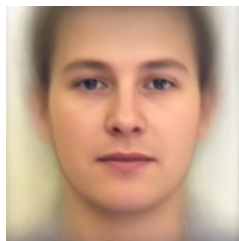


Machine Learning HW7 Report

學號：B05901040 系級：電機三 姓名：蔡松達

1. PCA of color faces:






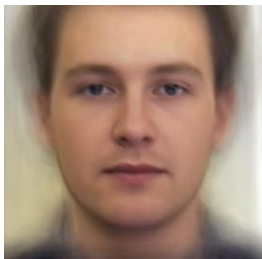
a. 請畫出所有臉的平均。


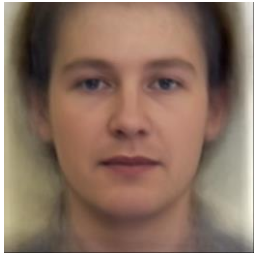




b. 請畫出前五個 Eigenfaces, 也就是對應到前五大 Eigenvalues 的 Eigenvectors。
(繪於下方 d 小題處)






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。

基本上從結果可以發現雖然 reconstruct 出來的臉頗為相似，但是若仔細從頭髮、嘴角、眼神、衣服的部分觀察就可以發現五張照片的差異，表示單純透過 PCA 的概念降維再進行重建雖然不能 100% 完美，卻已經能夠得到較為相像的結果。

	Original	Reconstruction
10.jpg		
22.jpg		
34.jpg		

40.jpg		
160.jpg		

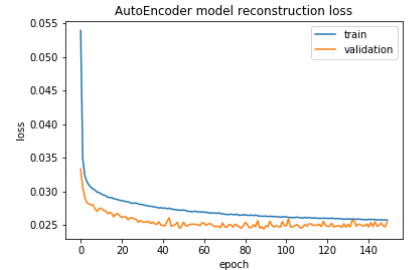
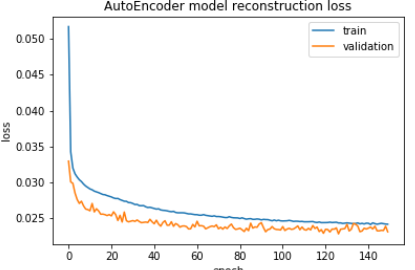
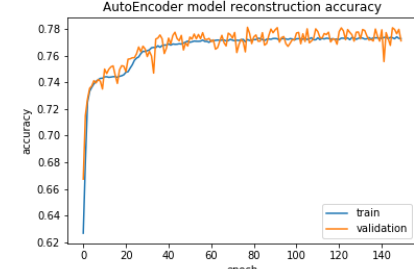
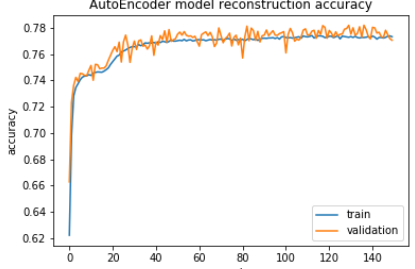
- d. 請寫出前五大 Eigenfaces 各自所佔的比重，請用百分比表示並四捨五入到小數點後一位

	eigenfaces	比重
第一大 eigenvalue		4.1%
第二大 eigenvalue		2.9%
第三大 eigenvalue		2.4%
第四大 eigenvalue		2.2%
第五大 eigenvalue		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)

此處使用兩種不同的降維方法，第一種是只使用三層 Conv2D 的 Autoencoder 進行降維，第二種是使用四層 Conv2D 的 Autoencoder 進行降維，reconstruction loss 使用 mse 作為計算方式，cluster 方式皆使用 Kmeans。(所有圖片皆經過 normalize，normalize 方式為將所有 pixel 除以 255，而 training data 為原始圖片經過 data augmentation 的圖片，validation 取原始圖片)

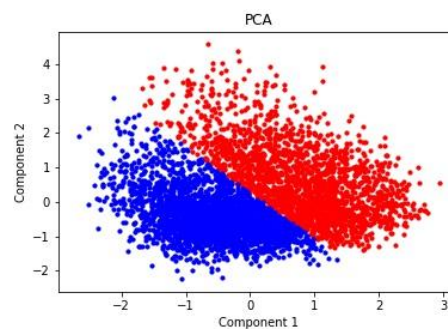
	方法一	方法二
架構	參考 c 小題之架構，只差在少放 conv4，conv3 層直接接到 flatten(1024)，再接到 Dense(10)，decoder 則直接接回 1024，Reshape 後就接回 deconv3 層。	與 c 小題之架構完全相同
reconstruction loss		
reconstruction accuracy		
對於原始圖片 reconstruction loss (mse) / accuracy	0.0258 / 0.7711	0.0242 / 0.7751

