2011 International Conference on Signal, Image Processing and Applications
With workshop of ICEEA 2011
IPCSIT vol.21 (2011) © (2011) IACSIT Press, Singapore

Implementation of Binary PSO Based Face Recognition System using Image Preprocessing

Nitish Sinha[1], Priyanka.G[2]

[1,2]Dept. of Electronics and Communication, M.S. Ramaiah Institute of Technology, Bangalore, Karnataka

Abstract. Face Recognition has always been a challenging issue. Selecting less and proper number of features to represent an image with reduced redundancy and noise affects the performance of pattern recognition system. This paper presents a novel algorithm for feature extraction using the discrete cosine transforms (DCT) and feature selection using Binary Particle Swarm Optimization (BPSO) for the Face Recognition System. Major contribution lies in application of suitable image preprocessing techniques depending upon image acquisition conditions. Face Recognition System thus devised is compared with the Face Recognition System using Binary PSO without preprocessing and was found to yield excellent recognition rate using lesser number of DCT coefficients in the feature subset.

Keywords: Face Recognition System, Feature Selection, Image Preprocessing Techniques, Discrete Cosine Transform.

1. Introduction

Face Recognition System [3], [4], [5] is one of the most successful applications of enhanced computational ability and image processing. Automatic face recognition is intricate primarily because of difficult imaging conditions, ageing, facial expression, occlusion etc. Thus, image preprocessing is used to resize (to reduce the dimensionality of feature subset), adjust contrast, brightness and filter the noise in an image. The concept of preprocessing is applied to Face Recognition System and results are tabulated over ORL Database and YALE B Database.

Particle Swarm Optimization [1] was basically designed to simulate the social behavior like bird flocking and fish schooling and later the algorithm was simplified and observed to perform optimization too. PSO has been extensively used in fields like Pattern Recognition, Fuzzy System etc. Binary PSO is very much suitable for selecting minimum number of features which can represent an image with least loss of information.

2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a swarm intelligence technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [1]. In PSO, the swarm consists of particles which move around the solution space of the problem. These particles search for the optimal solution of the problem in the predefined solution space till the convergence is achieved.

2.1 PSO algorithm

- Initialize the particle position by assigning location $p = (p_0, p_1 \ldots, p_N)$ and velocities $v = (v_0, v_1, \ldots, v_N)$.
- Determine the fitness value of all the particles: $f(p) = (f(p_0), f(p_1), \ldots, f(p_N))$.
- Evaluate the location where each individual has the highest fitness value so far: $p = (p_0^{best}, p_1^{best}, \ldots, p_N^{best})$.
- Evaluate the global fitness value which is best of all p^{best} : $G(p) = \max(f(p))$.

---

[1] First author. Mob: +91-9880041761
Email address: nitish.snh@gmail.com

- The particle velocity is updated based on the p^{best} and g^{best}.
- $v_i^{new} = v_1 + c_1 \times rand\ ( ) \times (p_i^{best} - p_i) + c_2 \times rand\ ( ) \times (p_g^{best} - p_i)$ for $1 < i < N$. (1)
- where c_1 and c_2 are constants known as acceleration coefficients and rand () are two separately generated uniformly distributed random numbers in the range [0, 1].
- Update the particle location by: $p_i^{new} = p_i + v_i^{new}$ for $1 < i < N$.
- Terminate if maximum number of iterations is attained or minimum error criteria is met.
- Go to step 2.

2.2 Binary PSO

For binary discrete search space, Kennedy and Eberhart [2] have adapted the PSO to search in binary spaces by applying a sigmoid transformation to the velocity component in the equation (1) to squash the velocities into a range [0,1] and force the component values of the positions of the particles to be 0's or 1's. The sigmoid expression is given by:

$$\text{sigmoid}(p_{id}^k) = \frac{1}{1 - e^{-v_{id}^k}} \tag{2}$$

where $$p_{id}^k = \begin{cases} 1, \text{if rand}(\ ) < \text{sigmoid}(p_{id}^k) \\ 0, \text{otherwise} \end{cases}$$

3. Face Recognition

The four major steps in the proposed Face Recognition system are: Preprocessing (PPR), feature extraction, feature selection and matching. The block diagram of proposed system is shown in Fig. 1.

