405410081 邱莉淩

CV HW1

How to use this code

Open hw1_main and run this code in matlab.

Method Description

(a)

I loaded every image (120, 165, 3) in train set and test set in the memory, and used rgb2gray() to turn them into grayscale images(120, 165). After reshaping them into array (19800, 1), I merged them into one matrix respectively.

First, I computed the covariance matrix of the matrix A (19800, 1300) merged by train images after every column subtracted the mean image. Second, I used eig() in matlab to compute eigenvalue and eigenvector and keep the eigenvector with the highest 9 eigenvalue as the eigenface.

```
function vectors = computeEigenface(A, d)
    n = size(A, 1);

    A_ = mean(A.');

    As = [];
    for i = 1:size(A, 2)
        As = [As A(:, i) - A_.'];
    end

    C = As * As.' / (size(A, 2) - 1);

    [eigenvector eigenvalue] = eig(C);
    eigenvalue = diag(eigenvalue);

    eigens = [eigenvalue eigenvector.'];
    eigens_sorted = sortrows(eigens, 'descend');
    featureVector = eigens_sorted(:, 2:n+1).';
    vectors = featureVector(:, 1:d);
end
```

(a)-i

Reshape eigenvectors with the highest 9 eigenvalues into matrix (120, 165). I used mat2gray() to turn the range of eigenvector into [0, 1] before I use imwrite() to write images to graphics files.

```
function outputEigenfaces(vectors, h, w)
    for i = 1:size(vectors, 2)
        img = vectors(:, i);
        img = reshape(img, h, w);

        img = mat2gray(img);

        num = convertStringsToChars(num2str(i, '%02d'));
        filename = ['.\ans(a)-i\' num '.bmp'];
        imwrite(img, filename);
    end
end
```

(a)-ii

I used rand() to choose a random image in test image every person and turned them into matrix (1, 19800). Consequently, multiply this and the different dimension of eigenvector (19800, d). The projected data would be a matrix (1, d). The output would be projected data multiply the correspond eigenvector, add them all and add the mean image.

```
d_vectors = vectors(:, 1:d(t));
projData = randImg.' * d_vectors;

projFace = zeros(h*w, 1);
for i=1:size(projData, 2)
   tmp = d_vectors(:, i) * projData(i);
   projFace = projFace + tmp;
end

projFace = projFace + mean(TrainImage.').';
projFace = reshape(projFace, h, w);
projFace = mat2gray(projFace);
```

(b)

I computed projected data in different dimension of eigenvector of matrix composed by train images and test image, and used knnsearch() to find every nearest neighbor of test image in train images.

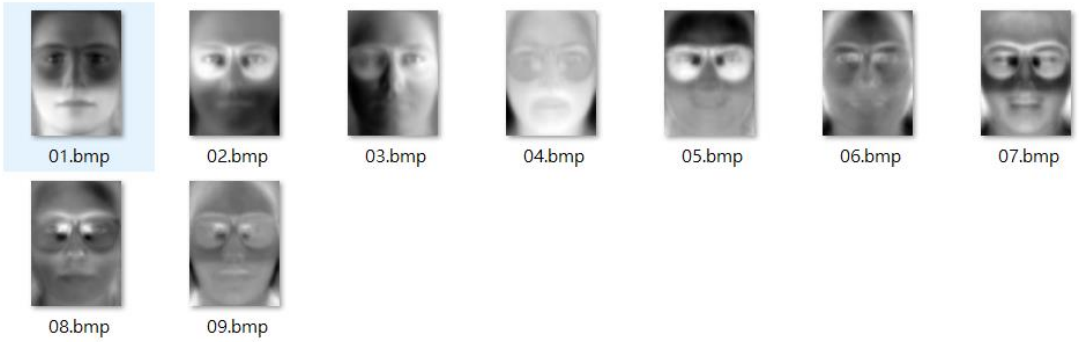
```
%compute projData of train image and test image
d = [1:20];
reg = [];
for i = 1:size(d, 2)
    projDataTrain = TrainImage.' * vectors(:, 1:d(i));
    projDataTest = TestImage.' * vectors(:, 1:d(i));

    index = knnsearch(projDataTrain, projDataTest);
    reg = [reg index];
    regPer = uint8(ceil(reg / 13));
end
```

