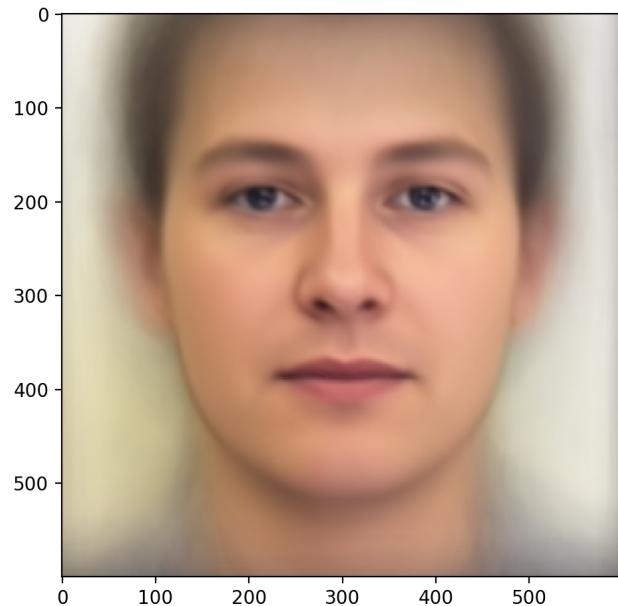
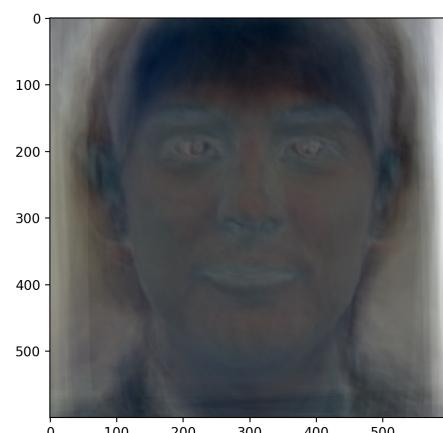
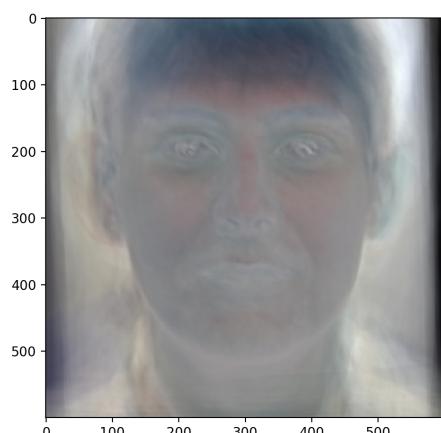
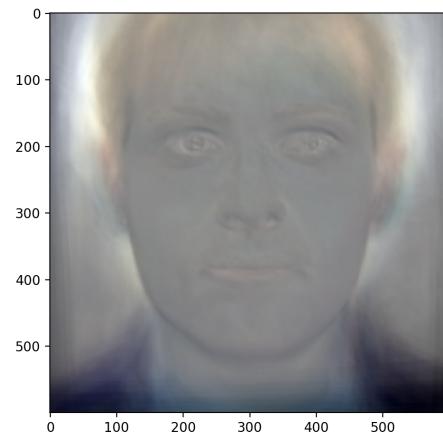
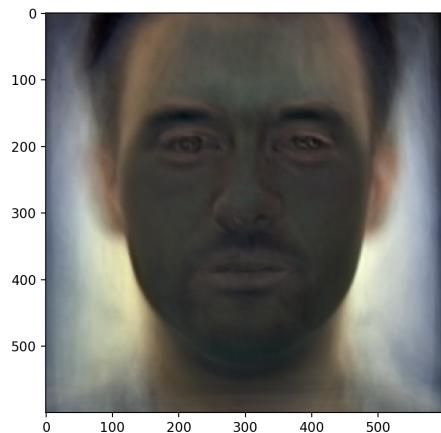


