STATS 231A Project 1

Bruce Zhu (UCLA)

Oct 20, 2017

Part 1:

(1). Compute the mean and first k eigen-faces for the training images with no landmark alignment. Display the first K=20 eigen-faces and use them to reconstruct the remaining 27 test faces. Plot the total reconstruction error (squared intensity difference between the reconstructed images and their original ones) per pixel (i.e. normalize the error by the pixel number, and average over the testing images) over the number of eigen-faces k.

I computed the mean and first k eigenfaces for the training images. The mean face is shown below, representing the mean feature of the training faces.

I also used function pca() in MATLAB to compute the eigenfaces. the image below shows the first 20 eigen-faces.(notice: since the pca has different algorithm (svd, pairwise) inside, the eigenfaces may be a little different)

we can see that the first two eigenfaces are similar to real human face, which indicates that these two faces contain most information of faces, and the others may only represent partial information of the faces.

Then I used eigen-faces to reconstruct the remaining 27 test faces. Let the ei denote the ith eigenfaces:

$$\langle I-m, ei \rangle = bi,\ i=1,\ldots,k.$$

where m is the mean face of the training image, Then we can reconstruct faces as follows:

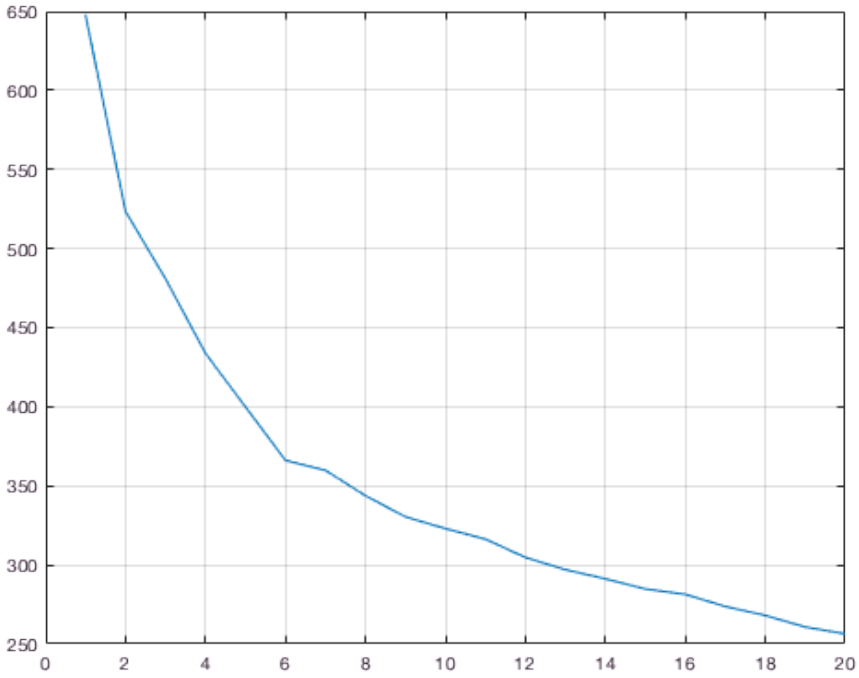
$$I\_hat = mI + sum(bi * ei) \quad \text{where i from 1 to k}$$

I used first k=20 eigenfaces and the testing images, we can see the image below. We found that compared to the testing images, the reconstructed faces lose some details in both geometry and appearance perspectives. Since we don't align faces through landmarks so that pca( ) cannot work so well.

