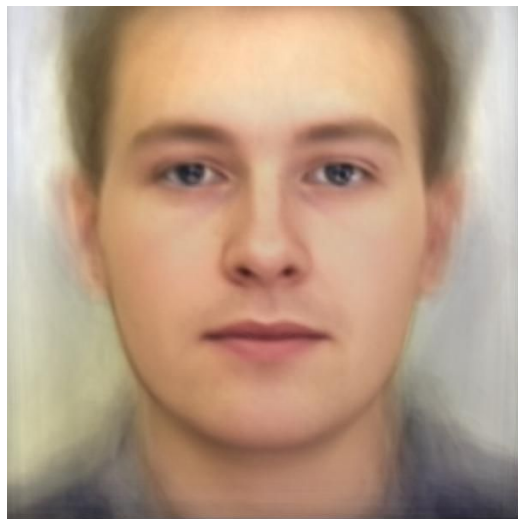
22.jpg



37.jpg



72.jpg



- 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。

	λ_1	λ_2	λ_3	λ_4	λ_5
ration (%)	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. `autuencoder + PCA(n_components=200, copy=False, whiten=True, svd_solver='full')` :

accuracy

Private Score	Public Score
0.96127	0.96109

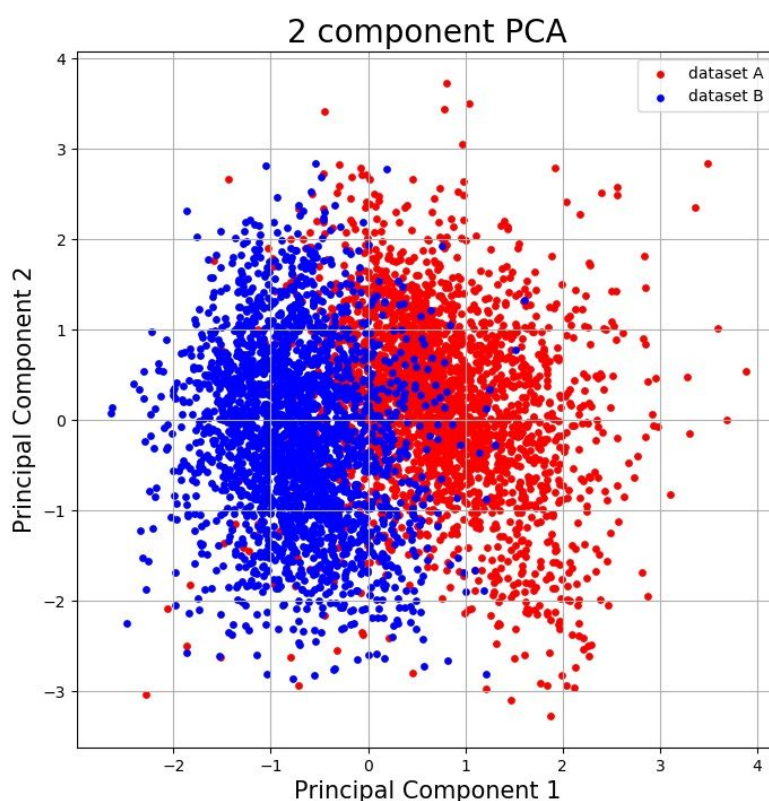
2. `autoencoder + TSNE(n_components=2, init='pca', random_state=0)`:

accuracy

Private Score	Public Score
0.91220	0.91143

因為兩種方法所用的autoencoder的模型相同，所以兩者的 reconstruction loss應該相等。

- 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。



由圖中可觀察到全部的dataset A和dataset B大致分佈在右上和左下，但仍有一些點的分界不是那麼清楚，所以實際用cluster做分類時，準確率為 88.62 %，而這可能是因為我們為了把data視覺化，將他們降到只剩兩維，造成一些重要的feature消失。

- 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。

```
input_img = Input(shape=(32,32,3)) # adapt this if using `channels_first` image data format

x = Conv2D(64, (3, 3), activation='relu', padding='same')(input_img)
x = BatchNormalization()(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(16, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x)
encoded = MaxPooling2D((2, 2), padding='same')(x)

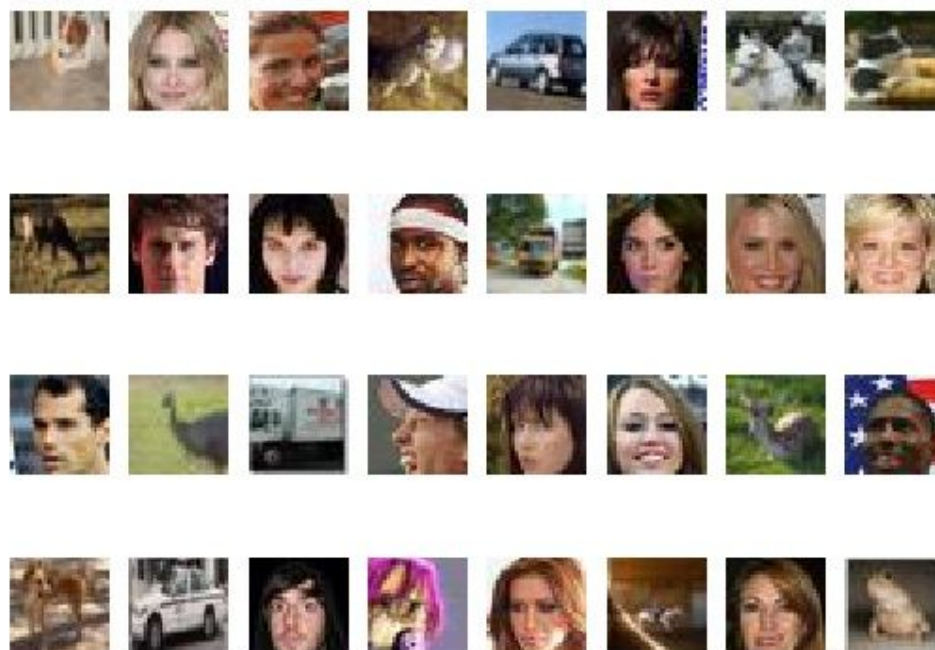
# at this point the representation is (4, 4, 8) i.e. 128-dimensional

x = Conv2D(16, (3, 3), activation='relu', padding='same')(encoded)
x = BatchNormalization()(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = BatchNormalization()(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(x)

autoencoder = Model(input_img, decoded)
autoencoder.summary()
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
```

1~32.jpg

original images



reconstruction images

