CS 276A / STATS M231 Project 1

Yutong Lu, ID: 005027184

October 28, 2017

1 Active Appearance Model (AAM) for face reconstruction and synthesis

1.1

a. Compute the mean face for the training images with no landmark alignment.
By calculating the mean value of a position of each training image, we can get the mean face simply.

Figure 1: Mean face

b. Compute the first k-eigen faces for the training with no landmark alignment.
If directly calculating the eigenvectors of the scatter matrix in Matlab, it will go wrong due to the large dimension of the scatter matrix. Therefore, we apply a tricky way to calculate the eigenvectors.

$$Ax = \lambda I x$$

Let $A = diff * diff^T$, where $diff$ is the difference between training images and the mean image. Then, the above equation can be expressed into

$$diff * diff^T * x = \lambda I x.$$

By multiplying $diff^T$ on both sides of the equation, we can get

$$diff^T * diff * diff^T * x = \lambda * diff^T * x.$$

Now, $diff^T * v_i$ becomes the eigen-vector of $diff^T * diff$, so that the eigenvectors we need will be easily to compute and it is nothing but $diff * diff^T * v_i$.
Therefore, in this way, we transform the high-dimensional eigenvector problem into a low-dimensional problem and make it easy to solve.

c. Display the first K=20 eigen-faces.

From the method mentioned above, we hence can produce eigen-faces for the training images. Figure 2. shows the first 20 eigen-faces.

Figure 2: First 20 eigen-faces

d. Use first 20 eigen-faces to construct the remaining 27 test faces.
For testing images, we can get the reconstructed faces faces by projecting them into the eigen-faces $e_1, e_2, ...e_{20}$, which is given by

$$\hat{X}_k = mean + \sum_{i=1}^{d'} a_i * e_{Xi},$$

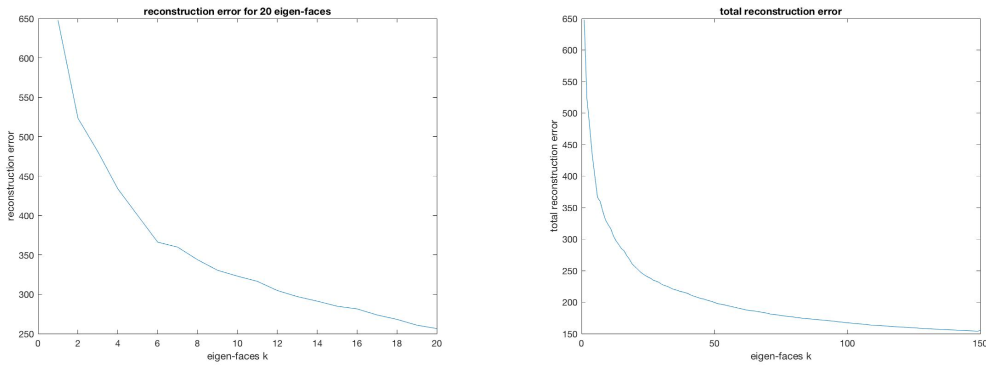
where d' is the no. of eigen-faces used. Figure 3. shows the 27 reconstructed testing images.

Figure 3: Reconstructed testing images

e. Plot the reconstruction error.
Reconstructed error can be defined as:

$$Error = \frac{1}{256 * 256} \sum_{i,j} (\hat{X}(i,j) - X(i,j))^2.$$

We follow this equation to build the codes and reconstruction error over 20 eigen-faces and reconstruction error over 149 eigen-faces are shown in Figure.4.



