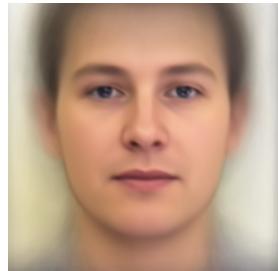


A. PCA of colored faces

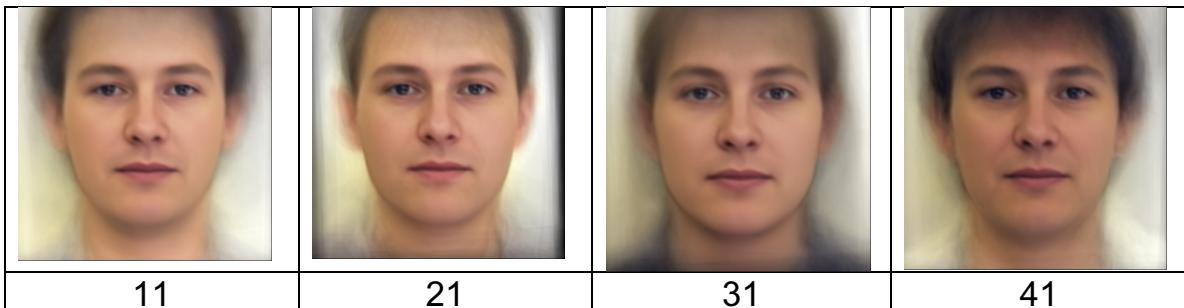
1. (.5%) 請畫出所有臉的平均。



2. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。



3. (.5%) 請從數據集中挑出任意四個圖片，並用前四大 Eigenfaces 進行 reconstruction，並畫出結果。



4. (.5%) 請寫出前四大 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。

第一大：4.1% 第二大：2.9% 第三大：2.4% 第四大：2.2%

B. Image clustering

1. (.5%) 請比較至少兩種不同的 feature extraction 及其結果。(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)

(I) 利用 AutoEncoder

Encoder
Dense(1024,activation='relu')
Dense(512,activation='relu')
Dense(512,activation='relu')

```

Dense(256,activation='relu')
Dense(256,activation='relu')
Dense(64,activation='relu')
Dense(64,activation='relu')
Dense(64,activation='relu')
Dense(128,activation='relu')
Decoder
Dense(128,activation='relu')
Dense(256,activation='relu')
Dense(512,activation='relu')
Dense(512,activation='relu')
Dense(784,activation='relu')

```

再透過 Kmeans 來分類，分成 2 類

(2) 利用 PCA

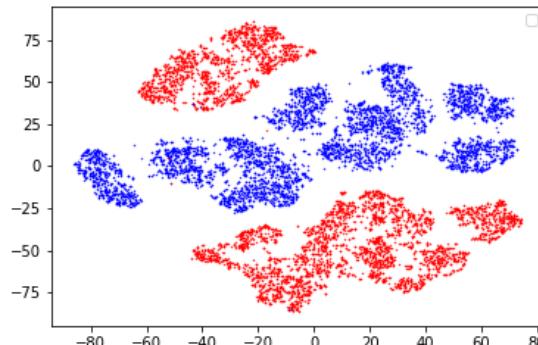
利用 PCA 降維至 125，在利用 Kmeans 分至 10 類

	Public score	Private score
Autoencoder + Kmeans	0.99708	0.99708
PCA + Kmeans	0.99000	0.98990

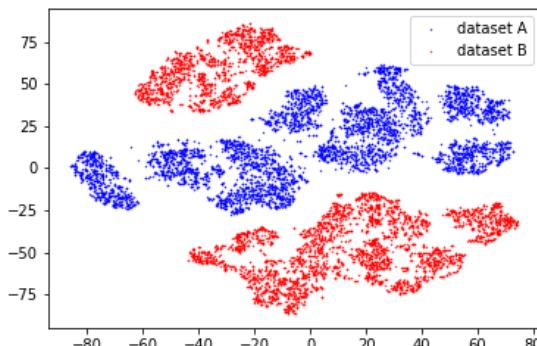
我們 PCA 在取的時候需要先選擇降維的維度，Kmeans 也需要選擇分類的維度，因此在實驗上為

PCA n=	Kmeans n=	Public Score	Private Score
10	16	0.98538	0.98552
10	18	0.98703	0.98717
10	20	0.98777	0.98779
10	30	0.98936	0.98923
10	40	0.98978	0.98972
10	60	0.98979	0.98966
10	80	0.98997	0.98997
10	100	0.98999	0.98993
10	125	0.99000	0.98990
10	150	0.99000	0.98990
10	200	0.98998	0.98990

2. (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。



3. (.5%) visualization.npy 中前 5000 個 images 跟後 5000 個 images 來自不同 dataset。請根據這個資訊，在二維平面上視覺化 label 的分佈，接著比較和自己預測的 label 之間有何不同。



Autoencoder 在 Kaggle 上分數為 0.99708，因此在分類上其實有蠻高的準確率，因此發現分錯的點數極少，大致上是相同的。

C. Ensemble learning

1. (1.5%) 請在 hw1/hw2/hw3 的 task 上擇一實作 ensemble learning，請比較其與未使用 ensemble method 的模型在 public/private score 的表現並詳細說明你實作的方法。（所有跟 ensemble learning 有關的方法都可以，不需要像 hw3 的要求硬塞到同一個 model 中）

	Public score	Private Score
Model1	0.68929	0.67651
Model2	0.68264	0.67288
Ensemble method1	0.67567	0.66898

方法一

- (1) 直接將 model1 以及 model2 對 testing data predict 的結果做平均值

利用平均的方式效果並未比較好，我們可以從資料當中發現

	Model1	Model2	Model1+ Model2
第 01 筆	Class 3: 1.0 Class 5: 2.14e-16	Class 0: 0.3984 Class 5: 0.3249	Output: Class 3
第 04 筆	Class 6: 0.4165 Class 4: 0.2944	Class 5: 0.3442 Class 0: 0.3262	Output: Class 2: 0.2216

發現用平均的方式，極有可能分類到 model1 以及 model2 都不在最高分的狀況，如第四筆資料一樣，因此才會導致分數變低。