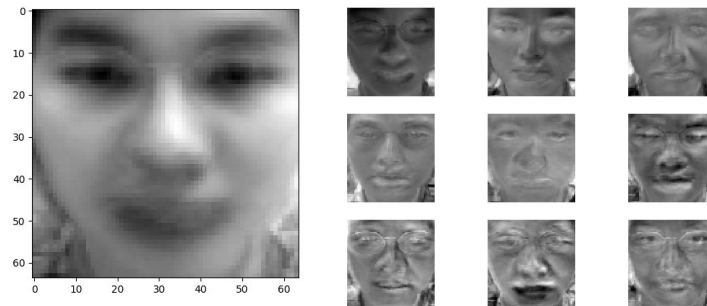


1.1. Dataset 中前 10 個人的前 10 張照片的平均臉和 PCA 得到的前 9 個 eigenfaces:

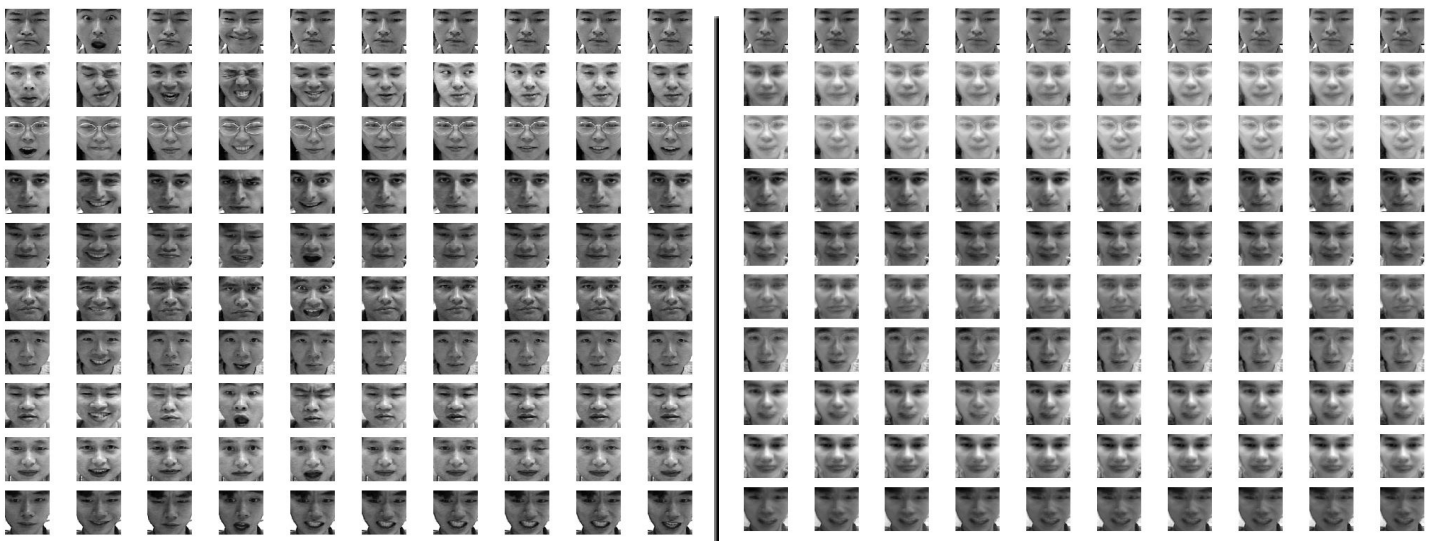
答：(左圖平均臉，右圖為 3x3 格狀 eigenfaces, 順序為 左到右再上到下)



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：

(original) (reconstruct)



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：

$k = 59$

2.1. 使用 word2vec toolkit 的各個參數的值與其意義:

答：

`min_count = 5`. This discards words than appear less than `min_count`. I chose 5 which is the default value.

`wordvec_dim = 300`. Size of a word vector. The default value is 100 and I picked 300. Usually a large vector size will result in a better prediction.

