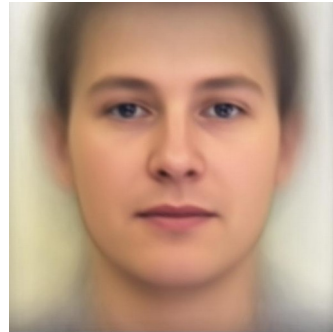


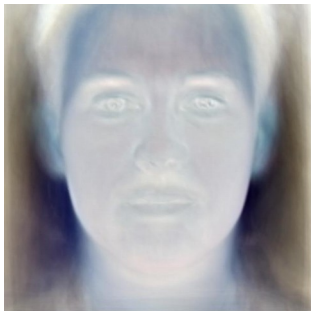
A. PCA of colored faces

A.1. (.5%) 請畫出所有臉的平均。

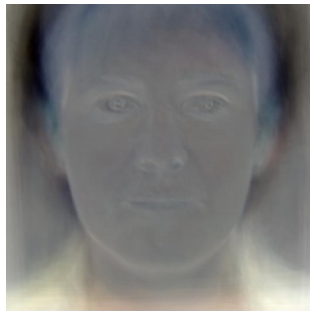


A.2. (.5%) 請畫出前四個 **Eigenfaces**，也就是對應到前四大 **Eigenvalues** 的 **Eigenvectors**。

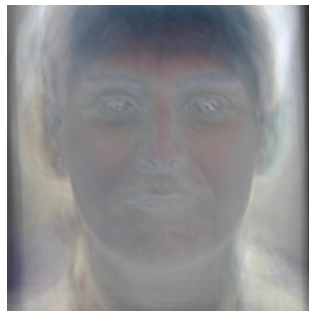
1:



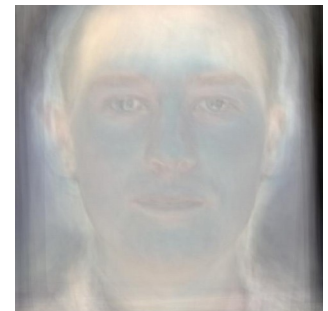
2:



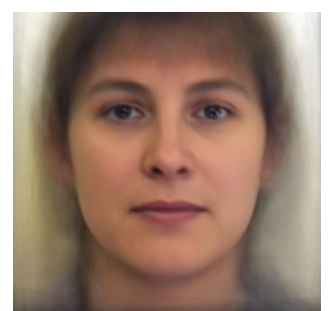
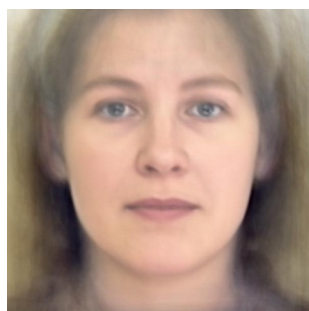
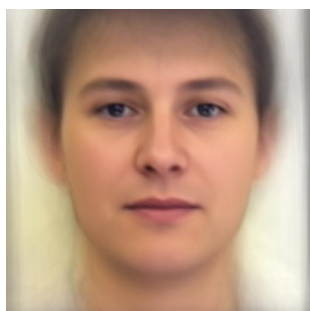
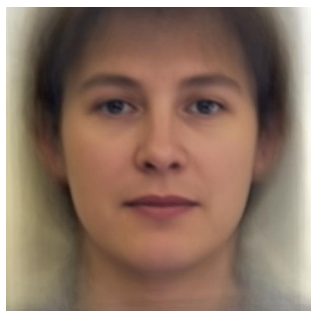
3:



4:



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。



A.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。

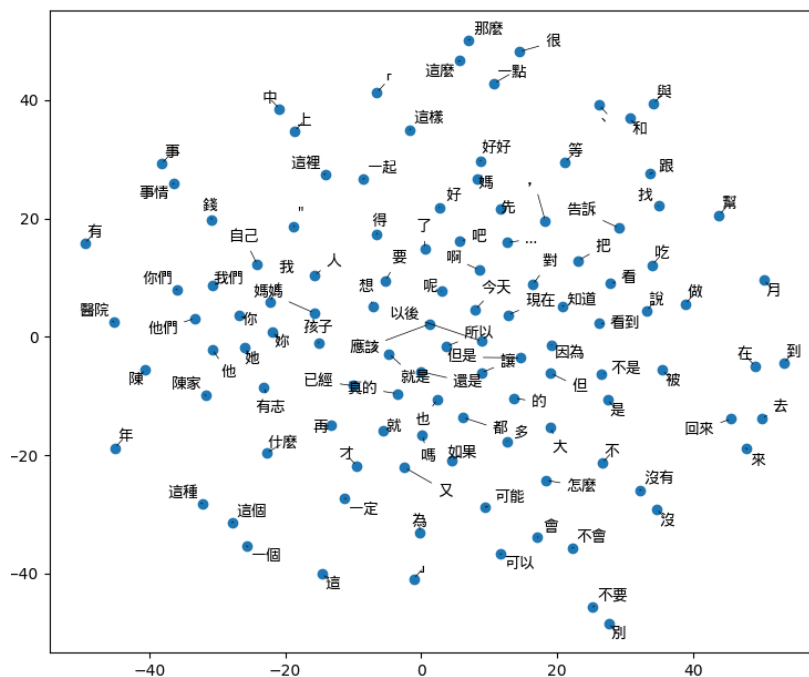
4.1 % , 2.9 % , 2.4 % , 2.2 %

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 gensim 的 word2vec. 只調整 embedding dimension. embedding dimension 即為將 word embed 到高維度空間的 dimension.

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



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

關聯性較大的字會聚集在一起,比方說 "我","我們" ; "你","妳" ; "事","事情" ...等

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)

1: 直接對 **data** 作 **PCA** 降維至 **dim=10** ,然後對其用 **Birch clustering**.

2: **Auto-encoder** 降維至 **dim = 4** ,然後扣掉 **mean** 後用 **Birch clustering**. **Auto-encoder** 架構如下, **training with 10 epoch with learning rate 0.0002, Adam optimizer, L2 regularization 1.0e-10** :

Autoenc (

(encoder): Sequential (

(0): Linear (784 -> 256)

(1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)

(2): ReLU ()

(3): Linear (256 -> 128)

(4): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True)

(5): ReLU ()

(6): Linear (128 -> 64)

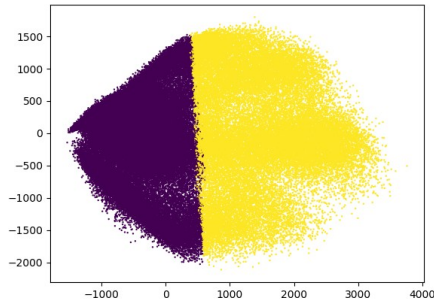
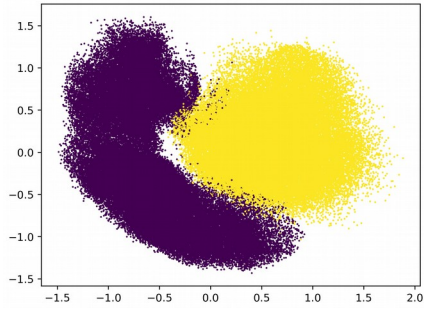
(7): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True)

(8): ReLU ()

```

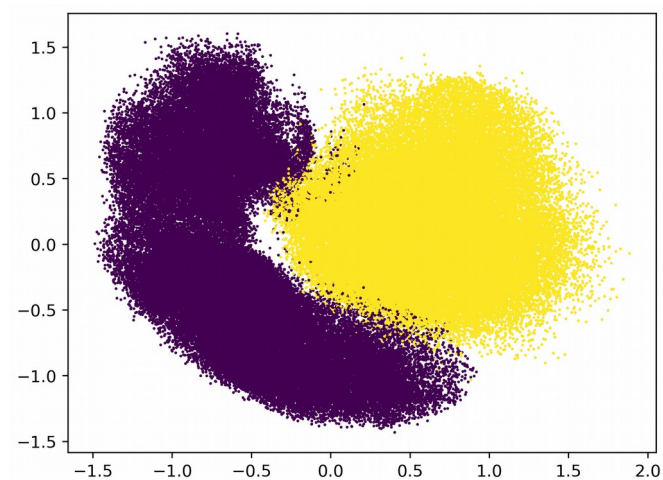
(9): Linear (64 -> 4)
)
(decoder): Sequential (
(0): Linear (4 -> 64)
(1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True)
(2): ReLU ()
(3): Linear (64 -> 128)
(4): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True)
(5): ReLU ()
(6): Linear (128 -> 256)
(7): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True)
(8): ReLU ()
(9): Linear (256 -> 784)
(10): ReLU ()
)
)

```

	PCA+Birch	Autoenc + Birch
Feature visualization (using PCA)		
Kaggle score (public)	0.02372	0.84438

可以看到 Auto-encoder 的分羣較明顯, 使得 clustering 能夠有效分羣, 預測準確度較高

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



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。

如下圖,左邊為我的 model 預測,右邊為正確 label. 可以發現可以正確 label ,除了在兩個 cluster 邊界有少數 data 標錯

