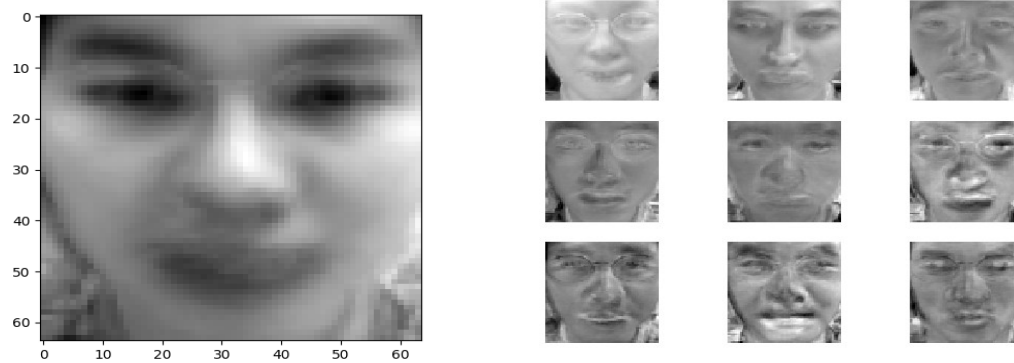


1.1. Dataset 中前 10 個人的前 10 張照片的平均臉和 PCA 得到的前 9 個 eigenfaces:

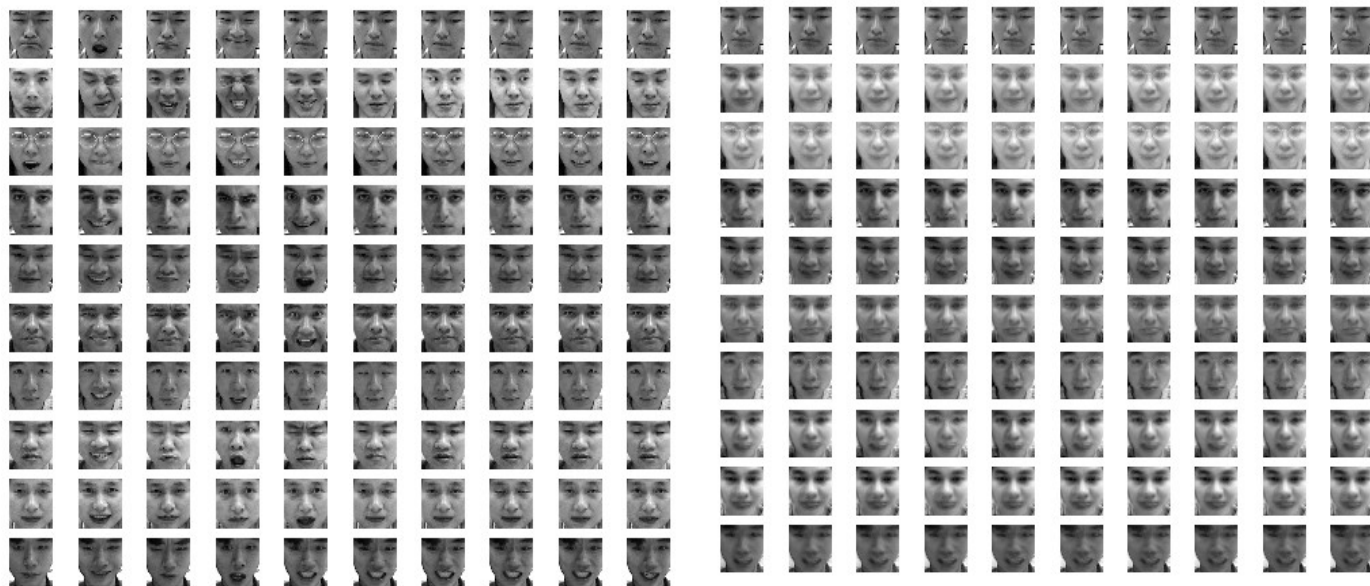
答:



1.2. Dataset 中前 10 個人的前 10 張照片的原始圖片和 reconstruct 圖 (用前 5 個 eigenfaces):

答:

原圖為:

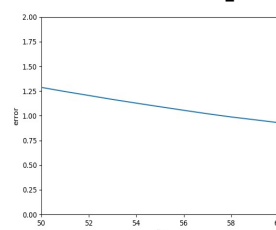


左圖為原本的圖片，又圖為用 5 個 eigenfaces 後, 所得到的圖片, 很明顯的發現 reconstruct 後的圖片跟原圖有許多的不同, 那是因為我們只提取 5 個 feature 去還原, 假如提取多一點的 feature, 那麼就會愈跟原圖相進.

