Face Recognition by Computing Eigenvectors of Image

Aravind Menon(MIT2020049), Shubham Pandey(MIT2020050), Shubham Sahu(MIT2020051)
Group Code : G17_EV&V_A2
Indian Institute Of Information Technology, Allahabad

BRIEF REPORT

1. Abstract

In Computer Vision, face recognition is one of the most important problems to solve. The image of a human face is a highly complex structure that needs to be transformed into a collection of much simpler metrics which capture all the important features of that image. In this paper, we will be looking into eigenfaces, how they can be created from a training set of face images and how they can help in recognising a person's face.

Index Term - Principle component analysis, eigenvector, eigenvalue, eigenface, faces recognition.

1. Introduction

Face recognition is based on the fact that it is possible to uniquely identify people because we all have different faces. However, there are many challenges to face recognition algorithms such as difference in lighting of the images, the images have to be rescaled and resized to be able to compare and calculate variations in features.[1]

In simple terms, face recognition is just what the name conveys, recognizing a person using an image of his/her face.Facial recognition is a biometric software application capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person's facial contours. The technology is mostly used for security purposes, though there is increasing interest in other areas of use. Facial recognition technology has received significant attention as it has the potential for a wide range of applications related to law enforcement as well as other enterprises

2. Related Work About Topic

2.1 Eigenvalues.

A matrix in the most simplest way of defining things, is a transformation. It transforms one coordinate system to another one. Every vector in our domain gets transformed to some other vector. Now an EIGENVECTOR is one such vector that retains its direction. So we have a vector that, in the transformed and the original coordinates is pointing in the same direction. The magnitude of it may vary. Now this scaling of the magnitude is the eigenvalues.

2.2 Eigenfaces.

Eigenfaces is an application of Principal Components Analysis (PCA), a well-known method for data compression. In my opinion, the first and the most straighforward problem of using this analysis that it works with faces as holistic images and it doesn't take into account any face's specific aspects like face parts (eyes, nose, lips, etc.).

2.3 Principal Component Analysis.

PCA gives us a systematic way to figure out which combinations of features appear to be more responsible for the variance of data than others and thus provide some guidance about how to reduce the number of dimensions in our input. Note, this does not necessarily mean PCA tells us which features are unnecessary; rather, it tells us how we might combine features together into a smaller subspace without losing (much) information.

It is the process of reducing the dimensionality of training data so that we are able to view it in a low dimensional space without the loss of too much information. [2]

3. Methodology

Face images are complex as they consist of a lot of pixels which carry the information about all its features. The relevant information or dominant features of a face image can be identified using a much smaller set of eigenvectors of its covariant matrix.

$$I = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{m \times n} \xrightarrow{\text{CONCATENATION}} \begin{bmatrix} x_{11} \\ \vdots \\ x_{1n} \\ \vdots \\ x_{2n} \\ \vdots \\ x_{mn} \end{bmatrix}_{1 \times N} = x.$$

Assume that the matrix obtained from concatenating the images, that is, converting the total pixels of each training image to one row, is A(Fig(1)). The covariance of the matrix can be obtained by,

$$C = A^T A$$

Now, calculate eigenvectors of C and take the k largest eigenvectors in the resultant matrix. These images give the variance among all images in the training set and are termed as Eigenfaces.

4. Algorithm Analysis

Input: A set of facial images known as Training set (k) .
Output: Form feature vectors for each image.

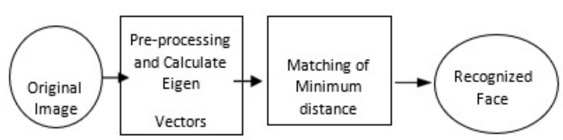


Fig. 1. Basic Steps used for Face Recognition

Steps.

