Performance Evaluation of Ant Colony Optimization and Genetic Algorithm for Facial Feature Selection

Adigun Abimbola Adebisi[A], Adegun Adekanmi Adeyinka[B] and Asani Emmanuel Oluwatobi[B]

[A]Ladoke Akintola University of Technology, Nigeria
[B]Landmark University, Nigeria

Abstract

Feature selection is one of the most imperative steps towards developing any robust facial recognition system. Researches carried out so far have shown that of all the evolutionary algorithms used for the purpose of facial feature extraction, Ant Colony Optimization (ACO) and Genetic Algorithm (GA) are the most widely used. ACO is widely used because of its ability to build its solutions sequentially while GA is widely embraced because of its ability to exploit accumulating information about an initially unknown search space in order to bias subsequent search into promising subspaces. This research work presents a performance analysis and evaluation of both algorithms for facial feature extraction using accuracy, computational time, false acceptance rate and false rejection rate as performance metrics. Results show that GA performs better in terms of accuracy, computational time, false acceptance rate and false rejection rate. Hence, it can be concluded that GA is a better algorithm for facial feature extraction in terms of the performance metrics used.

Keywords: *Feature selection, Ant colony optimization, facial feature selection*.

I. Introduction

How robust a facial recognition system will be could be predicted from how successful the feature selection stage is. Feature Selection (FS) aims to reduce the feature set dimensionality (as in [5]) by selecting a subset of features that performs the best under some classification criterion. FS achieves this by eliminating irrelevant, redundant and noisy features from the feature space which results in a reduction of the training time, computational cost and further improves the performance of classifiers [7]. In general, feature selection is a search problem according to some evaluation criterion and it has three main components which are: Generating subsets of features, Evaluation criteria and Stopping criteria [2].

In generating subsets of features, sequential or random approach can be employed. Starting with an empty subset and gradually adding one feature at a time is called sequential forward selection while starting with a full set and removing one feature at a time is called sequential backward selection. Alternatively, a random subset of features could be generated. To evaluate a criterion, some measures need to be used to decide which feature subset to keep. Each subset needs to be evaluated by a certain evaluation criterion and compared with the previous best one with respect to this criterion. Evaluation criteria can be categorized broadly into two groups which are wrapper and filter approaches (also called open-loop, reset bias, front end methods, or independent criterion) based on their dependence on the learning algorithm applied on the selected feature subset [3]. Stopping criteria involves forcing the loop of generating and evaluating feature subsets to terminate. For exhaustive or sequential feature subset generation, the loop will naturally stop when a full feature subset becomes empty or an empty subset becomes full [2].

Finding the optimal feature selection is an NP-hard optimization problem that involves searching the space of possible feature subsets to identify the optimal one. There are 2n states in the search space where n is the number of features in the dataset. For large n values, evaluating all the states is computationally infeasible [1]. Many algorithms such as GA [4, 8, 10, 12, 13, 14, and 15], Tabu Search (TS), Simulated Annealing (SA) and ACO [4, 6, 7, 8, 9, 11, 15, and 17] have been widely used for solving facial feature selection. In this paper, a performance analysis of GA and ACO evolutionary algorithms for facial feature selection is presented. The performance evaluation metrics used includes accuracy, computational time, False Acceptance Rate (FAR) and False Rejection Rate (FRR).

II. ANT COLONY OPTIMIZATION (ACO)

ACO is an evolution simulation algorithm proposed by authors in [15] as a novel nature-inspired meta-heuristic solution for hard Combinatorial Optimization (CO) problems. ACO is inspired by the foraging behavior of some ant species in their search for the shortest paths to food sources. These ants deposit same amount of pheromone on their paths in order to mark some favorable path that should be followed by other members of the colony, so shorter paths will receive more pheromone per unit time. The ability of real ants to find shortest routes is mainly due to their depositing of pheromone as they travel; each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others diminished until all ants follow the same shortest path (here, it is said that the "system" has converged to a single solution) [16]. ACO has been successfully applied in the selection of optimal minimal subset of feature and can be used in any optimization problem, if the following aspects are provided [15]:

Probabilistic forward ants and solution construction: Forward ants build a solution by choosing probabilistically the next node to move to among those in the neighborhood of the graph node on which they are located.

