1

Given m = 5 data points configuration in Figure 1. Assume K = 2 and use Euclidean distance. Assuming the initialization of centroid as shown, after one iteration of k-means algorithm, answer the following questions.
(a) Show the cluster assignment;

1: [2,2]
2: [-1,1]
3: [3,1]
4: [0,-1]
5: [-2,-2]

inital centroids:
A:[-3,-1]
B:[2,1]

Thus the Euclidean distance from each point to each centroids can be computed as follow:
$dist = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$

After calculation, the distance is:

centroid	A	B
1,	[5.8	1.0]
2,	[2.8	3.0]
3,	[6.3	1.0]
4,	[3.0	2.8]
5,	[1.4	5.0]

Then we just need to choose the min for each point. Thus the cluster assignemnt is as follow:

A : 2, 5
B: 1, 3, 4

(b) Show the location of the new center;
The new center can be calculated as follow:

For each cluster, get the coordinates of each point, sum up then divide by number of points.
$X_{new} = \frac{1}{n}\sum_{i=1}^{n} X_i$
$Y_{new} = \frac{1}{n}\sum_{i=1}^{n} Y_i$

A: [1.5, -0.5]
B: [1.67, 0.67]

(c) Will it terminate in one step?
Run the code one more time, to see whether the cluster assignment changed or not. Since 2nd iteration changed the cluster assignment to:
A: 2, 4, 5
B: 1, 3
Thus it will not terminate in one step.

Now answer the above questions using Manhattan distance
(a) Show the cluster assignment;

1: [2,2]
2: [-1,1]
3: [3,1]
4: [0,-1]
5: [-2,-2]

inital centroids:
A:[-3,-1]
B:[2,1]

Thus the Manhattan distance from each point to each centroids can be computed as follow:

$dist = |X_1 - X_2| + |Y_1 - Y_2|$

After calculation, the distance is:

centroid	A	B
1,	[8	1]
2,	[4	3]
3,	[8	1]
4,	[3	4]
5,	[2	7]

Then we just need to choose the min for each point. Thus the cluster assignemnt is as follow:

A: 4,5

B: 1,2,3

(b) Show the location of the new center;
The new center can be calculated as follow:

For each cluster, get the coordinates of each point, sum up then divide by number of points.
$X_{new} = \frac{1}{n}\sum_{i=1}^{n} X_i$
$Y_{new} = \frac{1}{n}\sum_{i=1}^{n} Y_i$

A: [-1.0, -1.5]
B: [1.33, 1.33]

(c) Will it terminate in one step?
Run the code one more time, to see whether the cluster assignment changed or not. Since 2nd iteration changed the cluster assignment to:
A: 2, 4, 5
B: 1, 3
Thus it will not terminate in one step.

2

Consider the data point setting in Figure 2. We will use spectral clustering to divide these points into two clusters. Our version of spectral clustering uses a neighborhood graph obtained by connecting each point to its two nearest neighbors (breaking ties randomly), and by weighting the resulting edges between points x_i and x_j by $W_{ij} = \exp(-\|x_i - x_j\|)$. Indicate on Figure 2b the clusters that we will obtain from spectral clustering. Please provide an argument for your answer. Any reasonable answer will be given credits.