由結果可以得知若使用較深的架構可以得到較好的重建結果，而且在最後進行得到的 kaggle 結果方法一為 0.928，方法二為 0.967，可得知運用較深的架構或許所損失的資訊較少較能夠表達出原始資訊。

- 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 feature 的前兩維)

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。

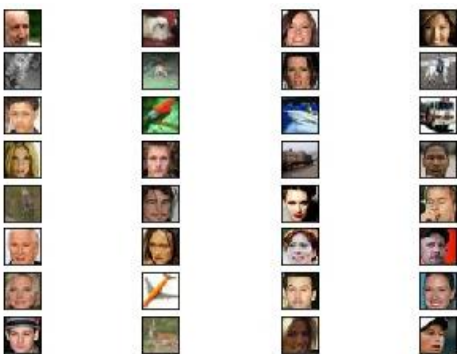
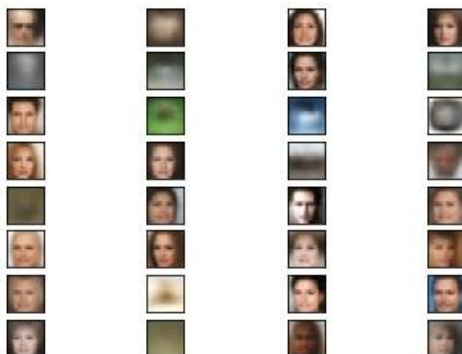
首先先對於檔案中的值進行 normalize，接著使用 PCA 將原本架構 (c 小題) 中的 10 維使用 PCA 降至二維，再使用 Kmeans 進行分類，下圖藍色與紅色點分屬預測結果不同的兩類，可以發現透過 PCA 分析幾乎能夠將兩類的 data 分開的頗清楚，只要從左上往右下畫線做區隔幾乎就能夠得到不錯的分類效果。而下表列出分群結果，加總而言總共有 2937 張圖片被分為同一類，另一類則有 2063 張圖，分類完成後有 12.5% 的圖片預測的結果與原先所屬的 label 不同，而有 87.5% 則成功預測，說明此 model 有滿大的機會能將其分到正確的類別，不過也還是有進步的空間，由於此處是直接將維度由 10 維降到 2 維，或許是因為丟失了不少資訊，而造成預測結果上還有一些進步空間，這個部分只要不要一次用 PCA 降太多維度應該就能達到更好的效果。



	預測為 label 0 (藍色點)	預測為 label 1 (紅色點)
前 2500 張圖片	2406 (96.24%)	94 (3.76%)
後 2500 張圖片	1969 (78.76%)	531 (21.24%)

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。

架構如下圖，基本上就是對於原始圖片做 convolution 4 次，然後進行 flatten 再接到 output 層(512→10)為 encoder，decoder 則是先接回原先的維度(10→512)，reshape 後再做 Convolution 的 transpose 4 次得到重建之後的結果。(Loss function: 原圖片和新圖片 pixel 間的 mse, batch size=100, epochs=150, 有使用 normalization 以及 data augmentation 對 training data 做轉換)

Encoder 架構			Decoder 架構		
Layer (type)	Output Shape	Param #	dense_1 (Dense)	(None, 512)	5632
input_1 (InputLayer)	(None, 32, 32, 3)	0	reshape_1 (Reshape)	(None, 2, 2, 128)	0
conv1 (Conv2D)	(None, 16, 16, 16)	1216	deconv4 (Conv2DTranspose)	(None, 4, 4, 64)	73792
conv2 (Conv2D)	(None, 8, 8, 32)	12832	deconv3 (Conv2DTranspose)	(None, 8, 8, 32)	18464
conv3 (Conv2D)	(None, 4, 4, 64)	18496	deconv2 (Conv2DTranspose)	(None, 16, 16, 16)	12816
conv4 (Conv2D)	(None, 2, 2, 128)	73856	deconv1 (Conv2DTranspose)	(None, 32, 32, 3)	1203
flatten_1 (Flatten)	(None, 512)	0	=====		
embedding (Dense)	(None, 10)	5130	kernel size		
kernel size			(照 deconv1~deconv4 順序): 5, 5, 3, 3		
(照 conv1~conv4 順序): 5, 5, 3, 3			strides: 2*2		
strides: 2*2			padding: same		
padding: same			reconstruct 結果		
原圖片					
					

由上圖可知原圖片中是人臉的圖片經過 reconstruct 之後絕大多數都能得到較好的重建效果，雖然臉的樣子不見得完全相同，但是皆能夠表達出人臉的樣貌，但如果不是人臉的圖片則幾乎無法重建出樣貌，絕大多數呈現模糊的結果，因此也可推知想要進行分類時，大致就分為重建出較顯著的人臉樣貌、重建出模糊的圖片兩類；此外之所以部分圖片重建效果較差猜測可能是因為降的維度較低(10)，損失資訊較多所導致。