Assignment 5

Face Recognition Using PCA and LDA
Computer Vision

Submitted By:
Abhay Singh

1. Face Recognition using Principal Component Analysis

<u>Training</u>:

Reading all training images and reshaping images into a single matrix, where each row represents an image reshaped into a single vector.

```
mainfolder = dir(datapath);
numberofsubfolders = size(mainfolder,1);
X = [];
for i=3:numberofsubfolders
        str = strcat(datapath,'/',mainfolder(i).name);
        content = dir(str);
        sizeofcurrentfolder = size(content,1);
        for j=3:sizeofcurrentfolder
                currentfile = content(j).name;
                img = imread(currentfile);
                [r c] = size(img);
                temp = reshape(img' , r*c ,1);
                X = [X temp];
        end
end
```

Subtracting pixel-wise mean from each of the vector,

```
A = [];
for i=1 : imgcount
        temp = double(X(:,i)) - m;
         A = [A temp];
end
```

Calculating the eigen vectors of *A'A, and choosing the significant ones ignoring the ones whose eigen value is less than 1.*

```
L= A' * A;
[V,D]=eig(L);

M_high = [];
for i = 1 : 135
        if( D(i,i) > 1 )
                M_high = [M_high V(:,i)];
        end
end
```

Calculating the required eigen vector of AA’ and plotting the subset of eigen spaces onto the same figure:

```
eigenfaces = A * M_high;


for l = 1:9
        subplot(3,3,l);
        a = reshape(eigenfaces(:,l),c,r);
        B = imrotate(a,270);
        B = mat2gray(B);
        imshow(B);
end
```

Projecting all the images onto the facespace:

```
projectimage = [];
for i = 1 : 135
        temp = eigenfaces' * A(:,i);
        projectimage = [projectimage temp];
end
```

Testing

Reading all the testing images in a similar fashion as did in training and performing the preprocessing steps:

1. Subtracting the mean from the test image.
2. Projecting the test image onto the facespace.
3. Calculating the euclidean distance from all the training images.
4. Assigning the same class to the test image as of the image for which the euclidean distance is minimum.

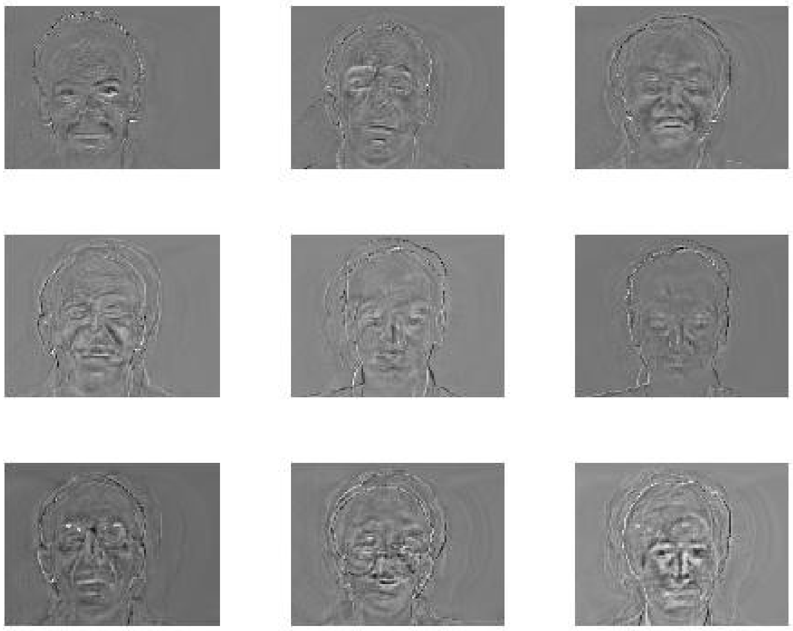
Code Snippet for the above steps:

```
test_image = imread(currenttestfile);
%imshow(test_image)
[r c] = size(test_image);
temp = reshape(test_image',r*c,1);
temp = double(temp)-m;
projtestimg = eigenfaces' * temp;

euclide_dist = [];
for i = 1 : 135
  temp = (norm(projtestimg-projectimage(:,i)))^2;
  euclide_dist = [euclide_dist temp];
end

[min_dist,ind] = min(euclide_dist);
index = [index, ind];
```

Here index vector contains the matched image index for all the testing images.
Accuracy comes out to be **73.34**

EigenSpace

Showing a subset of eigenspace:



2. Face Classification using Fisher Linear Discriminant

Initial Steps are same as of the above method, so now we have all the projected images onto the facespace using pca which is in the vector *projectimage*.

Calculating the mean of each Class:

Total mean in eigenspace:

```
        mean_PCA = mean(projectimage,2);
```

Now we need to estimated two matrix known as Within Scatter Matrix and Between Scatter Matrix which are calculated as follows:

```
for i = 1 : numberofclasses
        m_fisher(:,i) = mean(( projectimage(:,((i-1)*9+1):i*9) ), 2 )';

        S  = zeros(126,126);
        for j = ( (i-1)*9+1 ) : ( i*9 )
                S = S + (projectimage(:,j)- m_fisher(:,i))*(projectimage(:,j)-m_fisher(:,i))';
        end

        Sw = Sw + S;
        Sb = Sb + (m_fisher(:,i)-mean_PCA) * (m_fisher(:,i)-mean_PCA)';
end
```

Here, Sw is within Scatter Matrix
Sb is Between Scatter Matrix
Now we need to estimate fisher discriminant basis's by maximizing the ratio of projected between-class and within-class variations.

Significant directions are obtained as follows:

```
        [F_eigenvec, F_eigenval] = eig(Sb,Sw);
        F_eigenvec = fliplr(F_eigenvec);
```

Here, fliplr is used to flip the columns from left 2 right in the matrix *F_eigenvec*

Projecting all the training images onto Fisher linear space using following command:

```
        for i = 1 : 135
                projectimage_Fisher(:,i) = V_Fisher' * projectimage(:,i);
        end
```

Please Note here image is projected from the eigenspace created using ***PCA*** *onto fisher space.*

Testing

Reading all the testing images in a similar fashion as did in training and performing the preprocessing steps:

1. Subtracting the mean from the test image.
2. Projecting the test image onto the FisherSpace.
3. Calculating the euclidean distance from all the training images.
4. Assigning the same class to the test image as of the image for which the euclidean distance is minimum.

Code Snippet for the above steps:

```
InputImage = imread(currenttestfile);
temp = InputImage;

[irow icol] = size(temp);
test_image = reshape(temp',irow*icol,1);
test_image = double(test_image)- m;
ProjectedTestImage = V_Fisher' * eigenfaces' * test_image;


Euc_dist = [];
for i = 1 : 135
   q = projectimage_Fisher(:,i);
   temp = (norm( ProjectedTestImage - q))^2;
   Euc_dist = [Euc_dist temp];
end


[min_dist,ind] = min(Euc_dist);
index = [index, ind];
```

Here index vector contains the matched image index for all the testing images.
Accuracy comes out to be **56.67**