Problem 2: Principal Component Analysis (40%)

Principal component analysis (PCA) is a technique for dimensionality reduction which performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In this problem, you will perform PCA on a dataset of face images.

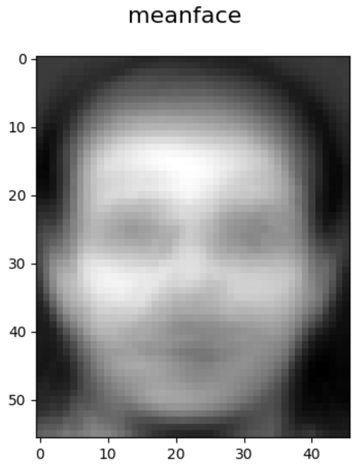
The folder `p2_data` contains face images of 40 different subjects (classes) and 10 grayscale images for each subject, all of size $(56, 46)$ pixels. Note that `i_j.png` is the j-th image of the i-th person, which is denoted as $\textbf{person}_i\textbf{image}_j$ for simplicity.

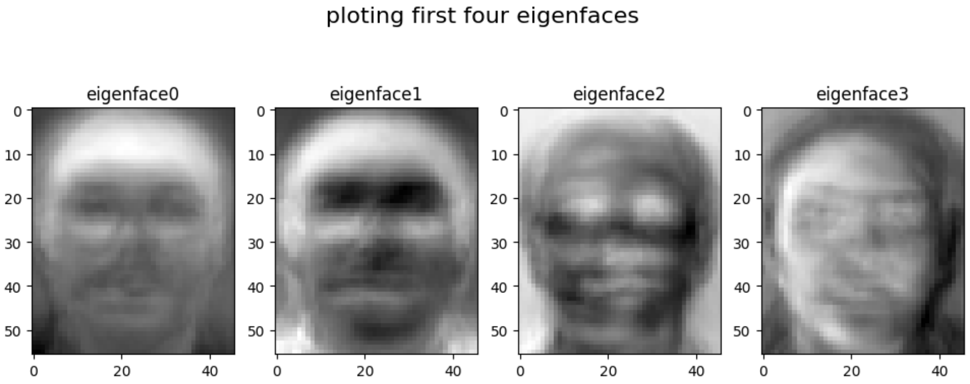
First, split the dataset into two subsets (i.e., training and testing sets). The first subset contains the first 6 images of each subject, while the second subset contains the remaining images. Thus, a total of $6 \times 40 = 240$ images are in the training set, and $4 \times 40 = 160$ images in the testing set.

In this problem, you will compute the eigenfaces of the training set, and project face images from both the training and testing sets onto the same feature space with reduced dimension.

1. (10%) Perform PCA on the training set. Plot the mean face and the first four eigenfaces.

Ans of problem2.1:

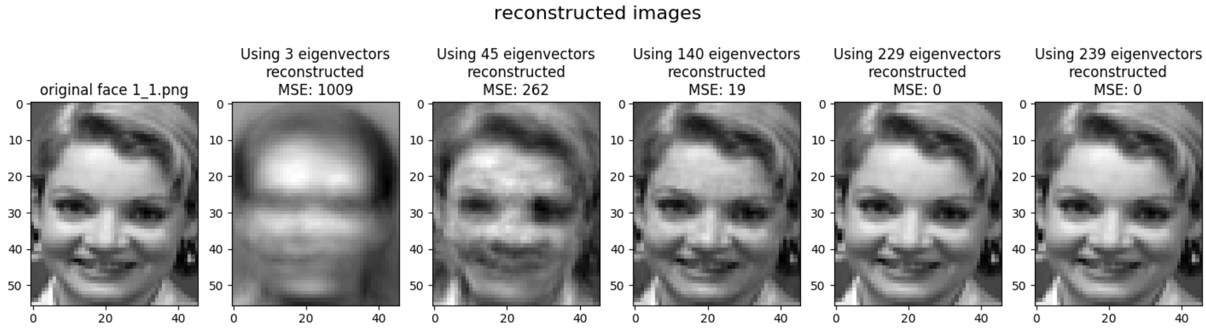




2. (8%) Take **person$_1$image$_1$**, and project it onto the PCA eigenspace you obtained above. Reconstruct this image using the first $n = 3, 45, 140, 229$ eigenfaces. Plot the four reconstructed images.