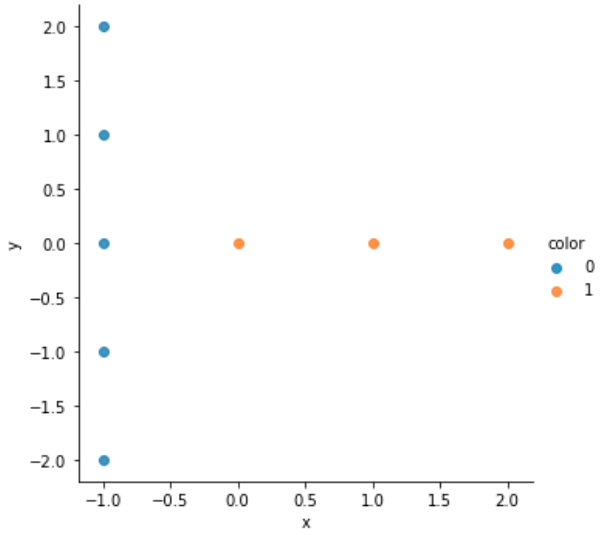

Figure 1: spectral clustering

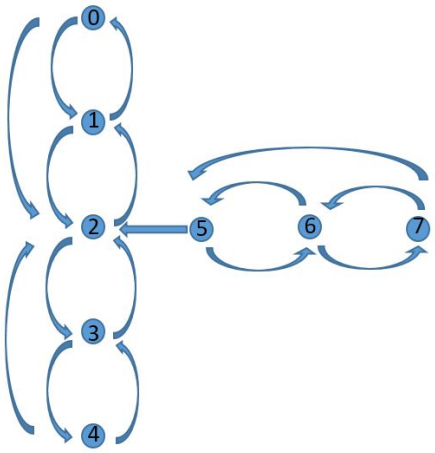

Figure 2: directed graph

	0	1	2	3	4	5	6	7
0	0.367879	-0.367879	-0.135335	0.000000	0.000000	0.000000	0.000000	0.000000
1	-0.367879	0.735759	-0.367879	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	-0.367879	1.374309	-0.367879	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	-0.367879	0.735759	-0.367879	0.000000	0.000000	0.000000
4	0.000000	0.000000	-0.135335	-0.367879	0.367879	0.000000	0.000000	0.000000
5	0.000000	0.000000	-0.367879	0.000000	0.000000	0.503215	-0.367879	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	-0.367879	0.735759	-0.367879
7	0.000000	0.000000	0.000000	0.000000	0.000000	-0.135335	-0.367879	0.367879

Figure 3: Laplacian matrix of the directed graph

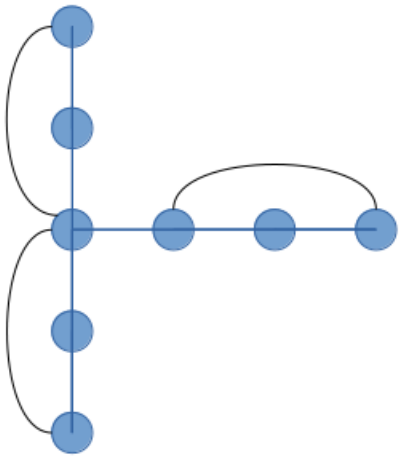

Figure 4: undirected graph

	0	1	2	3	4	5	6	7
0	0.503215	-0.367879	-0.135335	0.000000	0.000000	0.000000	0.000000	0.000000
1	-0.367879	0.735759	-0.367879	0.000000	0.000000	0.000000	0.000000	0.000000
2	-0.135335	-0.367879	1.374309	-0.367879	-0.135335	-0.367879	0.000000	0.000000
3	0.000000	0.000000	-0.367879	0.735759	-0.367879	0.000000	0.000000	0.000000
4	0.000000	0.000000	-0.135335	-0.367879	0.503215	0.000000	0.000000	0.000000
5	0.000000	0.000000	-0.367879	0.000000	0.000000	0.871094	-0.367879	-0.135335
6	0.000000	0.000000	0.000000	0.000000	0.000000	-0.367879	0.735759	-0.367879
7	0.000000	0.000000	0.000000	0.000000	0.000000	-0.135335	-0.367879	0.503215

Figure 5: Laplacian matrix of the undirected graph

1. calculate the Euclidean distance between one points and its two nearest neighbors by using the following formula:
   $dist = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$

2. calculate the $weight_{ij}$:
   $Weight_{ij} = exp(-dist)$
   Thus we can get the adjacency matrix A.

3. sum up each column of A to form a diagonal degree matrix D:

4. calculate the Laplacian matrix L:
   L = D-A

5. perform eigen decomposition, find the eigenvalues and eigenvectors:
   eigenvectors: $v_1, v_2, v_3, ..., v_n$
   eigenvalues: $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_n$

6. find 2 eigenvectors v_1, v_2 corresponding to the 2 smallest eigenvalues. Form a new matrix Z:
   Z = (v_1, v_2)

7. perform Kmeans algorithm on Z with k=2.

There are two situation:1. directed graph; 2. undirected graph.