a. Reconstructed testing faces

b. Original testing faces

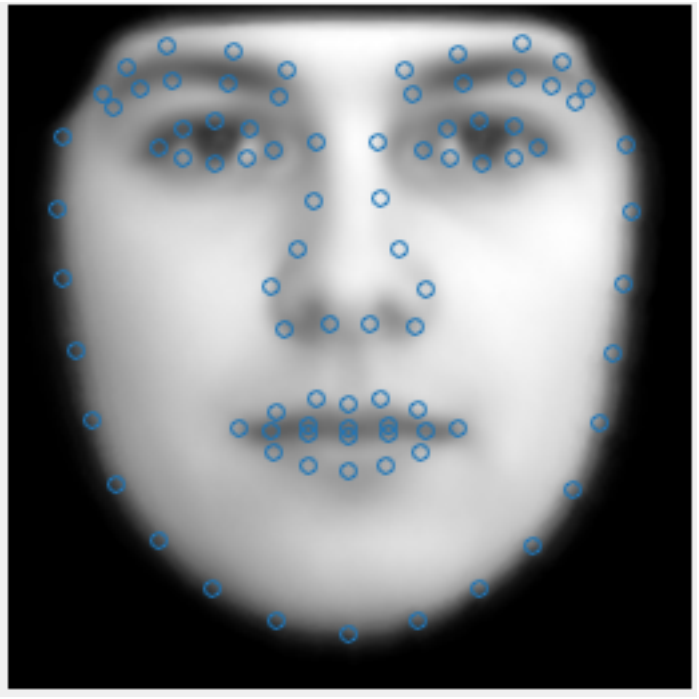
I plot the total reconstruction error over the number of eigenfaces k (from 1 to 20),



and we could see, as we increase the number of the eigenfaces the total reconstruction error will go down; however, the slope become more and more smaller because the first couple eigenfaces contain more information.

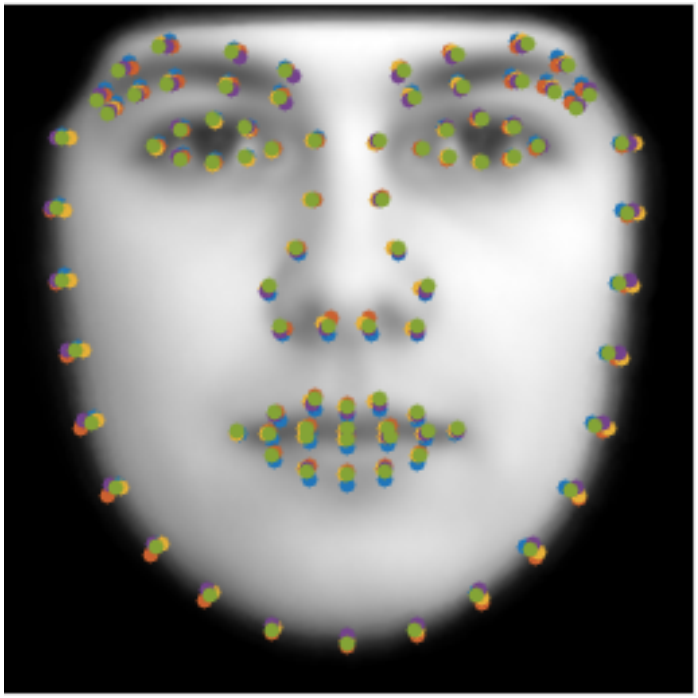
(2) .Compute the mean and first k-eigen-warpping of the landmarks for the training faces. Here warping means displacement of points on the face images. Display the first 5 eigen warppings (you need to add the mean to make it meaningful), and use them to reconstruct the landmarks for the test faces. Plot the reconstruction error (in terms of distance) over the number of eigen-warppings k (again, the error is averaged over all the testing images).

I compute the mean landmarks of training images as below.



and again, similar as the first quesition to use pca() to compute the eigen-landmarks (I also multiplied the sqare root of the respective eigen-values

and then added the mean landmarks) as shown below we summarize 5 eigen landmarks, which are very similar to each other. This is probably because the difference between eigen landmarks are very small when add mean landmarks up.



