International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 5, Issue 4, Jun - Jul 2016; 1-10

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

FACE RECOGNITION USING NEURO-FUZZY AND EIGENFACE

SALAM HAMDAN & ADNAN SHAOUT

Department of Computer Science, Princess Sumaya University For Technology, Amman, Jordan

ABSTRACT

Face recognition is actively growing area of research because it is applied in numerous practical applications such as bank identification, security cameras, criminal investigations, security monitoring, and surveillance system and helps to aid national security. Face is a type of biometric characteristic of the human which is unique to each human. The face is recognized by considering features i.e. eye distance, nose distance, lip distance, etc. In this paper, a human presence is detected by extracting the skin region by using the Eigenface approach. After detecting the skin region, the individual face is recognized using a Neuro-Fuzzy. The experimental results show the effectiveness of the proposed technique.

KEYWORDS: Skin Detection, Neuro-Fuzzy, Eigenface, Face Recognition

I. INTRODUCTION

Eigenface technique is a straightforward and strong method for face detection and recognition problems[1]. It can classify faces accurately with very short time, in comparison with other techniques such as learning Vector Quantization (LVQ) or Self Organizing Map (SOM)[2]. Previously, the algorithms that target the image recognition focused on some features of the face. However, the Eigenface method focus on more features of the face to register more information to classify faces by considering general facial patterns[3].

Some of the main patterns that the Eigenface uses include specific facial features. Registering more information can naturally analyze the images in an enhanced way, in comparison with feature-based face recognition methods[4].

Eigenface is basically a bases vector for real faces, which relates to some transformation algorithms such as Fourier analyses, that sums the sinusoid weights at different frequencies and then recomposes the signal to its original shape. Eigenface follows the same way by summing the weights of Eigenfaces in order to rebuild distinct features of the face of a person.

Eigenface recognition system consists of two phases:

- Creating the Eigenface bases and recognition, and
- Detecting new faces

The rest of this paper is organized as follows: section II describes a brief literature review on face detection systems, section III provides a description of the Eigenface approach, section IV describes the Neuro-fuzzy approach, section V presents the simulation of the new approach, section VI provides the performance evaluation and finally section VII presents the conclusion of the paper.

II. LITERATURE REVIEW

Face recognition techniques can be divided into two main categorizes. The first category is the face recognition from intensity images. This category has a number of approaches such as feature-based sub-category, which depends on eliciting the main facial features such as nose, mouth or eyes [5]. Many approaches follow under this category. Kande et al [6] employed simple image processing methods to extract a vector of 16 facial parameters, which were ratios of distance, areas and angles. Brunelli et al [7]made an enhancement on the previous approach by making the vector of 35 geometric features. The second class is holistic. This approach depends on identifying the face using the entire image instead of local features of the face. Sivovich et al [8]used the principal components analysis (PCA). The Turk et al [9] proposed to use Eigenface to recognize faces. Thai et al[10] used artificial neural network for face recognition. They proposed to combine AdaBoost and artificial neural network as a hybrid methodology to detect faces[10]. Shamla et al proposed a face recognition system based on neural network and was implemented using Matlab. They proposed to label a self-organizing feature maps (SOM) to measure image similarity[11]. Aborisade, D.O, proposed a novel fuzzy logic based edge detection technique which used three linear spatial filters to generate 3 edges strength value at each pixel of a digital image through spatial convolution process[12]. Rizon et al[13] proposed an approach to detect faces by using Eigenface to recognize the face and neural networks to recognize to whom the image belongs to. They reported that they raised the speed of recognition.

III. EIGENFACE RECOGNITION

The steps for detecting a face based on Eigenface technique is shown in the flowchart in figure 1.

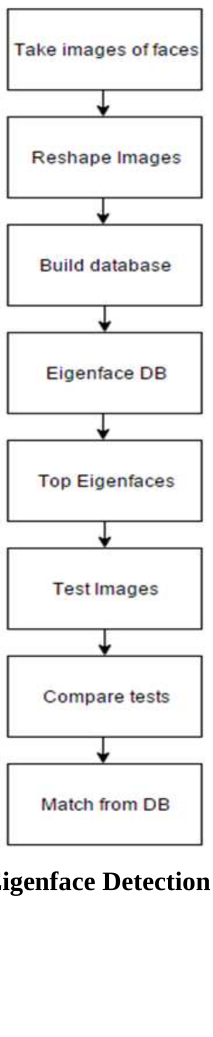

Figure 1: Flowchart of Eigenface Detection and Recognition System

In order to determine the Eigenfaces, initially dataset of face images need to be built. The collected images are then considered as the database of known faces. All of the collected images in the database should be of the same type and size. In this work, the images are converted to gray scale ranging from 0 to 255.

