

Genetic Algorithms for Feature Selection in Face Recognition

Jyoti¹, Sakshi², Divesh³

^{1,2,3}Department of Computer Science & Engineering, Guru Jamabeshwar University of Science and Technology, Hissar, INDIA

ABSTRACT

This paper presents study of simple genetic algorithm and Parallel Genetic Algorithmic approach for feature selection in Face Recognition. This casts the recognition problem in Feature Extraction followed by Feature Selection. The former, adopts PCA technique to extract the eigenvectors of the face images whereas latter exploits the Parallel Genetic Algorithm to select the best set of eigenvectors. ORL face database has been used for experimentation. The simulation results reveal the efficiency and efficacy of the approach when compared with PCA and Simple Genetic Algorithm.

Keywords: Face Recognition, Principal Component Analysis, Simple Genetic Algorithm, Parallel Genetic Algorithm.

I. INTRODUCTION

Face Recognition has always been a fascinating research area. It has drawn the attention of many researchers because of its various potential applications such as security systems, entertainment, criminal identification in a wide range of applications like identity authentication, smart cards, security, access control, surveillance and intelligent human computer interaction [16][18]. For smart interaction, person identification is one of the most crucial building blocks. In past years, interests and research activities in face recognition have increased significantly [1] [2] [3] [4] [18]. Since then, many approaches have been proposed, among them eigenface technique has shown outstanding results in Face Recognition system using the Principal Components Analysis (PCA) method presented by Turk and Pentland in 1991[4]. PCA is used for Feature Extraction from face images and shows that faces can be represented in eigenfaces, as a linear combination of small set of faces. Dimensions of feature space are reduced drastically by PCA technique [5].

There are numerous meta-heuristic techniques which have been implemented for feature selection in Face Recognition and Genetic Algorithm is one of them. It started early in 1970's, when Holland' first proposed Genetic Algorithms [8]. And since, it provides efficacious techniques for optimization and machine learning applications. It uses direct search algorithm based on the

mechanism of evolution such as inheritance, mutation, natural selection, and recombination (or crossover). It is a heuristic method that uses the idea of survival of the fittest [21].

Much of the analysis of genetic algorithm were concerned with implicit parallelism of genetic search that is many alternative subspaces are dynamically sampled in parallel, resulting in an efficient use of the information implicit in the evaluation of the members of the population (Holland, 1975). The distributed nature of parallel genetic over the search space provide power for searching in changing environments and a good reason to adapt with objective function as long as the population remains distributed [12]. As such, there has been extensive research in this field.

This paper illustrates a face recognition approach based on PCA and Parallel Genetic Algorithm (PGA). PCA is used to extract face image features. After feature extraction Parallel GA is used to select the optimal feature vectors and Euclidian Distance is used as a classifier. The primary goal of this paper is to present the performance evaluation of the proposed system using Parallel GA. The rest of the paper is organized as follows: PCA and PGA basics are reviewed in section 2. Proposed Algorithm is presented in section 3. Results and Discussions are reported in section 4 and Conclusion is covered in section 5.

1. PCA and PGA BASICS:

PCA and PGA basics are as follows:

1.1 Principal Component Analysis

The main idea of principle component being the vector recognition; is a linear combination of original face images which best accounts for face image distribution involving the entire image space length N^2 , describes an $N \times N$ image while defining subspace face image called face space. Therefore, as these vectors being the eigenvectors of the covariance matrix corresponding to the original face images and face like in appearance are referred to as eigenfaces[7][17].

In practice, as accurate reconstruction of the face image is not a required, a smaller $M'(M' < N^2)$ best faces having largest eigenvalues is found to be sufficient for recognition. As the eigenfaces span an M' dimensional subspace of the original N^2 image space, the M' significant eigenvectors of the matrix having the largest eigenvalues are sufficient for representation of the reliable faces in face space, which are

characterized by eigenfaces. The classification is done using the Euclidean distance. [10] [5] [7].

