Genetic Algorithm based Human Face Recognition

Ravi Subban¹, Dattatreya Mankame², Sadique Nayeem³, P. Pasupathi³ and S. Muthukumar³
¹ Department of Computer Science, School of Engg. and Tech., Pondicherry University, Pondicherry, India
Email: {sravicite@gmail.com, dpmankame@gmail.com, sadique Nayeem@gmail.com}
²Department of Information Science and Engg., K.L.E. Institute of Technology, Hubli, India
³ Centre for Information Technology and Engg., Manonmaniam Sundaranar University, Tirunelveli, India
Email: {pp.cite.msu@gmail.com, sm.cite.msu@gmail.com}

Abstract—This paper proposes a novel algorithm for recognizing human faces using Genetic Algorithms. This proposed method consists of three major phase namely, face representation, face detection and face detection. In addition, a comparative study is made using Principle Component Analysis and Linear Discriminant Analysis using the commonly used face databases such as Essex Face Database-Face94, Indian Face Database [16], Yale Database, FACE 1999 [18] and UMIST Face Database [19]. The proposed method produces better results than PCA and LDA for one sample per person. For Face 1999, which contained images with different background, the recognition rate is not quite good. But for all the other databases the test produces more than 90% recognition accuracy rate. From the result obtained it is observed that the proposed system does not need more number of images per person. Here the same recognition rate is achieved in around 5 images per person instead of around 10 images per person in basic PCA and LDA.

Index Terms—Genetic Algorithms, Principle Component Analysis, Linear Discriminant Analysis, Face Representation

I. INTRODUCTION

The input of a face recognition system is always an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video. Generally, there are three phases in a face recognition system, viz., face representation, face detection and face recognition [1] as shown in figure 1. The input of a face recognition system is always an image or video stream. The output is an identification or verification of the subject or subjects that appear in the image or video. Face detection is the second step in artificial face recognition and it deals with locating a face in a test image and to separate it from the remaining scene and background. Face detection can be divided into four categories; knowledge-based method [4], feature invariant-based method [5] [6] [7], template-based method [8] [9] and appearance-based method [5] [10] [11] [12].

Face representation is the first step in artificial face recognition and it deals with the modeling of faces. The representation of a face determines the successive algorithms of detection and identification. There are different approaches used for face representation, which can be classified into three categories: template-based, feature-based and appearance-based. Template-matching approach is the simplest approach to represent the complete face using a single template, i.e., a 2-D array of intensity, which is generally an edge map of the original face image. In feature-based approaches, the face is represented by the geometric
features, such as position and width of eyes, nose, and mouth, thickness of eyebrows and arches face breadth, or invariant moments. This approach requires smaller memory and has higher recognition speed than template-based approach. In the appearance-based approaches face images are projected onto a linear subspace of low dimensions. Such a subspace is first constructed by principal component analysis on a set of training images, with eigenfaces as its eigenvectors [2].

Face recognition is the third step in face recognition system and it is performed at the subordinate-level. At this stage, a new face is compared to a set face images stored in a database and then classified as a known individual if a match is found. Here, performance of face recognition system is affected by several factors such as scale, pose, illumination, facial expression, and disguise.

Figure 1. Components of Face Recognition system

Face detection is the second step in artificial face recognition and it deals with locating a face in a test image and to separate it from the remaining scene and background. Face detection can be divided into four categories; knowledge-based method [4], feature invariant-based method [5] [6] [7], template-based method [8] [9] and appearance-based method [5] [10] [11] [12]. Di Huang, Mohsen Ardabilian, Yunhong Wang, and Liming Chen [13] proposed a method that derives an upper bound for recognition error arising from the proposed weighted feature fusion to justify theoretically its effectiveness for recognition from videos. This method has been successfully evaluated under the conditions that are similar to those in real-world. But this method works well when a small number of misaligned images. Yogachandran Rahulamathavan, Raphael C.W. Phan, Jonathon A. Chambers, and David J. Parish [14] proposed a method to perform facial expression recognition on images in the encrypted domain, based on local FLDA. The recognition rate of the largest class size is in this experiment is 93.71 percent.

II. FACE REPRESENTATION DETECTION AND RECOGNITION USING GENETIC ALGORITHM

Genetic algorithm (GAs) is based on the most popular Darwinian’s theory of survival of the fittest. In computer world language it is said that for a problem there is a population of candidate solution. Out of these solutions only those solutions which are good enough are selected and new solutions are created from the selected solution. According to Michalewicz [3], Genetic Algorithms are defined as a very powerful and stochastic search broadly applicable, and in general it must have five basic components:

- A representation of genetic solutions to the problem,
- A way to create an initial population of solutions,
- An evaluation function for the classification of solutions in terms of their levels of fitness,
- Genetic operators that alter the genetic makeup children during playback, and
- Values for the parameters of genetic algorithms.

