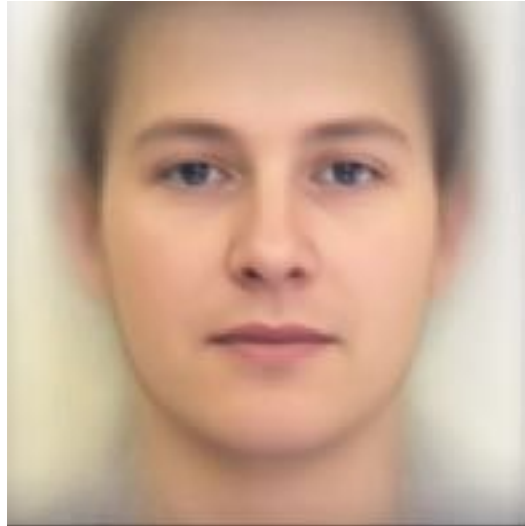
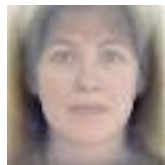
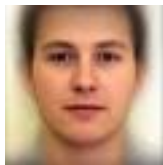


1. PCA of colored faces

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。



eigenface

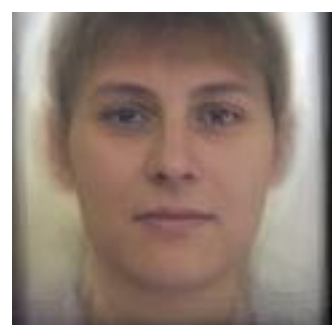


eigenface取負號

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。

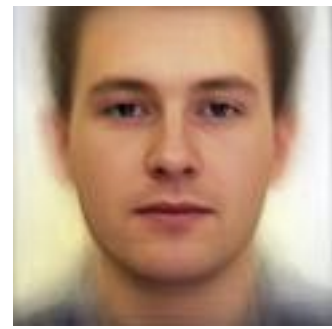


→

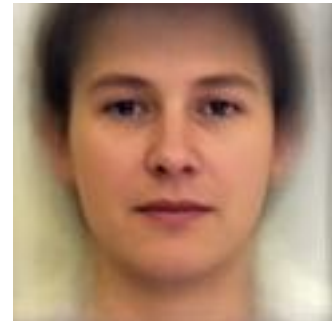




→



→



→



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。
(下列數字是resize size = 60的版本)

四大eigenface對應的s為	比例為(除以s.sum)
# S0 = 396.512383386 →	7.4%
# S1 = 199.906933537 →	3.7%
# S2 = 151.76995039 →	2.8%
# S3 = 123.458163746 →	2.3%