1.2 Genetic Algorithm

2.2.2.1. Simple Genetic Algorithm

Genetic algorithms are population based search algorithms. These begin with a set of random trial solutions, called as initial population. The trial solutions maintained in population are termed chromosomes. The chromosomes are encoded into bit strings and each bit is called as a gene. Within the iteration (also termed generations) chromosomes compete to survive, and to find the optimal solutions. Genetic operators are used to alter the population and those chromosomes satisfying optima criterion are likely to survive by selection operator [6]. New solutions are created (searched) by randomly altering the existing chromosomes in population by mating two chromosomes (crossover operator) and producing mutant chromosomes (mutation operator). The fitness of each chromosome is assessed by the measure of, fitness function or objective function. Those chromosomes having the largest fitness value have the higher probability of its survival [8] [9] [15].

2.2.2.2. Parallel Genetic Algorithm

The distributed nature of the genetic search should provide a natural source of power for searching in changing environments and with this; Parallel Genetic Algorithm (PGA) is employed for Feature Selection. PGA has been reported to outperform the implementation of Genetic Algorithm on both serial and parallel machines [15] [8]. The reason behind this improved search quality is that the different "islands" maintain some degree of independence and thus explore different regions of the search space while at the same time sharing information by means of migration. This can also be seen as a means of sustaining genetic diversity [14] [12] [13]. The Island Model [19] [20] of parallel Genetic Algorithm maintains its own subpopulation (called as islands) for search and each island executes a genetic algorithm. The model key concept is that the islands will exchange a portion of their population periodically in a process called migration. A total population N_{Total} could be spread across M islands by giving each island a population size of $N_{Island} = \frac{N_{Total}}{M}$. The Island Model uses two terms: migration interval that is the number of generations after which migration between islands will take place and migration size, the number of individuals in the population to be migrated. Each Island follows Simple genetic Algorithm independently [8] [9].

II. PROPOSED ALGORITHM

The Algorithm begins with a set of random trial solutions generated on all three islands.

Phase1: Chromosome Representation:

The solutions of formulated problem are the index of M' eigenvectors among the chosen eigenvectors after

applying PCA. Numbers of solutions vectors (M') are termed as Length of chromosome. Number of Chromosomes in Population is defined by parameter Population Size (P).

Each Chromosome is represented as

$C^P = [e_1^c e_2^c \dots e_{M'}^c]$ where $c=1, 2 \dots \text{Length of chromosome}$. The FRR of each Chromosome is stored in Fitness matrix.

Population matrix is given by

$$\text{Population} = \left\{ \begin{array}{l} \text{Length of Chromosome}(c) \\ \times \text{Population Size}(P) \end{array} \right\} \quad (1)$$

$$\text{Population} = \left\{ \begin{array}{l} e_1^1 e_2^1 \dots e_{M'}^1 \\ e_1^2 e_2^2 \dots e_{M'}^2 \\ \dots \\ e_1^P e_2^P \dots e_{M'}^P \end{array} \right\}$$

After this initialization, Fitness is computed that is given by

$$\text{Fitness} = \text{Max}[FRR(C^1), FRR(C^2), \dots, FRR(P)] \quad (2)$$

where FRR is Face Recognition Rate.

The fitness of each chromosome is assessed by the measure of above described fitness function, or objective function. Those chromosomes having the largest fitness value on one island are being replaced by the worst fitness value chromosome in its neighbor. After a fixed migration interval, the migration between worst and best chromosomes is allowed. Chromosomes that are to be transferred between islands are fixed through migration size parameter. This process is continued until the stopping criterion of the algorithm is not reached that is the; Maximum number of iterations.

Phase 2: Genetic Operators:

The Operators used in genetic algorithm for stochastic search in this paper are Selection, Crossover and Mutation. Within the iterations; chromosomes compete to survive, to find the optimal solutions and genetic operators are used to alter the population and those chromosomes satisfying optimal criterion are likely to survive by selection operator. New solutions are created (searched) by randomly altering the existing chromosomes in population by mating two chromosomes (crossover operator) and producing mutant chromosomes (mutation operator).

1) Selection:

Roulette-wheel strategy is being used for each chromosome in population. The probability of selection is

$$P_s = \frac{F(c)}{\sum_{i=1}^N F(c)} \quad (3)$$

For a population of N probability of selection of chromosome will be given by above equation.

2) Crossover Operator:

Each chromosome is randomly picked for mating. For each mating, a crossover point is selected randomly. The two selected chromosomes are exchanged up to this point accordingly.

3) Mutation Operator:

Each chromosomes gene position is assigned a probability for undergoing mutation. Flipping is done in order to maintain diversity within the population and avoid premature convergence.

