

Homework 4 Report - Unsupervised Learning & Dimension Reduction

學號: R06942018, 姓名: 何適楷, 系級: 電信碩一

1 PCA of colored faces

(Collaborators : 陳致維 b04901165)

1.1

(.5%) 請畫出所有臉的平均。

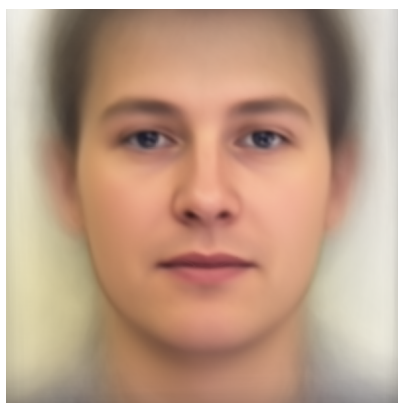
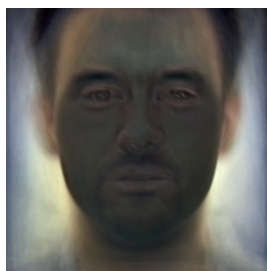


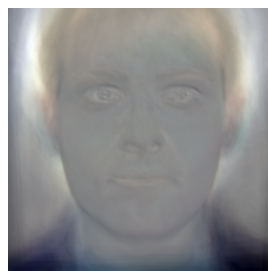
Figure 1: mean face

1.2

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



(a) first eigenface



(b) second eigenface



(a) third eigenface (b) second eigenface

Figure 2: 前四大 eigenface

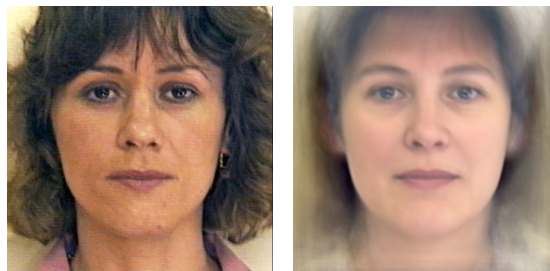
1.3

(.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。



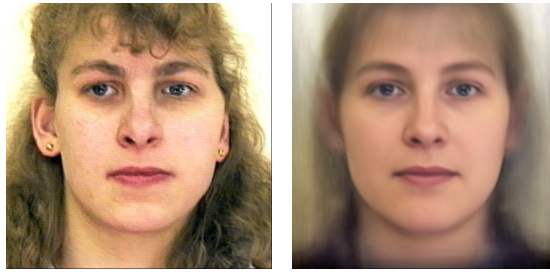
(a) original image (b) reconstructed image

Figure 3: 0.jpg 的 reconstruction



(a) original image (b) reconstructed image

Figure 4: 4.jpg 的 reconstruction



(a) original image (b) reconstructed image

Figure 5: 172.jpg 的 reconstruction



(a) original image (b) reconstructed image

Figure 6: 375.jpg 的 reconstruction

1.4

(.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。

Eigenfaces	比重
First	4.1%
Second	2.9%
Third	2.4%
Fourth	2.2%

2 Image clustering

(Collaborators : 陳致維 b04901165)

2.1

(.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)

	public accuracy	private accuracy
PCA	0.51666	0.51742
autoencoder	0.98392	0.98355

我使用的這兩種方法都是把原始資料投影到 32 維，第一種方法是利用 PCA，第二種方法是用 autoencoder，架構如下圖，從結果可以得知 autoencoder 在這個作業上，效能遠好於 PCA，可能是因為這個 data set 在高維空間具有高度的非線性，所以線性的 PCA 很難去表達，導致判斷率很低，然而 autoencoder 可以學出非線性的 function，故判率較高。

autoencoder Architecture		#
Layer (type)	Output Shape	Param #
Dense	128	100480
Dense	64	8256
Dense	32	2080
Dense	64	2112
Dense	128	8320
Dense	784	101136
Total params: 222,384		
optimizer: Adam		
Learning Rate: 5×10^{-4}		
epochs: 50(early stopping)		
batch size: 256		

2.2

(.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。我使用 TSNE 來畫出利用 autoencoder 降維所產生的 32 維空間中的主要兩維