In genetic algorithms, an individual can be represented as a numeric value that may or may not be the best solution for a problem and a set of individuals is called population. Each individual is represented by a chromosome. The chromosome in biological systems is a sequence of DNA, thus in the same way, chromosomes in genetic algorithms is represented by a bit sequence, where each chromosome has $n$ bits. Coding of individuals or obtaining the sequence of bits that represent an individual is an important step in
implementing the genetic algorithms. This coding is done by converting the decimal value of an individual to binary, thus resulting in a sequence of 0 and 1. For example, for a certain population that its individuals have $n = 8$ bits, the X chromosome of an individual is represented by Equation (1).

$$X = x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1$$ (1)

where $x_8$ and $x_1$ are the most and least significant bits, respectively.

There are three genetic operators that can be applied to the chromosomes of individuals of a population and these operators are:

A. Selection

The selection operator is used to select the individuals from the current population on which the other genetic operators are applied. The selection operator works as the same as that of natural selection performed in biological systems where less fit individuals are eliminated from the population and fitter individuals have a higher chance to pass their details on to the future generations. Some of the traditional forms of selection are Proportionate selection, Tournament selection, Rank Based selection.

B. Crossover

The crossover operator allows individuals to exchange the information, similarly as done in the sexual reproduction of natural organisms. Crossover operator is also called as the recombination operator and it helps to generate the offspring for the next generation. It does so by combining bits of selected individual chromosome to form new and possibly better offspring in order to continue the population. The children generated will not be identical to any of its parents and contain the combined parental traits. There are three type of crossover namely: one point crossover, two point crossover and uniform crossover. To show the operation of the crossover operator, consider two individuals with the numerical values equal to 134 and 201, where the binary sequences of these individuals are 1000110 and 11001001 respectively, both individuals or chromosomes have $n = 8$ bits. Let us take one point crossover and cutting point on the 3rd bit ($p=3$). Figure 2(a) shows the crossover, figure 2(b) shows the bits to be switched and figure 2(c) shows the new children binary sequences as 1001001 and 1100110, whose decimal values are 137 and 198, respectively. The child individuals are shown in Figure 2(c).

One way to implement the operator’s crossover is via Boolean algebra. For solving the operator crossover, consider the two individuals parent1 and parent2 as in equation 2 and 3.

$$parent1 = a_n ... a_3 a_2 a_1$$ (2)

and

$$parent2 = b_n ... b_3 b_2 b_1$$ (3)

which have chromosomes with n bits.

Applying the single-point crossover operator to the parent parent1 and parent2, child individuals are generated, as shown in Equations (4) and (5).

$$offspring1 = c_n ... c_3 c_2 c_1$$ (4)

and
The bits of offspring1 and offspring2 are obtained from the Equations (6) and (7), for i = 1, 2, 3... n. Operations shown in these two equations are logical operations, that is, the signs + and · represent logic functions AND and OR, respectively.

\[ \text{offspring1} = a_i.s_i + b_i.r_i \] (6)

and

\[ \text{offspring2} = b_i.s_i + a_i.r_i \] (7)

where \( a_i, b_i, c_i, d_i, s_i \) and \( r_i \) are the i-th bit of the individuals \( \text{parent1, parent2, offspring1, offspring2} \) and binary sequences S and R, respectively. The binary sequences S and R have the same amount bit the chromosomes of the individuals of the population. These two bit sequences are defined by Equations (8) and (9).

\[ S_i = \begin{cases} 0, & \text{if } i \leq p \\ 1, & \text{if } i > p \end{cases} \] (8)

\[ r_i = \begin{cases} 1, & \text{if } i \leq p \\ 0, & \text{if } i > p \end{cases} \] (9)

C. Mutation

In biological systems, mutation refers to a change in characteristics of an individual. In order to have mutation, the DNA sequence of an individual is needed to be altered. So, in genetic algorithm the mutation operator tries to bring change in the individual’s characteristics by changing a value of the bits within the chromosome. Doing so, a new individual is generated.

In the genetic algorithm fitness function plays the most important role. It is used to see whether the new generation is converging to a solution of the problem or not. It is a mathematical expression and this expression depends on the problem to be solved. The fittest individual is chosen on the basis of the value of fitness function.

Thus the whole genetic algorithm can be summarized in the following steps:

1. **Step 1:** \( T \) is defined by user. (\( T \) is the maximum number of generation).
2. **Step 2:** Form an initial population.
3. **Step 3:** For each binary sequence representing an individual of initial population, decodes up these sequences in a binary integer value \( x \).
4. **Step 4:** Use the selection operator and choose pairs of parent individuals.
5. **Step 5:** Apply the crossover operator for pairs of individuals selected in the previous step, thereby generating the individual children.
6. **Step 6:** Apply the mutation operator to individual children changing 0 to 1 or vice versa.
7. **Step 7:** Calculate the values of the fitness function for individual parents and children.
8. **Step 8:** Select the better half (individuals with higher function values of fitness) of the population and these individuals will be selected as the population of the next generation.
9. **Step 9:** Return to step 2 until the number of generations is \( T \) achieved.