Phase3: Check the stopping criterion:

Algorithm terminate when the maximum number of improvisations is reached. The current best chromosome is selected from the population after the termination criterion is satisfied. Feature selection using Parallel GA can be summarized using the diagram:

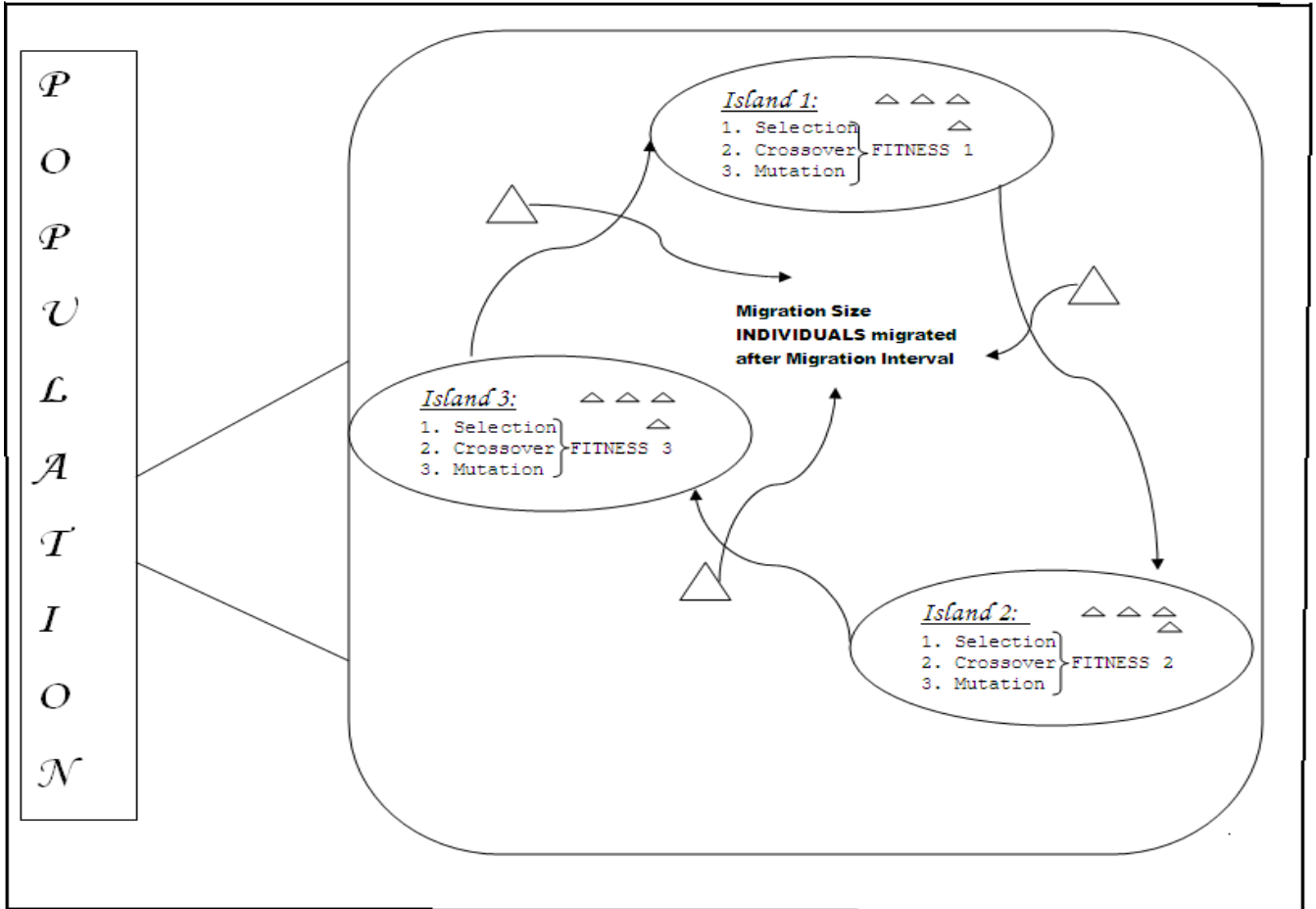


Figure1: Parallel Genetic Algorithm Using Island Model

III. RESULTS AND DISCUSSIONS

Experiments were conducted using Olivetti Research Laboratory (ORL) database [11]. The original image was resized into 40×40 before computation of the face images. Experiments were conducted for different sets consisting of Five training and Five test images for the **First**, Three training and Seven test images for

Second, and Seven training and Three test images for the **Third** of each subject respectively. 40% of the total number of eigenvectors was chosen after feature extraction by using PCA, corresponding to highest eigenvalues.