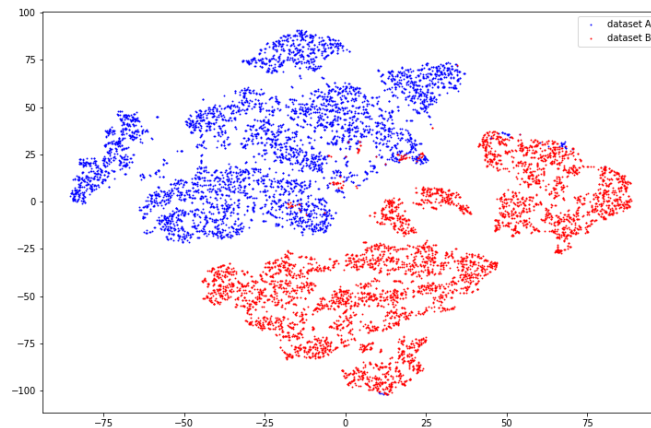


Figure 7: My prediction

2.3

(.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。

可以觀察到，用預測的 label 在藍點群中會混入一些紅點，label data 中藍點中幾乎都沒有紅

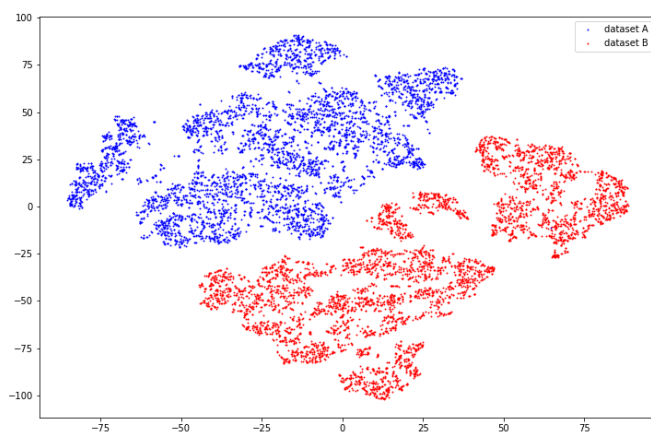


Figure 8: label data

點，而肉眼可以明顯發現，紅藍點之間有明顯的鴻溝，這代表這個 model 還有些點無法分辨開來，可能是這幾個點利用 autoencoder 在高維空間反而比較接近另一種 class 的緣故。

3 Ensemble Learning

(Collaborators : 陳致維 b04901165) (1.5%) 請在 hw3 的 task 上實作 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)

我使用的方法是 Ensemble 中的 Boosting，Boosting 的精神在於只要在 training set 上能得到低於 50%error rate 的 model，就可以藉由改變 weights 的方式提高在 training set 上的正確率，因為 Boosting 的精神在於比較弱的 model 可以組出比較好的 model，所以我設計了一個比較小的 model(如下頁表)。

以下是我得到的結果：

	train accuracy	public accuracy	private accuracy
original model	0.78330	0.57954	0.58372
4 adaboost model	0.84873	0.58038	0.58846

從表中可以看到，經過 4 個 adaboost 的 model 結果在 training set 上得到的表現如預期的比原本的 model 好。public 和 private 的結果也比較好 (不過 Boosting 並沒有保證這點，所以不多做討論)。

Layer (type)	Output Shape	Param #
ZeroPadding2D	(50, 50, 1)	0
Conv2D	(48, 48, 28)	320
MaxPooling2D	(24, 24, 28)	0
Conv2D	(24, 24, 56)	18496
Conv2D	(24, 24, 56)	36928
MaxPooling2	(12, 12, 65)	0
Conv2D	(12, 12, 108)	73856
Conv2D	(12, 12, 108)	147584
MaxPooling2	(6, 6, 108)	0
Flatten	(4608)	0
Dense	(169)	903364
Dropout(0.5)	(169)	0
Dense	(169)	38612
Dropout(0.5)	(169)	0
Dense	(7)	1379
Total params: 889,513		
optimizer: Adam		
Learning Rate: 4×10^{-4}		
epochs: 20		
batch size: 256		