For directed graph, since one point has two neighbors, we can draw the graph like Fig 2. We see that the connection(edge) between point 2 and 5 is very weak. There is only one edge from 5 to 2. Also, as shown in the Laplacian matrix (Figure 3), the matrix is composed of two blocks. The first block is made of data points 0 to 4; the second block is made of data points 5 to 7. There is only one exception that point 2 and point 5 has a connection which is highlighted by blue square. Thus information stored in the Laplacian matrix can be captured by the spectral clustering algorithm through eigen decomposition. In summary, all the points can be divided into two clusters as shown in Fig 1.

For undirected graph, the result is similar as shown in Fig 1, 4 and 5.

3

Suppose we have 4 points in 3-dimensional Euclidean space, namely (4, -2, 4), (5, -3, 5), (2, 0, 2), and (3, -1, 3).

(a) Find the first principal direction.

1. given 4 data points, find the mean for each column (feature):
   $\mu = \frac{1}{4}\sum_{i=1}^{4} x^i$

2. find the covariance matrix for the data:
   $C = \frac{1}{4}\sum_{i=1}^{4}(x^i - \mu)(x^i - \mu)^T$

3. find the eigenvector w_1 of the covariance matrix C corresponding to the largest eigenvalue λ_1. Then, this eigenvector is the first principle direction.

The first principal direction is:
(0.57735027, -0.57735027, 0.57735027)

(b) When we reduce the dimensionality from 3 to 1 based on the principal direction you found in (a), what is the reconstruction error in terms of variance?

1. Since principle direction satisfies:
   $Cw = \lambda w$

2. Variance in the principle direction is:
   $w^T C w = \lambda w^T w = \lambda$

3. As we can see that the reconstruction error in terms of variance is the sum of the rest of the eigenvalues, thus the variance is 0.

(c) You are given the following 2-D datasets, approximately draw the first and second principal directional on each plot.

The two principle directions are shown as below.

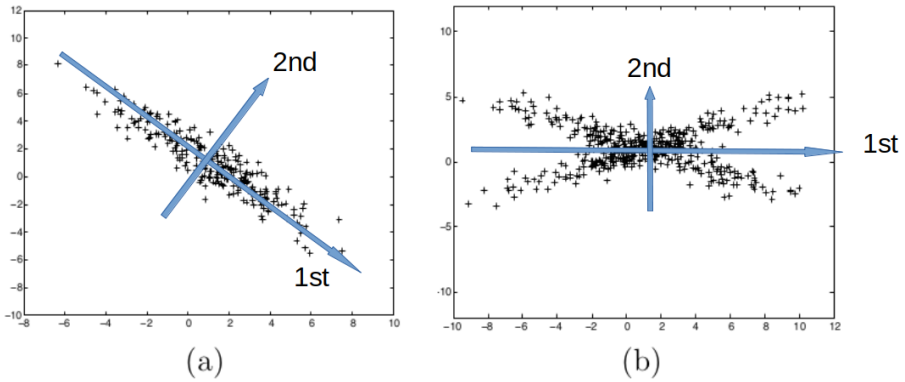

Figure 6: principle direction

4

This question is a simplified illustration of using PCA for face recognition using a subset of data from the famous Yale Face dataset.

Remark: you have to perform downsampling of the image by a factor of 4 to turn them into a lower resolution image before we do anything.

1. First, given a set of images for each person, we generate the so-called eigenface using these images. The procedure to obtain eigenface is explained as follows. Given n images of the same person denoted by x_1 , . . . , x_n . Each image originally is a matrix. We vectorize each image to form the vector $x_i \in \mathbb{R}^p$. Now form a matrix.

   $X = [x_1, ..., x_n] \in \mathbb{R}^{p \times n}$

   The eigenfaces correspond to the largest k eigenvector of the data matrix XX^T. Perform analysis on the Yale face dataset for subject 14 and subject 01, respectively, using all the images EXCEPT for the two images named subject01-test.gif and subject14-test.gif. Plot the top 6 eigenfaces for each subject. When visualizing the eigenvalues,you have to reshape the eigenvectors into images with the same dimension as the original images.

   What is the interpretation of the top 6 eigenfaces?

   The top 6 eigenfaces capture a set of basis features of all the facial images. The first eigenface has the most important common feature of all the facial images, then the importance decreases as the number of eigenfaces goes up.

