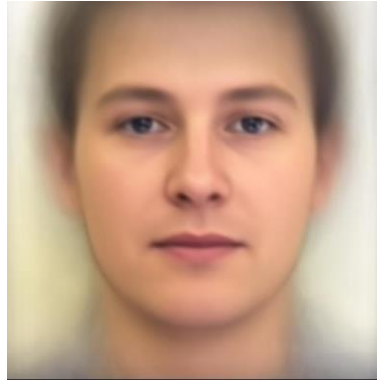


## Machine Learning HW7 Report

學號：B06902066 系級：資工二 姓名：蔡秉辰

### 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。



由左上到右下分別為前五大 Eigenvalues 對應到的圖片。

- 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。  
取 index 1, 10, 22, 37, 72 的圖片(即為手把手之 code 所使用的圖片)  
Reconstruction 結果依序為以下：



- 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。  
分別為：4.1%, 3.0%, 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) 使用兩相同的 **autoencoder**，架構如下：

Input( ) + Dropout(0.1)  
Conv2D(32, (3, 3), 'relu') + BatchNormalization( )  
Conv2D(64, (3, 3), 'relu') + BatchNormalization( )  
MaxPooling2D((2, 2))  
Conv2D(128, (3, 3), 'relu') + BatchNormalization( )  
Conv2D(256, (3, 3), 'relu') + BatchNormalization( )  
MaxPooling2D((2, 2))  
Flatten( ) + Dense(64, 'relu')

(2) 以下為兩個方法之差異：

方法一：使用 **t-SNE** 降成兩維，再使用 **sklearn** 中的 **K-means**

方法二：直接使用 **sklearn** 中的 **K-means**

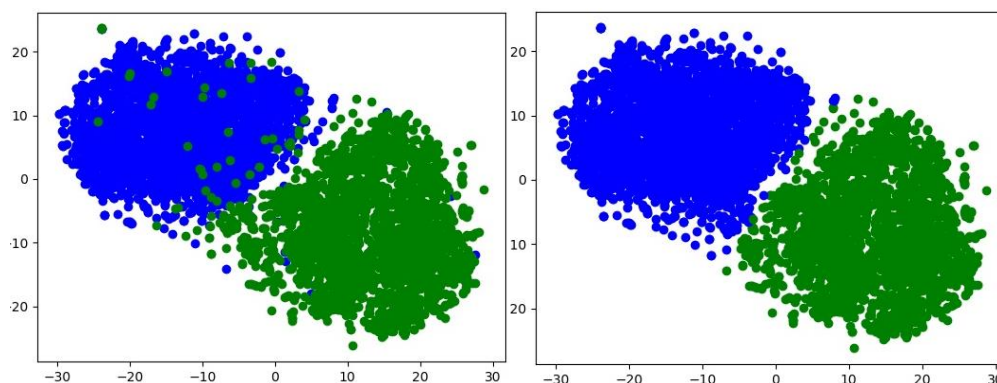
(3) 結果：

	Private accuracy	Public accuracy
方法一	0.96183	0.96222
方法二	0.70525	0.70432

註：因 **auto-encoder** 相同，故 **reconstruction loss** 會相同。

- 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。

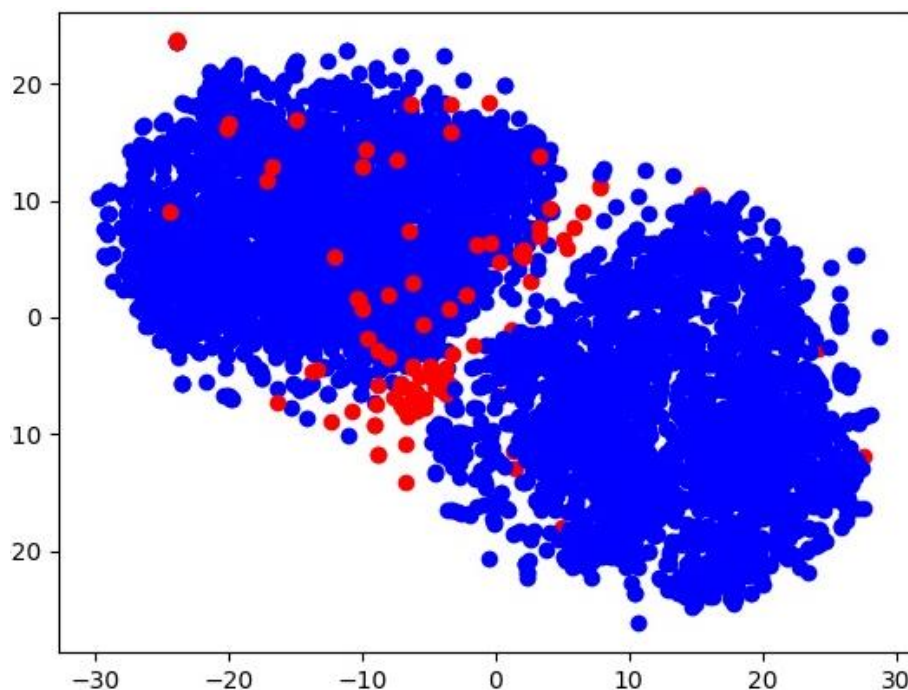
使用 auto-encoder 降到 64 維，再使用 MulticoreTSNE 降到兩維：