III. PROPOSED METHOD

The above mentioned problem can be solved with the help of application of genetic algorithm for the face recognition. In this paper, a method has been proposed for face recognition using genetic algorithm. First of all, a set of training images is given:

\[ I_{\text{train}} = I_1, I_2, I_3 ... I_n \] (10)

and a set of testing images is given:

\[ I_{\text{test}} = I_1, I_2, I_3 ... I_n \] (11)

**Steps:**

1. Convert all the images of the training set into gray scale then into column vector as shown in the figure 3.
2. Select the image (to be tested) from the testing set, convert the image into gray scale then into column vector as shown in the figure 4.
3. For more than one sample per person, apply crossover operator (as in figure 5) to produce more number of images per person otherwise go to step 4.
4. For one sample per person apply mutation (as in figure 6) at the least significant bits of chromosome.

5. Determine the fitness function value by using the Euclidean distance between the test image and the training set images.

6. If any individual obtain a value of the fitness function below the threshold one (defined by user), the system recognizes the image same as the test image, otherwise.

7. Increase the generation count. Go to step 3 and repeat step 3 to 7 till the counter has reached a maximum number generation T (defined by the user).
IV IMPLEMENTATION OF GENETIC ALGORITHM FOR FACE RECOGNITION

In this experiment the genetic algorithm has been applied for the face recognition and this concept has been implemented and tested with five different databases whose description is given in the table I.

<table>
<thead>
<tr>
<th>Name of database</th>
<th>Image format</th>
<th>Image size</th>
<th>Image type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFD [14]</td>
<td>jpg</td>
<td>110X75</td>
<td>Color</td>
</tr>
<tr>
<td>Face94 [13]</td>
<td>jpg</td>
<td>90X100</td>
<td>Color</td>
</tr>
<tr>
<td>Yale [17]</td>
<td>gif</td>
<td>320X243</td>
<td>Gray</td>
</tr>
<tr>
<td>Face 1999 [18]</td>
<td>jpg</td>
<td>300X198</td>
<td>Color</td>
</tr>
<tr>
<td>UMIST [19]</td>
<td>jpeg</td>
<td>92X112</td>
<td>Gray</td>
</tr>
</tbody>
</table>

To perform the tests of the proposed method, five different databases which contained 4095 images of 273 individuals are used. The experiment is carried out using MATLAB R2008a with different number of samples kept in the training set. The test images were kept in the testing set. Both sets were loaded and test image is provided for the recognition. Test is carried out by first using 1 sample per person, then 2 samples per person till 5 samples per person. Whole experiment is carried out till 4th generation. In generation 0 the test image is recognized with the training set image. In generation 1 the training set image is mutated as per step 4 of the proposed method. In the rest of the other generation, training set image is crossovered as per step 3 of the proposed method.

V. EXPERIMENTAL RESULTS

The experimental results are given in the table 2. Test is carried out by first using 1 sample per person, then 2 samples per person till 5 samples per person. Whole experiment is carried out till 4th generation. In generation 0, as shown in figure 7, the test images are recognized using original test images. This test gives almost the same result like PCA or LDA. But it failed to a large extent for recognizing images with different background. In generation 1, as shown in figure 8, the entire pixels are mutated by reversing their least significant bit. At this generation recognition rate is much improved specially for the gray level image. In subsequent generations the same trend followed and in generation 4 as shown in the figure 11, the proposed method produces better result than PCA and LDA for one sample per person. The figures 7 through 11 shows the chart showing the results obtained (Recognition Accuracy Rate) in the generations 0 through 4. For Face 1999, which contained images with different background, the recognition rate is not quite good. But for all the other databases the test produces more than 90% recognition rate. From the results obtained, it is observed that the proposed system does not need more number of images per person. Here the same recognition rate is achieved in around 5 images per person instead of around 10 images per person in basic PCA and LDA.
## Table II. Experimental Results of Genetic Algorithm Based Human Face Recognition