1. PCA of colored faces

1.(.5%) 請畫出所有臉的平均。

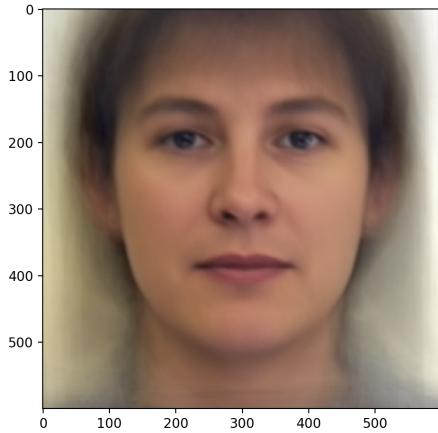


2.(.5%) 請畫出前四個 Eigenfaces，也就是對應到前四大 Eigenvalues 的 Eigenvectors。

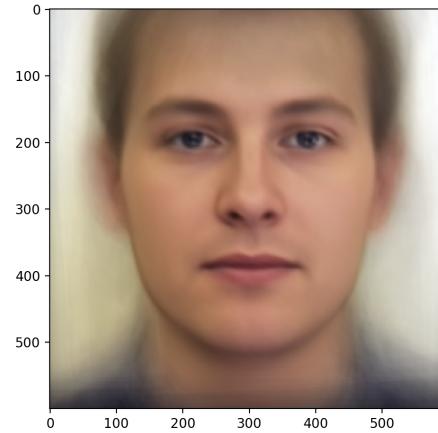


3.(.5%) 請從數據集中挑出任意四個圖片，並用前四大 Eigenfaces 進行 reconstruction，並畫出結果。

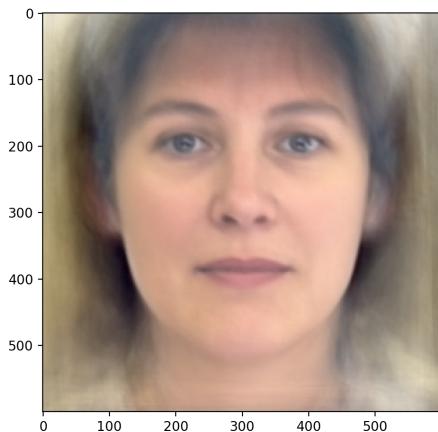
第一張：



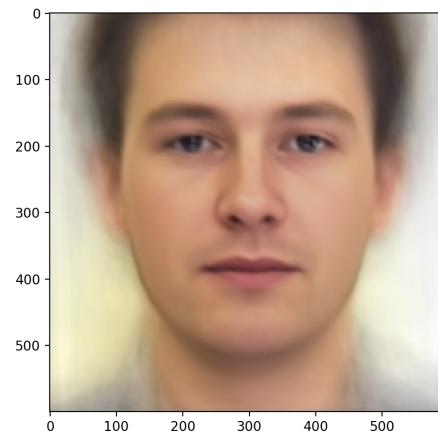
第三張：



第四張：



第二十張：



4.(.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。

第一：4.1%

第二：3.0%

第三：2.4%

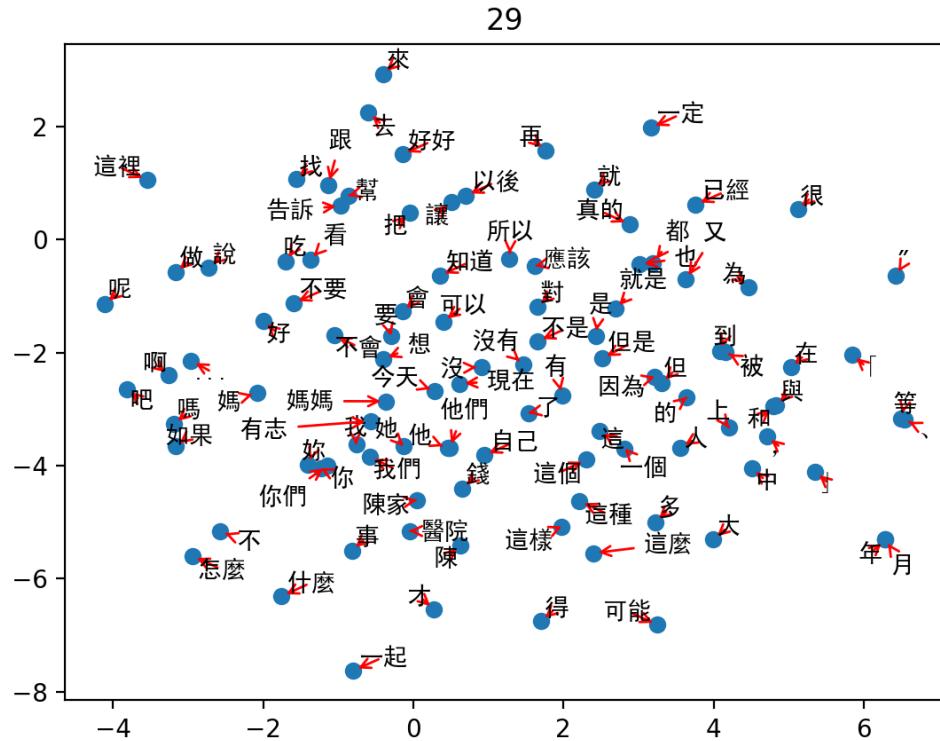
第四：2.2%

2. 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。

我是用gensim來實作word2vec，min_count調為4000，意思是不考慮出現低於4000次的字。而size調為100，意思是output出來的vector維度會有100維。

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