Deterministic backward ants and pheromone update: The use of an explicit memory allows an ant to retrace the path it has followed while searching the destination node.

Pheromone updates based on solution quality: In ACO, the ants memorize the nodes they visited during the forward path, as well as the cost of the arcs traversed if the graph is weighted. They can therefore evaluate the cost of the solutions they generate and use this evaluation to modulate the amount of pheromone they deposit while in backward mode.

Pheromone evaporation: In real ant colonies, pheromone intensity decreases over time because of evaporation. In ACO, evaporation is simulated by applying an appropriately defined pheromone evaporation rule. Pheromone evaporation reduces the influence of the pheromones deposited in the early stages of the search, when artificial ants can build poor-quality solutions.

A. ACO ALGORITHM USED FOR THE FACIAL FEATURE SELECTION

Given a feature set of size n, the feature selection problem is to find a minimal feature subset of size s where s < n while maintaining a fairly high classification accuracy in representing the original features. Adopting an ant optimization algorithm on a discrete search space represented by a digraph with only O(n) arcs as shown in Figure 1, the nodes represent features and the arcs connecting two adjacent nodes indicates the choice of the next feature.

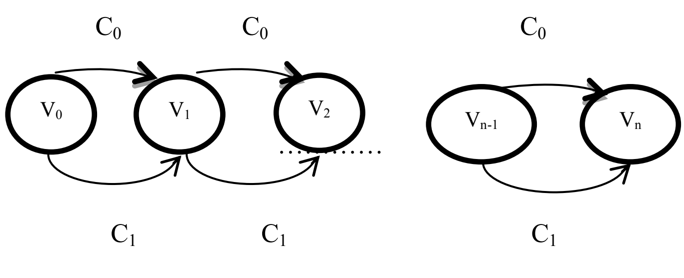

Figure 1: ACO digraph

Denoting the n features as $k_1, k_2 \ldots, k_n$, the i th node v_1 is used to represent feature k_1. An additional node v_0 was placed at the beginning of the graph where each ant starts its search. As shown in Figure 1, the ants travel on the digraph from v_0 to v_1 and then to v_2 and so on. The ant terminates its tour and outputs this feature subset as it reaches the last node v_n. When an ant completes the search from v_0 to v_n, the arcs on its trace form a solution.

Two arcs named C^o_j and C^i_j link two adjacent nodes v_{j-1} and v_j. If an artificial ant at v_j selects arc C^0_j (or C^1_j), the j^{th} feature is selected (or not selected). On each arc C^i_j, virtual pheromone value τ^j_i is assigned as the feedback information to direct the ants' searching on the graph. The pheromone matrix τ is initialized as

$$\tau^j_i = 1 \text{ for all } i= 1, 2\ldots n \text{ and } j = 0, 1.$$

A probabilistic function of transition, denoting the probability of an ant at node v_{i-1} to choose the path C_{ji} so as to reach v_i was designed by combining the heuristic desirability and pheromone density of the arc. The probability of an ant at node v_{i-1} to choose the arc C_{ij} at time t is:

$$P_i^{\,j}(t) = \frac{[\tau_i^{\,j}(t)(\eta_i^{\,j})^{\beta}}{[\tau_i^{\,0}(t)]^{\alpha}(\eta_i^{\,0})^{\beta} + [\tau_i^{\,1}(t)]^{\alpha}(\eta_i^{\,1})^{\beta}}$$

$$(i = 1,2,\ldots,n; \quad j = 0,1) \qquad (1)$$

Here, $\tau_i^{\,j}(t)$ is the pheromone on the arc $c_i^{\,j}$ between nodes v_{i-1} and v_i at time t, which reflects the possibility of an ant to follow arc $c_i^{\,j}$ (j=0, 1). $\eta_i^{\,j}$ is the heuristic information reflecting the desirability of choosing arc $c_i^{\,j}$. α and β are two parameters that determine the relative importance of the pheromone and the heuristic information.

From (1) the transition probability used by ACO depends on the pheromone intensity $\tau_i^{\,j}$ (t) and heuristic information $\eta_i^{\,1}$. To effectively balance the influences of positive feedback information from previous high-quality solutions and the desirability of the arc, proper values of the parameters α and β were chosen. When α= 0, no positive feedback information is used but when β=0, the potential benefit of arcs is neglected and it becomes entirely a random search.