A comparison is made between PCA, Simple Genetic Algorithm(SGA) and Parallel Genetic Algorithm(PGA) by varying the various parameters of GA like number

of generations (200, 400, 600, 800, 1000), population size i.e. for SGA (90,120,240) and for PGA that is 3 island (30, 40, 80) equivalent to SGA (30*3, 40*3, 80*3). Crossover Rate (0.1, 0.3, 0.6) and Mutation Rate (1/L, 0.1/L, and 10/L) where L is length of chromosome that is number of vectors selected from this M' solution of Eigen vectors were used for the experimentation. For Parallel GA island model; 3 islands were used for proposed algorithm and the migration interval was kept 10 with migration size parameter as 2.

The experiment was conducted for all the above described GA parameters but the paper yields the results only at fixed number of generations and number of chromosomes; at different mutation and crossover rate parameter for which the proposed algorithm yields the maximum FRR.

Experiment 1: First set (Five Training and Five Test Images)

a) Recognition Rate using 20% of Total Eigen Vectors

Table1.1: Recognition Rate using 20% Eigen Vectors

Number of features selected=4																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
		0.25			0.025			2.5			0.25			0.025			2.5		
	Mutation	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
98		86	88	86	86	90	94	94	88	94	94	94	94	94	94	94	94	94	94

b) Recognition Rate using 40% of Total Eigen Vectors

Table1.2: Recognition Rate using 40% Eigen Vectors

Number of features selected=8																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
		0.125			0.0125			1.25			0.125			0.0125			1.25		
	Mutation	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
98		94	96	98	98	98	98	96	96	98	98	98	98	98	98	98	98	98	98

c) Recognition Rate using 60% of Total Eigen Vectors

Table1.3: Recognition Rate using 60% Eigen Vectors

Number of features selected=12																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
		0.083			0.0083			0.83			0.083			0.0083			0.83		
	Mutation	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
98		98	98	98	98	98	98	98	98	98	98	98	98	98	98	98	98	98	98

d) Recognition Rate using 80% of Total Eigen Vectors

Table1.4: Recognition Rate using 80% Eigen Vectors

Number of features selected=16																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
		0.0625			0.00625			0.625			0.0625			0.00625			0.625		
	Mutation	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
98		98	98	98	98	98	98	98	98	98	98	100	100	100	100	100	98	98	100

In case (a) by selecting only 4 features; PCA Recognition Rate(RR) was more than SGA and PGA, even the number of generations for SGA and PGA were 800 and in (b) selection of only 8 features made

PCA and PGA Recognition rate equivalent, and SGA rates vary with mutation and crossover rates. In case (c) RR for all the techniques are equivalent and in case (d) PGA has outperformed with Recognition Rate as 100 by selecting 16 features and gives an

excellent result when mutation rate was 0.00625.

Experiment 2: Second set (Three Training and Seven Test Images)

a) Recognition Rate using 20% of Total Eigen Vectors

Table2.1: Recognition Rate using 20% of Total Eigen Vectors

Number of features selected=4																			
Number of Generations=800																			
PCA	SIMPLE GENETIC ALGORITHM(Popsize=240)										PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.25			0.025			2.5			0.25			0.025			2.5		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
98		86	88	86	86	90	94	94	88	94	94	94	94	94	94	94	94	94	94

b) Recognition Rate using 40% of Total Eigen Vectors

Table2.2: Recognition Rate using 40% of Total Eigen Vectors

Number of features selected=5																			
Number of Generations=800																			
PCA	SIMPLE GENETIC ALGORITHM(Popsize=240)										PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.2			0.02			2			0.2			0.02			2		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
92.85 71		97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29

c) Recognition Rate using 60% of Total Eigen Vectors

Table2.3: Recognition Rate using 60% of Total Eigen Vectors

Number of features selected=7																			
Number of Generations=800																			
PCA	SIMPLE GENETIC ALGORITHM(Popsize=240)										PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.142			0.0142			1.42			0.142			0.0142			1.42		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
92.85 71		97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	98.57 1	98.57 1	98.57 1	97.14 29	98.57 1	97.14 29