Each image then is converted to vector Γn with size of N;

Where N is the imagewidth * imagehigh

The dataset should include multiple face images for each person. In this paper, we are using 10 face images for 40 persons. The used images are from [15].This image dataset will assist in enhancing the accuracy of detecting and recognizing the faces. The dataset will increase the available information from the samples that can be used with Eigenface. This dataset maybe called "Facespace", with dimension N. Figure 2 shows a sample of facespace which contain two faceimages for two persons.

After building the dataset, the average face (η) should be calculated from the facespace using equation (1)

$$\eta = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \tag{1}$$

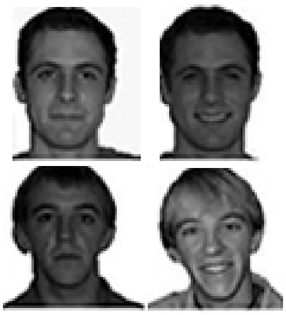

Figure 2: Sample of Facespace

Where M is the number of faces in the facespace.

The second step is to calculate the difference of each face from the average (ϑ) using equation (2).

$$\vartheta = \Gamma_i - \eta \tag{2}$$

Then covariance matrix (C) is required to be determined for the dataset by using the following equation:

$$C = \frac{1}{M}\sum_{n=1}^{M}\vartheta_n\vartheta_n^T = \frac{1}{M}\sum_{n=1}^{M}\begin{pmatrix} \mathrm{var}(p_1) & \cdots & \mathrm{cov}(p_1, p_N) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(p_N, p_1) & \cdots & \mathrm{var}(p_N) \end{pmatrix}_n = AA^T$$

Where p_i= pixel i of n.

The Eigenface of consideration is that of C where C is of N dimension. Solving for Eigenface will be impossible. In order to solve this dilemma, the Principal Component Analysis (PCA) method is applied.

Applying PCA can reduce the Eigenface vector from dimension N to M. It helps knowing that if we have only M images, then only M non-trivial Eigen vectors will be available which can be solved by having M X M matrix of eigenvectors as shown in equation 3:

$$L = A^T A \tag{3}$$

Where, $A = [\vartheta_1 \vartheta_2 \dots \vartheta_M]$. we then have,

$$A^T A_{v_i} = \mu_i v_i \tag{4}$$

$$A A^T A_{v_i} = \mu_i A v_i \tag{5}$$

Where $v_i$is the eigenvector of L. Then the result Av_i will be an eigenvector of C.

Finally the M eigenvector of L can be used from M eigenvectors $u_l$of C. This will result in forming the Eigenface basis.

$$u_l = \sum_{k=1}^{M} v_{lk} \vartheta_k \tag{6}$$

Which shows that only M_k Eigenfaces are required to output the complete facespace basis, where k is the number of persons.

Finally the image will be rebuilt using only few Eigenfaces (M') which correspond to the vectors with the highest eigenvectors and the most variance face space. Figure 1 shows the complete flowchart of the method.

IV. THE NEURO-FUZZY PROCESS

Figure 3 shows the use of neuro-fuzzy process flowchart. After getting the output from Eigenface, now all the data are available for LVQ neural network to train the clusters. However, LVQ is able to classify all the input images for the 40 persons. In order to enhance this process and to get accurate results, fuzzy logic is used for matching the classes. The k-mean algorithm is used for generating the fuzzy rules. In this work, we have 40 clusters, which are the same as the number of the persons. Moreover, Gaussian membership was used for the fuzzy membership due to the simplicity of its derivative expression. The model that was used in this work was based on Jang's [1] neuro-fuzzy classifier. The rule weights are determined based on the number of the samples that we used for each person, which are 10.

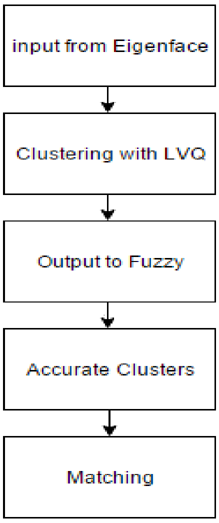

Figure 3: The Neuro-Fuzzy Process Flow Chart

Using this process, enhanced and more accurate results were obtained with shorter time relative to other methods as shown in table 1.