The heuristic information $\eta_i^{\,1}$ is the desirability of choosing the arc $c_i^{\,j}$ between nodes v_{i-1} and v_i, which means the preference of an ant to choose the feature k_i. There are many ways to define a suitable value of $\eta_i^{\,1}$. It could be any evaluation function on the discrimination ability of a feature k_i, such as rough set dependency measure or entropy-based measure. The value of $\eta_i^{\,1}$ using F-score, which was an easy measurement to evaluate the discrimination ability of feature k_i is defined as follows:

$$\eta_i^{\,1} = \frac{\sum_{c=1}^{m}\left(\overset{-c}{x_i} \quad \overset{-}{-x_i}\right)}{\sum_{c=1}^{m}\left[\frac{1}{N_i^{\,c}-1}\sum_{j=1}^{N_i^{\,c}}\left(x_{ij}^{\,c} - \overset{-c}{xi}\right)^2\right]}$$

$$(i = 1,\ldots n) \qquad (2)$$

Here, m is the number of classes of the image set; n is the number of features; $N_i^{\,c}$ is the number of samples of the feature k_i in class c, (c =1,2,…,m, i=1,2,…,n), $x_{ij}^{\,c}$ is the $j^{\,th}$ training sample for the feature k_i of the images in class c, (j=1,2,…,N_i^c), x_i is the mean value of the feature k_i of all images, x_i is the mean of the feature k_i of the images in class c.

As shown in equation 2, the numerator indicates the discrimination between the classes of the image set, and the denominator specifies the discrimination within each class. A larger η_i^1 value implies that the feature k_i has a greater discriminative ability.

For the value of η_i^0, set $\eta_i^0 = \frac{\xi}{n}\sum_{i=1}^{n}\eta_i^1$, where ξ ε (0, 1) is a constant.

III. GENETIC ALGORITHM (GA)

GA are population based search algorithms which begin with a set of random trial solutions known as the initial population, this trial solutions maintained in population are termed chromosomes [13]. GA is based on the mechanics of biological evolution such as inheritance, mutation, natural selection, and recombination (or crossover). As discussed in [17], each individual in the population known as chromosome represents a particular solution of the problem. Each chromosome is assigned a fitness value depending on how good its solution to the problem is. The fitness of each chromosome is assessed by the measure of fitness function or objective function where chromosomes having the largest fitness value have the higher probability of its survival [13]. After fitness allotment, the natural selection is executed and the 'survival of the fittest chromosome' can prepare to breed for the next generation. A new population is then generated by means of genetic operations highlighted below:

Cross-over and mutation: This evolution process is iterated until a near-optimal solution is obtained or a given number of generations are reached.

Fitness function: In order to identify the best individual during the evolutionary process, a function needs to assign a degree of fitness to each chromosome in every generation. Hence, in order to determine whether the assumed region of the input image is a face or not, the fitness value of the possible face region is computed by means of similarity.

Selection: Selection operator is a process in which chromosomes are selected into a mating pool according to their fitness function. Good chromosomes that contribute their gene-inherited knowledge to breed for the next generation are chosen.

Cross-over: Crossover is the process of exchanging portions of the strings of two ''parent'' chromosomes. This operator works on a pair of chromosomes and produces two offspring by combining the partial features of two chromosomes. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms [18].

Mutation: Mutation consists of changing an element's value at random, often with a constant probability for each element in the population; this operator alters genes with a very low probability. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA. However, values of between 0.001 and 0.01 are not unusual for mutation probability [18].

A. GA Used for the Facial Feature Selection

For GA-based feature selector, we set the length of chromosomes to L with gene g_i and the values for L were varied depending on our population.

If $g_i = 1$, this means the feature component was selected as one of the optimal components. Otherwise, $g_i = 0$ means the feature component was to be discarded, the probability of every bit being equal to 1 is set to 0.8 when the initial population of chromosomes is being created, this is done so as to speed up the convergence.