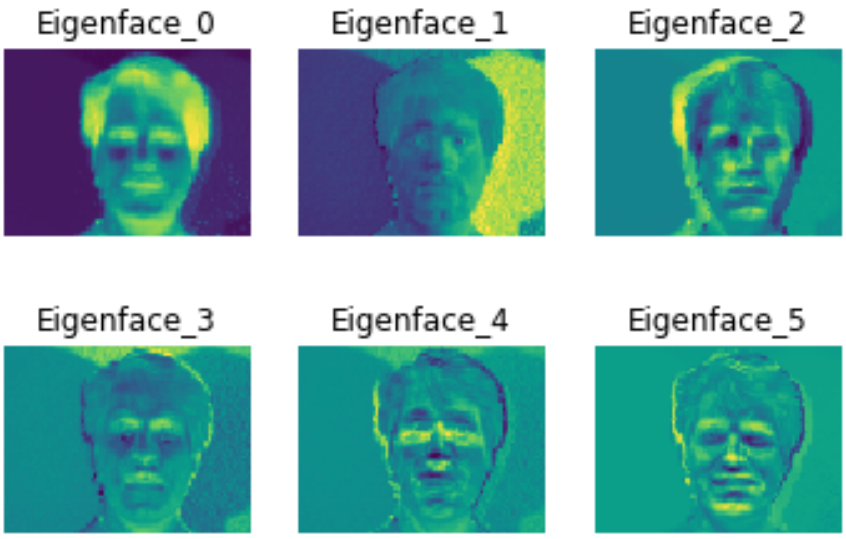

Figure 7: eigenfaces for subject 1

Figure 8: eigenfaces for subject 14

2. Now we will perform a face recognition task.
   For doing face recognition through PCA we proceed as follows. Given the test image subject01-test.gif and subject14-test.gif, we vectorize each image. Take the top eigen- faces of Subject 1 and Subject 14, respectively, project the 2 vectorized test images using the vectorized eigenfaces to obtain scores, respectively.
   (Hint: use $(eigenface_1)^T(testimage)$)
   Report four scores: (1) projecting test image of Subject 1 using eigenface of Subject 1; (2) projecting test image of Subject 1 using eigenface of Subject 14; (3) projecting test image of Subject 14 using eigenface of Subject 1; (4) projecting test image of Subject 14 using eigenface of Subject 14.
   Explain whether or not (and how) can you recognize the faces of the test images using these scores.

   $(Eigenface_1^T)(sub01)$: -9452.06
   $(Eigenface_1^T)(sub14)$: -8938.22
   $(Eigenface_14^T)(sub01)$: -8456.75
   $(Eigenface_14^T)(sub14)$: -9959.05

   We can recognize the faces of the test images using these scores. The geometic defination of two vector multiplication is:

   $$a \cdot b = \|a\|\|b\| \cos\theta$$

   We know that if two vectors are totally unrelated to each other (the similarity is 0), then they are orthogonal to each other, thus $\cos\theta = 0$, which means the result of the multiplication is $\|a\|\|b\|0 = 0$. If they are very similar to each other, the directions of both vector will align, thus $\cos\theta = 1$. The result of the multiplication will be $\|a\|\|b\|1 = \|a\|\|b\|$.

   In summary, the higher the absolute score is, the more similar between the test image and the eigenfaces.

5

The objective of this question is to reproduce the ISOMAP algorithm results that we have seen discussed in class. The file isomap.mat (or isomap.dat) contains 698 images, corresponding to different poses of the same face. Each image is given as a 64 64 luminosity map, hence represented as a vector in $\mathbb{R}^{4096}$. This vector is stored as a row in the file. [This is one of the datasets used in J.B. Tenenbaum, V. de Silva, and J.C. Langford, Science 290 (2000) 2319-2323]

(a) Choose the Euclidean distance between images (i.e. in this case a distance in $\mathbb{R}^{4096}$). Construct a similarity graph with vertices corresponding to the images, and connecting each image to the k nearest neighbors in the dataset, for k = 100. (Notice that as a result, each vertex is in general connected to more than k neighbors.) Visualize the similarity graph (e.g., plot the adjacency matrix where weights are shown using intensity).