d) Recognition Rate using 80% of Total Eigen Vectors

Table2.4: Recognition Rate using 80% of Total Eigen Vectors

Number of features selected=10																			
Number of Generations=800																			
PCA	SIMPLE GENETIC ALGORITHM(Popsize=240)										PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.1			0.01			1			0.1			0.01			1		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
92.85 71		97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	97.14 29	98.57 1	98.57 1	98.57 1	98.57 1	98.57 1	98.57 1

In case (a) the PCA outperformed SGA and PGA by selecting only 2 features and providing the Recognition Rate as 92.8571; in case (b) selection of only 5 features at 800 generations made the Recognition rate of SGA equivalent to PGA . The convergence to maximum recognition rate for SGA took time whereas PGA attained its maximum RR at only 200 generations. In

case (c) PGA has outperformed SGA and PCA by selecting only 7 vectors and providing the RR of 98.571% at mutation rate of 0.0142. When number of features selected were increased to 10 in case (d) PGA performance was increased by all the mutation rates i.e. 0.1, 0.01 and 1.0.

Experiment 3: **Third set** (Seven Training and Three Test Images)

a) Recognition Rate using 20% of Total Eigen Vectors

Table3.1: Recognition Rate using 20% of Total Eigen Vectors

Number of features selected=6																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.16			0.016			1.66			0.16			0.016			1.66		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
96.66		96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66
67		67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67

b) Recognition Rate using 40% of Total Eigen Vectors

Table3.2: Recognition Rate using 40% of Total Eigen Vectors

Number of features selected=11																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.09			0.009			0.9			0.09			0.009			0.9		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
96.66		96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	96.66	100	100	100	100	100	100	100	100	100
67		67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67

c) Recognition Rate using 60% of Total Eigen Vectors

Table3.3: Recognition Rate using 60% of Total Eigen Vectors

Number of features selected=17																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.058			0.0058			0.58			0.058			0.0058			0.58		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
96.66		96.66	96.66	100	96.66	100	96.66	96.66	96.66	96.66	100	100	100	100	100	100	100	100	100
67		67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67

d) Recognition Rate using 80% of Total Eigen Vectors

Table3.4: Recognition Rate using 80% of Total Eigen Vector

Number of features selected=22																			
Number of Generations=800																			
PCA		SIMPLE GENETIC ALGORITHM(Popsize=240)									PARALLEL GENETIC ALGORITHM(Popsize=80)								
	Mutation	0.045			0.0045			0.45			0.045			0.0045			0.45		
	CrossOver	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6	0.1	0.3	0.6
96.66		100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
67		67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67	67

In case (a) when 6 features were selected RR was the same for all the techniques and SGA attained its maximum RR of 96.6667% at 800 generations. In case (b), PGA outperformed PCA and SGA by selecting only 11 features, providing 100% Recognition Rate. RR for SGA is equivalent to PGA at crossover rate 0.6 and 0.1 with mutation as 0.058 and 0.0058 respectively in case(c). SGA took time to attain

its maximum Fitness whereas PGA attained that fitness in just 200 generations. In case (d), 22 features were selected and SGA attained its maximum fitness value which was equivalent to PGA. Both SGA and PGA outperformed the standard eigenface technique. PGA outperforms SGA when its mutation rate is 0.1/L, Crossover rate is 0.6, and number of chromosomes is 80, number of generations 800, number of islands 3,

migration rate 2 and migration interval 10. The results demonstrate that the recognition rate decreases for same number of classes as the number of training images is decreased (set-2) whereas on the other hand, there is substantial increase in recognition rate when the number of training images is increased (set-3). The results depict that the proposed technique gives better performance as compared to eigenface method and SGA. It was further observed that less population size on each island and more number of generations provide an increased value of recognition rate with crossover and mutation rate as 0.6, 0.1\|L respectively where L is Numbers of features selected.

