International Journal of Computer and Communication Technology

Volume 2 | Issue 3 Article 4

July 2011

Effective Image Clustering with Differential Evolution Technique

G Sudhakar
Swarnandhra College of Engineering & Technology, Aandra Pradesh, sudha.godi@gmail.com

Polinati Vinod Babu
Swarnandhra College of Engineering & Technology, Aandra Pradesh, vinodbabusir@gmail.com

Suresh Chandra Satapathy
ANITS,Vishakapatnam, Aandra Pradesh, sureshsatapathy@ieee.org

Gunanidhi Pradhan
Principal, Govt College of Engg, Kalahandi,Orissa, gunanidhipradhan@gmail.com

Follow this and additional works at: https://www.interscience.in/ijcct

Recommended Citation

Sudhakar, G; Babu, Polinati Vinod; Satapathy, Suresh Chandra; and Pradhan, Gunanidhi (2011) "Effective Image Clustering with Differential Evolution Technique," *International Journal of Computer and Communication Technology*: Vol. 2 : Iss. 3 , Article 4.
DOI: 10.47893/IJCCT.2011.1089
Available at: https://www.interscience.in/ijcct/vol2/iss3/4

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Effective Image Clustering with Differential Evolution Technique

G Sudhakar

Swarnandhra College of Engineering & Technology,
Aandra Pradesh.
Email: sudha.godi@gmail.com

Polinati Vinod Babu,

Swarnandhra College of Engineering & Technology,
Aandra Pradesh.
Email: vinodbabusir@gmail.com

Suresh Chandra Satapathy

ANITS,Vishakapatnam,
Aandra Pradesh.
Email: sureshsatapathy@ieee.org

Gunanidhi Pradhan

Principal, Govt College of Engg,
Kalahandi,Orissa,
Email: gunanidhipradhan@gmail.com
* Corresponding author

Abstract: The paper presents a novel approach of clustering image datasets with differential evolution (DE) technique. The differential evolution is a parallel direct search population based optimization method. From our simulations it is found that DE is able to optimize the quality measures of clusters of image datasets. To claim the superiority of DE based clustering we have compared the outcomes of DE with the classical K-means and popular Particle Swarm Optimization (PSO) algorithms for the same datasets. The comparisons results reveal the suitability of DE for image clustering in all image datasets.

Keywords: Image Clustering, K-means, PSO, Differential evolution

1 Introduction

Image clustering and categorization is a means for high-level description of image content [1]. The goal is to find a mapping of the archive images into classes (clusters) such that the set of classes provide essentially the same information about the image archive as the entire image-set collection. The generated classes provide a concise summarization and visualization of the image content that can be used for different tasks related to image database management. Image clustering enables the implementation of efficient retrieval algorithms and the creation of a user-friendly interface to the database.

Image clustering approaches can be broadly categorized to two classes: supervised and unsupervised. There are many well known supervised algorithm such as the minimum-distance-to-mean, Gaussian maximum likelihood etc. The main drawback of supervised approach is that if the class value is unknown then it is difficult to implement this approach. This is where the unsupervised approach is very much preferred. There are several algorithms that belong to unsupervised class. All these algorithms further can be classified into two groups: hierarchical and partitional [2][3]. In this paper only partitional algorithm is discussed in which the clustering is formed by minimizing some criteria i.e. squared error function. Hence it can be treated as an optimization problem. The objective here is to minimize the criteria function. K-means[4] is a well known approach for partitional clustering. However, the K-means algorithm is not always able to optimize the mean squared error criterion as it is dependent on initialization values. In this work we have explored the DE [5] approach for optimizing the mean squared error values. DE is a parallel direct search population based optimization method. This approach for numerical optimization is simple to implement and requires little or no parameter tuning, but gives a remarkable performance. Like all other evolutionary algorithms, the initial population is chosen randomly. Three benchmark image datasets are chosen for the clustering purpose. In this work we have implemented DE for clustering. To compare the results obtained with DE we have simulated K-means and PSO [6] for clustering same datasets. The results reveal that K-means algorithm is trapped in local minima in all the problems whereas PSO and DE present better results. Compared to PSO, DE is able to provide more accurate optimized results for all the investigated dataset. The major merit lies with DE is that it does not require many parameter tuning as compared to PSO.

The rest of the paper is organized as follows: An overview of K-means, PSO and DE image clustering are given in section 2. Section 3 describes the image data set and simulation results. Section 4 concludes the paper, and outlines further improvement.