1. Calculate mean matrix . Then subtract this mean from the original faces ( ) k to calculate the feature vector (k)

$$\psi = \frac{1}{N} \sum_{k=1}^{N} \Gamma_k$$

2. Find the covariance matrix c as follows:

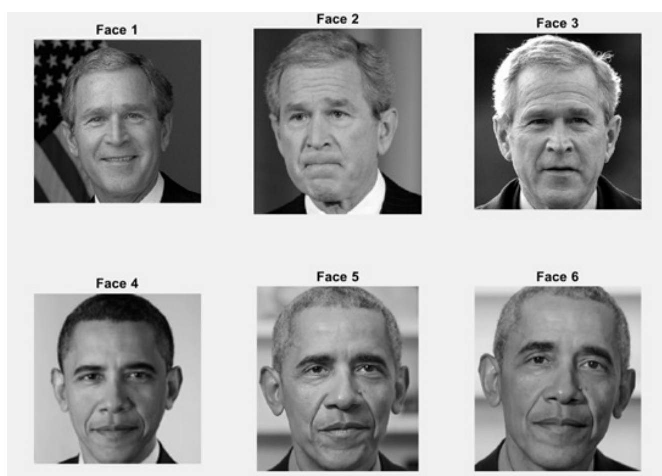
$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \\ &= AA^T \end{aligned}$$

where the matrix $A = [\ \Phi_1\ \Phi_2\ \ldots\ \Phi_M\ ]$.

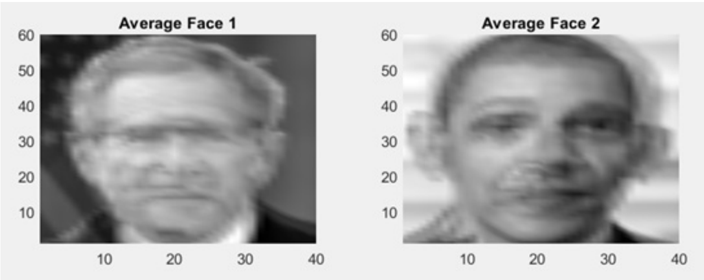
3. Compute the eigenvectors and eigenvalues of C
4. The N significant eigenvectors are selected on the basis of biggest equivalent eigenvalues.
5.Project all the face images into these eigenvectors and form the feature vectors of each face image

5. Experiment

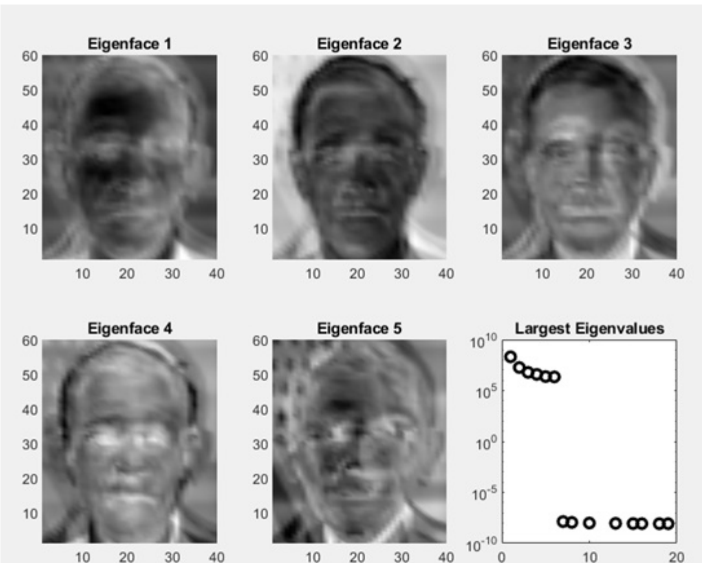
In Fig(1), we have the set of training images. There are three images of two people. Fig(2) shows the average face calculated by averaging the double values of the pixels of training data. After resizing the images, the covariance matrix is calculated and the highest eigenvalues and their corresponding eigenvectors are chosen. Fig(3) shows the eigenfaces generated. Fig(4) shows the two average faces projected onto the eigenvector space. Fig(5) shows the test images and the projection of these images onto the same vector space.