3.(.5%) 請討論你從 visualization 的結果觀察到什麼。

沒有、沒和有三個滿接近的，應該是會出現在句子中類似的地方。另外還有發現右下角年和月非常接近，原因應該是兩者都是在形容時間，在句子中說明差不多的意思。還有你、我、他、妳們、我們、他們，這些常用的主詞都在相近的地方。嗎、吧、阿這些語尾助詞在相近的地方。

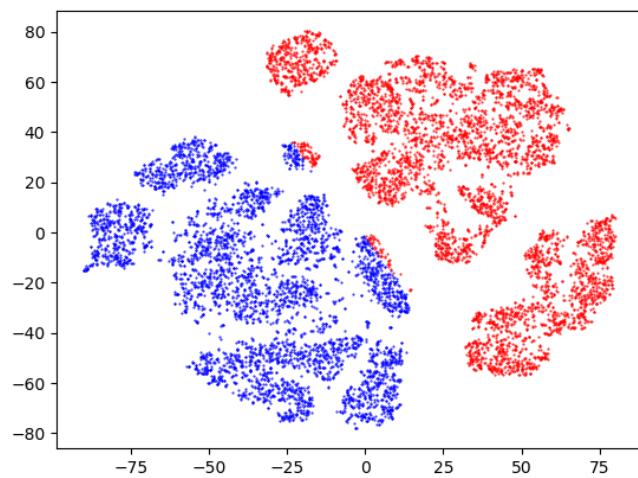
3. Image clustering

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)

	Kaggle public score	Kaggle private score
pca到48維->tsne到2維->kmeans	0.38214	0.38255
deep auto encoder -> kmeans	0.99950	0.99909

使用deep auto encoder的效果比pca和tsne的效果都好上很多，甚至可以做到接近1的正確率，讓我體會了deep的威力。

2.(.5%) 預測 visualization.npy 中的 label，在二維平面上視覺化 label 的分佈。
deep auto encoder到32維後，再用tsne到2維，在做kmeans



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。

預測的結果和實際結果除了中間一些資料外都分類正確。原因是在看tsne投影的結果後，就會發現資料還算分的滿開的，也有大致分為兩群，因此kmeans可以達到不錯的效果。

