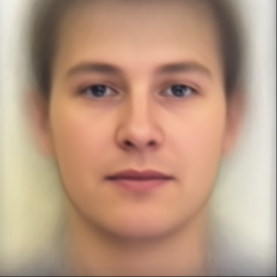
學號：R06725005 系級： 資管碩一 姓名：郝思喬

1. PCA of colored faces
   1. (.5%) 請畫出所有臉的平均。

將全部圖片的RGB分別相加除以平均，得到下圖。



* 1. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

下圖從左到右依序為最大的四個Eigenfaces。



* 1. (.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。

挑選的圖片為1, 101, 201, 301四張，結果為下圖。



* 1. (.5%) 請寫出前四大 Eigenfaces 各自所佔的比重 (explained variance ratio)，請四捨五入到小數點後一位。

前四大Eigenfaces佔的比例為4.1%、2.9%、2.4%、2.2%。

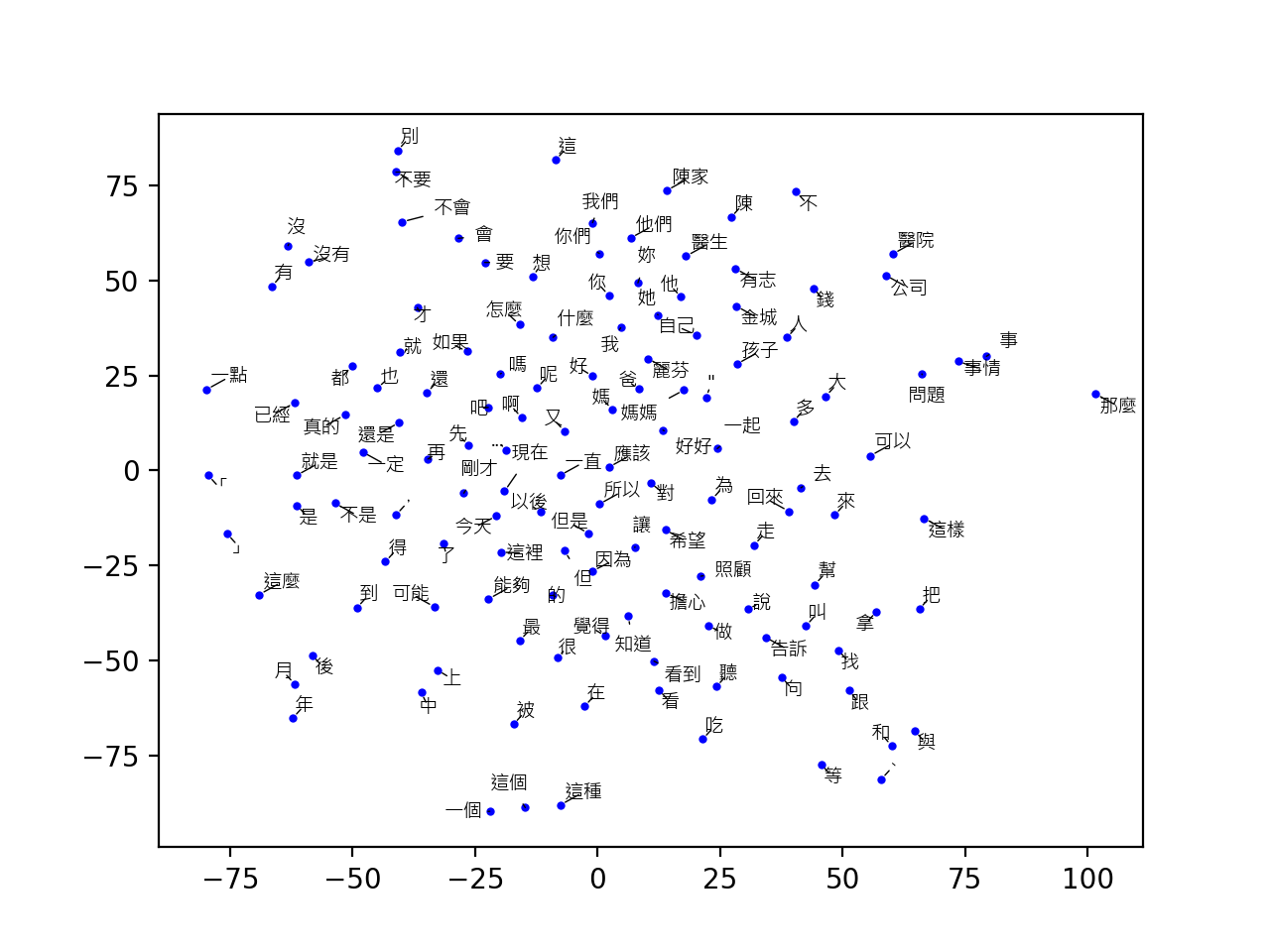
1. Visualization of Chinese word embedding
   1. (.5%) 請說明你用哪一個 word2vec 套件，並針對你有調整的參數說明那個參數的意義。

使用jieba做tokenize之後再用gensim的Word2Vec。

size=200: 表示做出來的每個token的維度有200維。

window=5: 為預設值，表示與目前這個word最大的預測距離。

min\_count=5: 為預設值，表示frequency低於五次的token不計算。

* 1. (.5%) 請在 Report 上放上你 visualization 的結果。
  2. (.5%) 請討論你從 visualization 的結果觀察到什麼。

在右下角的區塊較多動詞，左下方的部分為量詞，正上方的部分為人物的名詞或是代名詞，左方的部分為程度副詞。

1. 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)