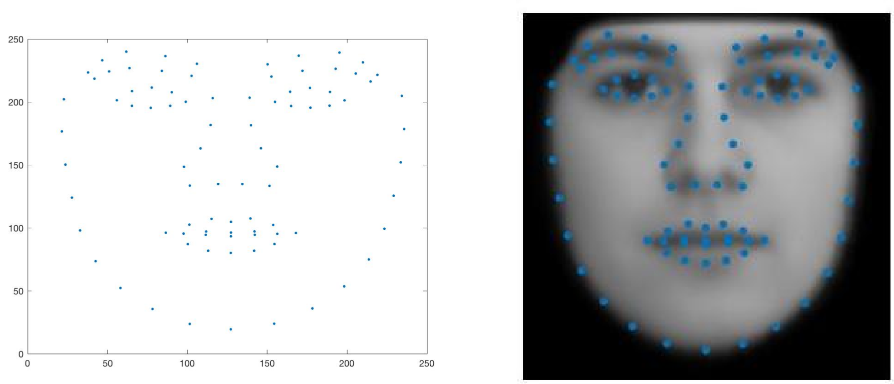
(a) k = {1, 2, ..., 20}

(b) k = {1, 2, ..., 149}

Figure 4: Reconstruction Error over k eigen-faces

1.2

a. Compute the mean of the landmarks for the training faces.
The codes are similar as part (1), the only difference is previously we calculate the images, but for this part we calculate the landmarks. Figure. 5 shows the mean of the landmarks and the mean landmarks with the mean face.



(a) Without the mean face

(b) With the mean face

Figure 5: Mean of the landmarks

b. Compute the first k eigen-warping of the landmarks of the training faces.
The calculating procedures are just as same as in subsection1.1. Therefore, it will not be described here again.

c. Display the first 5 eigen-warpings.

Figure 6. shows the first 5 eigen-warpings separately and Figure 7. shows all eigen-warpings in one picture with the mean face shown to see clearly how these eigen-warpings are located.

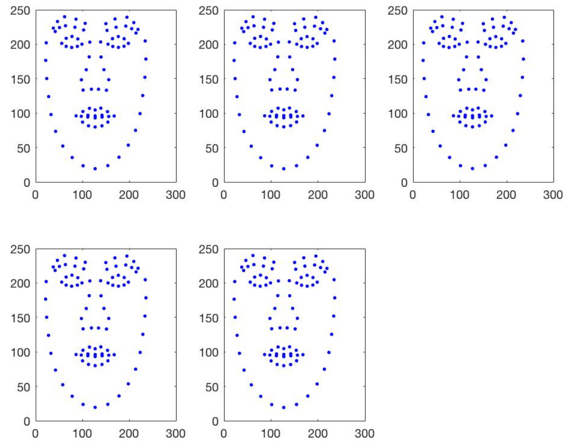


Figure 6: First 5 eigen-warpings shown separately

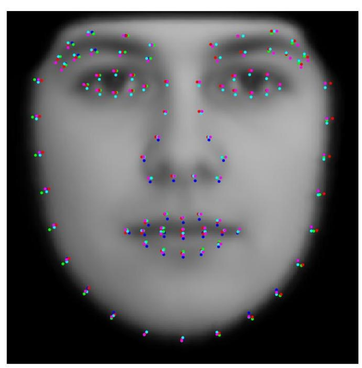


Figure 7: First 5 eigen-warpings shown together

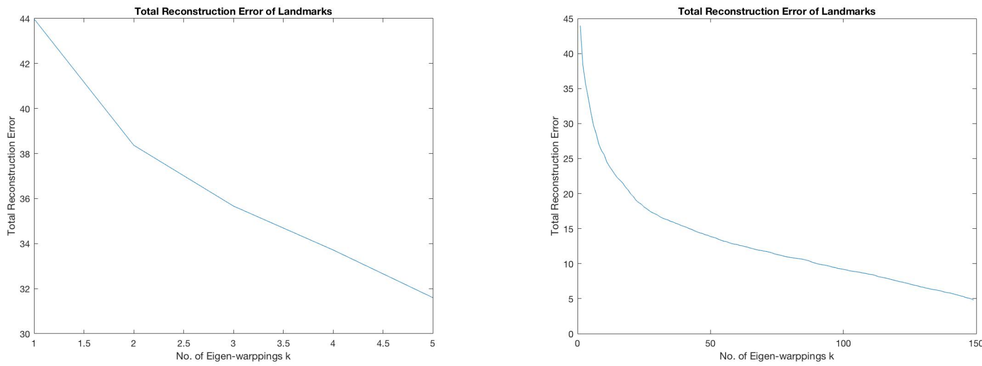
d. Use the first 5 eigen-warpings to reconstruct the landmarks for the test faces. Figure 8. shows both the reconstructed landmarks and the original landmarks in one picture together with the original testing faces for comparison. It can be seen that the reconstructed landmarks and are very clear and there are just small biases between the reconstructed landmarks and the original ones.