2 K-Means, Pso, And De: Image Clustering Overview

Following are the terminology used to describe the K-means, PSO and DE image clustering algorithms:

- N_d : number of dimension of image data vector
- *Np*: number of image pixels
- *Nc*: number of clusters(as provided by the user)
- *Zp*: pixel p having N_d dimension
- m_j : mean of cluster j

To measure the quality of above three algorithms we have chosen a quality metric called as Quantization error (Q_e) and is given by

$$Q_e = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall z_p \in c_{ij}} \frac{d(z_p, m_j)}{\mathrm{mod}(c_{ij})}\right]}{N_c} \quad \text{---------} \quad (1)$$

Where $d(z_p, m_j) = \sqrt{\sum_{k=1}^{N_\mathrm{d}} \left(z_{pk} - m_{jk}\right)^2}$ -------- (2)

and mod(c_{ij}) is the number of data vectors belonging to the cluster j.

We use pixels as the data objects for image clustering. The image is converted into its corresponding RGB values. The gray scale of these values are computed which represents the intensity of the brightness.

2.1 K-Means Algorithm:

In K-means algorithm data vectors are grouped into predefined number of clusters. At the beginning the centroids of the predefined clusters are initialized randomly. The dimensions of the centroids are same as the dimension of data vectors. Each pixel is assigned to the cluster based on the closeness, which is determined by the Euclidian distance measure given in equation (2). After all pixels are clustered, the mean of each cluster is recalculated. This process is repeated until no significant changes result for each cluster mean or for some fixed number of iterations.

The K-means algorithm is summarized as

1. Randomly initialize the N_c cluster centroid vectors

2. Repeat

(a) For each data vector, assign the vector to the class with the closest centroid vector, where the distance to the centroid is determined using equation (2)

(b) Recalculate the cluster centroid vectors, using

$$m_j = \frac{1}{n_j} \sum_{\forall z_p \in C_j} z_p$$

until a stopping criterion is satisfied

2.2 PSO Algorithm:

Particle swarm optimization (PSO) is a population-based stochastic search process, modeled after the social behavior of a bird flock [6]. The algorithm maintains a population of particles, where each particle represents a potential solution to an optimization problem. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. The aim of the PSO is to find the particle position that results in the best evaluation of a given fitness (objective) function.

Each particle represents a position in N_d dimensional space, and is: "flown" through this multi-dimensional search space, adjusting its position toward both

- The particle's best position found thus far. And
- The best position in the neighborhood of that particle.

Each particle ***i*** maintains the following information:

- x_i : The ***current*** *position* of the particle;
- v_i : The ***current velocity.*** of the particle;
- y_i : The ***personal best position*** of the particle.

Using the above notation, a particle's position is adjusted according to

$$v_{i,k}(t+1) = w v_{i,k}(t) + c_1 r_{1,k}(t)\left(y_{i,k}(t) - x_{i,k}(t)\right) + c_2 r_{2,k}(t)\left(\hat{y}_k(t) - x_{i,k}(t)\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

$r_{1,j}(t), r_{2,j}(t) \sim U(0,1)$ and k=1,….., N_d

Where w is the inertia weight, c_1, c_2 are the acceleration constants The velocity is thus calculated based on three contributions: (1) a fraction of the previous velocity, (2) the cognitive component which is a function of the distance of the particle from its personal best position, and (3) the social component which is a function of the distance of the particle from the hest particle found thus far (i.e. the best of the personal bests) The personal best position of particle *i* is calculated as

$$y_i(t+1) = \begin{cases} y_i(t) & if \;\; f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & if \;\; f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Where f() is the function evaluation.

Two basic approaches to PSO exist based on the interpretation of the neighborhood of particles [7-8]. Equation (3) reflects the *gbest* version of PSO where, for each particle, the neighborhood is simply the entire swarm. The social component then causes particles to be drawn towards the best particle in the swarm. In the *lbest* PSO model, the swarm is divided into overlapping neighborhoods, and the best particle of each neighborhood is determined. For the *lbest* PSO model, the social component of equation (3) changes to.

$$c_2 r_{2,k}(t)(\hat{y}_{j,k}(t) - x_{i,k}(t)) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (6)$$

where $\hat{y}_j$ is the best particle in the neighborhood of the ith particle.

The PSO is usually executed with repeated application of equations (3) and (4) until a specified number of iterations have been exceeded. Alternatively, the algorithm can he terminated when the velocity updates are close to zero over a number of iterations.

In the context of clustering, a single particle represents the N_c cluster centroid vectors. That is, each particle x_i is constructed as follows:

$$x_i = (m_{i,1},....,m_{ij},....,m_{iN_c}) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

where m_{ij} refers to the j-th cluster centroid vector of the i-th particle in a cluster. Therefore, a swarm represents a number of candidate clusters for the current data vectors. The fitness of particles is easily measured as the quantization error given in equation (1).

