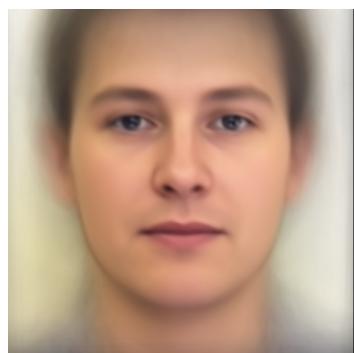


A. PCA of colored faces

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。

左上、右上、左下、右下依序為前 1~4 大的 Eigenfaces



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。

左圖為原圖，右圖為 reconstruct 的結果。



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。

1 : 4.1%

2 : 2.9%

3 : 2.4%

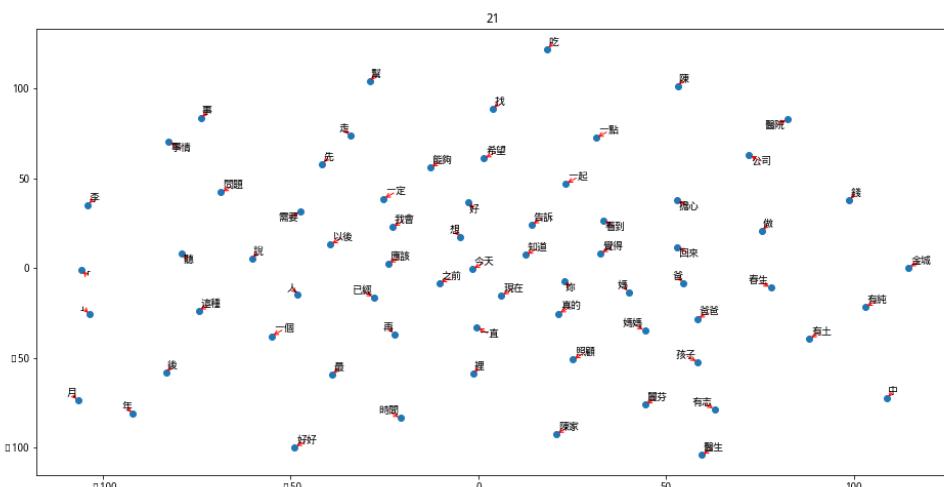
4 : 2.2%

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 套件，調整的參數為
---size = 250，表示訓練出來的詞向量維度為 250 維
--- sg = 0, 1 表示使用 skip-gram, 0 表示使用 CBOW
--- min_count=5，表示不考慮出現次數為 5 以下的詞
其餘皆使用預設值。

- 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。

具有類似關係的詞會有聚在一起的現象，例如右下爸爸、媽媽、孩子聚在一起，「事」和「事情」也在附近，中間的部分「今天」、「之前」、「現在」和時間有關的詞也都群聚在一起

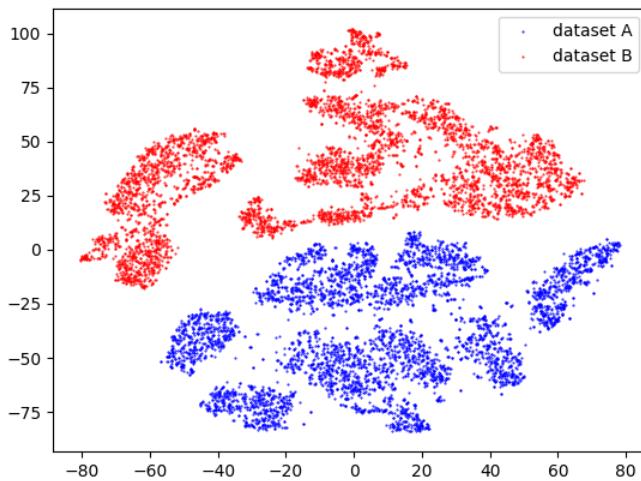
C. Image clustering

- 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)

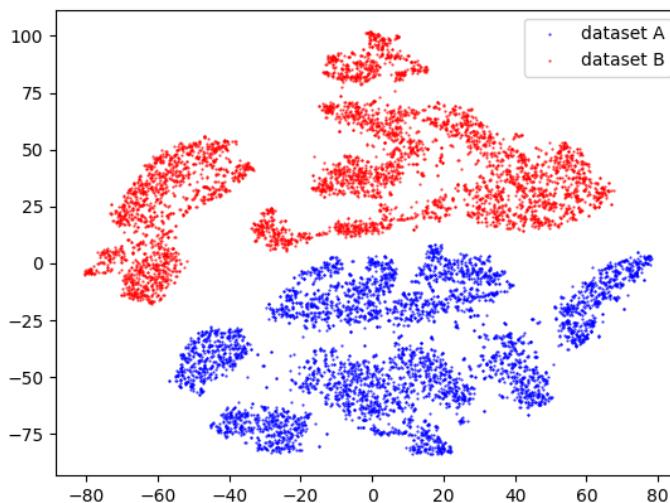
我使用 DNN 的 autoencoder 做 dimensional reduction 到 64 維再用 kmeans 分群，得到的結果在 kaggle 上 f1score 為 1。如果用 CNN 的 autoencoder 做的話 f1score 只有 0.03，有可能是因為 encode 時用了 convolution layer，則 decode 時也應該從 convolution 轉回去，如果用 dense layer 效果就會很差。用 sklearn 的 PCA 降維到 64 再用 kmeans 分群則 f1score 則大約只有

0.04。

- 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。



由於我的 model 預測出來的正確率為 100%，因此畫出來並沒有什麼不同。