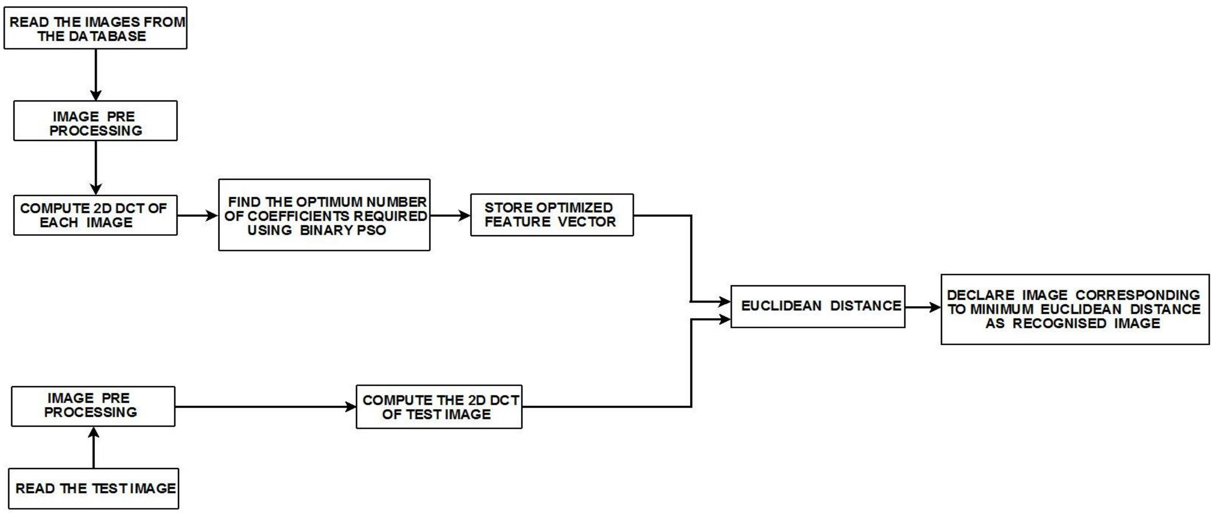

Fig. 1 Flow of the proposed Face Recognition System

3.1. Image preprocessing

Table 1: Preprocessing steps

Database	PPR1	PPR2	PPR3	PPR4	PPR5	No. of Training images/person	No. of Test images/person
ORL	Impyramid	Imadjust	Edge	Uint8	Imadd	4	6
YALE B	Imadjust	Fspecial	Imfilter	-	-	22	36

Referring to above table, we define the terms as follows:

1) Impyramid (I, direction): Impyramid (I, direction) computes a Gaussian pyramid reduction or expansion of I by one level, direction can be 'reduce' or 'expand'.

2)Imadjust(I,[],[],gamma): J = imadjust(I,[*lowin*; *highin*]*;* [*lowout*; *highout*],gamma) maps the values in I to new values in J, where gamma specifies the shape of the curve describing the relationship between the values in and J. If gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values. If you omit the argument, gamma defaults to 1 (linear mapping).

3)Edge(I,canny): J = edge(I) takes a gray scale or a binary image I as its input, and returns binary image J of the same size as I, with 1's where the function finds edges in I and 0's elsewhere. edge (I,'canny') specifies the Canny method.

4)Uint8 (I): uint8(I) returns the stored integer value of object as a built-in uint8. If the stored integer word length is too big for a uint8, or if the stored integer is signed, the returned value saturates to uint8.