V. SIMULATION

For the proposed system the training phase is split into two phases. However, before training the system, the data images that are used to train the system are preprocessed to be of the same size and type. Figure 4 shows the code used to perform the preprocessing step on the images before they are entered to the system.

```
w=NFuzzy();
currentFolder = pwd;
[F,currentFolder,pgm] = uigetfile({'*.*','All Files(*.*)'},
'Select your File ');
loadimage = strcat(currentFolder,F);
input = importdata(loadimage);
ri=round(400*rand(1,1));
r=w(:,ri);
v=w(:,[1:ri-1 ri+1:end]);
%[q,qq]=size(v);
%r=r(1:q,1:qq);
N=20;
%eigen=w;
% remove the mean from the input images( v)
O=uint8(ones(1,size(v,2)));
m=uint8(mean(v,2));
vzm=v-uint8(single(m)*single(O));
```

Figure 4: Image Preprocessing Code

The next step for the Eigenface approach is performing training using the preprocessed images. This is done by first compute the average of each image then computing the mean for the 10 images for each person. Thereafter, the face detection is done by comparing the average of the images with the mean of the images. Afterwards, the calculated means are entered into a Neuro-Fuzzy system to perform the classification. Figure 5shows the code for the Neuro-Fuzzy and the Eigenface system.

```
w=NFuzzy();
currentFolder = pwd;
[F,currentFolder,pgm] = uigetfile({'*.*','All Files(*.*)'}, 'Select your File ');
loadimage = strcat(currentFolder,F);
input = importdata(loadimage);
ri=round(400*rand(1,1));
r=w(:,ri);
v=w(:,[1:ri-1 ri+1:end]);
N=20;
O=uint8(ones(1,size(v,2)));
m=uint8(mean(v,2));
vzm=v-uint8(single(m)*single(O));
L=single(vzm)'*single(vzm);
[V,D]=eig(L);
V=single(vzm)*V;
% Choose the eignevectors related to the top 10 eigenvalues.
V=V(:,end:-1:end-(N-1));
cv=zeros(size(v,2),N);
for i=1:size(v,2);
    cv(i,:)=single(vzm(:,i))'*V;
end
subplot(121);
imshow(reshape(r,112,92));title('Selected Image','Fontsize',12,'color','Blue');
subplot(122);
p=r-m;
s=single(p)'*V;
z=[];
for i=1:size(v,2)
    z=[z,norm(cv(i,:)-s,2)];
    if(rem(i,20)==0),imshow(reshape(v(:,i),112,92)),end;
    drawnow;
end
[a,i]=min(z);
subplot(122);
imshow(reshape(v(:,i),112,92));title('Result','Fontsize',12,'color','blue');
```

Figure 5: Eigenface Code and Neuro-Fuzzy Code

VI. PRFORMANCE EVALUATION

In this paper an algorithm to recognize faces is proposed. First of all, skin is recognized by using an Eigenface approach. After the skin is detected, a Neuro-Fuzzy is used to recognize faces. 400 images were used as database from [15]. 40 persons were there and every person has 10 different images. From the 400 images an image is picked randomly and then the proposed algorithm was applied. Table 1 shows the results of the performance evaluation. The results show that when the proposed system is applied the performance accuracy was 96.45% when the tested images were pure with no additional noise added to them. However, the proposed algorithms were tested by adding noise to the images such as rotate the tested image and add salt and pepper noise. As shown in table 1 each person has 10 images and each image is tested and the average of the 10 images is shown in the table. The total accuracy when the salt and pepper noise was added is 94.6 and when rotating the image the total accuracy is 94%. Comparing our results to [3] the results were quite satisfying which it was 96% without any addition of noise to their images.

Some successful and unsuccessful search results are shown in Figures 6 and 7, respectively.