Figure 8: Reconstructed landmarks with original landmarks for the test faces

e. Plot the reconstruction error.
Reconstruction error can be defined:

$$Error = \sqrt{\sum_{u,v} (\hat{X}(u,v) - X(u,v))^2}.$$

Therefore, we plot the reconstruction error over 20 eigen-warpings, $k = 1, 2, ..., 20$ and also error over 149 eigen-warpings, $k = 1, 2, ..., 149$ in Figure. 9.



]

(b) k = {1, 2, ..., 149}

Figure 9: Reconstruction Error over k eigen-warpings

1.3

a.
The idea of this step is to align the training images by warping their landmarks to the mean position. Since each training image has its specific landmarks, so warping their landmarks, we can warp the training image. We therefore can show the aligned training images (totally 150 training images) in Figure. 10. After computing the eigen-faces for the aligend images, we show the top 10 eigen-faces in Figure. 11.

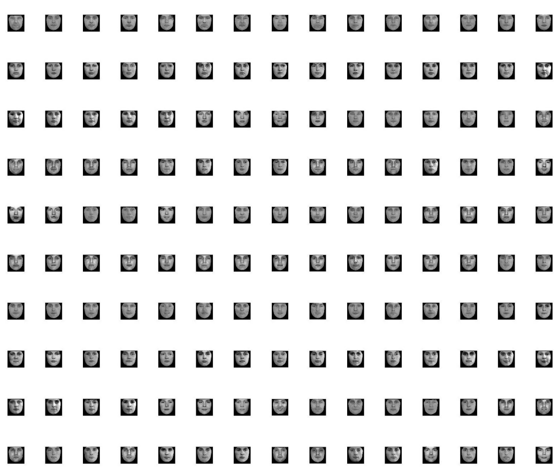


Figure 10: Aligned training images

Figure 11: Eigen-faces from the aligned images

b.

The idea of step is to firstly warp the testing images into the aligned images, and use k eigen-faces to reconstruct the aligned images. In the meantime, we also use the 10 eigen-warpings to get the reconstructed landmarks from the original landmarks. Secondly, we combine these aligned images with reconstructed landmarks to get the reconstructed testing images.

Figure. 12 shows the reconstructed landmarks. Figure. 13 shows the reconstructed images at mean position and Figure. 14 shows the reconstructed testing images using the landmarks got from Figure. 12. Figure. 15 shows the original testing images. It can be seen that the reconstructed images are more smooth than the original ones since we have loss on both geometry and appearance. This is because small features such as wrinkle on human faces cannot be reconstructed by our algorithm.

