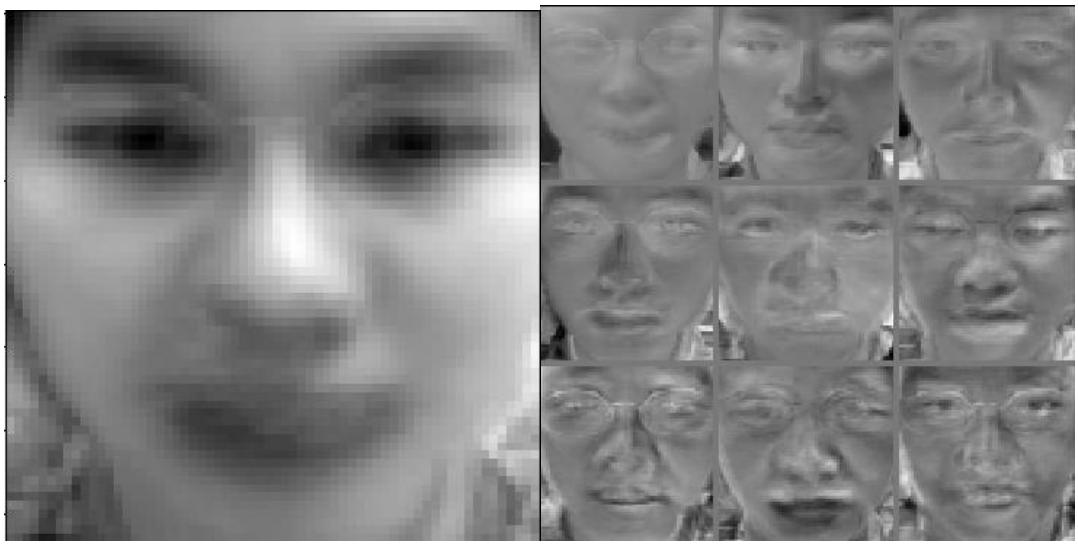
學號：R05921077 系級： 電機碩一 姓名：陳立杰

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： (，右圖為 3x3 格狀 eigenfaces, 順序為 左到右再上到下)



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：(左右各為 10x10 格狀的圖, 順序一樣是左到右再上到下)

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:

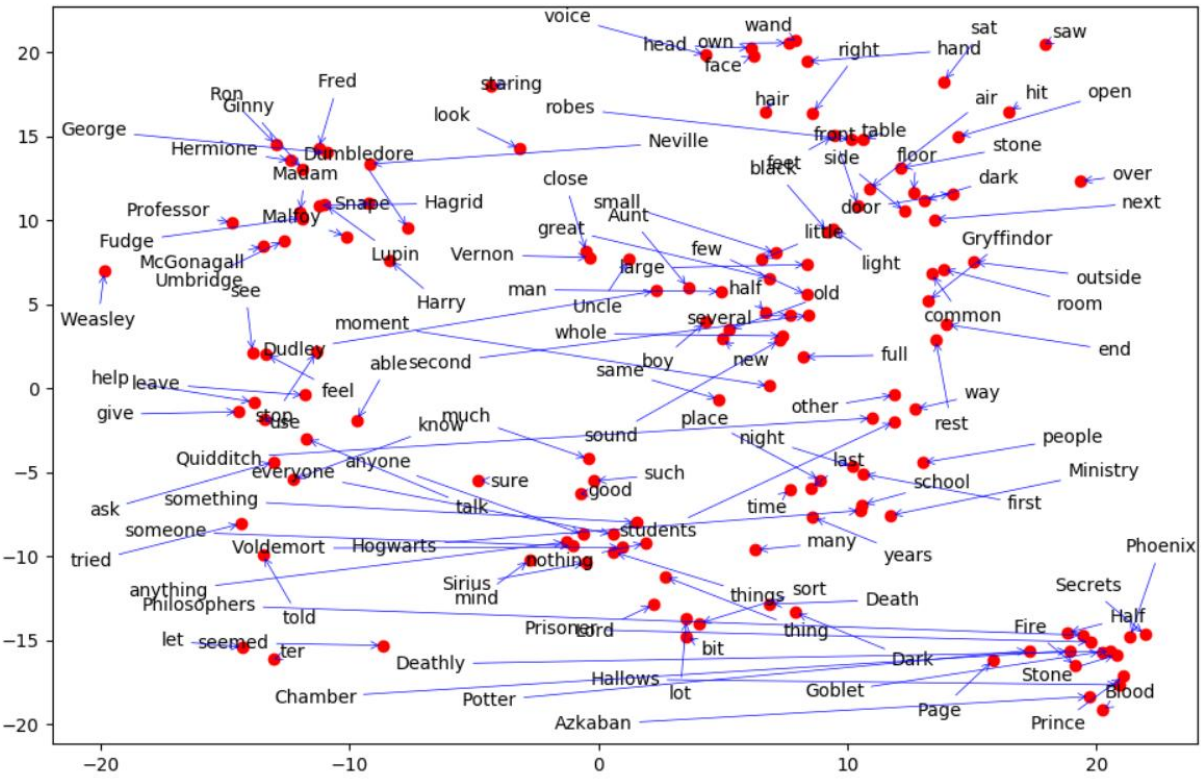
答：

w.word2vec('./all.txt', './text8.bin', size=50, verbose=True)

我試了很多參數，但感覺大部分影響不大，像 *min_count* 是忽略小於這個 frequency 的字，但是因為原本就已經挑出最多使用的數字，所以調整此參數沒什麼意義，*alpha* 是 learning rate，default 的設定就足夠使用了，*window* 是目前的 word 跟要 predict 的 word 的距離大小，一樣用預設的 5，最後我只保留原本的幾項參數。第一個參數是讀取要轉成 vector 的 word，第二個參數是是把 train 好的 model 存起來，第三個參數是 words 轉成 vector 後的 dimension，第四個參數是在 compile 時是否顯示詳細資料。

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：

從這個圖很明顯可以看到他做了各種有意義的分類，像是右下那一群全部都是跟每一集的名字有關。左上那一群的話就是一些人物的名字，而且越相近的人物越近，像石內普跟跩哥馬份兩個反派幾乎是黏在一起，然後教授等人也會比較靠近旁邊的 professor，左下那一群大概就是一些動詞。中間那邊有三群是代名詞、時間、量詞各一群。右上有兩個群，物品群還有跟身體有關的群

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：
我的作法是先產生多組 data 他把 transform 到 100 dimensions，在這個高維空間上隨意 sample 100 個點，然後再取它附近 300 個點來做 PCA，取出所有 eigenvalue 來看在第幾個 value 後會明顯下降，再把所有 sample 點之 eigenvalues 做平均，避免 sample 到曲率過大的點，然後在加上這組 data 的標準差，看這組 data 的離散程度如何，把些資料跟 data 的 label 對映好，用 SVR 做 linear regression，讓 input data 跟 label map 到一個最佳解，train 完後就可以 predict testing data 的維度。這個做法感覺可以用在許多地方，因為它可以局部 sample 多個點再做平均來避免一些非線性的曲面，所以就算是非線性 transform 的 data 用這個方法應該也是可以得到不錯的結果，只不過前提是要有辦法得到 training data 跟對應的 label。

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：
因為這題沒有辦法自動產生 label data，沒有辦法直接用 SVR 做 regression，所以我直接 sample 幾個點後取鄰近點算 PCA 再取他們的平均 eigenvalue，最後再全部列出來直接判斷，只不過在做 SVD 時參數要設 full_matrices=False 不然會有記憶體錯誤，下列是結果:
[1. 0.65424037 0.46392074 0.35731298 0.28098685 0.23927274
  0.21089253 0.18396492 0.15868999 0.14351788 0.13030794 0.1214479
  0.11230075 0.10564158 0.10062947 0.09610514 0.09087608 0.08808576
  0.08540917 0.08319101 0.081447 0.07971141 0.07834665 0.07705882
  0.07582945 0.07475372 0.0738856 0.07292033 0.00000094]
Eigenvalue 一直下降到第六個之後就變的很平緩了，結果我覺得蠻合理的，因為這些圖片感覺他的 dimension 不會超過六個，所以在的六個 eigenvalue 後大概都差不多小，從這個 distribution 我猜測 dimension 應該是在四左右。