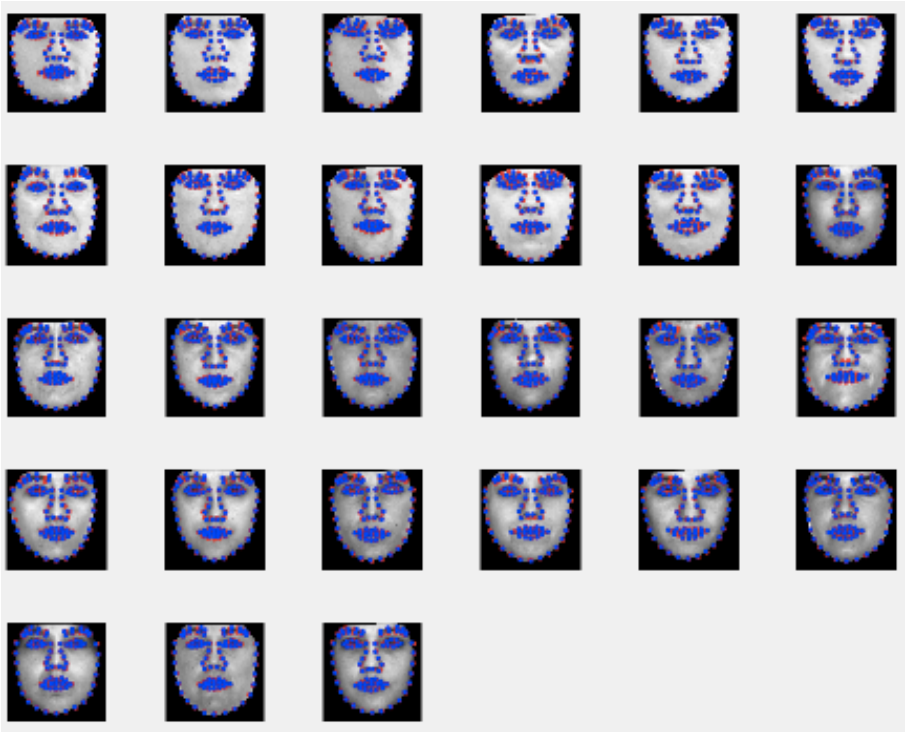
I also use the first k=5 eigen landmarks to reconstruct the test landmarks, similar as the above question

$$\langle I-m, ei\rangle = ci,\ i=1,\ldots,k.$$

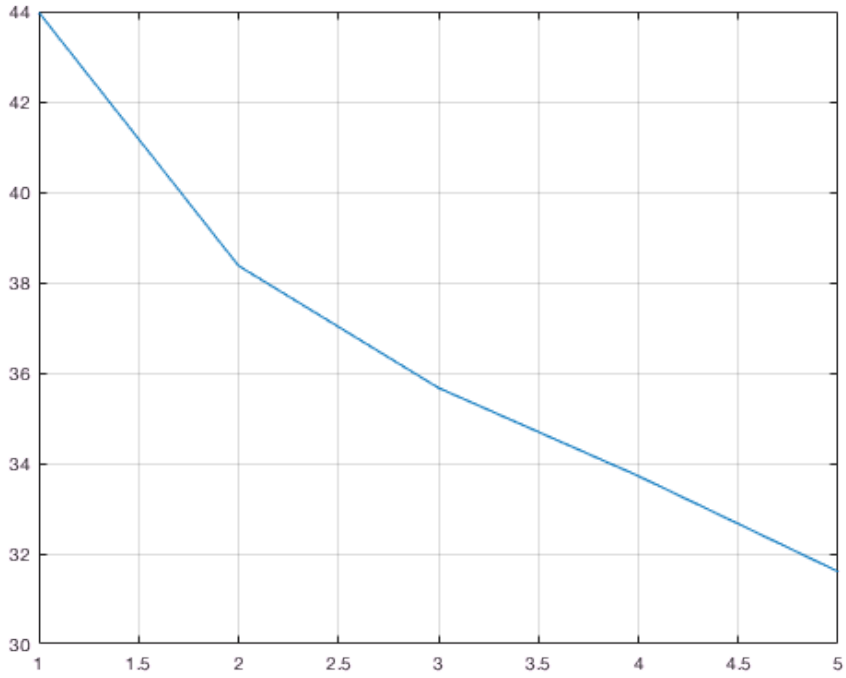
where m means the mean landmarks of training landmarks, and ei be the ith eigen-landmark. Then we construct the landmarks as follows:

$$L\_hat = m + \sum(ci * ei) \quad \text{where i from 1 to k}$$

Showing the reconstructed landmarks using k=5 eigen landmarks and original testing landmarks in the image below. I plot them in the same figure. The red points are the reconstructed landmarks while the blue points are the original landmarks, we can see that they are very similar to each other which means the first five eigen-landmarks include the major information.



I also average the reconstruction error over the first k=5 eigen-landmarks, and plot the error over the number of eigen-landmarks which is 5. The image shown below,



and we can find out as the number of eigen-landmark increase the total reconstruction error will go down, but the slope will become smaller, which means the first couple eigen-landmarks contain more information than the others.