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。

使用了 gensim.models 的 word2vec，

有調整了 window、workers、size、min_count

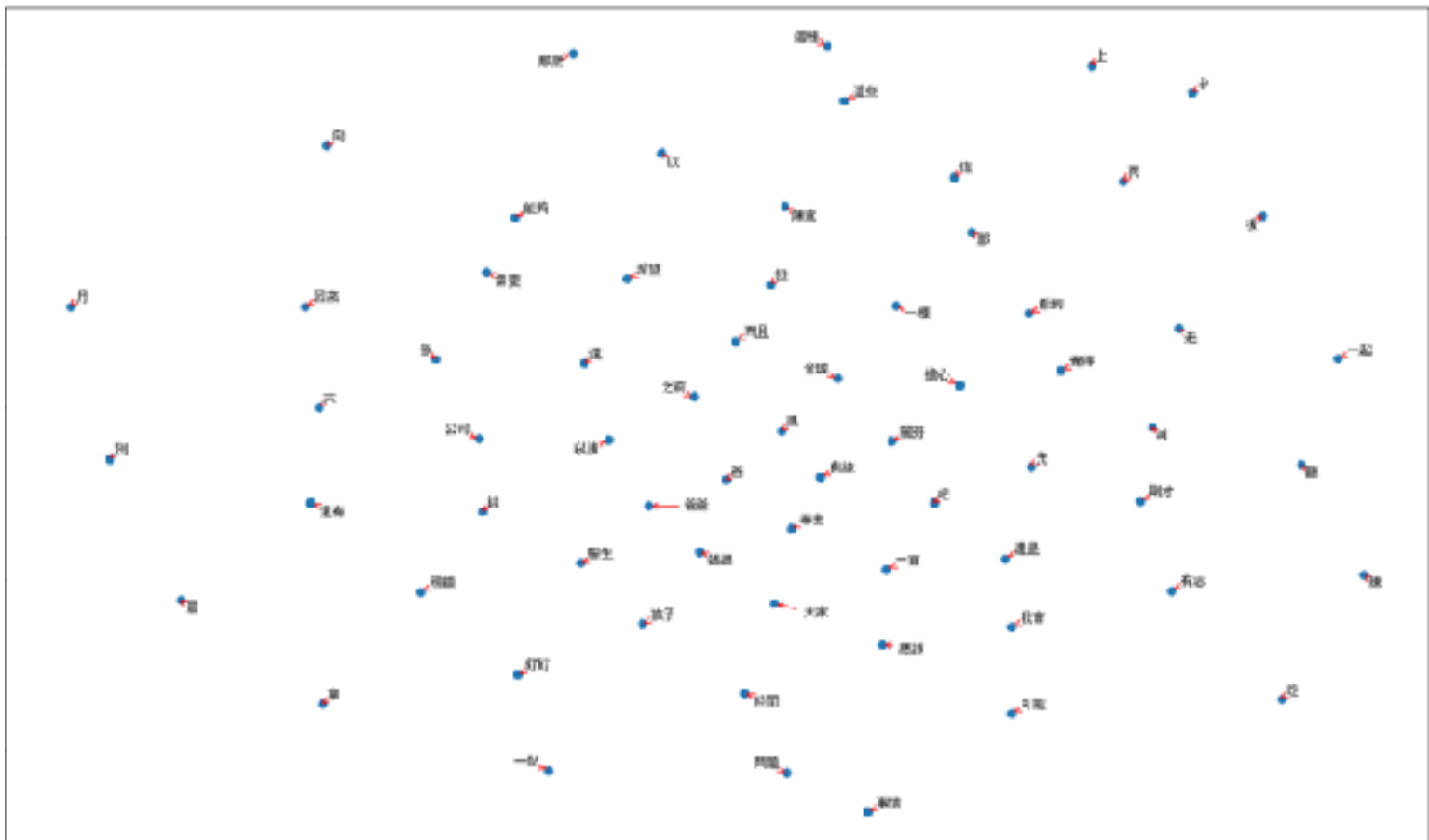
window：能夠看到左右兩邊的範圍有多少

workers：執行緒數量

size：訓練出來的維度

min_count：字如果沒有超過 min_count 時，不考慮訓練

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。
爸、爸爸 之於 媽、媽媽的向量有點平行
其他就找不太到了，另外讓我意外的是人名還蠻多的

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)

我用了兩種降維方式

1. `sklearn.decomposition.PCA(n_components = 10)`

將28*28降到五維，再使用kmeans分群

效果非常非常的差

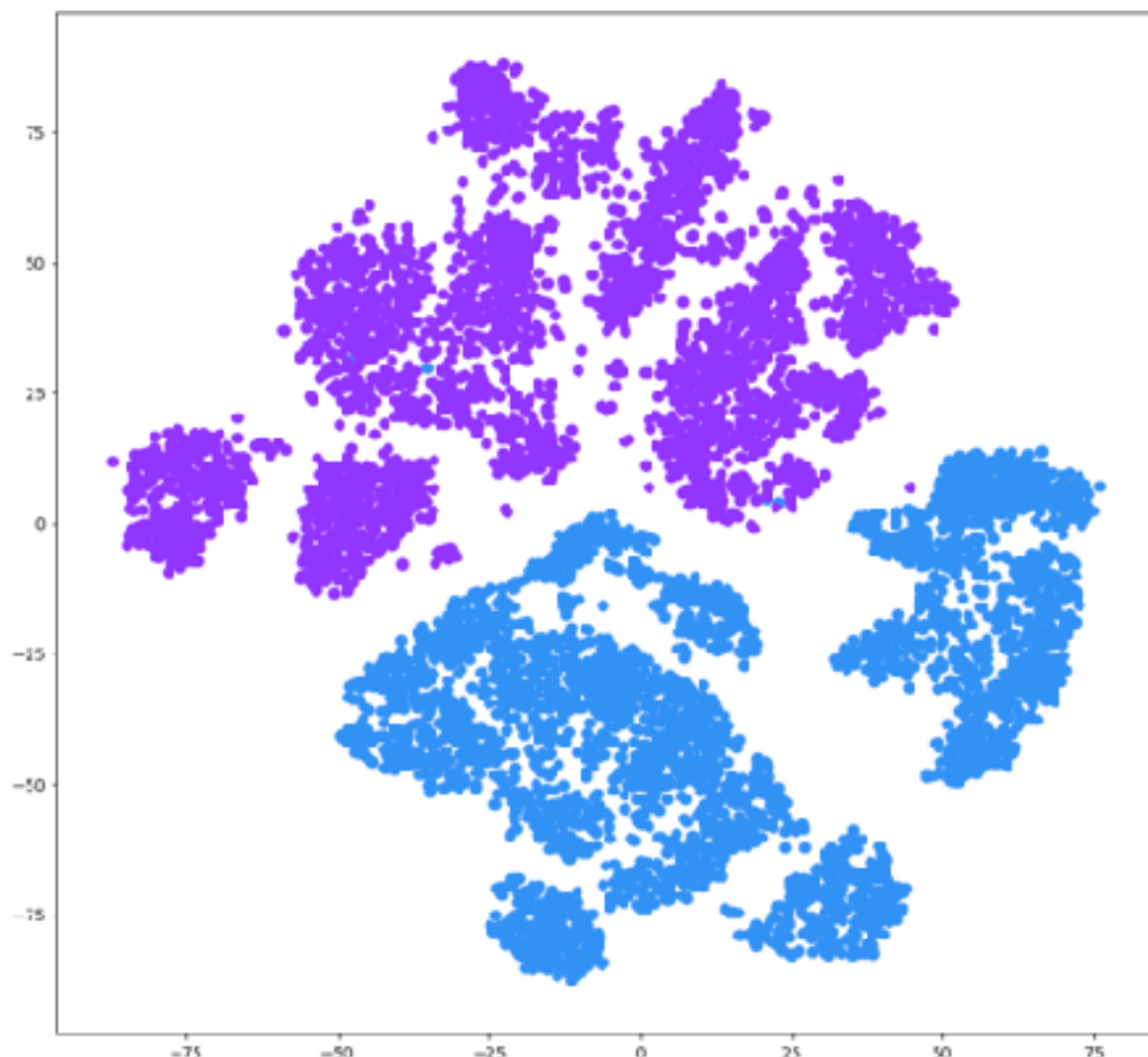
private	public
0.03047	0.03022

private	public
0.91524	0.91993

2. 使用keras的autoencoder降維降到32維，再使用kmeans分群

效果群拔

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



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。

除了零星的幾點會跑到另外一群之外，大部分的點都與自己做的相同