1.3. Dataset 中前 10 個人的前 10 張照片投影到 top k eigenfaces 時就可以達到 $< 1\%$ 的 reconstruction error.

答: (回答 k 是多少)

k = 59



2.1. 使用 word2vec toolkit 的各個參數的值與其意義:

答:

size:表示訓練出來的字詞向量會有幾個維度

alpha:表示訓練 model 的初始 learning rate

window:表示往前或往後看第幾個字

min_count:表示字詞需要出現過幾次, 才會把他丟進 train

iter:表示你要訓練多少次 model

cbow:表示用哪種 model, 例如: skim-gram

size : 35

iter : 60

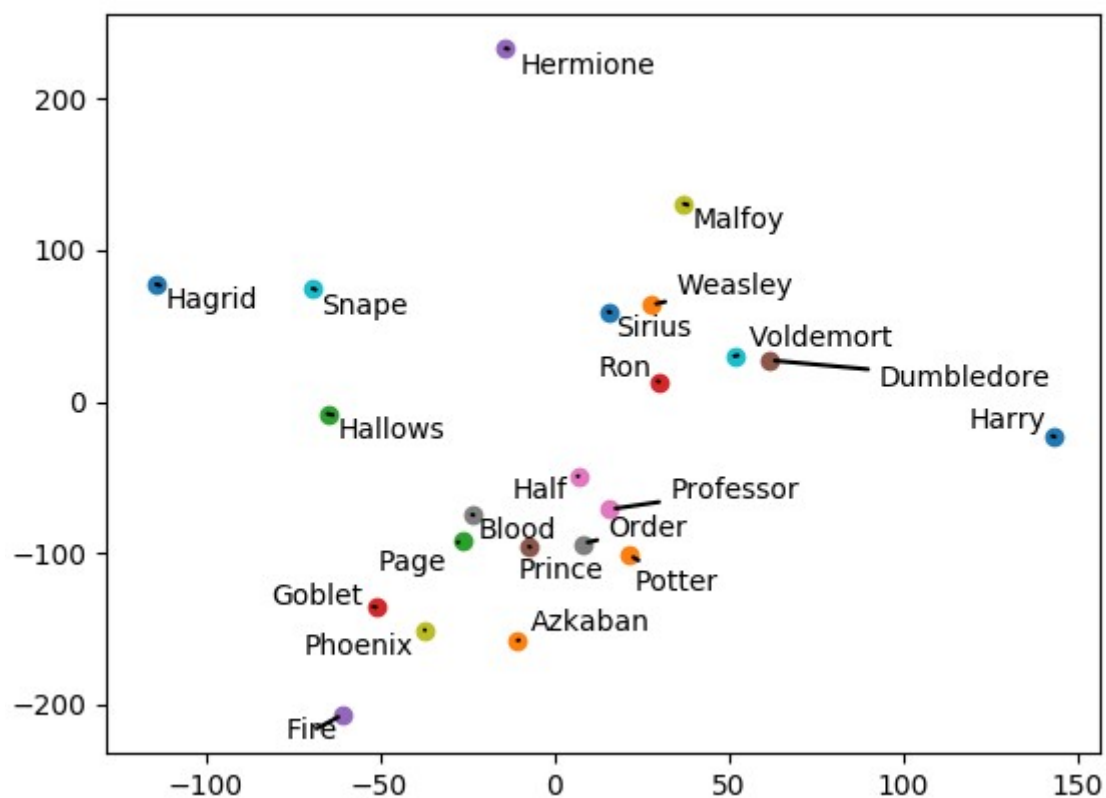
min_count: 550

window : 5

alpha and cbow 為 default

2.2. 將 word2vec 的結果投影到 2 維的圖:

答: (圖)



2.3. 從上題視覺化的圖中觀察到了什麼?

答:

書的標題名稱大部份都落於中間那一塊區域, 然後從 Harry 開始往左上拋物線形狀, 大多數都是人名。

3.1. 請詳加解釋你估計原始維度的原理、合理性，這方法的通用性如何？

答：

原始維度因為有給我們限定範圍為 1~60 之間，那麼我們只要逆推回去就可以了，計算方法是原始的資料，一定有他原本的特徵存在，再擴增矩陣時，原本矩陣的 **feature** 也不會消失，反而會因為通過 $w\mathbf{x}+b$ 更加增大，那麼我們只要抓取出這些 **feature** 就能得到原始的維度，由於取得的資料只有 **test** 的部份，沒有 **label**，那我們就需要自行生成一組 **train data**，並且取得所有維度的可能性，即 1~60 維度之間的所有 **datapoint** 可能性，後續我們要抓取出這些 **train data** 的 **feature**，用用 **Nearest Neighbor** 這演算法裡面的 **Ball Tree** 去提取出所有 **train data** 的 **feature**，然後再使用 **Support vector regression** 這個 **machine learning** 方法去訓練我們的 **train data**，並且得到 **model**，最後一樣先取得 **test data** 的 **feature**，再把他丟給 **model** 去預測出這些 **test data** 的原始維度。

這個方法有一個麻煩之處，在於需要自行生成一組 **train data**，如果我們不知道原始的維度範圍，那麼我們就需要慢慢測試出 **test data** 所預測出來的維度的範圍，這部份，如果給的 **test data** 維度很大，再創擴增矩陣時，很容易發生 **memory out** 的問題。

3.2. 將你的方法做在 **hand rotation sequence dataset** 上得到什麼結果？合理嗎？請討論之。

答：

在做 **hand rotation sequence dataset** 時，因為圖片的維度極大，所以，我再做處理時，先是把這個圖片的為度大小做切割，我把他切成 8 等份，由於每張圖片他的重點 **feature** 是固定的，故再切割時，不會影響到結果，例如一張 25*25 大小的圖片，他只有 4*4 的影像，其餘皆為白色，那麼再切割時，如果切割的範圍內沒有 **feature**，那麼這個切割區域的維度就是 0。

我們做降維度的方法是跟上訴一樣的，差別在於我是把 **test** 分割成 8 等份去測試出來個別等份的維度，之後再把這些維度都加起來，就得到原始資料的維度了

我所預測出來的維度經過 \ln 後得到 9.29172