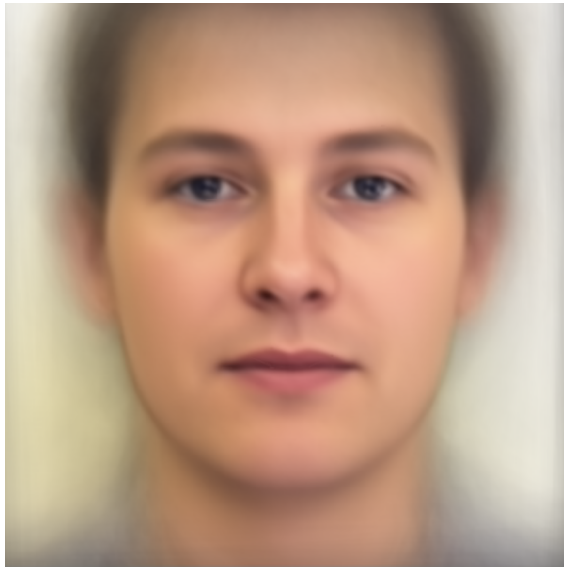


## A. PCA of colored faces

A.1. (.5%) 請畫出所有臉的平均。

答：（參見 p1.py）

將每一張圖片的 pixel 加起來取平均值作圖：



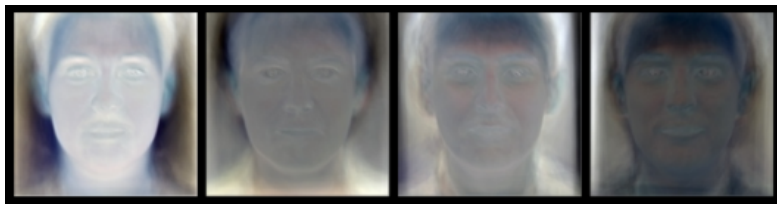
A.2.

A.3. (.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

A.4. 答：按照助教的投影片作 SVD，將 U 的前 4 個 column 作圖：



A.5.



A.6.

(反色)

A.7. (.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.8. 答：按照助教投影片的提示，選擇 0~3.jpg 的圖做 reconstruction



A.9.

A.10. (.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。

A.11. 答：將 SVD 結果的 S 前四個值去計算比例，再取小數點最後一位：

A.12. 0 st eigenvector ratio:  $0.04144624838262957 \approx 4.1\%$

A.13. 1 st eigenvector ratio:  $0.02948732225112061 \approx 2.9\%$

A.14. 2 st eigenvector ratio:  $0.02387711293208415 \approx 2.4\%$

A.15. 3 st eigenvector ratio:  $0.022078415569025393 \approx 2.2\%$

## B. Visualization of Chinese word embedding

(.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。

答：（參見 p2.py）

B.1. 我使用 gensim.Word2vec; 先用 jieba 分詞去掉符號，再用 gensim 轉 vector。

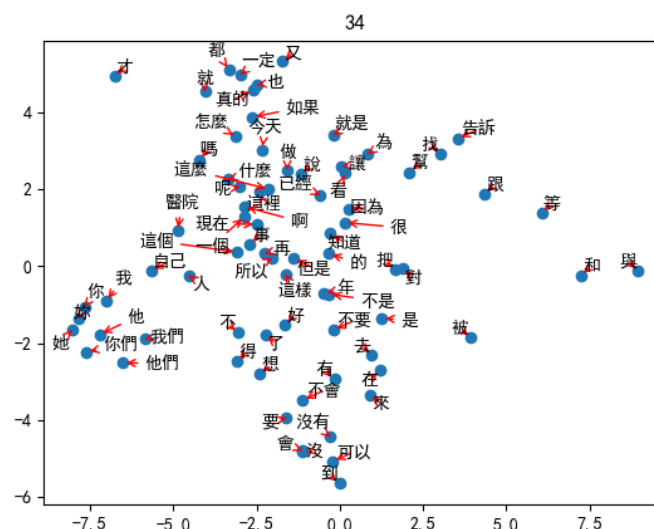
B.2. gensim 參數：

B.3. size=300, embedding 的維度為 300

B.4. windows=5, word2vec 的 window size 我設定為 5.