我使用的兩種方法為：

1. PCA降維+K-means分群

先利用sklearn的PCA將原本784維的圖再降成784維的feature，設定svd\_solver為randomized、whiten為True，再將結果丟入sklearn的k-means分為兩群。

1. Deep CNN Autoencoder+K-means分群

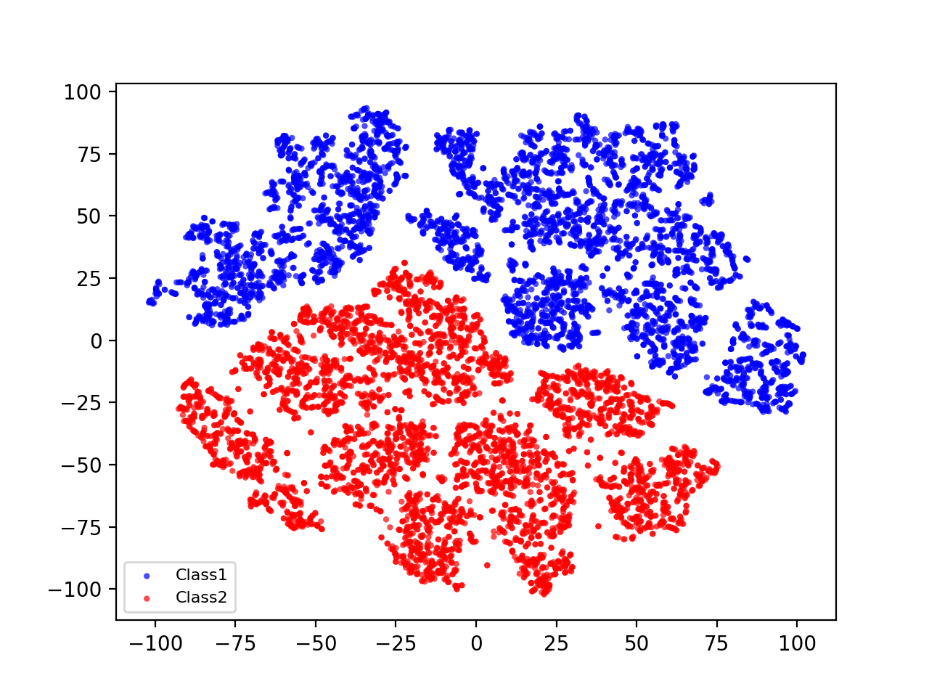
將input接4層Conv2D，除第三層之外使用MaxPooling2D，再接上3層Dense後得到64維的feature。接著再使用sklearn的k-means分為兩群。

兩種方法的兩群個數以及準確率為下：

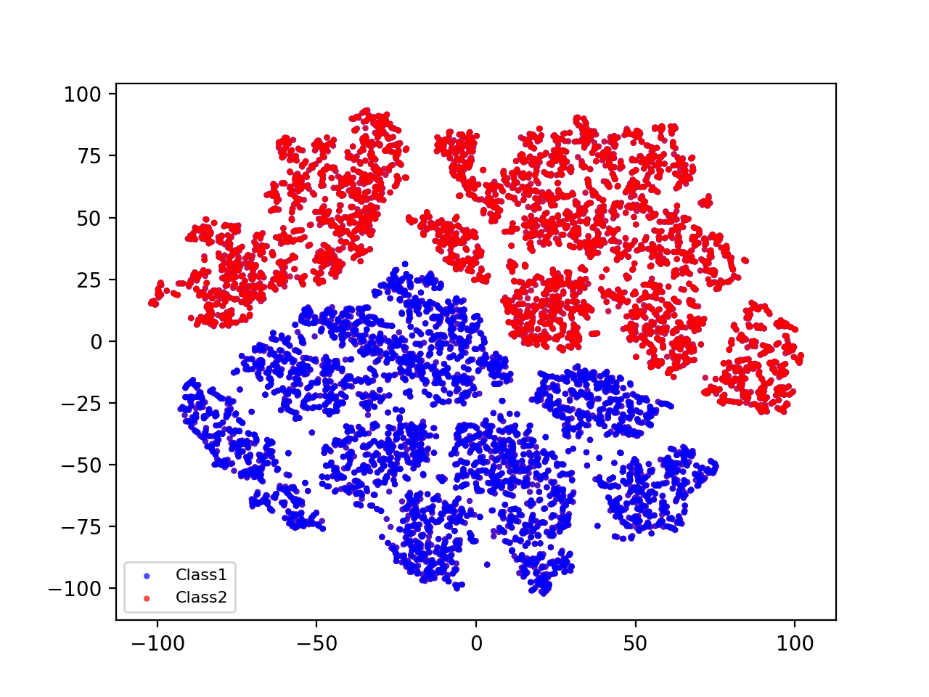
|  |  |  |
| --- | --- | --- |
| 方法 | 兩群數量 | F1 Score |
| PCA | 69994 / 70006 | 0.99165 |
| Deep CNN Autoencoder | 69995 / 70005 | 0.99451 |

從結果來看使用PCA以及autoencoder的方法皆能將feature精準的抽取出來。但是我在實驗PCA的各種維度時，與autoencoder一樣64維的PCA的feature並無法完整表達原本data，預測出來的分群分別為約30000/90000。因此可以得知在將原本data降為程某一特定的小維度時，autoencoder能夠更有效的表達原本data的特性。

* 1. (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。

根據預測的結果，兩個class皆為5000筆資料，畫出來的圖為以下：

* 1. (.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。

根據true label，得到的圖為以下，與自己預測的圖相同：