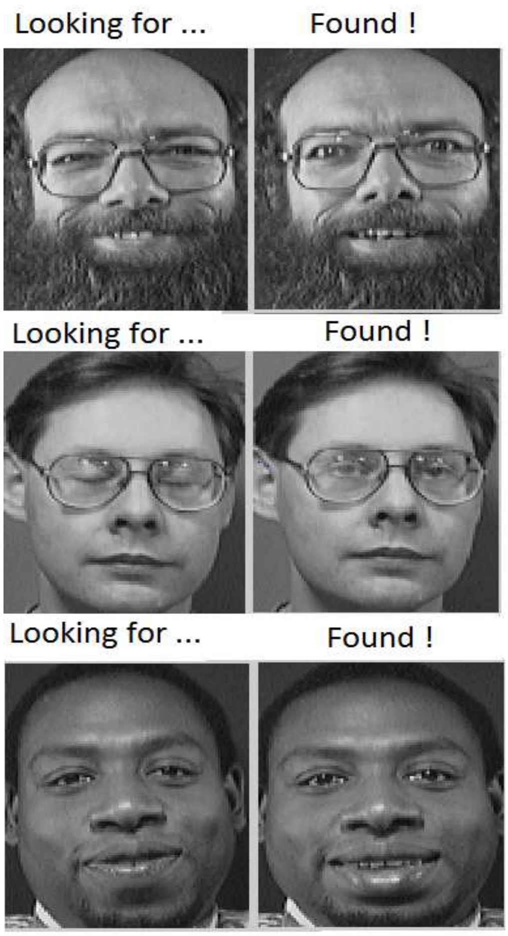

Figure 6: Some Successful Facial Recognition

Figure 7: Unsuccessful Result

VII. CONCLUSIONS

An algorithm for human face detection based on Eigenface for skin detection and Neuro-Fuzzy for face recognition was proposed in this paper and the overall performance of the system was 95% which was acceptable when compared with [3] on different images. Neuro-fuzzy method was used to increase the efficiency of face recognition approach. By the combination of Neuro-Fuzzy and Eigenface the accuracy of this system was improved. The algorithm did not work efficiently when the images were under salt and pepper noise and rotation. In the future, using Neuro-Fuzzy for skin detection and Eigenface for face recognition may give more accurate results.

REFERENCES

1. Zhang, J., Y. Yan, and M. Lades, *Face recognition: Eigenface, elastic matching, and neural nets.* Proceedings of the IEEE, 1997. 85(9): p. 1423-1435.
2. Lanitis, A., C. Draganova, and C. Christodoulou, *Comparing different classifiers for automatic age estimation.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2004. 34(1): p. 621-628.
3. Chowdhury, A. and S.S. Tripathy. Human skin detection and face recognition using fuzzy logic and Eigenface. in Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on. 2014. IEEE.
4. Turk, M. and A. Pentland, *Eigenfaces for recognition.* Journal of cognitive neuroscience, 1991. 3(1): p. 71-86.
5. Heisele, B., et al., *Face recognition: component-based versus global approaches.* Computer vision and image understanding, 2003. 91(1): p. 6-21.
6. Kanade, T., Picture processing system by computer complex and recognition of human faces. Doctoral dissertation, Kyoto University}, 1973. 3952: p. 83-97.
7. Brunelli, R. and T. Poggio, *Face recognition: Features versus templates.* IEEE Transactions on Pattern Analysis & Machine Intelligence, 1993(10): p. 1042-1052.
8. Sirovich, L. and M. Kirby, Low-dimensional procedure for the characterization of human faces. JOSA A, 1987. 4(3): p. 519-524.
9. Turk, M. and A.P. Pentland. Face recognition using Eigenfaces. in Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. 1991. IEEE.
10. Lawrence, S., et al., *Face recognition: A convolutional neural-network approach.* Neural Networks, IEEE Transactions on, 1997. 8(1): p. 98-113.
11. Le, T.H., *Applying artificial neural networks for face recognition.* Advances in Artificial Neural Systems, 2011. 2011: p. 15.
12. Mantri, S. and K. Bapat, *Neural Network Based Face Recognition Using Matlab.* IJCSET Vol 1 Issue 1, Feb, 2011.
13. Aborisade, D.O., *Novel fuzzy logic based edge detection technique.* International Journal of Advanced Science and Technology, 2011. 29: p. 75-82.
14. Rizon, M., Muhammad Firdaus, H., Puteh, S., Sazali, Y., Mohd Rozailan, M., Yeon, A., ... & Karthigayan, M. (2006). Face recognition using Eigenfaces and neural networks.Chicago.
15. http://www.cl.cam.ac.uk/research/dtg/attarchive/facesataglance.html and downloaded from http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip.

APPENDICES

Table 1: Performance Evaluation

Person	Average for 50 Trial %	With Salt and Pepper	With Rotation	
n1	90	90	88	
n2	95	93	92	
n3	99	95	95	
n4	100	99	98	
n5	97	94	93	
n6	100	96	96	
n7	98	92	90	
n8	98	95	95	
n9	100	96	96	
n10	99	96	94	
n11	96	94	93	
n12	95	94	93	
n13	100	97	93	
n14	90	91	92	
n15	90	93	90	
n16	100	99	98	
n17	99	97	95	
n18	97	94	93	
n19	96	93	92	
n20	96	94	91	
n21	99	95	94	
n22	100	98	97	
n23	93	90	90	
n24	93	96	97	
n25	96	97	95	
n26	94	94	93	
n27	99	93	92	
n28	90	81	90	
n29	100	99	97	
n30	94	93	92	
n31	94	95	94	
n32	100	99	100	
n33	100	98	98	
n34	100	97	98	
n35	94	93	94	
n36	96	97	96	
n37	97	97	96	
n38	93	93	92	
n39	99	98	98	
n40	92	90	91	**Total Average**
Average for FaceSpace	**96.45**	**94.625**	**94.025**	**95.03333333**