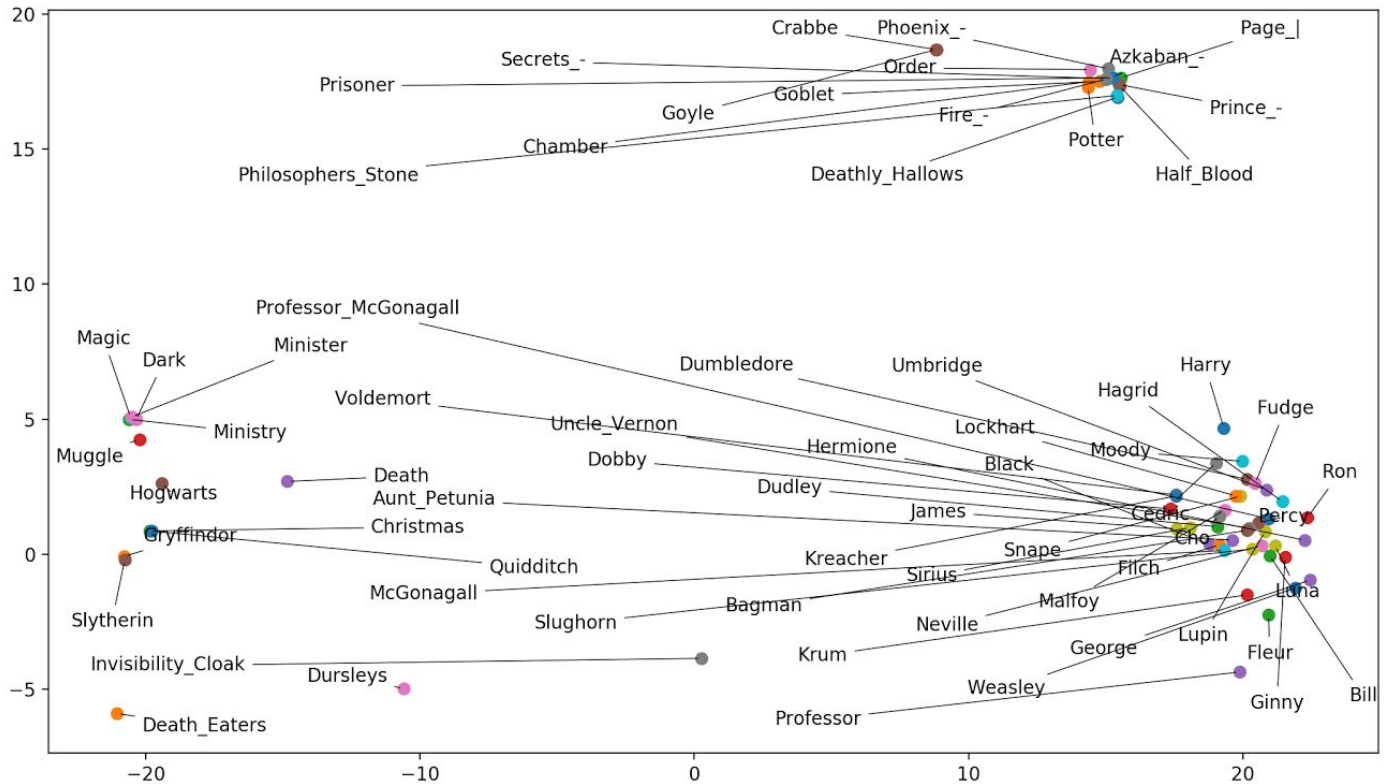
`window = 5`. The max skip length between words. 5 is the default value.

`negative_sample = 5`. Number of negative words. 5 is the default value.

`iteration = 6`. Training iterations. The default is 5 but I chose 6 to see if it gives a better result.

model = 0. 0 means skip-gram. This model uses the current word to predict surrounding context, which does a better job for infrequent words.

## 2.2. 將 word2vec 的結果投影到 2 維的圖:



答：

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？

My code is based on the example provided by TAs, which uses the nearest neighbors to estimate dimensions. It is a good method when decision boundary is irregular. The code first takes N random sample points and pick k neighbors to compute the eigenvalues. After normalizing the eigenvalues, it is then feed into a decision tree model for fitting. In the example, SVD is used but I change it to a decision tree model because non-linear

relationships between parameters will not affect the result as greatly. My kaggle accuracy is around 0.069 with SVD and 0.059 with decision tree. I also change y labels to be  $\ln(\text{dimensions})$  instead of just dimensions, this improves Kaggle accuracy. The number of examples and neighbors are 20 and 400 respectively. I have tried 512 neighbors but the result was not better. This might be because it makes boundaries between labels less distinct. Overall, my model did quite well.

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：

Because my trained model in 3.1 takes input data with dimension 100 so I need do some image preprocessing. First of all, the images are cropped to a square by taking the left 480\*480 and normalized. They are then resized to 10\*10 using PIL and the aspect ratio is maintained. The images are now converted to a dataset of 481 samples with 10\*10(=100) dimensions and can be fed into my model. Because the prediction value is  $\ln(d)$ , the real dimension =  $e^{\text{predicted}} = 4.00$  (rounded). In my opinion, the hand in the images seem to be rotating on a 1-d circle and no top or bottom views of the hand have appeared so the actual dimension should be around 3 instead. However, my model still gives a resonable prediction, which is 4.00. It is possible that re-scaling has affected the accuracy.