以上兩張圖：

- (1) 左圖中：藍色點為 dataset A 的點降到二維的分布，綠色點為 dataset B 的點降到二維的分布。
- (2) 右圖中：藍色點為 model 預測維 dataset A 的點，綠色點為 model 預測維 dataset B 的點。

將其作整合後，可以整理預測正確/錯誤的點如下：



其中，藍色的點為預測正確之 data；紅色的點為預測錯誤之 data。

- 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。

(1) Model 架構：

(i) Encoder:

Input( ) + Dropout(0.1)

Conv2D(32, (3, 3), 'relu') + BatchNormalization( )

Conv2D(64, (3, 3), 'relu') + BatchNormalization( )

MaxPooling2D((2, 2))

Conv2D(128, (3, 3), 'relu') + BatchNormalization( )

Conv2D(256, (3, 3), 'relu') + BatchNormalization( )

MaxPooling2D((2, 2))

Flatten( ) + Dense(64, 'relu')

(ii) Decoder:

Dense(128\*8\*8, 'relu') + Reshape((8, 8, 128))

UpSampling2D((2, 2))

Conv2DTranspose(256, (3, 3), 'relu') + BatchNormalization( )

Conv2DTranspose(128, (3, 3), 'relu') + BatchNormalization( )

UpSampling2D((2, 2))

Conv2DTranspose(64, (3, 3), 'relu') + BatchNormalization( )

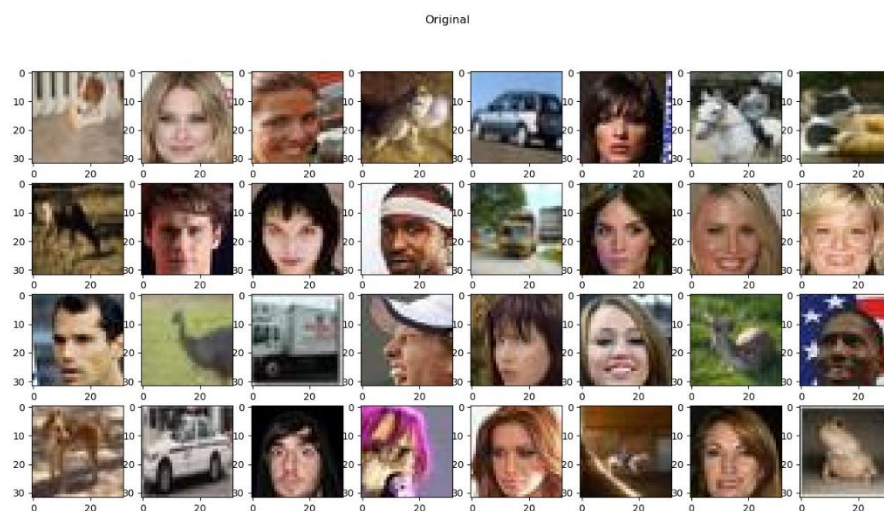
Conv2DTranspose(32, (3, 3), 'relu') + BatchNormalization( )

Conv2D(3, (3, 3), 'sigmoid')

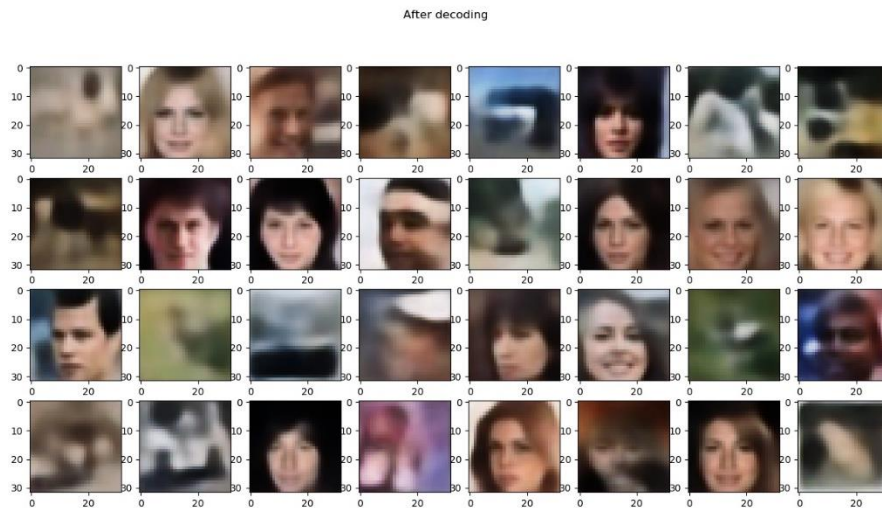
(iii) 使用參數：

Optimizer = 'adam', loss function = 'mse', epochs = 15

- (2) 取前 32 張圖來做比較。先印出原圖，再印出 reconstruct 之結果：



(原圖)



(reconstruct 後之圖)

可以看出 **reconstruct** 過後的圖會顯得較為模糊，尤其是非人臉的圖片。但仍然能大致看得出其類別以及原本大概的樣子(非人臉較難辨識)。可推斷此 **model** 應該是不错的。