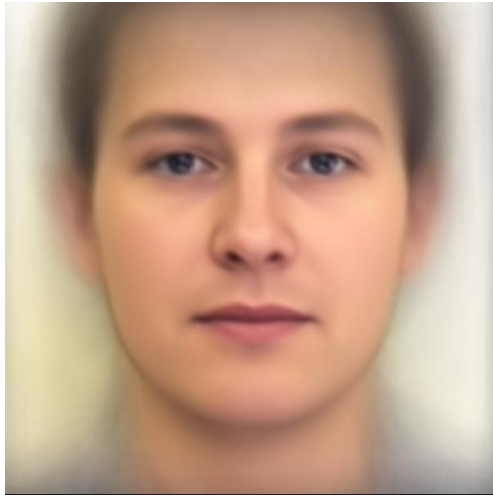


HW4 Report

學號：B03901015 系級：電機四 姓名：梅希聖

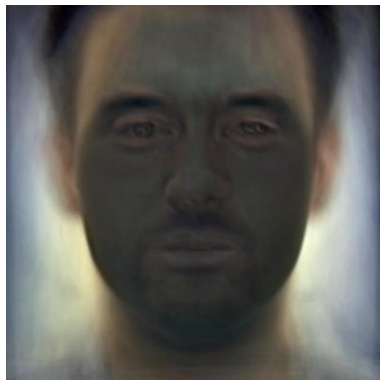
A. PCA of colored faces

1. (.5%) 請畫出所有臉的平均。

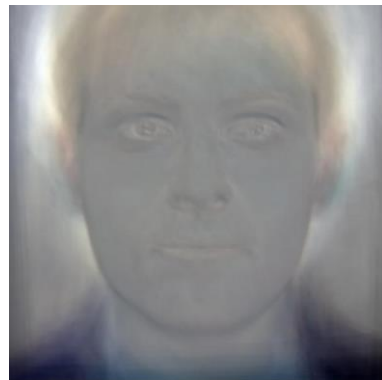


2. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

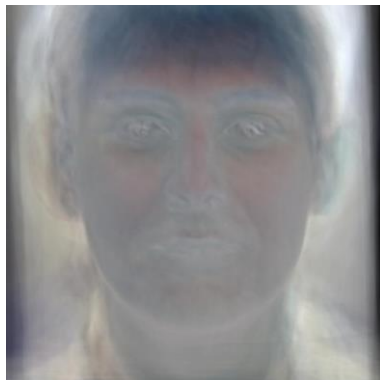
1st



2nd



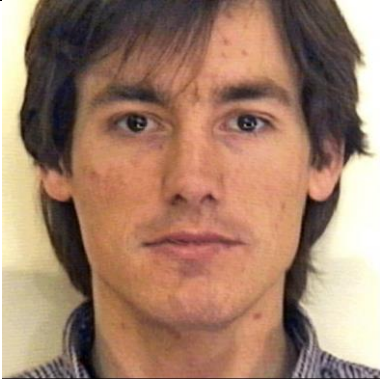






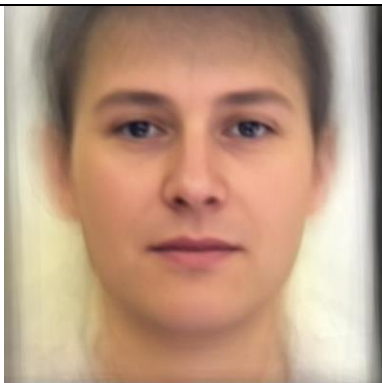
3rd



4th



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。

	原圖	重建後	
0.jpg			
20.jpg			
40.jpg			
60.jpg			

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。
- 取 SVD 解出之前 4 大 singular value，計算其所佔之比重，其結果為：[4.1%, 2.9%, 2.4%, 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)

Image clustering 的實作方法是先使用 PCA 降維，再使用 kmeans 分成 2 個類別，設定 PCA 降維後的維度，與準確率有很大的關係：

300 維：0.99998(private) / 0.99998(public)

200 維：0.99994(private) / 0.99994(public)

120 維：0.15010(private) / 0.15050(public)

50 維：0.15057(private) / 0.14998(public)

2 維：0.52214(private) / 0.52155(public)

並不是任意的 PCA 維度都能準確地將此筆 dataset 作分類，較高維的 data 得到較高的準確率似乎滿合理的，120 維與 50 維取出的 feature，雖然演算法相同，但取出的特徵可能並不是此筆 data 中可供分類的重要特徵。

