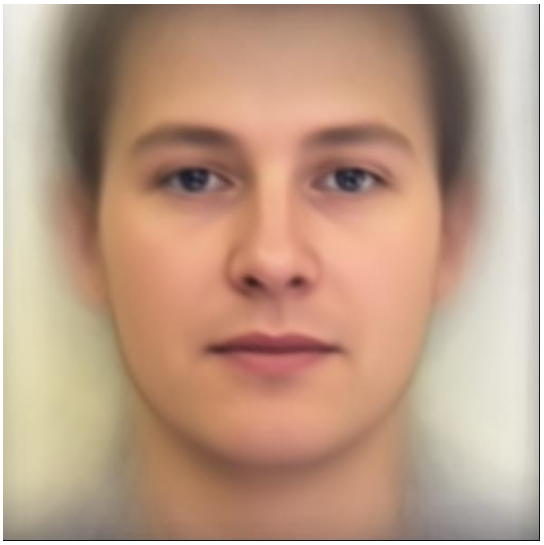
學號：B03901165　系級：電機四　姓名：謝世暐

A. PCA of colored faces

A.1.　(.5%) 請畫出所有臉的平均。(collaborators: b03901165 楊耀程)

ANS: 所有臉的平均:

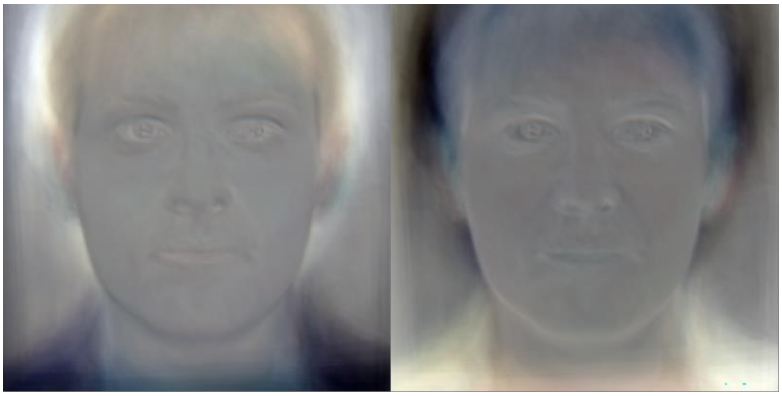


A.2.　(.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。(collaborators: b03901165 楊耀程)

ANS: 由於在決定對應到某個 eigenvalue 的 eigenvector 時，假設跟 PCA 一樣限定 vector 2-norm=1，eigenvector 仍然可以有正負兩種選擇，而畫圖上兩者互為對方的負片，由於不確定要畫哪一種，我將兩種都畫出來。以下是前四個 eigenfaces。

第一個:

第二個:



第三個:

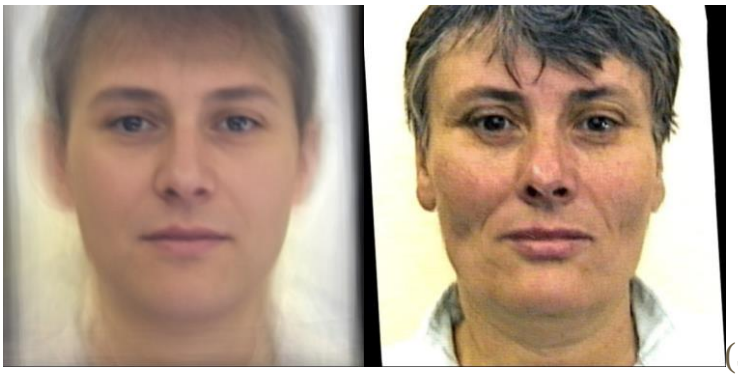
第四個:

A.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。(collaborators: b03901165 楊耀程)

ANS: 以下是隨意拿出四張圖片前四大 Eigenfaces 做 reconstruction 與其原圖的比較:


(4.jpg)


(9.jpg)


(52.jpg)


(120.jpg)

A.4. (.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。(collaborators: b03901165 楊耀程)