(3). Combine the two steps above. Our objective is to reconstruct images based on top 10 eigen-vectors for the warping and then top k (say 10) eigen-vectors for the appearance, in the context of compressing the face images and communicate through a network with small number of bits. For the training images, we first align the images by warping their landmarks into the mean position (interpolation between landmarks is needed), and then compute the eigen-faces (appearance) from these aligned images. For each testing face: (i) project its landmarks to the top 10 eigen-warpings, you get the reconstructed landmarks. (here you lose a bit of geometric precision of reconstruction); (ii) warp the face image to the mean position and then project to the top k (say k=10) eigen-faces, you get the reconstructed image at mean position (here you further lose a bit of appearance accuracy). (iii) Warp the reconstructed faces in step (ii) to the positions reconstructed in step (i). Note that this new image is constructed from 20 numbers. Then compare the reconstructed faces against the original testing images (here you have loss in both geometry and appearance). (iv) Plot the reconstruction errors per pixel against the number of eigen-faces k.

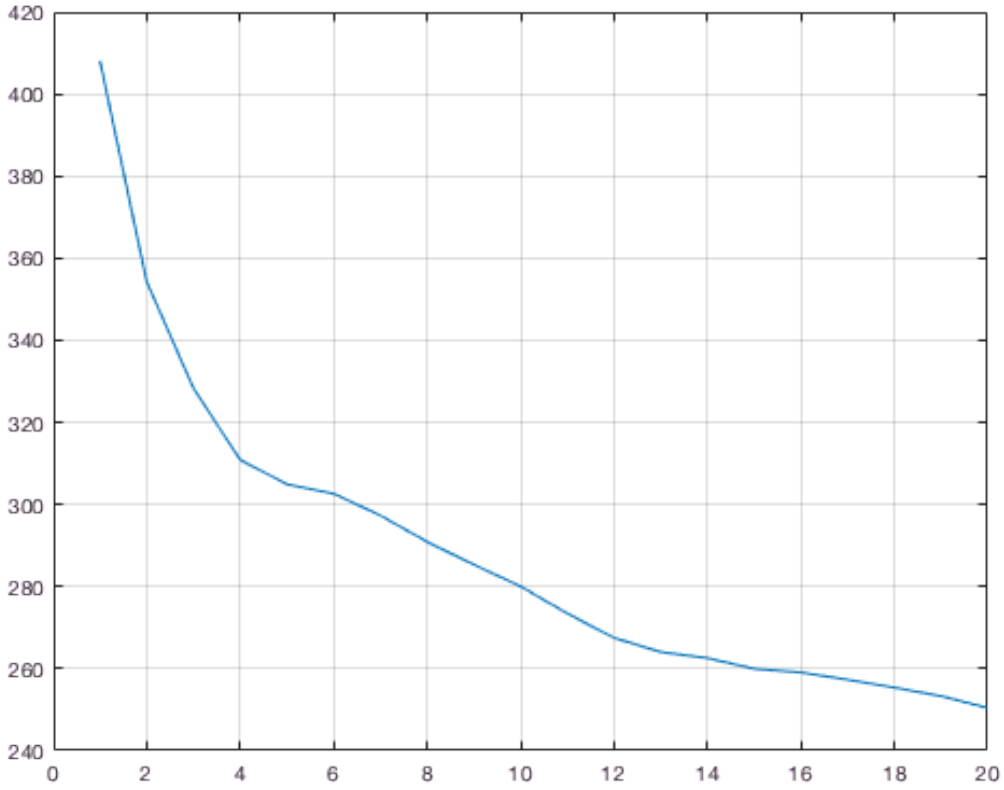
In this question, we need to reconstruct faces with landmark alignement. I plot the reconstruced faces and original testing faces respectively as below.