IV. CONCLUSION

This paper presents meta-heuristic techniques for feature selection using Simple GA (SGA) and Parallel GA (PGA) and in PGA specific Island Model (ring Topology) is being used to find the subset of eigenvectors. The results show that PGA yields better results as compared to standard eigenface method and SGA. SGA has displayed better search performance than PCA and Parallel Island Models has displayed better search performance than serial Single population models (e.g. Simple GA), in terms of quality of solutions found. Proposed Algorithm helps to get 100% recognition rate for only 11 numbers of features whereas, SGA and PCA method yields a 96.6667% RR for same number of features (In Set3: For 40% of eigenvalues). In PGA the RR was increased corresponding to the increase in number of training images. It shows better result from SGA even when the number of generations is less and population size is less on each island.

REFERENCES

- [1] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern Recognition", 25:65-77, 1992.
- [2] D. Valentin, H. Abdi, and G. W. Cottrell, A. J. O.Toole., "Connectionist models of face processing: A survey. Pattern Recognition", 27:1209-1230, 1994.
- [3] S. S. R. Chellappa, C. L. Wilson, "Human and machine recognition of faces: A survey", *Proc, IEEE*, 83:705-741, May 1995.
- [4] W.Zhao, R. CHELLAPPA, P. J. PHILLIPS, A. ROSENFELD, "Face Recognition: A Literature Survey", *ACM Computing Surveys*, Vol. 35, No. 4, pp. 399-458, December 2003.
- [5] M.N.Ahmadabadi, B.N.Araabi, C.Lucas, "Feature Selection using genetic algorithm and its application to face recognition", *Proceedings of the IEEE*, pp. 1368-1369, 2004.
- [6] D.E. Goldberg, "Genetic algorithms in Search, Optimization and Machine Learning", Addison-Wesley Longman Publishing Co, 1989.
- [7] Turk, M., Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71--86, 1991.
- [8] Srinivas M and Patnaik L, "*Genetic Algorithms: A Survey*", *IEEE Computer*, 27(6), pp.17-26, 1994.
- [9] Darrell Whitley, Soraya Rana, Robert B. Heckendorn, "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence", November 4, 1998.
- [10] Mohamed Rizon, Muhammad Firdaus Hashim, Putesh Saad, Sazali Yaacob, "Face recognition using eigenfaces and neural network", *American Journal of Applied Sciences*, 2006.
- [11] <http://www.cl.cam.ac.uk/ORL/face/database>.
- [12] H. Muhlenbein, "Evolution in Time and Space- The Parallel Genetic Algorithm", In G. Rawlins, editor, *FOGA -1*, pp. 316-337. Morgan Kaufmann, 1991.
- [13] D. Whitley and T. Starkweather, "GENTOR II: A Distributed Genetic Algorithm", *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 2, no. 3, pp. 189-214, 1990.
- [14] Timothy Starkweather, L. Darrell Whitley, and Keith Mathias, "Optimization Using Distributed Genetic Algorithms", In H.P. Schwefel and R. Manner, editors, *Parallel Problem Solving from Nature*, pp. 176-185, Springer/Verlag, 1990.
- [15] Theodore C. Belding, "The Distributed Genetic Algorithm Revisited", In Eshelman, D.(Ed), *Proc. of the 6th Int'l. Conf. on Gas*, Morgan Kaufmann, Kaufmann, San Francisco, CA, pp. 114-121, 1995.
- [16] Thomas Fromherz, Peter Stucki, Martin Bichsel, "A Survey of Face Recognition", *MML Technical Report*, vol. 97, pp. 1-18, 1997.
- [17] P. Aishwarya, "Face Recognition using Multiple Face Eigen Subspaces", *International Journal of Computer Applications*, vol. 1, no. 2, pp. 89-91, 2010.
- [18] G. D. Guo, H. J. Zhang, S. Z. Li, "Pairwise Face Recognition", *Technical Report*, MSR-TR-2001-12.
- [19] Erick Cantú-Paz, "A survey of parallel genetic algorithms", *Journal Alculateurs Paralleles, Reseaux ET Systems Repartis*, Vol.10, No.2, 1998.
- [20] Cantú-Paz, Erick, "Topologies, migration rates, and multi-population parallel genetic algorithms". *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-99)*. Edited by Banzhaf, W. et al. San Francisco, CA: Morgan Kaufmann Publishers, 1999.
- [21] Wikipedia, the free encyclopedia, Genetic algorithm, Available at: http://en.wikipedia.org/wiki/Genetic_algorithm