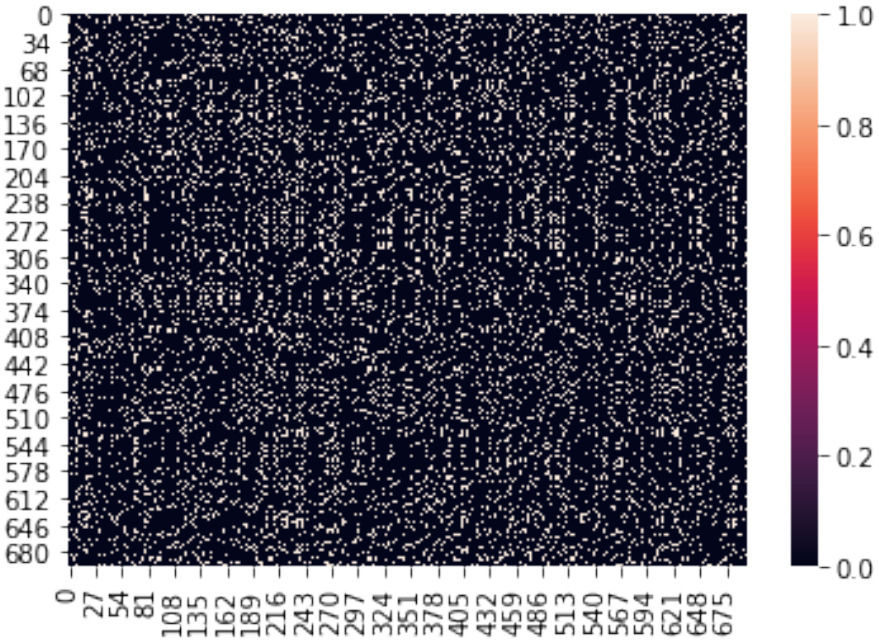

Figure 9: visualize the adjacency matrix

The adjacency matrix of the images is ploted as a heatmap.

(b) Implement the ISOMAP algorithm and apply it to this graph to obtain a d = 2- dimensional embedding. Present a plot of this embedding. Find three points that are close to each other and show what they look like. Do you see any similarity among them?

After finish the ISOMAP algorithm, I get the first two leading eigenvectors and eigenvalue, used them to reconstruct a 2-dimensional data matrix which is the result of non-linear dimension reduction. Plot the matrix as shown in Fig 10, then I chose two sets of three points that are close to each other. They are 691,140,154 and 406,631,278.

Images are shown as in Fig 11. It is obvious that there are high level of similarites among them. For 691,140,154, all heads are facing lower right. For 406,631,278, all head are facing left.