ANS:

就 Singular value 來計算:

第一大 eigenface 所占比重: 4.1%
第二大 eigenface 所占比重: 2.9%
第三大 eigenface 所占比重: 2.4%
第四大 eigenface 所占比重: 2.2%
(若以 eigenvalue 計算則為: 21.6%、10.9%、7.2%、6.1%)

B. Visualization of Chinese word embedding

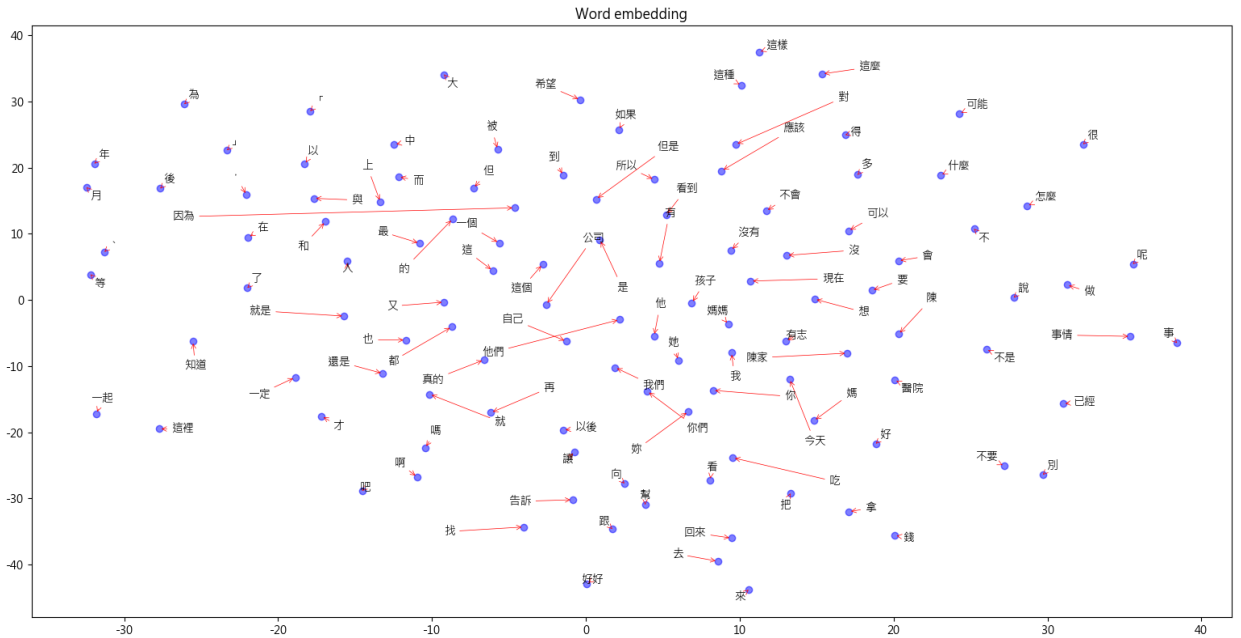
B.1. (.5%) 請說明你用哪一個 word2vec 套件，並針對你有調整的參數說明那個參數的意義。(collaborators: b03901165 楊耀程)

ANS:

我使用 gensim 來做 word embedding，embed 每個單字到 32 維，同投影片作法去掉句子長度<6 的，用剩餘句子來做 training，word2Vec 部分除此之外沒有再去更動其他參數。

B.2. (.5%) 請在 Report 上放上你 visualization 的結果。(collaborators: b03901165 楊耀程)

ANS:

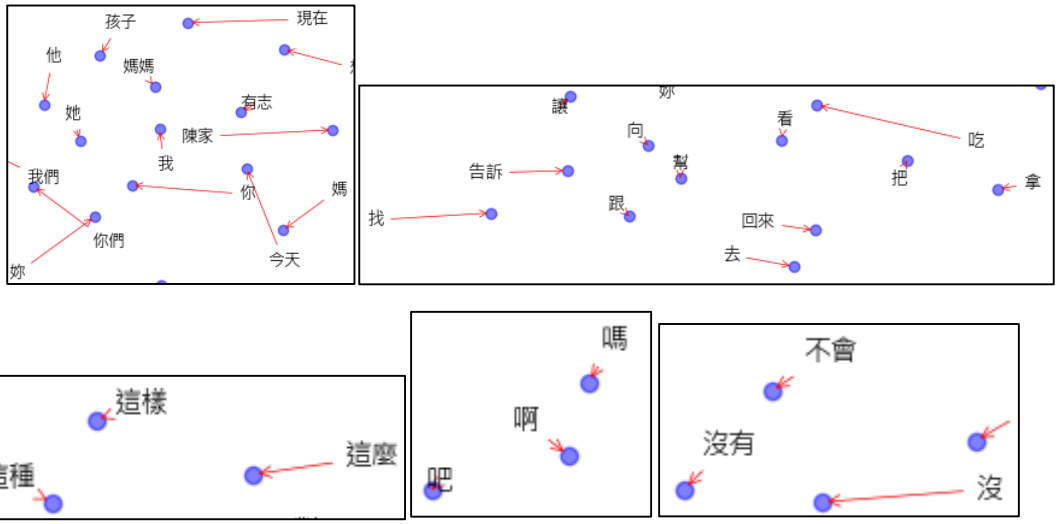


B.3. (.5%) 請討論你從 visualization 的結果觀察到什麼。(collaborators: b03901165 楊耀程)

ANS:

Visualization 只顯示出現次數>=3000 的單字。這些單字中，可以觀察到詞性相近的容易聚在一起，比如下圖中可以看出名詞和名詞

有聚在一起，動詞和動詞有聚在一起。同時，在句子中作用相同的詞，例如語助詞(嗎、啊、吧)，或是類似意思的詞(沒有、沒、不會)，在圖上也容易位在接近的位置。



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)

(collaborators: b03901165 楊耀程)

ANS:

我實作中結果最好的方法是用 DNN 搭一個 auto encoder，hidden layer 維度是 256->48->16->48->256，將影像降到 16 維，再用 Kmeans 分成兩群。訓練過程是一直 train 到能將兩群完全等分為止(70000:70000)。這個方法能在 kaggle 上得到 1.0000 的正確率。

我所實作的另一種方法是嘗試用 PCA 來降維，而分群仍然使用 KMeans。PCA 的部分我將影像同樣降到 16 維，以便和 auto encoder 做比較，然而用 PCA 做出來的結果則相當差，F1 分數僅有 0.03 左右。觀察分群比例，PCA 則是將兩群分成 0.74:0.26 的比例，顯然與第一種方式有著很大的差距。

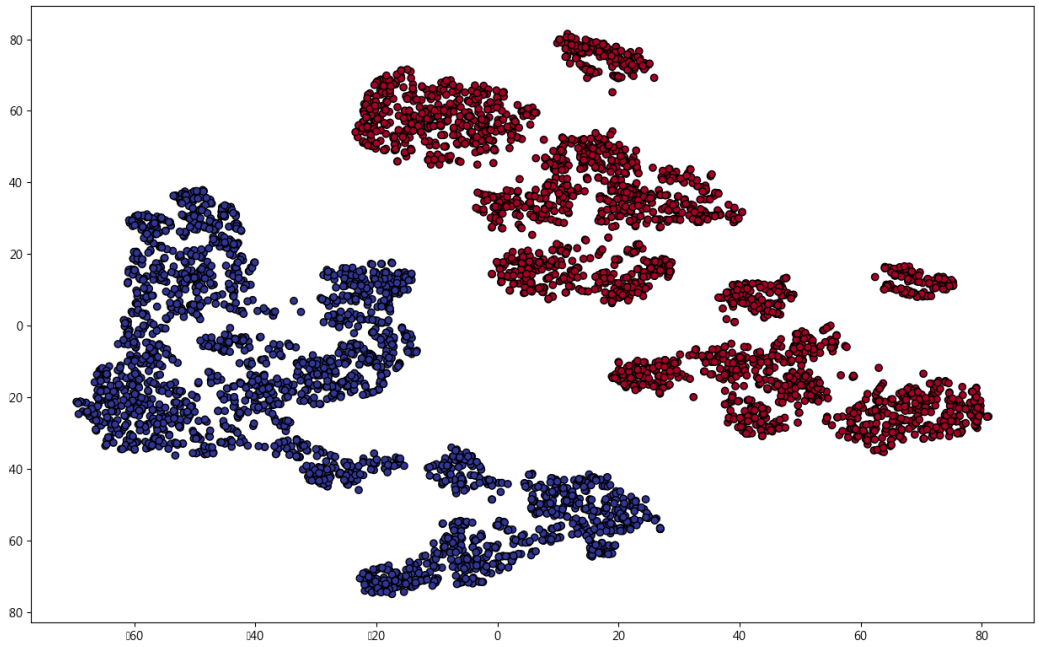
C.2. (.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。(collaborators: b03901165 楊耀程)