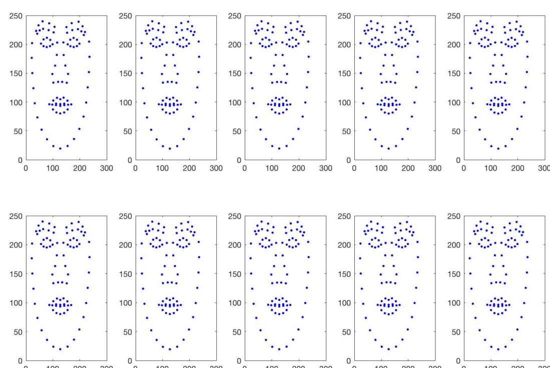


Figure 12: Reconstructed landmarks

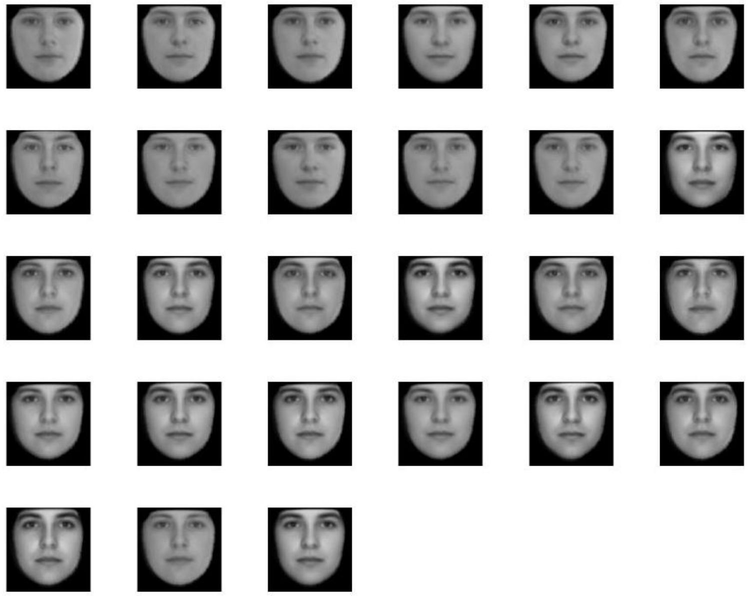


Figure 13: Reconstructed testing images at mean position

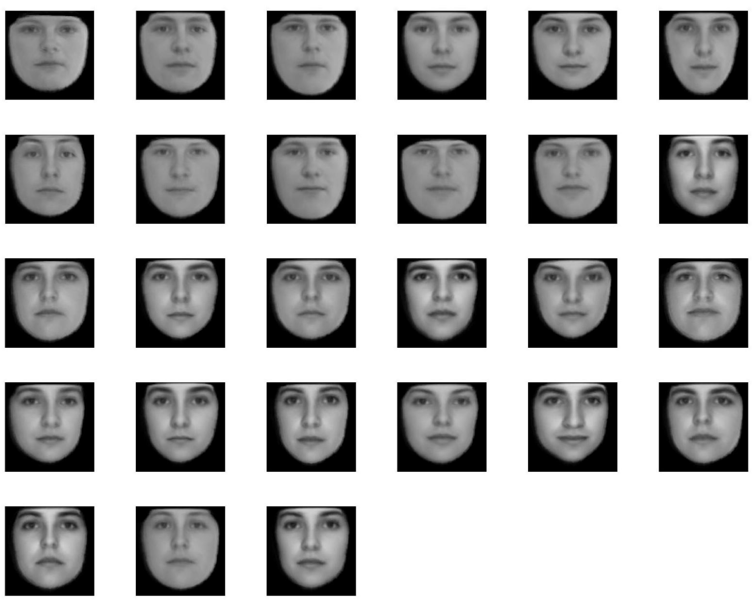


Figure 14: Reconstructed testing images using landmarks from Figure. 12

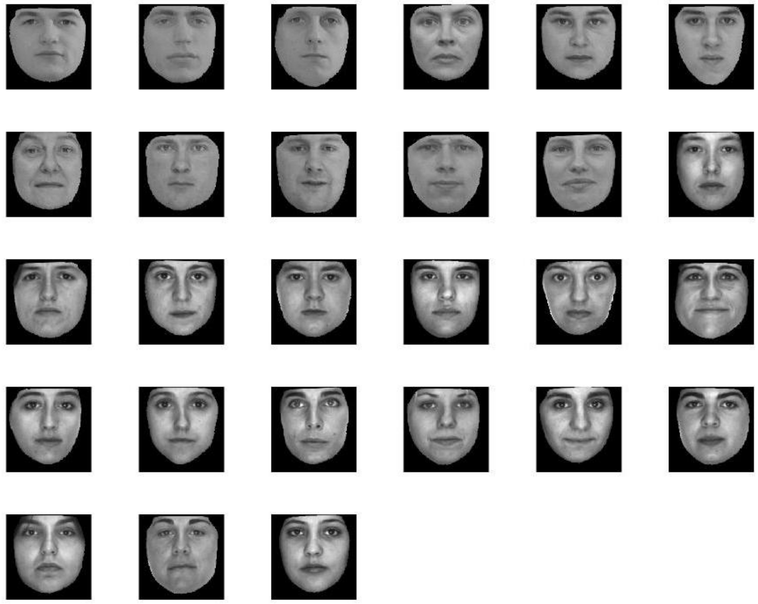
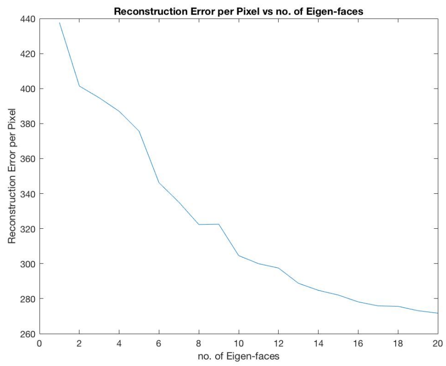
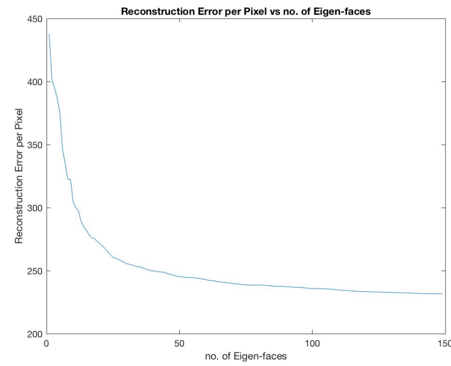


Figure 15: Original testing images

c. Plot the reconstruction errors per pixel against the number of eigen-faces k. We can see the reconstruction error (shown in Figure. 16) in this part is smaller than that in part 1.1 without warpings.



(a) k = {1, 2, ..., 20}



(b) k = {1, 2, ..., 149}

Figure 16: Reconstruction error of testing images with landmark alignment over k eigen-faces

1.4

Synthesize random faces by a random sampling of the landmarks (based on the top 10 eigen-values and eigen-vectors in the wrapping analysis) and a random sampling of the appearance (based on the top 10 eigen-values and eigen-vectors in the intensity analysis). Here, we adopt the normal distribution to project the appearance and geometry information on each axis. In detail, we can write:

$$a_i, b_i \sim N(0, \lambda_i), i = 1, 2, 3, ..., 10$$

where λ is the eigenvalue of each axis. A normal distribution is projected into the image space:

$$\hat{X} = m_X + \sum_{i=1}^{k} a_i * e_{Xi}$$

and also into the landmark space:

$$\hat{J} = m_J + \sum_{i=1}^{k} b_i * e_{Ji}.$$

Then, we can display 20 synthesized face images in Figure. 17.

Figure 17: Synthesized 20 random faces.

2 Fisher Linear Discriminant (FLD) for gender discrimination

2.1

In this question, 78 male faces and 75 female faces are made into training sets and 10 male faces and 10 female faces are made into testing sets.
As before, the dimension of the scatter matrix is pretty high (can be 65536*65536) so that Matlab would not respond to it. There are two ways to solve this problem: the first way is to use the algorithm provided by this course to find ω and the second way is to compute the Fisher faces over the reduced dimensions as previous questions. By implementing the first method into Matlab, the result is shown in Figure. 18. To distinguish male faces and female faces, we define:

$$\omega^T x + \omega_0 > 0 : malefaces$$
$$\omega^T x + \omega_0 < 0 : femalefaces$$

By setting ω_0 to 0, it can be seen that only one male sample is in the incorrect area so that we achieve an accuracy of 95%.
By using the second method, we can reduce the dimensions to a smaller number. For example, we show the Fisher results of using 10 reduced dimension and 50 reduced dimension in Figure. 19. It can be seen when using PCA method to reduce the dimensions, the