<table>
<thead>
<tr>
<th>Name of database</th>
<th>Total No. of unique person</th>
<th>No. of samples of each person in training set</th>
<th>No. of image in training set</th>
<th>Accuracy rate (%)</th>
<th>Gen: 0</th>
<th>Gen: 1</th>
<th>Gen: 2</th>
<th>Gen: 3</th>
<th>Gen: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFD</td>
<td>60</td>
<td>1</td>
<td>60</td>
<td>68.85</td>
<td>70.49</td>
<td>77.05</td>
<td>80.33</td>
<td>90.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>120</td>
<td>59.02</td>
<td>63.93</td>
<td>75.41</td>
<td>83.61</td>
<td>85.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>180</td>
<td>75.41</td>
<td>75.41</td>
<td>75.41</td>
<td>80.33</td>
<td>95.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>240</td>
<td>78.69</td>
<td>78.69</td>
<td>80.33</td>
<td>81.97</td>
<td>93.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>300</td>
<td>86.89</td>
<td>88.52</td>
<td>90.16</td>
<td>93.44</td>
<td>98.36</td>
<td></td>
</tr>
<tr>
<td>Face94</td>
<td>152</td>
<td>1</td>
<td>152</td>
<td>76.32</td>
<td>76.97</td>
<td>76.32</td>
<td>84.21</td>
<td>88.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>304</td>
<td>80.92</td>
<td>82.24</td>
<td>81.58</td>
<td>86.84</td>
<td>88.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>456</td>
<td>92.11</td>
<td>96.05</td>
<td>91.45</td>
<td>96.05</td>
<td>96.71</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>608</td>
<td>92.76</td>
<td>94.74</td>
<td>97.37</td>
<td>97.37</td>
<td>98.68</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>760</td>
<td>92.76</td>
<td>95.39</td>
<td>97.37</td>
<td>98.03</td>
<td>97.37</td>
<td></td>
</tr>
<tr>
<td>Yale</td>
<td>15</td>
<td>1</td>
<td>15</td>
<td>60.00</td>
<td>66.67</td>
<td>73.33</td>
<td>73.33</td>
<td>73.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>30</td>
<td>86.67</td>
<td>86.67</td>
<td>86.67</td>
<td>80.00</td>
<td>88.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>45</td>
<td>73.33</td>
<td>80.00</td>
<td>86.67</td>
<td>86.67</td>
<td>86.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>60</td>
<td>73.33</td>
<td>73.33</td>
<td>80.00</td>
<td>93.33</td>
<td>93.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>75</td>
<td>86.67</td>
<td>93.33</td>
<td>93.33</td>
<td>93.33</td>
<td>93.33</td>
<td></td>
</tr>
<tr>
<td>Face 1999</td>
<td>26</td>
<td>1</td>
<td>26</td>
<td>34.62</td>
<td>38.46</td>
<td>42.31</td>
<td>50.00</td>
<td>69.23</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>52</td>
<td>34.62</td>
<td>34.62</td>
<td>46.15</td>
<td>42.31</td>
<td>65.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>78</td>
<td>42.31</td>
<td>46.15</td>
<td>53.85</td>
<td>61.54</td>
<td>65.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>104</td>
<td>46.15</td>
<td>42.31</td>
<td>57.69</td>
<td>65.38</td>
<td>57.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>130</td>
<td>53.85</td>
<td>53.85</td>
<td>69.23</td>
<td>61.54</td>
<td>69.23</td>
<td></td>
</tr>
<tr>
<td>UMIST</td>
<td>20</td>
<td>1</td>
<td>20</td>
<td>65.00</td>
<td>60.00</td>
<td>65.00</td>
<td>85.00</td>
<td>85.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>40</td>
<td>60.00</td>
<td>65.00</td>
<td>75.00</td>
<td>80.00</td>
<td>90.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>60</td>
<td>70.00</td>
<td>70.00</td>
<td>70.00</td>
<td>85.00</td>
<td>85.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>80</td>
<td>80.00</td>
<td>75.00</td>
<td>80.00</td>
<td>90.00</td>
<td>95.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>100</td>
<td>90.00</td>
<td>85.00</td>
<td>90.00</td>
<td>90.00</td>
<td>95.00</td>
<td></td>
</tr>
</tbody>
</table>

### VI. CONCLUSIONS

Human face recognition plays a vital role in biometrics for human authentication as it is one of the biometric technique used for security purposes in banks, laptop and in industries. A comparative study is made using Principle Component Analysis and Linear Discriminant Analysis using the commonly used face databases, such as Essex face database-face94, Indian Face Database, Yale Database, FACE 1999 and UMIST Face Database. The proposed method produces better result as compared to other techniques like PCA and LDA for one sample per person. For Face 1999, which contained images with different background, the recognition rate is not quite good. But for all the other databases the test produces more than 90% recognition accuracy rate. From the result obtained it is observed that the proposed system does not need more number of images per person. Here the same recognition rate is achieved in around 5 images per person instead of around 10 images per person in basic PCA and LDA.
Figure 7: Recognition Accuracy Rate at Generation 0

Figure 8: Recognition Accuracy Rate at Generation 1

Figure 9: Recognition Accuracy Rate at Generation 2
ACKNOWLEDGMENT

This work is supported and funded by the University Grant Commission (UGC), India under Major Research Project to the department of Computer Science of Pondicherry University, Pondicherry, India.

REFERENCES