a).Reconstructed testing faces

b).Original testing faces

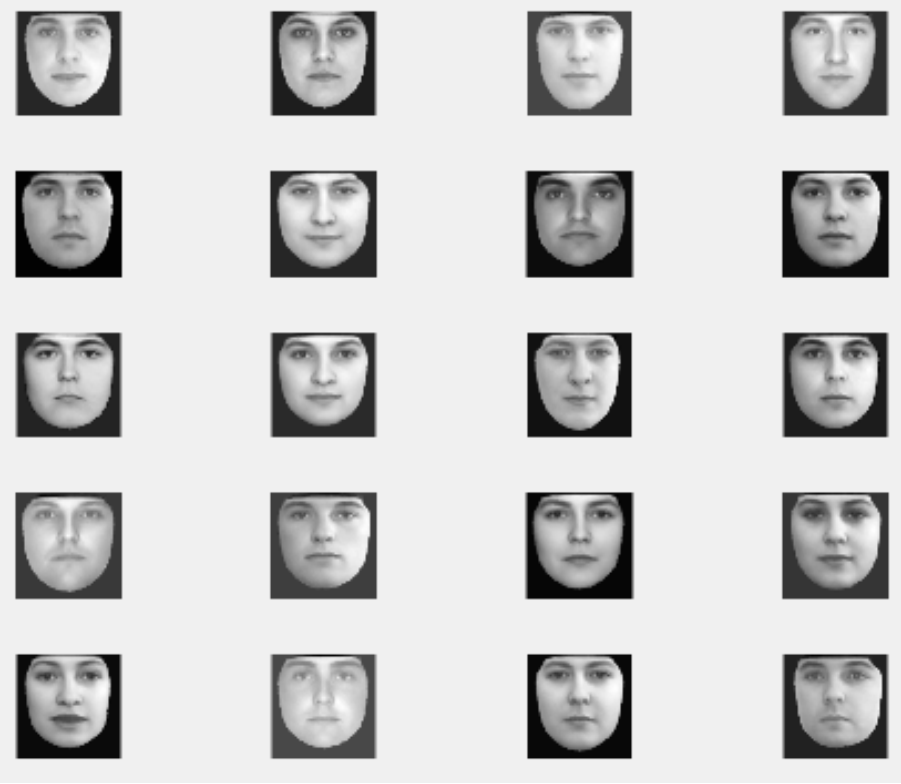
Compared to the reconstructed testing face in the first question, the reconstructed faces above are closer to the original testing faces in my eyes. I think this is due to the align process, which can make the PCA work better. However, our algorithm is still not able to well describe the small features, like beard or wrinkles due to the constraint of PCA.

I also plot the reconstruction error below, we can see the error rate decrease as the k goes up and the slope also become smaller and smaller as above. However I found the average error is not much smaller than the one in the first question. This probably because we lose information twice during the reconstruction process.



(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). Display 20 synthesized face images. (As we discussed in class, each axis has its own unit, that is, the square-root of its eigen-value).

In order to synthesize the random faces. I need to randomly sampling landmarks and appearance respectively based on their first 10 eigenvalues. I choose let f_i, l_i be the random numbers generated by Normal distribution with mean=0 and sigma equals the square root of eigenvalues, where i from 1 to 10 axis. Then we can sum the product of fi and eigenface (i), and then add the mean of the warped training images, giving us the randomly warped face dataset. I can also conduct the similar approach to the landmarks. Then we can use the method in the last question to generate random faces (at the end, I warped sample faces from mean landmark to specific positions according to sampled landmarks.) The Synthesized random faces are shown at below.



I found out that our Synthesized random faces are pretty similar to a real human face, which prove that our algorithm worked well.

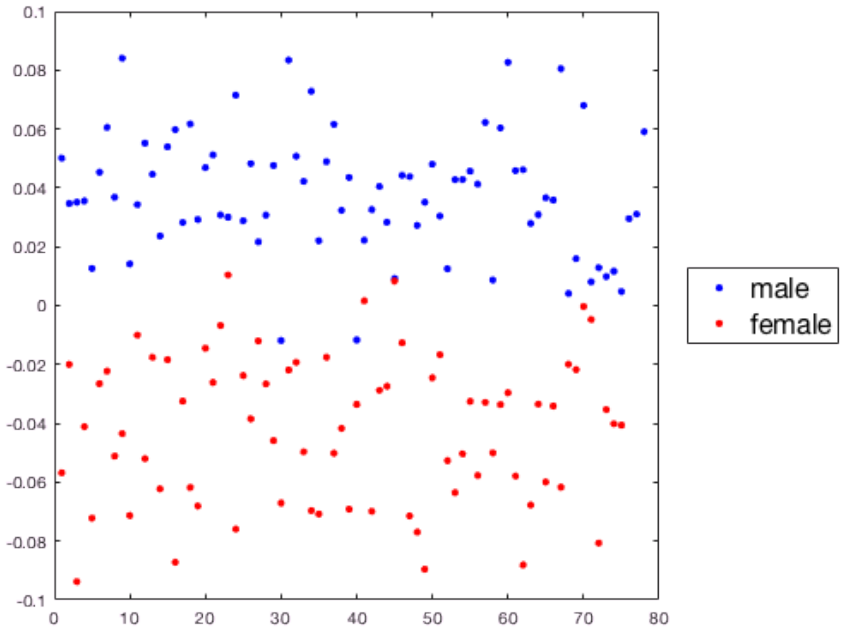
Part 2: Fisher Linear Discriminant (FLD) for gender discrimination