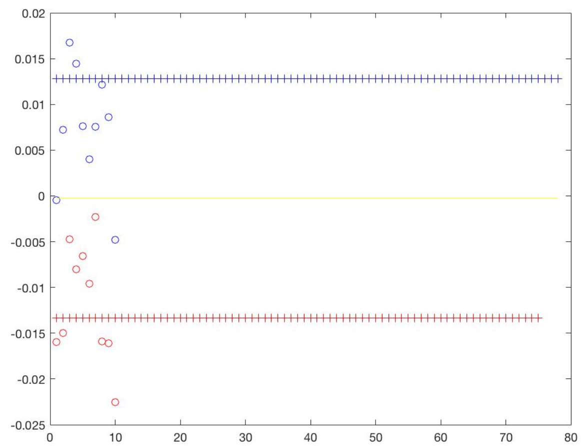
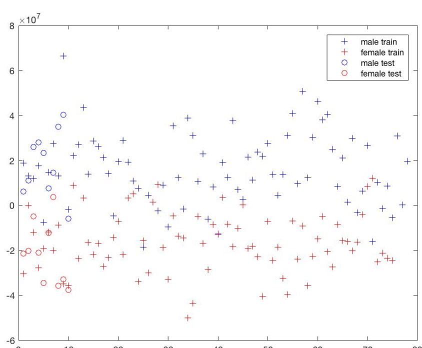
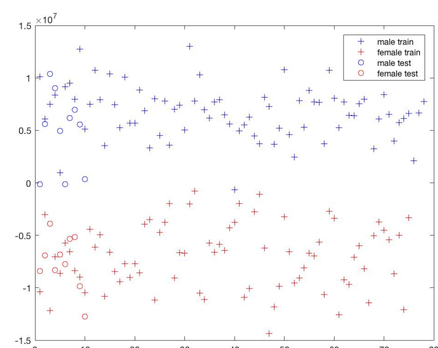


Figure 18: Fisher face result.



(a) reduced dimension to 10 by PCA



(b) reduced dimension to 10 by PCA

Figure 19: Fisher result over reduced dimensions by PCA

boundary will be clearer if we just decrease the dimensions to a moderate number. If we reduce it to just only 10 dimension, the boundary can be quite vague since we may lose too much information, however in this case the boundary still can be recognized.

2.2

Compute the Fisher face for the key point (geometric shape) and Fisher face for the appearance (after aligning them to the mean position) respectively. Each face is projected to a 2D-feature space, and visualize how separable these points are.
In this problem, we use PCA to get the Fisher result and before this, we need to firstly warp both images and landmarks to their mean value. We use

$$(w_{img}^T * img,\ w_{landmark}^T * landmark) \quad -> \quad (x, y)$$

and then plot them into a 2-D feature space.
Figure. 20 shows the Fisher result for the training sets with 10-dimension reduced by PCA.
Figure. 21 shows the Fisher result for the testing sets with 10-dimension reduced by PCA.
Figure. 22 shows the Fisher result for the training sets with 50-dimension reduced by PCA.
Figure. 23 shows the Fisher result for the testing sets with 50-dimension reduced by PCA.