Given a chromosome q, the fitness function F (q) is defined as:

$$F(q) = \frac{1}{\sum_{x=\Omega}\delta(x,q)}$$

Here Ω is the training image set for GAs and δ (x, q) is defined as:

$$\delta(x,q) = \begin{cases} 1, & \text{if x is classified correctly} \\ 0, & \text{if x is classified incorrectly} \end{cases}$$

In order to select the individuals for the next generation, GA's roulette wheel selection method was used.

3.1.1 Genetic Operators

Selection, Crossover and Mutation were the genetic operators used in this paper, during the iteration process, chromosomes used compete to survive so as to find the optimal solutions. Genetic operators were mainly used to alter the population and to enable those chromosomes satisfying optimal criterion to survive due to the selection operator used. New solutions were created by randomly altering the existing chromosomes in population; this was achieved by mating two chromosomes based on the crossover operators and producing mutant chromosomes based on the mutation operator.

1) Selection:

Roulette-wheel strategy was used for each chromosome in population. The probability of selection is defined as:

$$Ps = \frac{F(c)}{\sum_{i=1}^{N} F(c)}$$

2) Crossover Operator:

Each chromosome used was randomly picked for mating. For each mating process, a crossover point was selected randomly. The two selected chromosomes were exchanged at this point accordingly.

3) Mutation Operator:

Each chromosome's gene position was assigned a probability value before undergoing mutation. Flipping was done in order to maintain diversity within the population and also to avoid premature convergence.

4) Checking the stopping criterion:

The algorithm terminates when the maximum number of stipulated improvisations are reached and the current best chromosome was selected from the population after the termination criterion has satisfied

IV. RESEARCH METHODOLOGY

Two popular facial databases were used, they are Dr. Libor Spacek (also referred to as GRIMACE) and Database of Faces (formerly called 'The ORL Database of Faces') facial images were used. GRIMACE is a collection of twenty (20) images each of eighteen (18) individuals. Hence, GRIMACE contains three hundred and sixty (360) facial images of both male and female individual. The second database called the database of faces contains ten (10) different images each of forty (40) distinct subjects. Hence, database of faces contains four hundred (400) facial images. Therefore, a total of seven hundred and sixty (760) facial images were used.

ACO and GA based feature selection methods were applied to each feature set, the length of the selected feature vector and classifier performance were considered for evaluating the two algorithms. Accuracy, computational time, FAR and FRR were the metrics used to evaluate the performance of the two algorithms.

V. RESULTS AND DISCUSSION

A. Performance of ACO and GA based on Recognition Accuracy (in %)

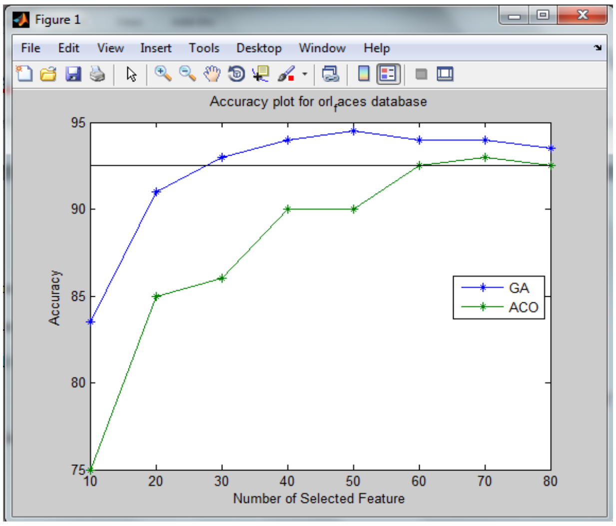

Figure 2: ACO and GA Accuracy Graph for ORL Faces Database

Figure 2 above shows the graph of the accuracy of the dataset in ORL database, it can be seen clearly that the percentage of accuracy in GA increases as the number of selected feature increases which shows that GA performs better in terms of accuracy in the ORL database.

B. Performance of ACO and GA based on Computational Time

Figure 3 shows the graph of ACO and GA's computational time for ORL face database. It can be seen that GA computes faster than ACO.