We have divided the 177 faces into male (88) [in the male face folder, Face 57 is removed as a duplicate] and female faces (85), plus 4 unknown (which is hard even for human eyes to tell based on the faces alone). Then you choose 10 male and 10 female as the testing set. The remaining faces (except the 4 unknown) will be used as training sets.

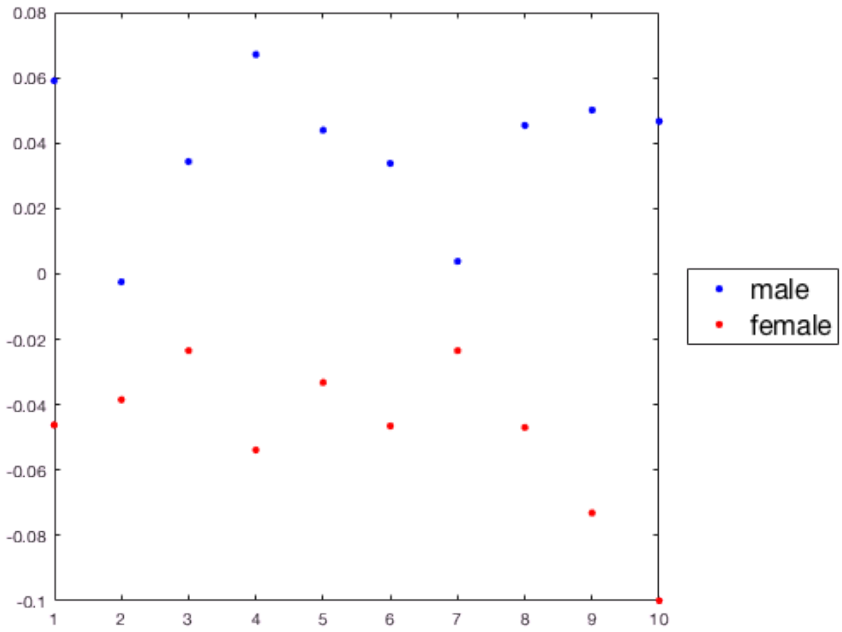
(5) . Find the FLD or Fisher face that distinguishes male from female using the training sets, and then test it on the 20 testing faces and see how much percentage is right. This Fisher face mixes both geometry and appearance difference between male and female.

I want to use the training dataset to calculate the Fisher face to distinguish male and female faces; however, the dimensions of the within-class scatter matrix are too high, which is 65536*65536. Calculating the inverse of within-class scatter matrix will cause out-of-memory problem in the MATLAB. Thus, we computed the Fisher faces over the reduced dimensions in Question1. For each face image, it has been reduced to 20 dimensional appearance vectors. Therefore, we can change the within-class scatter matrix to 20*20 dimensions, which can be inversed easily.

We calculate the direction vector W and plot the projected points of male and female faces in the figure below. Let W0=0, we could see that Fisher face line transpose (w)*X+W0 separates faces of different gender points very well.



I use the same Fisher face line to discriminate testing faces, if (transpose (w)*X+W0) bigger than 0, we could think it is a Male face, and if smaller than 0, we could think it is a female face.



As shown above, among the 20 test images, we see only the second male face is misclassified, thus the accuracy is 95%.

(6) In this question, we calculate Fisher face for the key point (geometric shape) and Fisher face for the appearance (after aligning them to the mean position) respectively, and each face is projected to a 2D space. I plot the graph below, and we can see most of the male and female points are separated well, but there are still some points in the middle, which are difficult to classify.