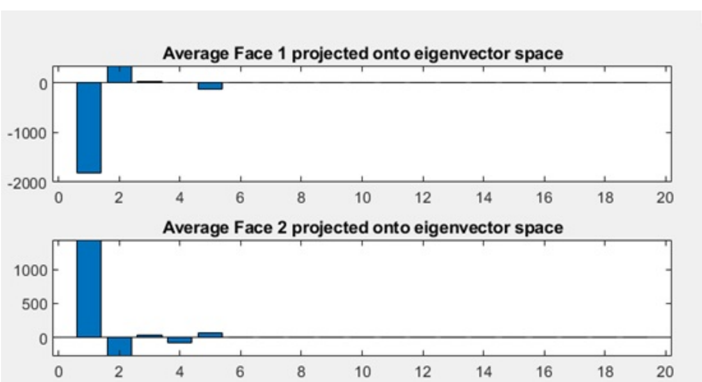
Fig(1)



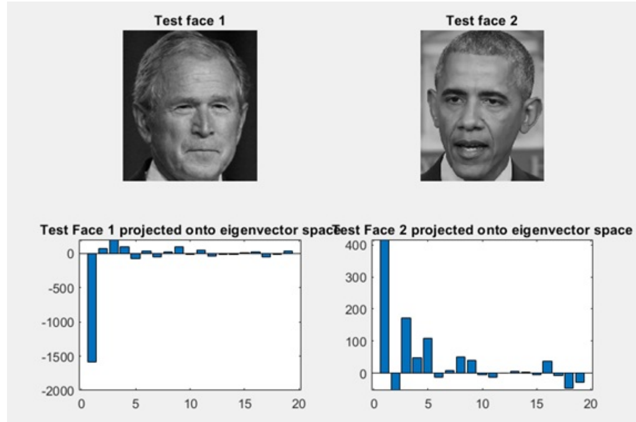
Fig(2)



Fig(3)



Fig(4)



Fig(5)

6. Conclusion

The projection of average of face 1 matches with the test face 1 and the average of face 2 matches with test face 2.

7. References

[1] Müge Çarıkçı a, Figen Özen A Face Recognition System Based on Eigenfaces Method Procedia Technology 1 (2012) 118 – 123]"

[2] [Marijeta Slavković1, Dubravka Jevtić1 Face Recognition Using Eigenface Approach Article in Serbian Journal of Electrical Engineering · February 2012]

[3] Md. Al-Amin Bhuiyan ,Towards Face Recognition Using Eigenface , (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016