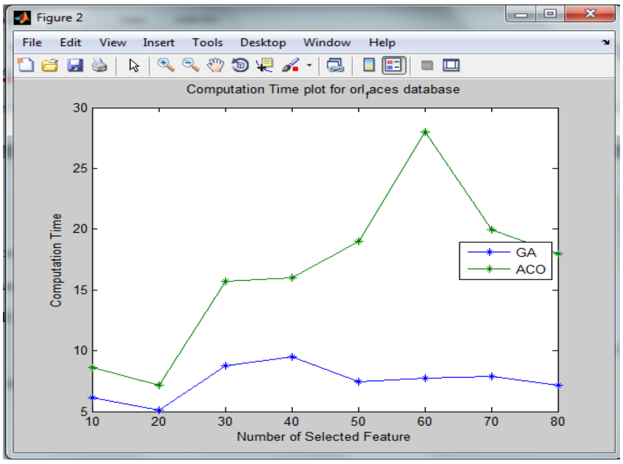

Figure 3: ACO and GA Computational Time Graph for ORL Faces Database

Figure 4 shows the graph of computational time for both ACO and GA using the GRIMACE facial database. It could be seen that the computational time of ACO increases widely as the number of selected feature increases compared to that of GA. This clearly shows the GA performing much better than the ACO.

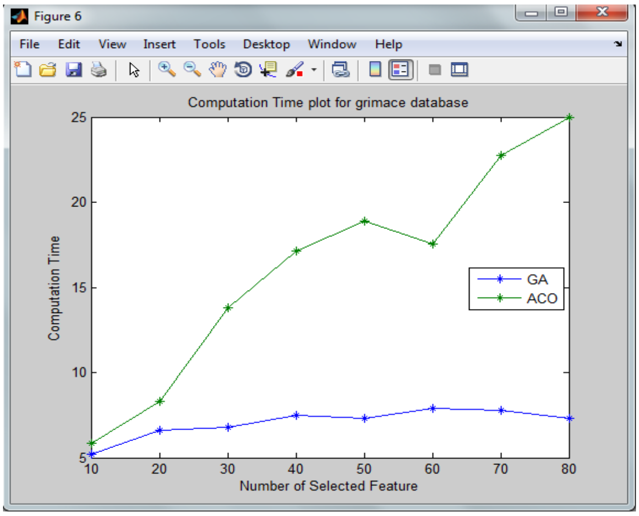

Figure 4: ACO and GA Computational Time Graph for GRIMACE Database

C. Performance of ACO and GA based on False Rejection Rate

Figure 5 shows the graph of ACO and GA for False Rejection Rate (In percentages) when applied to facial images in ORL database. It can be seen that the percentages of FRR in GA are lower than that of ACO. Also showing that the GA performs better than the ACO.

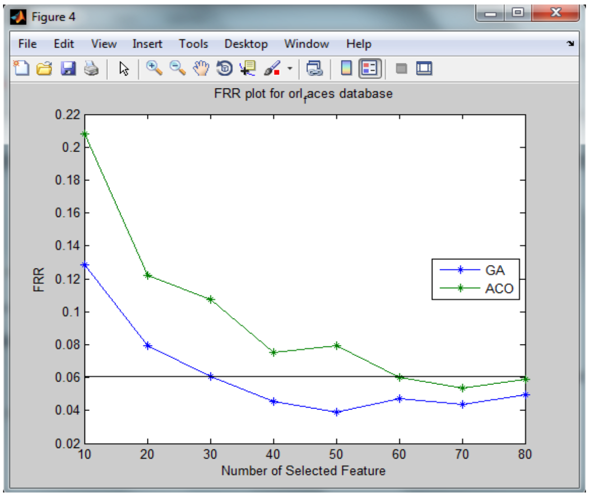

Figure 5: Graph of ACO and GA for False Rejection Rate using ORL Facial Database

Figure 6 shows the graph of ACO and GA for the FRR when applied to facial images in the GRIMACE database. It can be observed that there is a noticeable difference in the value of FRR when the number of selected features is twenty, at this point, GA is preferable to ACO but when the number of features is seventy (70), ACO seems to perform better than GA.

