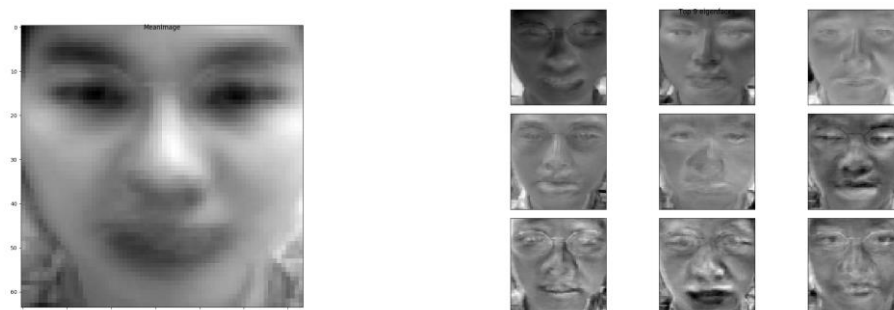


學號：R05921033 系級： 電機所碩一 姓名：余政穎

**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:**

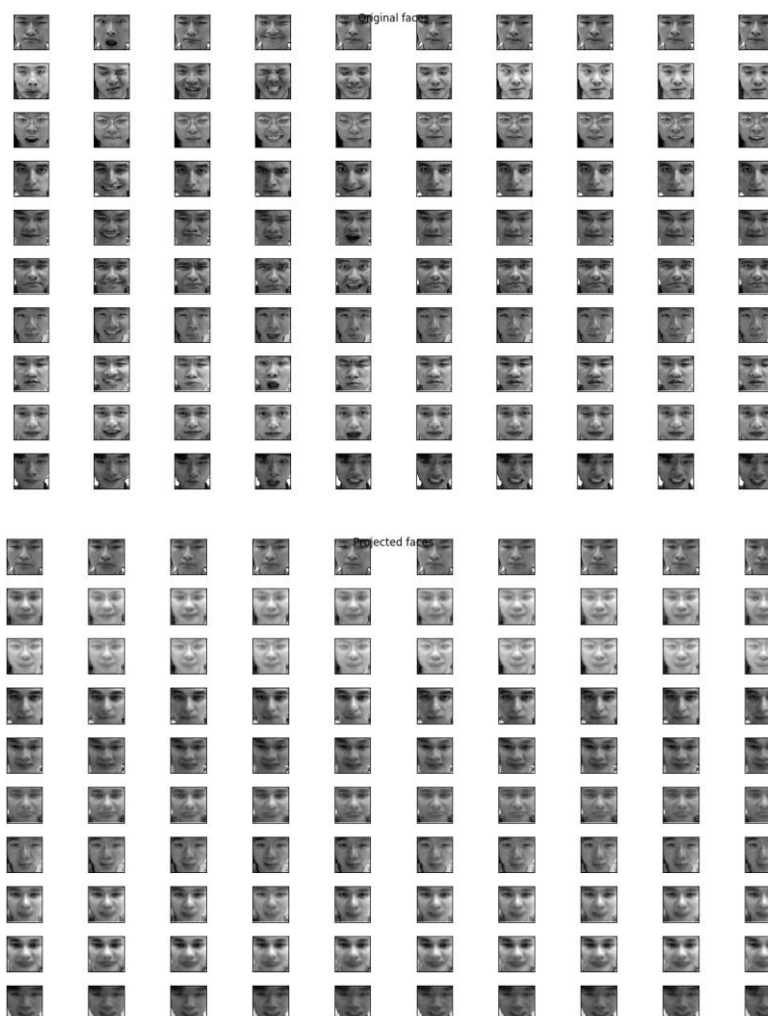
答：左圖是平均臉，右圖為 3x3 格狀 eigenfaces, 順序為左到右再上到下



可以觀察到，平均臉的圖形其實可以看的出一般正常人臉的形狀，而 eigenfaces 長相較特別(不是一般正常臉型)，推測是因為有特別放大某些特徵值的緣故。

**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：第一張為 10x10 格狀的原始圖片，順序一樣是左到右再上到下；第二張為 10x10 格狀的 reconstruct 圖片，順序一樣是左到右再上到下。



可以觀察到，使用 top 5 eigenfaces 所重建出來的圖，基本上同一個人都長得差不多，不像原圖有多種表情，這是因為只取重要 component 重建回來的原因，當我們取越多的 eigenfaces，便越能重建回原來的形狀，我嘗試跑過 top 100 eigenfaces 的結果，基本上重建回的圖，視覺上看起來跟原圖差不多！

**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. (相對值計算除以 256)**

答：我從 top 1 eigenfaces 開始往下跑，檢查其重建的 eigenfaces，記錄每 top 10n(n 屬於正整數)的 error 以及最小的 k 使得 RMSE 小於 0.01 的 RMSE 值。

k	10	20	30	40	50	58	59
RMSE	0.04922	0.03307	0.02445	0.01815	0.01343	0.0103	0.00997

