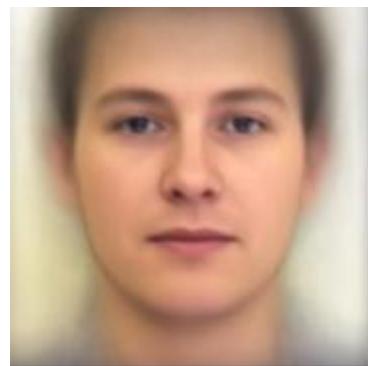


# ML2019SPRING HW7

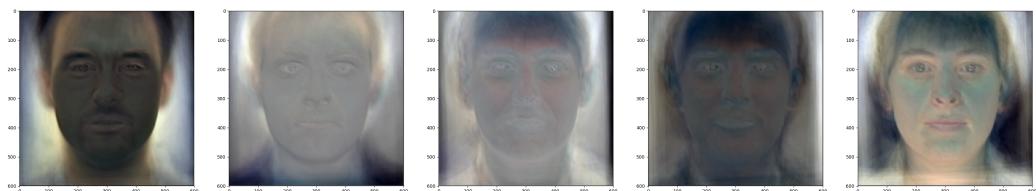
B06902074 資工二 柯宏穎

## PCA of color faces

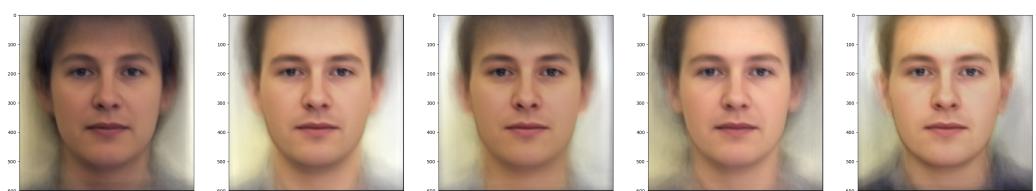
a. 請畫出所有臉的平均。



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。



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。

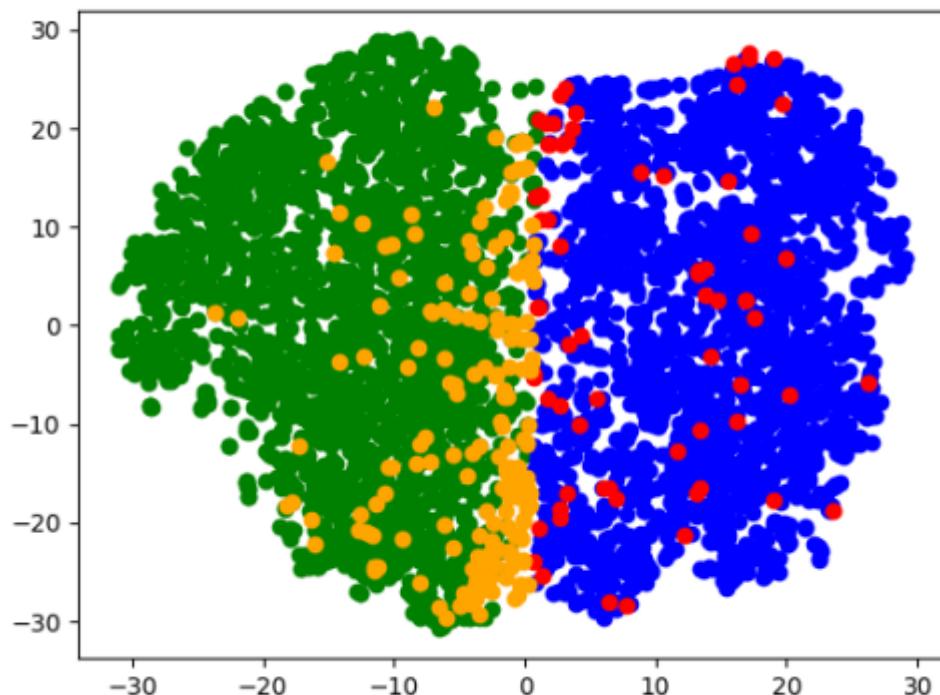
	rate
1	4.2
2	3.0
3	2.4
4	2.2
5	2.1

## 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）。

在同樣的架構，model下，我使用不同的降維方法。一開始我使用PCA降至兩維，或直接使用model訓練出來的32維資料下去，再利用K-means做分群。不過效果並沒有很理想，甚至有無降維並無太大的差別。最好的準確率在0.93左右。後來助教開放使用multicore的TSNE後，其降維時間下降很多，能更有效率地運算，成效也很好，同樣一個model(reconstruction loss(mse)0.007)，準確率來到0.97。另外還有train了一顆差異更大的model(reconstruction loss(mse)0.004，比最好model多兩層Conv2D與一層Maxpooling)，在沒有使用TSNE的情形下，只用flatten後32維做判斷，準確率只有0.55左右。做完更進一步的降維後，準確率來到0.935。一個好的降維方法，能幫助K-means好分類非常多。

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。



我先利用先前train好的encoder將圖片降至32維，再用TSNE降至2維。圖中左半部屬於dataset A的領域，綠色為來自dataset A且預測正確的圖，橘色則為來自dataset B但預測為A的圖。另一邊則為dataset B的領域，藍色為來自dataset B且預測正確的圖，紅色為來自dataset A但預測錯誤的圖。

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))

encoded = Conv2D(16, (4, 4), padding = "same", activation = "relu")(input_img)
encoded = Conv2D(32, (4, 4), padding = "same", activation = "relu")(encoded)
encoded = MaxPooling2D((2, 2), padding = "same")(encoded)
encoded = Conv2D(64, (4, 4), padding = "same", activation = "relu")(encoded)
encoded = Conv2D(128, (4, 4), padding = "same", activation = "relu")(encoded)
encoded = MaxPooling2D((2, 2), padding = "same")(encoded)
encoded = Flatten()(encoded)
encoded = Dense(32, activation = "relu")(encoded)

decoded = Dense(8 * 8 * 128, activation = "relu")(encoded)
decoded = Reshape((8, 8, 128))(decoded)
decoded = UpSampling2D((2, 2))(decoded)
decoded = Conv2D(128, (4, 4), activation = "relu", padding = "same")(decoded)
decoded = UpSampling2D((2, 2))(decoded)
decoded = Conv2D(64, (4, 4), activation = "relu", padding = "same")(decoded)
decoded = Conv2D(32, (4, 4), activation = "relu", padding = "same")(decoded)
decoded = Conv2D(3, (4, 4), activation = "sigmoid", padding = "same")(decoded)
adam = Adam(lr=0.0005, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0,
amsgrad=False)
autoencoder.compile(optimizer = adam, loss = "mean_squared_error")
```

我利用unit數量從小到大的convolution layer做卷積，中間適當地Maxpooling做降維，其中卷積時，我將padding設成same以避免圖片在卷積時逐漸降維，而造成之後decoding的困難。最後將圖片用flatten與fully connected壓成32維的向量，方便之後的預測。