B.5. (.5%) 請在 Report 上放上你 visualization 的結果。

B.6. 答：我選出出現頻率大於 6000 的詞語，用 SVD(取前兩維)降維到 2 維，如下圖：

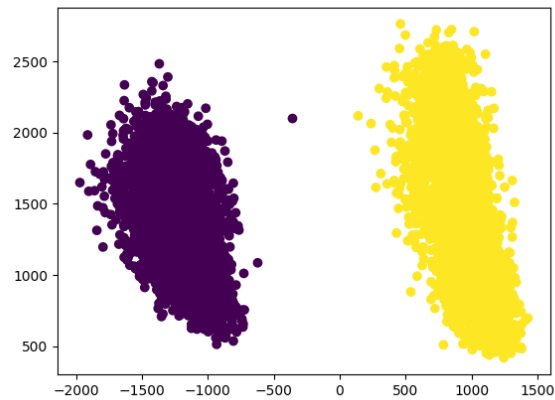


B.7.

- B.8. (.5%) 請討論你從 visualization 的結果觀察到什麼。
- B.9. 從上圖中可以發現，「你」，「我」，「他」三者之間的距離與「你們」，「我們」，「他們」三者之間的距離很類似，都形成了一個三角形，並且主要都聚在一起。而意思接近的詞語，像是「沒有」與「沒」也很接近，「這個」、「這裡」和「現在」接近。

## C. Image clustering

- C.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)
- C.2. 答：
- C.3. 1. 方法一，自己抽取了兩個 features:
- (1) 像素值大於 240 的數量 除以 像素值大於 10 的數量 比例
- C.3.1. (數字的圖中，有顏色的往往比衣服的颜色數字大)
- C.4. (2) 像素值大於 100 的數量 除以 整張圖片像素數量 比例
- C.4.1. (衣服的圖中，有顏色的像素往往比數字的顏色像素多)
- C.5. 抽取 feature 之後再用 kmeans 分兩群。得到 public 0.39710，private 0.39579.
- C.6. 2. 方法二，使用 autoencoder，由 784 維 Dense 分別接 128 維、64 維、32 維，再 Dense 接回 64 維、128 維、784 維，最後用 binary\_crossentropy 陪 adam,經過 800epoch 之後，在用 kmeans 分兩群得到 public 0.69184, private 0.69050.
- C.7. 3. 方法三，使用 autoencoder，由 784 維 Dense 分別接 256 維、128 維、64 維，再 Dense 接回 128 維、256 維、784 維，最後用 binary\_crossentropy 陪 adam,經過 2400epoch 之後，在用 kmeans 分兩群得到 public 0.99940, private 0.99914。
- C.8.
- C.9.
- C.10. (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。
- C.11. 答：我使用 SVD 用方法三(public 0.99940, private 0.99914)，對所有圖片作 encoder 之後，用 SVD 降維到 2 維，在用 Kmeans 的分群結果去上色，如下圖：



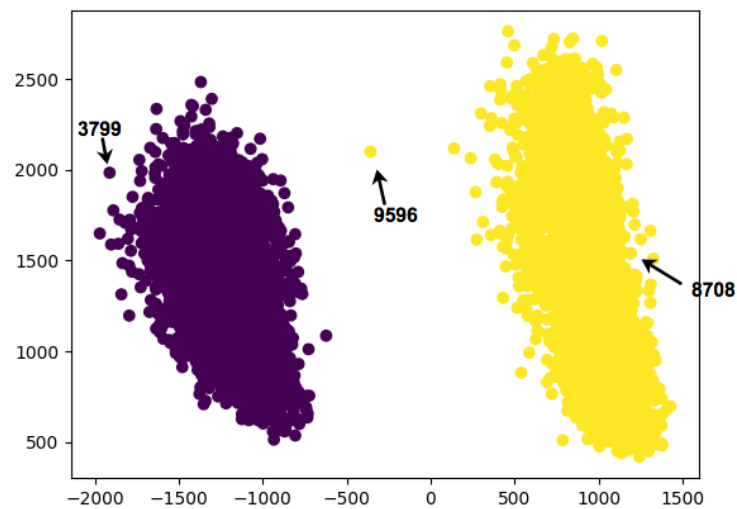
(紫:數字, 黃:衣服)

C.12.

C.13.

C.14. (.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。

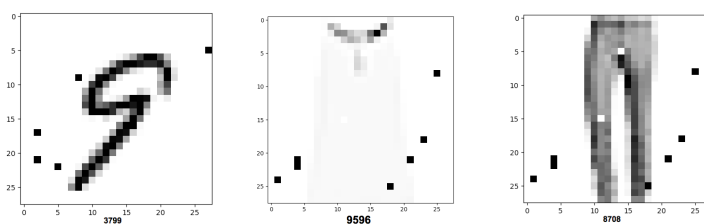
C.15. 答：前 5000 為第一群，後 5000 位第二群，如下圖：



C.16.

C.17. 中間有一個點(第 9596 張圖片)在自己的預測分到了數字，而正確答案是衣服。將第 9596 張圖片畫出來如下圖。

C.18.



C.19.

(反色)