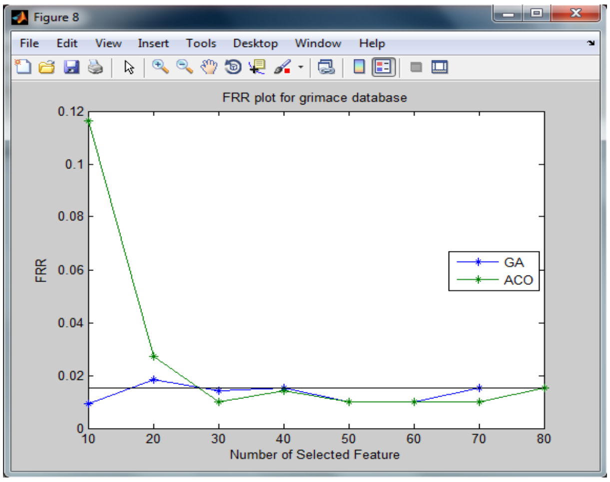

Figure 6: Graphical Representation of ACO and GA for False Rejection Rate using GRIMACE Facial Database

D. Performance of ACO and GA based on False Acceptance Rate

Figure 7 shows the graph of ACO and GA for FAR when applied to facial images in ORL database. It can be seen that the percentage of FAR in GA is lower than that of ACO which shows that GA performs better than ACO in terms of the FAR in ORL database.

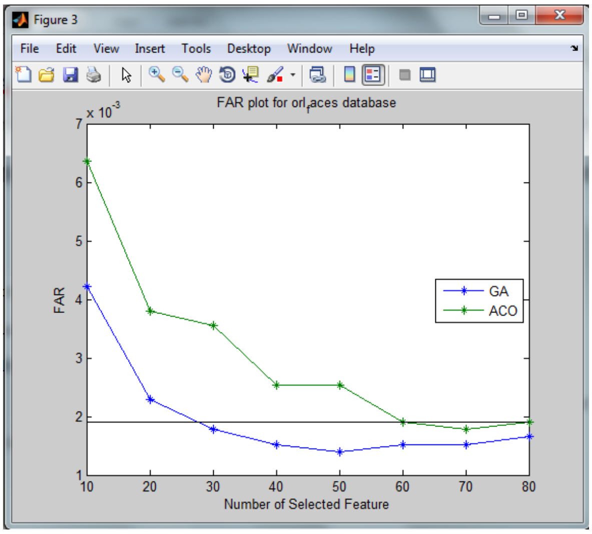

Figure 7: Graph of ACO and GA for False Acceptance Rate using the ORL Database

Figure 8 shows the graph of ACO and GA for FAR when applied to facial images in the GRIMACE database. It can be observed that for up to 25 features selected, GA performed better than ACO and vice versa for 25-40 features selected using the FAR. However, they both had the same FAR for 40-60 features selected and for 60-80 selected features, ACO performed better than GA.

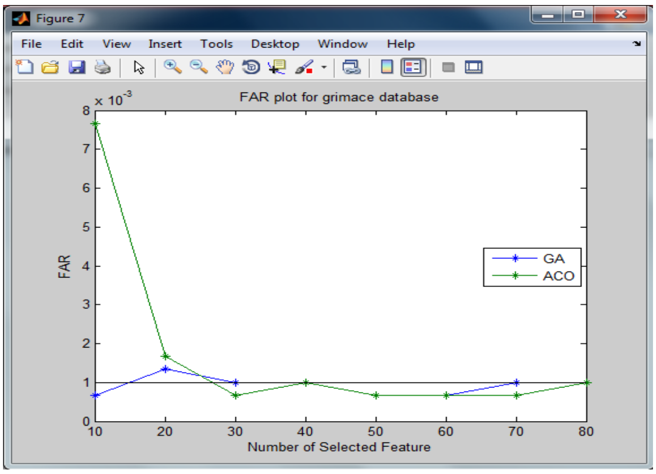

Figure 8: Graph of ACO and GA for False Acceptance Rate using GRIMACE Database

VI. CONCLUSION

In this research study, a Face Recognition System was developed. GA and ACO were used as the facial feature extraction algorithms. A performance evaluation on the basis of accuracy, computational time, false acceptance rate and false rejection rate were carried out on two separate facial databases (ORL and GRIMACE). Results show that GA performs better in terms of accuracy, computational time, false acceptance rate and false rejection rate with the exception of fluctuating performances of both algorithms for the FAR when used on the GRIMACE database. Hence, it can be concluded that GA is the better algorithm for facial feature extraction.