5)Imadd(I1,I2): Z = imadd(X,Y) adds each element in array X with the corresponding element in array Y and returns the sum in the corresponding element of the output array Z. X and Y are real, non-sparse numeric arrays with the same size and class, or Y is a scalar double. Z has the same size and class as X, unless X is logical, in
which case Z is double.

6) Fspecial ('sobel'): h = fspecial('sobel') returns a 3-by-3 filter h that emphasizes horizontal edges using the smoothing effect by approximating a vertical gradient. If it's needed to emphasize vertical edges, transpose the filter 'h'.
7)Imfilter (I, H,'same'): Filters the multidimensional array I with the multidimensional filter H. The array I can be logical or a non-sparse numeric array of any class and dimension. The result has the same size and class as I.'Same' means the output array is the same size as the input array. This is the default behavior when no output size options are specified.

3.2. Feature extraction using DCT

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of the sum of cosine functions oscillating at different frequencies. Discrete cosine transform in this context is used to select the most distinguishing feature of the human face [7], [8], [9]. For most of the images, much of the signal energy lies at low frequencies; these appear at the upper left corner of the DCT matrix. The DCT of the processed image is computed for the cropped version of an input image containing a face and only a small subset of the coefficients is retained as a feature vector. The DCT expression is given as:

$$F(u,v)=a(u)\times a(v)\sum_{x=0}^{N-1}\sum_{x=0}^{M-1}\cos\left[\frac{\pi\times u}{2N}(2x+1)\right]\times\cos\left[\frac{\pi\times u}{2M}(2y+1)\right]\times f(x,y) \tag{3}$$

where

$$a(u),\,a(v)=\begin{cases}\dfrac{1}{\sqrt{N}}, \text{ for } (u,v)=0\\ \dfrac{2}{\sqrt{N}}, \text{ for } (u,v)\neq 0\end{cases}$$

3.3. Feature selection using binary PSO

Feature selection [6] is performed to reduce the dimensionality of facial image so that the features extracted are as representative as possible. Method employed here is Binary PSO. Consider a database of L subjects or classes, each class W_1, W_2, W_3,…..,W_L with N_1,N_2,N_3,…N_L number of samples. Let M_1, M_2, M_3, ..., M_L be the individual class mean and M_0 be mean of feature vector. Fitness function is defined so as to increase the class separation equation. By minimizing the fitness function, class separation is increased. With each iteration, the most important features are selected. Binary value of 1 of its position implies that the feature is selected as a distinguishing feature for the succeeding iterations and if the position value is 0 the feature is not selected. The expressions for class, individual mean and mean of feature of feature vector are shown below.

$$W_j^{(i)}, \text{ for } j = 1, 2, \ldots, N_i \tag{4}$$

$$M_i = \frac{1}{N_i} \sum_{j=1}^{N_i} W_j^{(i)}, \text{ for } i = 1, 2, \ldots, L \tag{5}$$

$$M_0 = \frac{1}{N} \sum_{i=1}^{L} N_i \times M_i \tag{6}$$

3.4. Matching

Matching is done by calculating minimum of Euclidean distances of features of the test image with feature of each image in the database using the equation: $\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{N} (f_I - f_T)^2}$ where f_I and f_T are the feature vectors of database image I and test image T respectively. Minimum Euclidean distance gives the closest matching image from the database.

4. Experiments and Results

Table 2 shows the various parameters of Binary PSO during the experiment.

