

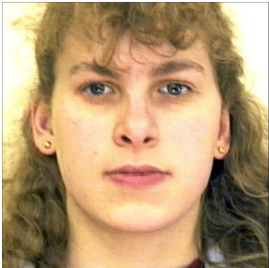

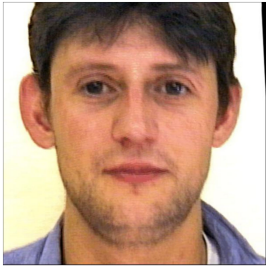

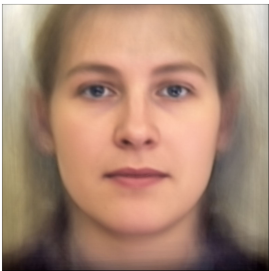
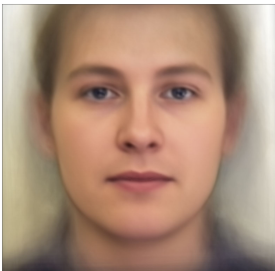
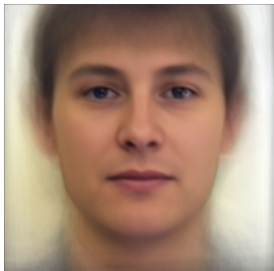
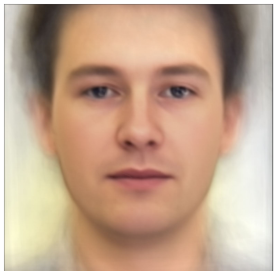
A. PCA of colored faces

A.1. (.5%) 請畫出所有臉的平均。

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

id	0	1	2	3
				

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。

id	168	314	357	404
origin				
reconstruction				

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。

sum of s = 13029227

s值	540384	384472	311315	287858
比重	4.1%	3.0%	2.4%	2.2%

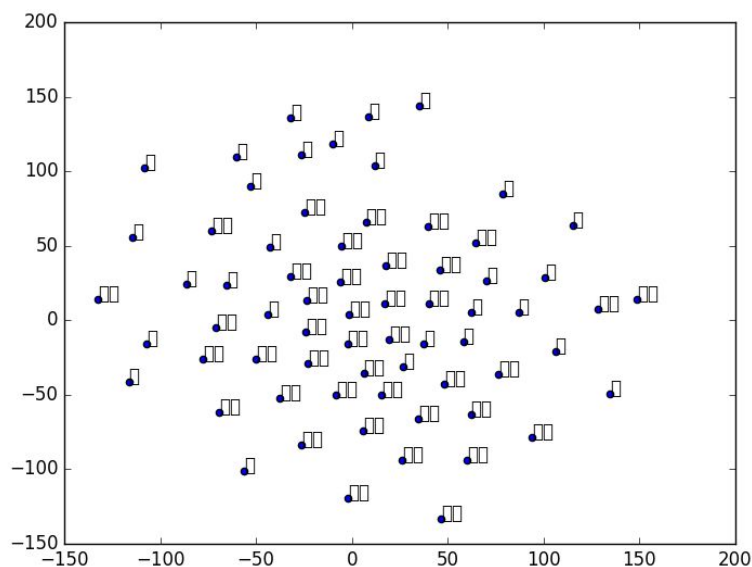
B. Visualization of Chinese word embedding

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。

我使用jieba分詞之後用gensim的word2vec做。word2vec中需要指定訓練輸出的維度數量，我設定為250維度。另外有min_count可以設定，因為語料庫較大，我設定出現頻率10次以下的詞不用裡他。

B.2. (.5%) 請在 Report 上放上你 visualization 的結果。

可惜編碼問題還是沒有辦法解決



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

原本期待可以讓類似的詞的點更加靠近，但由這張圖看來出來的結果意外的平均？

C. Image clustering

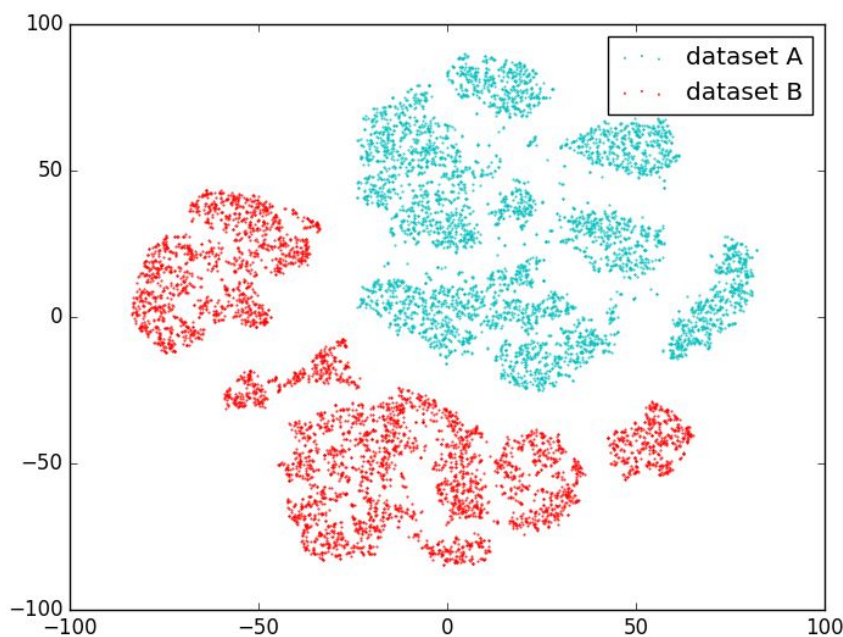
C.1. (.5%) 請比較至少兩種不同的 feature extraction 及其結果。(不同的降維方法或不同的 cluster 方法都可以算是不同的方法)

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 784)	0
dense_1 (Dense)	(None, 512)	401920
dense_2 (Dense)	(None, 32)	16416
dense_3 (Dense)	(None, 512)	16896
dense_4 (Dense)	(None, 784)	402192

DNN模型如上圖。Training的過程當中使用input_1輸入圖片，並取出Dense_4的output，將輸出對原本的圖片運算loss。使用這個model的時候將圖片從input_1輸入，並取出dense_2輸出的32個維度，將每張圖片的32個維度的資料做k-means。經過100個epochs之後可以讓loss降至0.1790左右，透過K-means演算法分類之後上傳至kaggle可以取得0.91左右的成績。我相信再透過微調這個model以及相關參數，可以讓準確率更上層樓。

另外我嘗試過用SVD對圖片降維，再用k-means分類後上傳至kaggle。嘗試過數種不同的降維後的維度數量，可惜上傳至kaggle成績始終沒有突破0.05。後來有將k-means分類後的兩個集合的數量分別寫入log檔案，發現無論設定SVD要將資料降至幾個維度，k-means分類出來的集合都是10000多跟3000多張圖片。推測SVD降維後得到的特徵和人看圖片看到的特徵相比應該有不小的差異。

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



此圖是將原始資料以DNN降至32維之後，以tsne降至2為所繪製而成。

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。

我是使用DNN將圖片的784個像素降維至32個維度，再依據題目使用tsne降維至2個維

度。有點令人意外但又不是那麼奇怪的是，降維後的資訊果真讓同一個dataset的資料都分再一起了。將降至32個維度的資料送入K-means後取得的答案非常漂亮，恰恰好前500筆被分再同一類，後5千筆被分再同一類。另外值得一提的事，若使用tsne降至2維的資料去跑k-means，反而分類出4574筆和5426筆資料。推測因為資訊過少，導致其中一群當中較為分離的資料被歸累到另一群了。