REFERENCES

[1] Nadia Abd-Alsabour and Atef Moneim (2012): Diversification with an Ant Colony System for the Feature Selection Problem, 2nd International Conference on Management and Artificial Intelligence, (IPEDR), Vol.35, 35-39.

[2] W., Klosgen and J. M., Zytkow (2002): Handbook of data mining and knowledge discovery, Oxford University Press.

[3] N. Ye (2003): The Handbook of Data Mining, Lawrence Erlbaum Associates, Inc.

[4] S.Venkatesan and Srinivasa Rao Madane (2010): Face Recognition System with Genetic Algorithm and ANT Colony Optimization, International Journal of Innovation, Management and Technology, Vol. 1, No. 5, ISSN: 2010-0248

[5] Veerabhadrappa, Lalitha Rangarajan (2010): Multi-level Dimensionality Reduction Methods using Feature Selection and Feature Extraction, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.1, No.4, 54-68.

[6] Yuanning LIU, Gang WANG, Huiling CHEN, Zhengdong ZHAO, Xiaodong ZHU and Zhen LIU (2011): An Adaptive Fuzzy Ant Colony Optimization for Feature Selection, Journal of Computational Information Systems Vol. 7, No 4, 1206-1213

[7] Bolun Chen, Ling Chen, Yixin Chen (2012): Efficient ant colony optimization for image feature selection, Journal of Signal Processing, Vol. 93, 1566 –1576

[8] C.Sunil Kumar, C.N Ravi and J.Dinesh (2014): Human Face Recognition and Detection System with Genetic and Ant Colony Optimization Algorithm, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 4, e-ISSN: 2278-0661, 11-15

[9] Shailendra Kumar Shrivastava and Pradeep Mewada (2011): ACO Based Feature Subset Selection for Multiple k-Nearest Neighbor Classifiers, International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 5, 1831-1838, ISSN: 0975-3397.

[10] Pratibha P. Chavan, M Murugan and Pramod U. Chavan (2014): Genetic Algorithm based Feature Subset Selection in Face Detection, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), ISSN: 2278-2834-, ISBN: 2278-8735, 27-31.

[11] Yuanning LIU, Gang WANG, Huiling CHEN, Zhengdong ZHAO, Xiaodong ZHU and Zhen LIU (2011): An Adaptive Fuzzy Ant Colony Optimization for Feature Selection, Journal of Computational Information Systems Vol. 7, No 4, 1206-1213.

[12] Manisha Satone and Gajanan Kharate (2014): Feature Selection Using Genetic Algorithm for face recognition based on PCA, Wavelet and SVM, International Journal on Electrical Engineering and Informatics, Volume 6, Number 1, 39-52.

[13] Jyoti, Sakshi, Divesh (2014): Genetic Algorithms for Feature Selection in Face Recognition, International Journal of Engineering and Management Research, Volume 4, Issue 1, ISSN No.: 2250-0758

[14] Ashkan Parsi, Mehrdad Salehi and Ali Doostmohammadi (2012): Swap Training: A Genetic Algorithm Based Feature Selection Method Applied on Face Recognition System, World of Computer Science and Information Technology Journal (WCSIT), Vol. 2, No. 4, 125-130, ISSN: 2221-0741.

[15] Dorigo.M, Birattari M., Stützle.T (2006): Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, Vol. 11, 28-29.

[16] Maryam Bahojb Imani, Tahereh Pourhabibi, Mohammad Reza Keyvanpour and Reza Azmi (2012): A New Feature Selection Method Based on Ant Colony and Genetic Algorithm on Persian Font Recognition, International Journal of Machine Learning and Computing, Vol. 2, No. 3, 278-282.

[17] S.N. Palod, S. K. Shrivastav, P. K. Purohit (2011): Review of Genetic Algorithm based face recognition, International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 2, ISSN: 0975-5462, 1478-1483

[18] Basiri, M. E. and Nemati, S. (2009): A Novel Hybrid ACO-GA Algorithm for Text Feature Selection, IEEE Congress on Evolutionary Computation (CEC 2009); 2561-2568.