ANS:

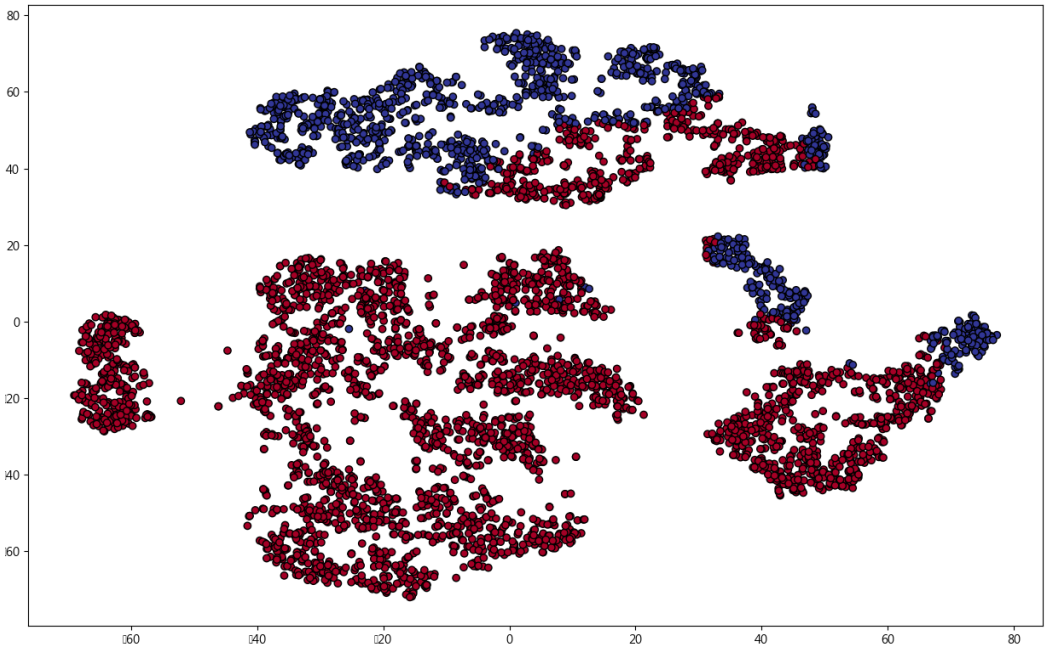
我用 tsne 將 data 從 16 維降到 2 維來視覺化，這邊貼上 auto-encoder 和

PCA 分類的視覺化結果，由於 tsne 挺花時間的，就只對前 5000 組做視覺化了。下兩圖可以看出 AE 的結果經過 tsne 降維後，兩群已經確實地完全分開，但 PCA 結果卻相當糟，顯然沒有分群分得很好。

Auto-encoder:



PCA:

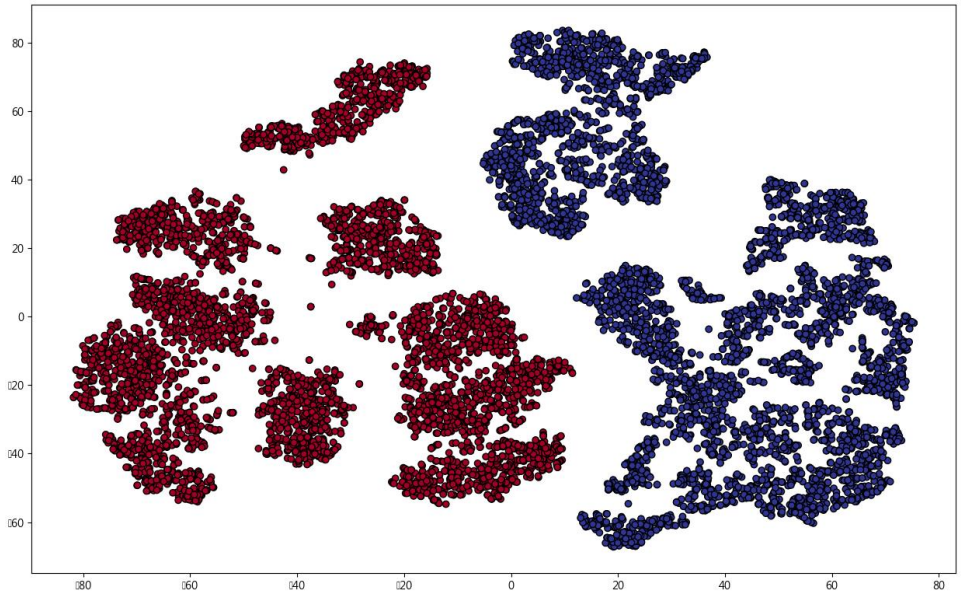


C.3. (.5%) visualization.npy 中前 5000 個 images 跟後 5000 個 images 來自不同 dataset。請根據這個資訊，在二維平面上視覺化 label 的分佈，接著比較和自己預測的 label 之間有何不同。(collaborators:

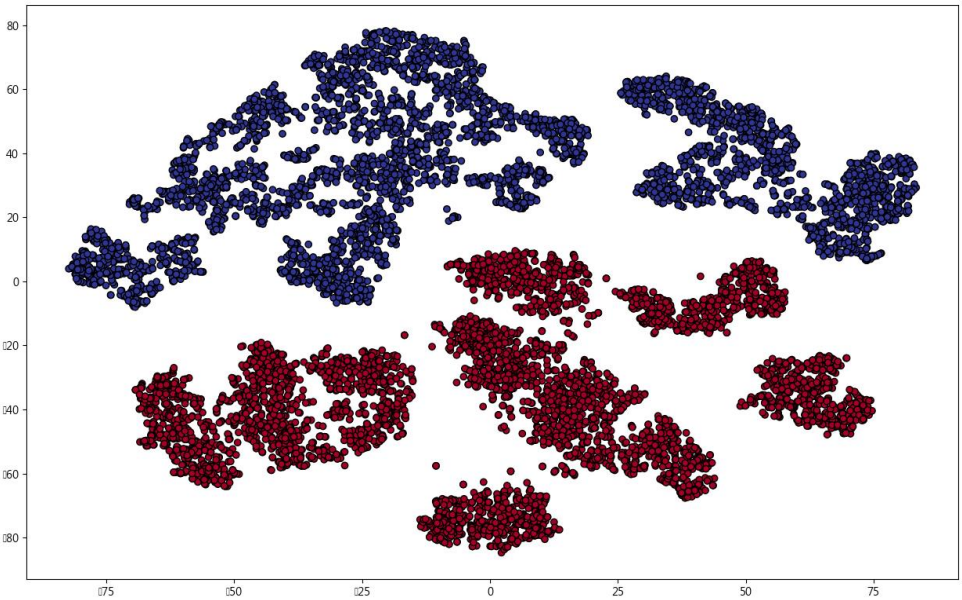
b03901165 楊耀程)

ANS:

下圖是用我的 model 預測的視覺化結果:



下圖則是用 ground truth label 做的視覺化結果:



由於我的 model 有做到 100%的正確率，所以出來結果並沒有什麼不同，可以看的出來確實都有將兩群分開。