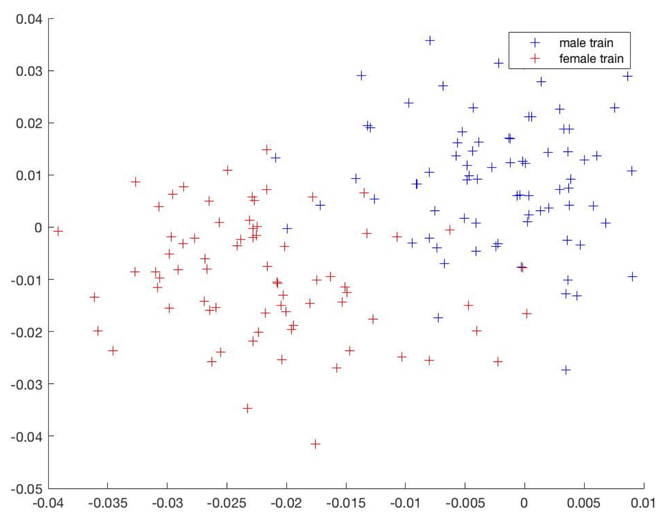


Figure 20: 2D projected points of training sets with 10-dimension by PCA

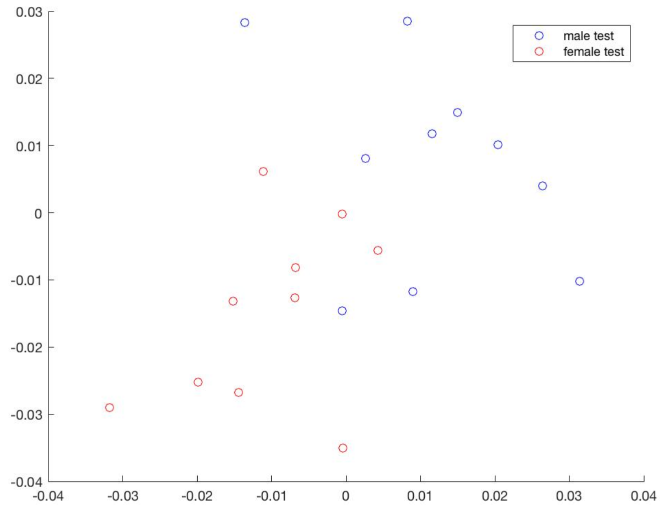


Figure 21: 2D projected points of testing sets with 10-dimension by PCA

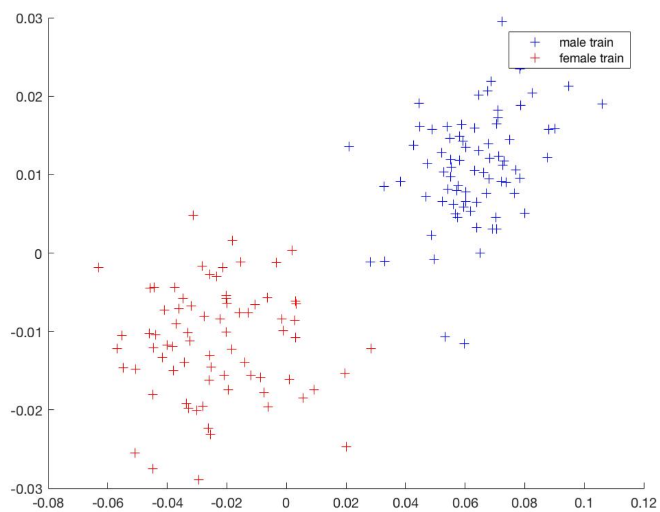


Figure 22: 2D projected points of training sets with 50-dimension by PCA

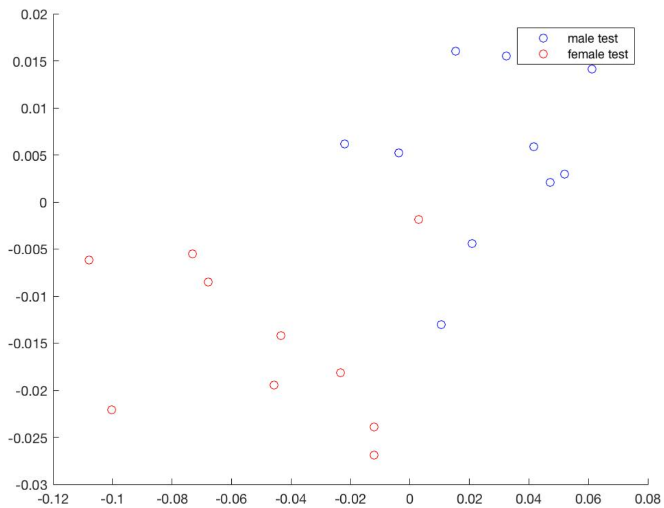


Figure 23: 2D projected points of testing sets with 50-dimension by PCA

By projecting both training sets and testing sets into 2-D space, we still can see the boundary between male faces and female faces and 50-dimension PCA perform better than 10-dimension PCA.