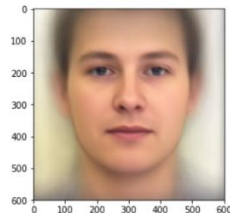


HW4

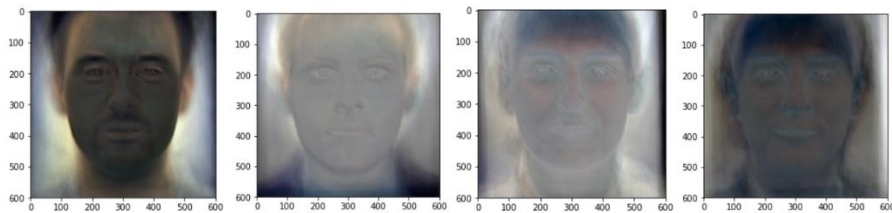
學號：b03505052 系級：工程科學及海洋工程學系四年級 姓名：李吉昌

A. PCA of colored faces

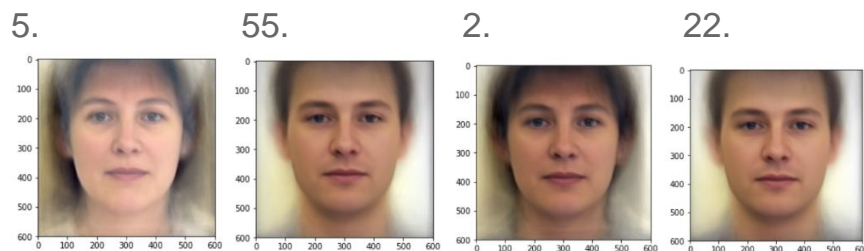
A.1. (.5%) 請畫出所有臉的平均。



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



A.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。



A.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。

1. 0.04144624838262974 : 4.1%
2. 0.029487322251120787 : 2.9%
3. 0.023877112932084086 : 2.4%
4. 0.022078415569025466 : 2.2%

B. Image clustering

B.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)

| | private | public |
|-----|---------|---------|
| DNN | 0.99966 | 0.99964 |
| CNN | 0.99584 | 0.99593 |

```

input_img = Input(shape=(28, 28, 1))
x = Conv2D(128, (3, 3), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
encoded = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(32, (3, 3), activation='relu', padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(128, (3, 3), activation='relu')(x)
x = UpSampling2D((2, 2))(x)
decoded = conv2d(1, (3, 3), activation='sigmoid', padding='same')(x)
autoencoder = Model(input_img, decoded)

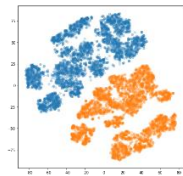
model = Sequential()
model.add(Dense(392, input_dim=784, activation='relu'))
model.add(Dense(196, activation='relu'))
model.add(Dense(98, activation='relu'))
model.add(Dense(49, activation='relu'))
model.add(Dense(98, activation='relu'))
model.add(Dense(196, activation='relu'))
model.add(Dense(392, activation='relu'))
model.add(Dense(784, activation='sigmoid'))

```

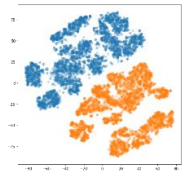
比較兩種 Autoencoder 的架構，分別實作 DNN 以及 CNN，

Kaggle 上的分數目前做到最好的架構是 DNN，因為 NN 的架構是做非線性的處理，因此在 clustering 的效果會做得比純粹只有 SVD 降維要好，至於 CNN 效果為什麼比較差，個人猜測是因為 Neural 數以及 layer 數沒有抓好。

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



B.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。



兩張圖一樣，因為自己在 kaggle 上分數已經接近 1，預測出來的結果應該也會滿接近全部預測正確。

C. Ensemble learning

C.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）

| | private | public |
|--------------|---------|---------|
| ensemble | 0.72053 | 0.71914 |
| non-ensemble | 0.69908 | 0.70047 |

我作業三就有使用 ensemble，用同一種架構但不同的資料處理以及不同的 opt，optimizer 選用兩種分別是 adam (default) 以及 SGD (lr=0.005, decay=0.00001, momentum=0.9) adam 得 0.70047 / SGD 得 0.69908

另外再將資料作處理 ($x = x - 127.5$) 再用兩種 optimizer 得出另外兩個 model Kaggle 的 best acc: 0.71524 則是將四種 model 做 ensemble 得出的結果。