Experimental Results

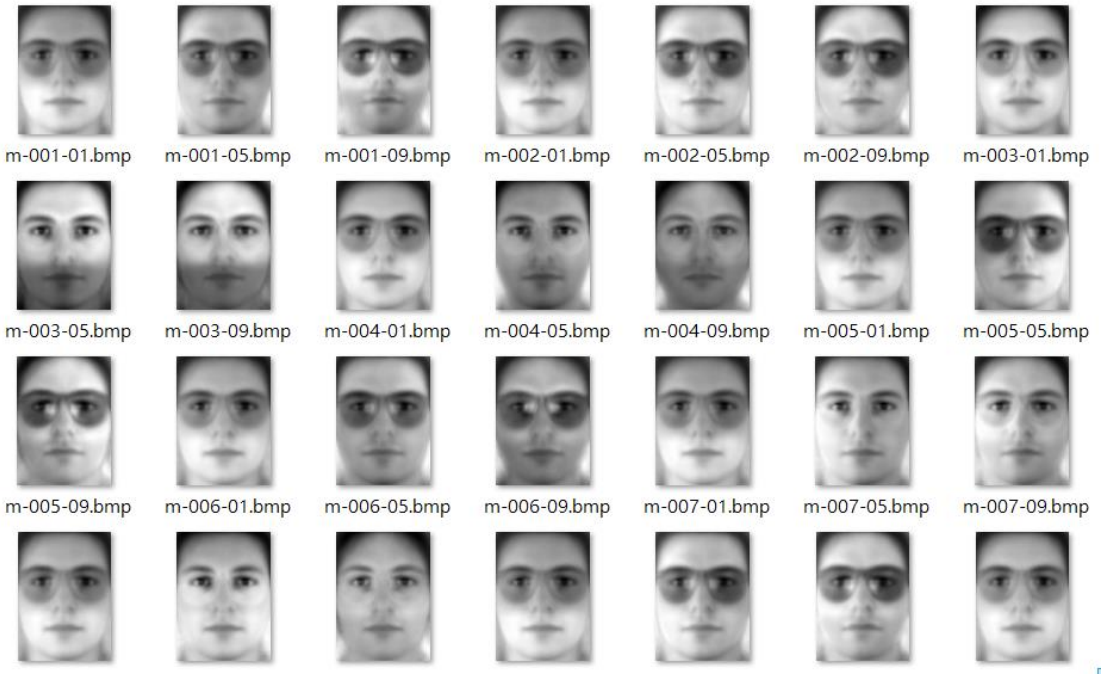
(a)-i

Eigenvectors, named by the order of eigenvalue.



(a)-ii

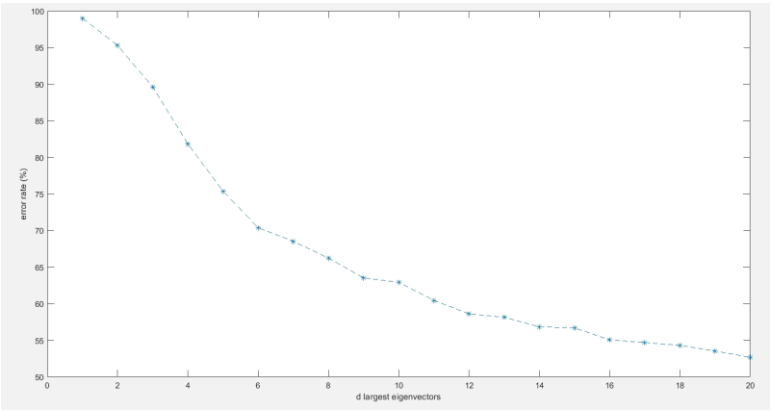
The approximated images by d eigenfaces, named by gender, person number and the dimension. Apparently that using higher dimension can get output more similar.



Error rate using dimension 1, 5, 9.

```
d = 1, error rate = 99.00%
d = 5, error rate = 75.38%
d = 9, error rate = 63.54%
Elapsed time is 209.584887 seconds.
```

Error rate using dimension 1-20. It could be seen that higher dimension can get better correctness.



Discussion

Every picture is in different condition such that light, glasses and scarf, but using PCA to recognize people can get an appropriate performance. It is said that if we have enough data, wearing mask or sunglasses can’t protect ourselves from face recognition.

Problem and Difficulties

Using different function in matlab would get different answer. It would be a little confusing. Otherwise, the performance of my computer is a little bad, so each compute would be 3 or more minutes and kind of wasting times.