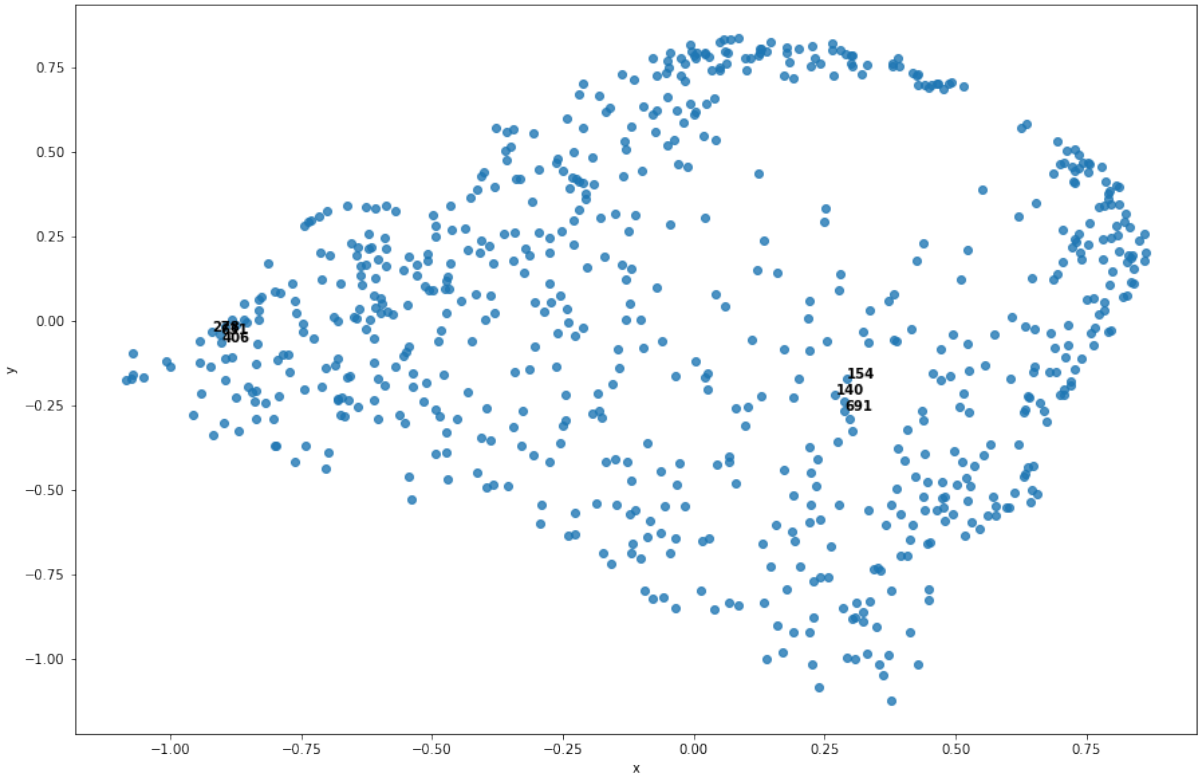

Figure 10: reconstruction of data after ISOMAP algorithm

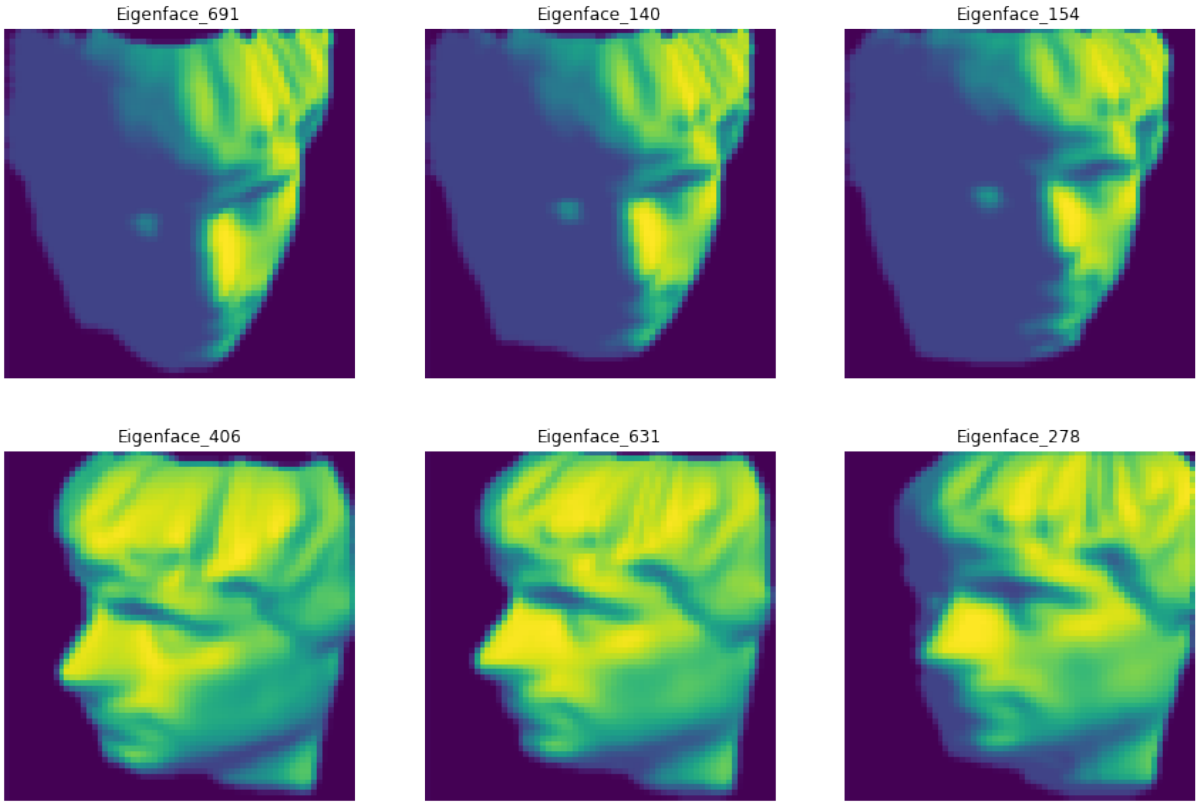

Figure 11: images of three close points