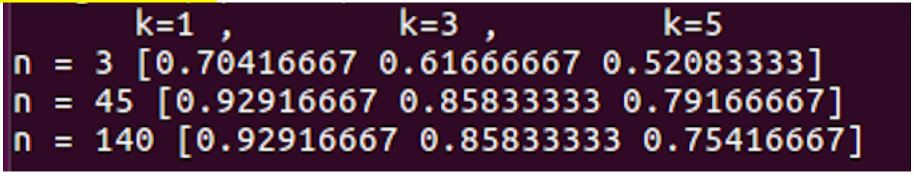
3. (4%) For each of the four images you obtained in 2., compute the mean squared error (MSE) between the reconstructed image and the original image. Record the corresponding MSE values in your report.

Ans of problem2.2 and 2.3:



4. (10%) Now, apply the k-nearest neighbors algorithm to classify the testing set images. First, you will need to determine the best k and n values by 3-fold cross-validation. For simplicity, the choices for such hyperparameters are $k = \{1, 3, 5\}$ and $n = \{3, 45, 140\}$. Show the cross-validation results and explain your choice for (k, n).

Ans of problem2.4:

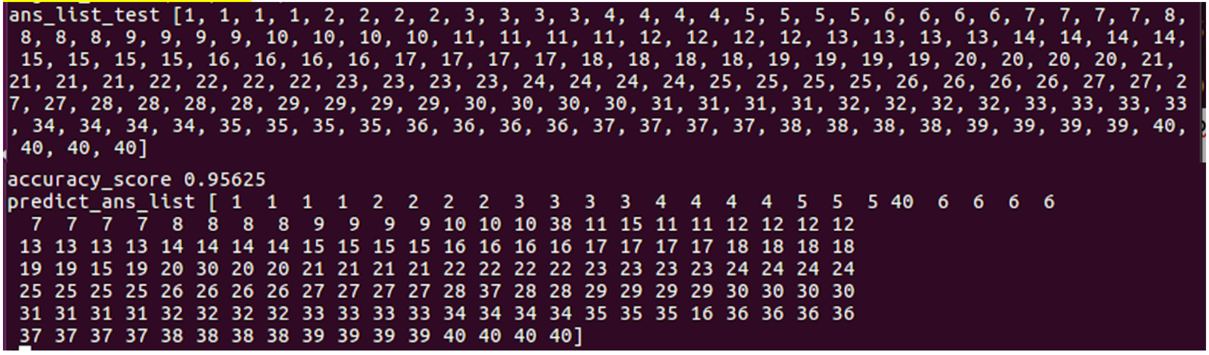


```
      k=1 ,         k=3 ,         k=5
n = 3 [0.70416667 0.61666667 0.52083333]
n = 45 [0.92916667 0.85833333 0.79166667]
n = 140 [0.92916667 0.85833333 0.75416667]
```

I use the function “GridSearchCV” from sklearn.model_selection.
Picking the mean testing score of each parameters set (k , n).
We can find that when k=1, the performance is better than k= 3 or 5. On the other hand, the performance is also better when n is larger. However, depending on the computing resource consuming, I prefer to choose the lower n if the performance is nearly the same(n=45 or 140 at k=1). So that we could save both the computing resource and time but obtain nearly the same performance. Ans: choose k=1, n=45

5. (8%) Use your hyperparameter choice in 4. and report the recognition rate of the testing set.

Ans of problem2.5:



```
ans_list_test [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7, 8,
 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13, 14, 14, 14, 14,
 15, 15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 17, 18, 18, 18, 18, 19, 19, 19, 19, 20, 20, 20, 20, 21,
21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26, 26, 26, 27, 27, 2
7, 27, 28, 28, 28, 28, 29, 29, 29, 29, 30, 30, 30, 30, 31, 31, 31, 31, 32, 32, 32, 32, 33, 33, 33, 33
, 34, 34, 34, 34, 35, 35, 35, 35, 36, 36, 36, 36, 37, 37, 37, 37, 38, 38, 38, 38, 39, 39, 39, 39, 40,
 40, 40, 40]
accuracy_score 0.95625
predict_ans_list [ 1  1  1  1  2  2  2  2  3  3  3  3  4  4  4  4  5  5  5 40  6  6  6  6
  7  7  7  7  8  8  8  8  9  9  9  9 10 10 10 38 11 15 11 11 12 12 12 12
 13 13 13 13 14 14 14 14 15 15 15 15 16 16 16 16 17 17 17 17 18 18 18 18
 19 19 15 19 20 30 20 20 21 21 21 21 22 22 22 22 23 23 23 23 24 24 24 24
 25 25 25 25 26 26 26 26 27 27 27 27 28 37 28 28 29 29 29 29 30 30 30 30
 31 31 31 31 32 32 32 32 33 33 33 33 34 34 34 34 35 35 35 16 36 36 36 36
 37 37 37 37 38 38 38 38 39 39 39 39 40 40 40 40]
```

Accuracy = 0.95625