Appendix

```
B1=rgb2gray( imread('gbush1.jpg'));

B2=rgb2gray( imread('gbush2.jpg'));

B3=rgb2gray( imread('gbush3.jpg'));

BS4=rgb2gray( imread('gbush4.jpg'));

O1=rgb2gray( imread('ob1.jpg'));

O2=rgb2gray( imread('ob2.jpg'));

O3=rgb2gray( imread('ob3.jpg'));

OB4=rgb2gray( imread('ob4.jpg'));

figure(1)

subplot(2,3,1),imshow(B1),title('Face 1');

subplot(2,3,2), imshow(B2),title('Face 2');

subplot(2,3,3),imshow(B3),title('Face 3');

subplot(2,3,4),imshow(O1),title('Face 4');

subplot(2,3,5),imshow(O2),title('Face 5');

subplot(2,3,6),imshow(O3),title('Face 6');


B1=imresize(double(rgb2gray( imread('gbush1.jpg'))),[60 40]);

B2=imresize(double(rgb2gray( imread('gbush2.jpg'))),[60 40]);

B3=imresize(double(rgb2gray( imread('gbush3.jpg'))),[60 40]);

O1=imresize(double(rgb2gray( imread('ob1.jpg'))),[60 40]);

O2=imresize(double(rgb2gray( imread('ob2.jpg'))),[60 40]);

O3=imresize(double(rgb2gray( imread('ob3.jpg'))),[60 40]);

AvgB=(B1+B2+B3)/3;

AvgO=(O1+O2+O3)/3;

figure(2)

subplot(2,2,1); pcolor(flipud(AvgB)); colormap(gray); shading interp; title('Average Face 1');
```

```
subplot(2,2,2); pcolor(flipud(AvgO)); colormap(gray); shading interp; title('Average Face 2');
D = [reshape(B1,1,60*40)
    reshape(B2,1,60*40)
    reshape(B3,1,60*40)
    reshape(O1,1,60*40)
    reshape(O2,1,60*40)
    reshape(O3,1,60*40)
    ];
[rowsx, colsx] = size(D);
Y = zeros(colsx, rowsx);
for row = 1 : colsx
  for col = 1 : rowsx
      Y(row,col)=D(col,row);
  end
end
[rowsx, colsx] = size(Y);
[rowsy, colsy] = size(D);
A = zeros(rowsx, colsy);
for row = 1 : rowsx
  for col = 1 : colsy
    s = 0;
    for k = 1 : colsx
      s = s + Y(row, k) * D(k, col);
    end
    A(row, col) = s;
  end
end
```

```
[V,E]=eigs(A,20,'lm');

figure(3)

subplot(2,3,1), face1=reshape(V(:,1),60,40); pcolor(flipud(face1)); colormap(gray); shading interp; title('Eigenface 1');

subplot(2,3,2), face1=reshape(V(:,2),60,40); pcolor(flipud(face1)); colormap(gray); shading interp; title('Eigenface 2');

subplot(2,3,3), face1=reshape(V(:,3),60,40); pcolor(flipud(face1)); colormap(gray); shading interp; title('Eigenface 3');

subplot(2,3,4), face1=reshape(V(:,4),60,40); pcolor(flipud(face1)); colormap(gray); shading interp; title('Eigenface 4');

subplot(2,3,5), face1=reshape(V(:,5),60,40); pcolor(flipud(face1)); colormap(gray); shading interp; title('Eigenface 5');

subplot(2,3,6), semilogy(diag(E),'ko','linewidth',[2]); title('Largest Eigenvalues');


VecB=reshape(AvgB,1,60*40);

VecO=reshape(AvgO,1,60*40);

[rowsx, colsx] = size(VecB);

[rowsy, colsy] = size(V);

ProjB = zeros(rowsx, colsy);

for row = 1 : rowsx

 for col = 1 : colsy

  s = 0;

  for k = 1 : colsx

   s = s + VecB(row, k) * V(k, col);

  end

  ProjB(row, col) = s;

 end

end
```

```
[rowsx, colsx] = size(VecO);
[rowsy, colsy] = size(V);
ProjO = zeros(rowsx, colsy);
for row = 1 : rowsx
  for col = 1 : colsy
    s = 0;
    for k = 1 : colsx
      s = s + VecO(row, k) * V(k, col);
    end
    ProjO(row, col) = s;
  end
end

figure(4)
subplot(3,1,1), bar(ProjB(2:20)); title('Average Face 1 projected onto eigenvector space');
subplot(3,1,2), bar(ProjO(2:20)); title('Average Face 2 projected onto eigenvector space');
B4=imresize(double(rgb2gray( imread('gbush4.jpg'))),[60 40]);
O4=imresize(double(rgb2gray( imread('ob4.jpg'))),[60 40]);
S=reshape(B4,1,60*40);
T=reshape(O4,1,60*40);
[rowsx, colsx] = size(S);
[rowsy, colsy] = size(V);
p = zeros(rowsx, colsy);
for row = 1 : rowsx
  for col = 1 : colsy
    s = 0;
    for k = 1 : colsx
      s = s + S(row, k) * V(k, col);
```

```
    end
    p(row, col) = s;
  end
end

[rowsx, colsx] = size(T);
[rowsy, colsy] = size(V);
o = zeros(rowsx, colsy);
for row = 1 : rowsx
  for col = 1 : colsy
    s = 0;
    for k = 1 : colsx
      s = s + T(row, k) * V(k, col);
    end
    o(row, col) = s;
  end
end

figure(5)
subplot(2,2,1), imshow(BS4); title('Test face 1');
subplot(2,2,2), imshow(OB4); title('Test face 2');
subplot(2,2,3), bar(p(2:20)); title('Test Face 1 projected onto eigenvector space');
subplot(2,2,4), bar(o(2:20)); title('Test Face 2 projected onto eigenvector space');
```