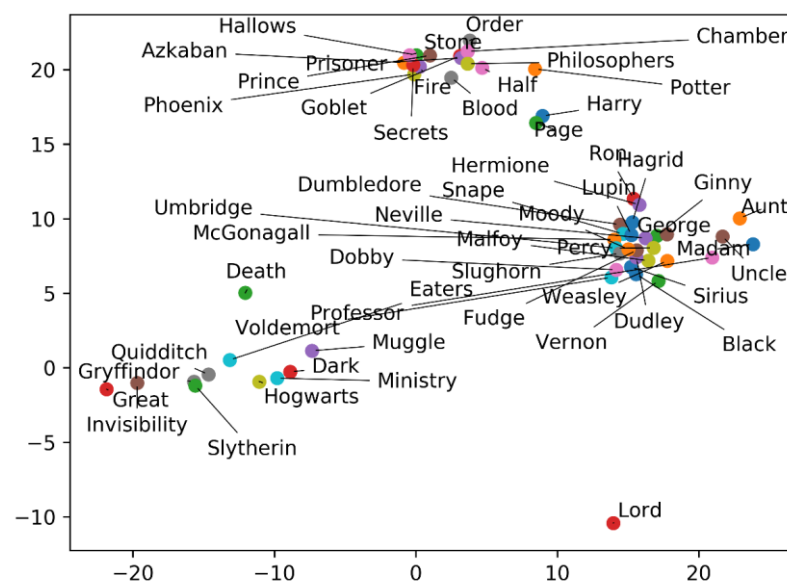
最小的 k 使得 RMSE 小於 0.01 的 k 是 59。

**2.1. 使用 word2vec toolkit 的各個參數的值與其意義：**

答：MIN\_COUNT = 2：忽略出現頻率小於此字詞的詞，由於這邊設 2，只出現一次的詞都不會被訓練到，減少一些字。WORDVEC\_DIM = 200：字詞 vector 的維度，越多代表模型的 fit 的資料越多，但也可能容易 overfit；WINDOW = 1：目前的字詞以及想要預測的字詞在句子隔的字數，這是設定字詞所影響的範圍，越大代表目前的字詞可影響越遠的其他字；NEGATIVE\_SAMPLES = 10：設定是否有 noise 資料已清除多餘的雜訊；ITERATIONS = 1000：訓練的 iterations 數量；MODEL = 0：若為 0，表示使用 CBOW、若為 1，表示使用 skip-gram；LEARNING\_RATE = 0.0001：一開始的 learning rate。

**2.2. 將 word2vec 的結果投影到 2 維的圖：**

答：以上述參數訓練後投影 600 個維度，所呈現的結果如下圖：



**2.3. 從上題視覺化的圖中觀察到了什麼？**

答：從上題的圖我們可以觀察到，字詞可被分為幾個區塊，主要都是名詞的部分，首先是上半部，基本上是書的標題(如：stone、Azkaban、Fire、Phoenix、Prince...等等)，由於其出現的地方在七本書之中應該相當類似，故被認為是同類是滿合理的；再來是左下角，這邊比較是霍格華茲的一些專用術語，像是 Gryffindor 跟 Slytherin 這最常出現的兩個學院，以及熱門的運動 Quidditch；最後是右邊的區域，基本上就是書中主要角色的人名，除了 Harry 由於也常常出現在標題中，故出現在比較遠的地方，其他的人名都集中在此，包含 Dumbledore、Hermione、Ron、Ginny、Snape 等知名角色，而且更進一步，可以觀察到衛斯理一家在較右邊，Hermione、Ron 在此區域的上緣，Snape、Dumbledore、McGonagall 等教授名在此區域的左邊，表示比較常一起出現的字詞，會聚集在附近的現象，這表示我們的訓練還不錯，呈現相當合理的結果！

### 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：

我主要是參考助教於 TA hour 時使用的方法，先使用 gen.py 產生與測試資料類似的 samples(分別從 1~60 維作為初始層，接著第二層的維度 random 從 61~80 維中挑一個數字，每一個 set 產生出 10000、20000、50000、80000、100000 個資料點)；接著，在每一個 set 中，我 random 挑出 10 個資料點(因為這樣便已足夠代表這個 set 的分布，也可節省時間)，找出在全部資料中，離這些點最近的 150 個點(使用 ball-tree)，用這些點計算其 eigenvalues，準備當作 linear SVR 模型的資料，以剛剛算出的 eigenvalue 作為訓練模型的 x，實際維度作為 y 得到一個類似於測試資料的模型，最後以同樣方法將測試資料隨機 sample、找離每個點最近的 10 個點並取 eigenvalue 丟入訓練好的模型，以用來預測其維度。這個方法需建立在已知資料點的產生方式為何才能使用，需知道測試資料的原始維度區間及升維方式，現實世界資料，有些不一定能知道這些資訊，所以我認為通用性有限。

原本我也有使用 sklearn 套件中的 Isomap 方法對可能的維度做降維，並計算 reconstruction error(此法可接近 baseline，但沒有超過)，若在合理的 threshold 下便猜測此維度，此法雖也能猜出某些原本較低維度的資料，但對於較高維度的資料估計不準，且 threshold 的調整也沒有一個依據，故此法亦不是非常通用。

### 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：針對 hand rotation sequence dataset，將整組照片當成一個 dataset，我直接採用 3.1 的方法，假設這個 dataset 的初始維度與升維方式一樣，再用訓練出來的模型去預測，並得到的降維後的結果為 11；至於此方法的合理性，對於手部轉動這個 dataset，圖片之間的變化性不大，所以可用低維度的資料表示也是合理的，用 PCA 也有可能可以不需要太多 eigenfaces 就重建回結果；只是，畢竟我們實際上並不知道 intrinsic dimension 跟圖片呈現出的 feature dimension 之間的轉換關係，所以這樣的假設其實還是有可能估計的不太準確。