Table 2: Parameters of Binary PSO

Parameters	C_1	C_2	Swarm Size	Iterations per Trial	Number of Trials	Threshold
Values	0.618	1.618	30	100	100	0.5-0.92

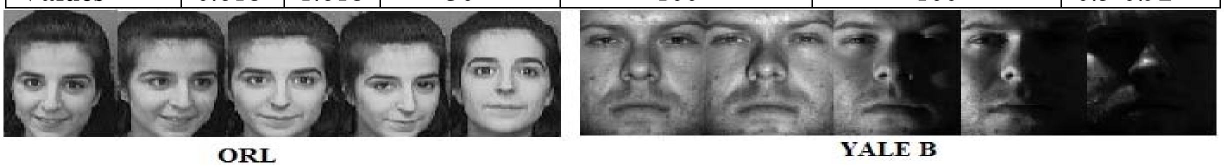

Fig. 2: Sample images from ORL and YALE B Databases

The proposed Face Recognition System was compared with that using the Binary PSO without any image preprocessing. Databases used were ORL and YALE B. Images are preprocessed. Only a part of 2D-DCT coefficients like 20×20, 30×30, 40×40 and 50×50 matrix were used, out of which only the minimum required DCT coefficients or features were further selected using the Binary PSO. Simulation was done in MATLAB 7.9.0.

Table 3 illustrates the results obtained during simulation. For ORL database the maximum recognition reached is 97.9% and the reduction in number of coefficients is 37% and in YALE B database the maximum recognition rate is 99.3% and the reduction in number of coefficients is 7%. The proposed method gives excellent results as compared to Binary PSO without preprocessing.

Table 3: Simulation results

Database	**DCT Coeff**	**20×20**		**30×30**		**40×40**		**50×50**	
		Unmo-dified	Mod-ified	Unmo-dified	Mod-ified	Unmo-dified	Mod-ified	Unmo-dified	Mod-ified
YALE B	Recognition Rate(%)	48.02	97.76	52.26	98.59	53.64	98.7	51.53	99.12
	Non Zero Coefficient	193	156	397	368	784	611	1153	1011
ORL	Recognition Rate(%)	92.51	93.52	92.57	93.87	92.6	94.68	92.45	95.7
	Non Zero Coefficient	218	50	336	99	801	159	1244	264

5. Conclusion

Preprocessing of images and fine tuning of binary PSO parameters were found to yield better results in both ORL and YALE B databases in terms of recognition rate and number of DCT coefficients in the final subset (Non zero coefficients). The improvement in the recognition rate was very good in YALE B database as images required more image preprocessing. Number of DCT coefficients was very less in ORL as images were resized to half. Thus proper use of image preprocessing depending upon image sets helps in increasing the recognition rate as well as reducing the number of DCT coefficients in the final subset. Future studies can be made to implement the face recognition system in real time.

6. References

[1] J. Kennedy and R. Eberhart, Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Network*s, vol. 6, pp. 1942–1948, Piscataway, NJ, December 1995.

[2] J. Kennedy and R. C. Eberhart. A Discrete Binary Version of the Particle Swarm Algorithm.*Proc. IEEEInternational Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 4104-4108, Oct. 1997.

[3] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[4] Gregory Shakhnarovich and Baback Moghaddam. Face recognition in subspaces. Technical Report TR2004-41, Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, Massachusetts 02139, 2004.

[5] A. S. Tolba, A.H. El-Baz, and A.A. El-Harby. Face Recognition: A Literature Review. *International Journal of Signal Processing, vol*. 2, no. 2, pp. 88-103. 2006

[6] Rabab M. Ramadan and Rehab F. Abdel Kader. Face Recognition using Particle Swarm Optimization-based selected features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(2):51-66, 2009.

[7] Ziad M. Hafed and Martin D. Levine. Face Recognition using the Discrete Cosine Transform. *International Journal of ComputerVision*, 43(3):167–188, 2001.

[8] F. M. Matos, L. V. Batista, and J. Poel. Face Recognition Using DCT Coefficients Selection. *Proc. of the 2008 ACM Symposium on Applied Computing, (SAC'08)*,pp. 1753-1757, March 2008.

[9] Dr H B Kekre, Tanuja Sarode, Prachi Natu and Shachi Natu. Performance comparison of Face Recognition using DCT against Face Recognition using Vector Quantization Algorithms LBG, KPE, KMGC, KFCG. *International Journal Of Image Processing*, 4(3):377–389,2010.