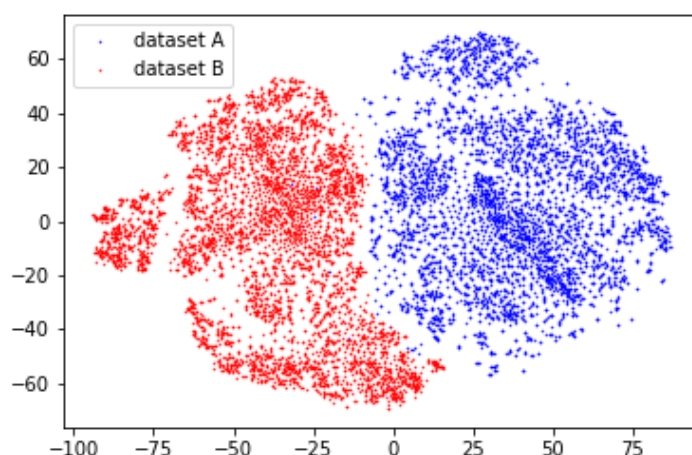
我嘗試的另一種方法是先使用 PCA 降至較低維後(ex: 50)，再使用 TSNE 降至二維，最後再用 kmeans 分類，得到的 f1-score 是 0.88614(private) / 0.88613 (public)，分數並不如 PCA 的最高分來的高。

另外，如果想對這個 dataset 直接使用 TSNE 降維似乎不太實際，因為 TSNE 演算法的運算複雜度極高。在第二部分中，使用從 50 維降至 2 維在我的電腦上大約要跑 3 個小時，但使用 PCA 在數十秒內即可完成。

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

使用 PCA 降至 200 維再以 kmeans 進行分類，為了視覺化，再使

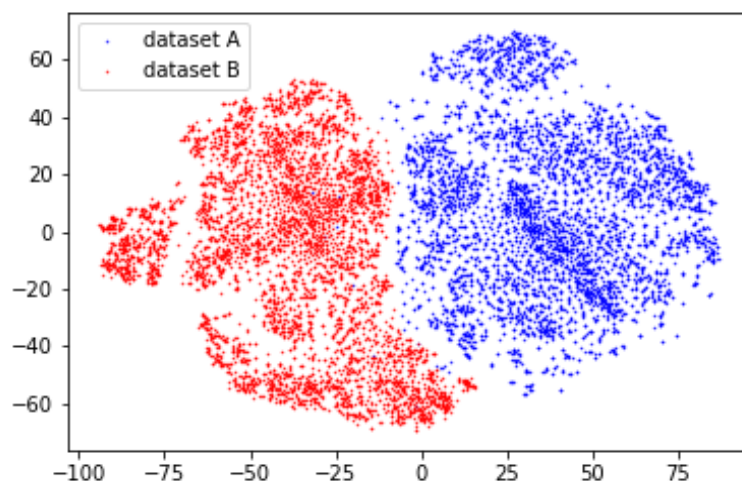
用 TSNE 降至 2 維後作圖如下：



從圖中可發現，2 個類別間有一條明顯的界線，分類相當清楚。

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。

自己的分類器認為前 5000 與後 5000 個圖片分別屬於不同分類，預測結果與實際情況完全相符，作圖結果也與上圖相同，代表原本的 PCA 降維對於原本的 dataset (image.npy) 與 visualization.npy 可能抽出了共同的特徵，或是兩筆資料根本來自同個 dataset，才會有如此精準的分類。



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）

Reference: <https://medium.com/randomai/ensemble-and-store-models-in-keras-2-x-b881a6d7693f>

本次實作 ensemble 的方法與 hw3 繳交之作法相同，使用多個 model output 取平均的方式，模組使用 `keras.layers.average`。加入這個 layer 後，會將這些 model 的 output 作平均，得到最後的輸出。由於不同的 model 在訓練時看到的 feature 可能會不同，也容易產生 overfitting 的問題，因此將 output 取平均有助於降低 overfit 帶來的效果，也讓 model 更加 generalize。

在實作 hw3 時，我分別使用不同參數與資料集訓練了 6 個 CNN 的 model，其中在 kaggle 上最高得到 0.67372 (public) / 0.66146 (private) 的分數，連 strong baseline 都沒有通過。經過 ensemble 之後，最高得到 0.71301 (public) / 0.70771(private) 的分數。由此可見使用這個方法具有將許多較弱的分類器集成一個強分類器的效果。