Using the standard *gbest* PSO, data vectors can be clustered as follows:

1. Initialize each particle to contain N_c, randomly selected cluster centroids. For example Iris data set has Four dimension and three clusters. Hence each particle should have 12 i.e 4*3 dimensions. Here n is number of particles and m is the dimension of particles.

$$\begin{array}{ccccccc}
X_{11} & X_{12} & X_{13} & \text{- - - - - - -} & X_{1i} & \text{- - - - - -} & X_{1m} \\
X_{21} & X_{22} & X_{13} & \text{- - - - - - -} & X_{2i} & \text{- - - - - -} & X_{2m} \\
X_{31} & X_{32} & X_{33} & \text{- - - - - - -} & X_{3i} & \text{- - - - - -} & X_{3m} \\
| & | & | & \text{- - - - - - -} & | & \text{- - - - - -} & | \\
| & | & | & \text{- - - - - - -} & | & \text{- - - - - -} & | \\
| & | & | & \text{- - - - - - -} & | & \text{- - - - - -} & | \\
| & | & | & \text{- - - - - - -} & | & \text{- - - - - -} & | \\
| & | & | & \text{- - - - - - -} & | & \text{- - - - - -} & | \\
X_{n1} & X_{n2} & X_{n3} & \text{- - - - - - -} & X_{ni} & \text{- - - - - -} & X_{nm}
\end{array}$$

```
2. For $t = 1$ to $t_{max}$ do
  (a) For each particle i do
      (b) For each data vector $z_p$
         i) Calculate the Euclidean distance $d(z_p, m_{i,j})$ to all cluster centroids $C_{ij}$
         ii) Assign $z_p$ to cluster $C_{ij}$ such that $d(z_p, m_{ij}) = \min_{\forall c=1,...N_c}\{d(Z_p, m_{ic})\}$
    iii) Calculate the fitness using equation (1)
      (c) Update the global best and local best positions
      (d) Update the cluster centroids using equations (3) and (4)
 where t,,, is the maximum number of iterations.
```

The population-based search of the PSO algorithm reduces the effect that initial conditions have, as opposed to the K-means algorithm; the search starts from multiple positions in parallel.

C. DE Algorithm:

Although PSO is attractive due to its simple concept, few parameters, and easy implementation, it also suffers from the problem of premature convergence, especially in the large scale and complex problem. And it requires many parameters tuning for obtaining optimal results. However, Differential Evolution (DE) is a parallel direct search method developed by Storn and Price in 1997 which is a population-based global optimization algorithm. It uses a real-coded representation [5]. This approach for numerical optimization is simple to implement and requires little or no parameter tuning, but gives a remarkable performance. Like all other evolutionary algorithms, the initial population is chosen randomly.

Like all other evolutionary algorithms, DE method also consists of three basic steps:
(i) Generation of population with N individuals in the d-dimensional space, randomly distributed over the entire search domain
$\overrightarrow{X_i}(t) = [x_{i,1}(t), x_{i,2}(t), x_{i,3}(t) \ldots x_{iD}(t)]$, where t=0,1,2,….t,t+1
(ii) Replacement of this current population by a better fit new population, and
(iii) Repetition of this replacement until satisfactory results are obtained or certain criteria of termination is met.
The basic scheme of evolutionary algorithms is given below:

a) Mutation

After the random generation of population, in each generation, a Donor vector $\overrightarrow{V_i}(t)$ is created for each $\overrightarrow{X_i}(t)$.This donor vector can be created in different ways (see DE mutation schemes).

b) Recombination

Now a trial offspring vector is created by combining components from the Donor vector $\overrightarrow{V_i}(t)$ and the target vector $\overrightarrow{X_i}(t)$. This can be done in the following way

$$U_{i,j}(t) = V_{i,j}(t) \quad \text{if rand}_{i,j}(0,1) <= Cr$$
$$= X_{i,j}(t) \quad \text{otherwise}$$

Where C_r is the probability of recombination.
c) Selection

Selection in DE adopts Darwinian principle "Survival Of the Fittest". Here if the trail vector yields a better fitness value, it replaces its target in the next generation; otherwise the target vector is retained in the population. Hence the population either gets better (w.r.t. the fitness function) or remains constant but never deteriorates.

$$\overrightarrow{X_i}(t+1) = \overrightarrow{U_i}(t) \quad \text{if } f(U_i(t)) \le f(X_i(t)),$$
$$= \overrightarrow{X_i}(t) \quad \text{if } f(X_i(t)) < f(U_i(t)) \quad (8)$$

DE mutation Schemes

The five different mutation schemes suggested by Price [5] is as follows:

Scheme 1-DE/rand/1

In this scheme, to create a donor vector $\overrightarrow{V_i}(t)$ for each ith member, three other parameter vectors (say the o_1, o_2, and o_3th vectors) are chosen randomly from the current population. A scalar number F is taken. This number scales the difference of any two of the three vectors and the resultant is added to the third one. For the ith donor vector, this process can be given as

$$\overrightarrow{V_i}(t+1) = \overrightarrow{X_{o_1}}(t) + F * \left(\overrightarrow{X_{o_2}}(t) - \overrightarrow{X_{o_3}}(t)\right)$$

Scheme 2-DE/rand to best/1

This scheme follows the same procedure as that of the Scheme1. But the difference is, now the donor vector is generated by randomly selecting any two members of the population (say the $\overrightarrow{X}_{0_2}(t)$, and $\overrightarrow{X}_{0,3}(t)$ vectors) and the best vector of the current generation (say $\overrightarrow{X}_{best}(t)$). For the ith donor vector, at time t=t+1, this can be expressed as

$$\vec{V_i}(t+1) = \vec{X_i}(t) + \lambda * \left(\vec{X}_{best}(t) - \vec{X_i}(t)\right) + F * \left(\vec{X}_{o_2}(t) - \vec{X}_{o_3}(t)\right)$$

Where λ is a control parameter in DE and ranges between [0, 2] . To reduce the number of parameters, we consider λ =F.

Scheme 3-DE/best/1

This scheme is identical to Scheme 1 except that the result of the scaled difference is added to the best vector of the current population. This can be expressed as

$$\vec{V}_i(t+1) = \vec{X}_{best}(t) + F * \left(\vec{X}_{o_1}(t) - \vec{X}_{o_2}(t)\right)$$

Scheme 4-DE/best/2

In this scheme, the donor vector is formed by using two difference vectors as shown below

$$\vec{V}_i(t+1) = \vec{X}_{best}(t) + F * \left(\vec{X}_{o_1}(t) - \vec{X}_{0_2}(t)\right) + F * \left(\vec{X}_{o_3}(t) - \vec{X}_{o_4}(t)\right)$$

Scheme 5-DE/rand/2

Here totally five different vectors are selected randomly from the population, in order to generate the donor vector. This is shown below

$$\vec{V}_i(t+`) = \vec{X}_{o_1}(t) + F_1 * \left(\vec{X}_{o_2}(t) - \vec{X}_{o_3}(t)\right) + F_2 * \left(\vec{X}_{o_4}(t) - \vec{X}_{o_5}(t)\right)$$

Here $F1$ and $F2$ are two weighing factors selected in the range from 0 to 1. To reduce the number of parameters we may choose $F1 = F2 = F$.
The experiment we conducted in this study uses Scheme 1-DE/rand/1

Procedure for DE

1. Randomly initialize the position of the particles
2. Evaluate the fitness for each particle
3. For each particle, create Difference-Offspring
4. Evaluate the fitness of the Difference-Offspring

5. If an offspring is better than its parent then replace the parent by offspring in the next generation;
6. Loop to step 2 until the criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

DE Clustering

There are different DE Schemes available [5]. Here we stick to the classical DE algorithm with mutation Scheme 1-DE/rand/1.

Data vectors can be clustered using classical DE as follows:

```
i.   Initialize each vector to contain Nc number of randomly selected cluster centers

ii.  For $I=1$ to $I_{max}$ do

  a) For each vector i do
  b) For each object in the data set $Z_p$
  i.   Calculate the Euclidean distance $d(z_p, m_j)$ to all cluster centroids $C_{ij}$ using equation 2
  ii.  Assign $Z_p$ to the cluster $C_{ij}$ such that $\min_{\forall k=1.....N_k}\{d(z_p, m_j)\}$
  c) Change the population members according to the DE algorithm outlined in this section. Use the vectors fitness to guide the evolution of the population.
iii. Report cluster centers and the partition obtained by globally best vector at time $I=I_{max}$
```

IMAGES AND SIMULATION RESLUTS

The three image clustering algorithms namely K-means, PSO, and DE have been applied to three types of imagery data, namely MRI brain, Lena, and Mandrill. These data sets have been selected for testing and comparing above three algorithms. The three images chosen comprises of 250x250 8-bit gray scale pixels. The figure 1, figure 5, and figure 9 are the original images of MRI brain, Lena, and Mandrill respectively. A total no. of clusters of 8, 8, and 6 were randomly chosen respectively for MRI brain, Lena, and Mandrill images.

The performances of three chosen algorithms are computed by the quantization error given in equation (1) and the intra and inter cluster distances as in [9].

The clustered images of MRI brain, Lena, Mandrill using K-means are shown in figure 2, figure 6, and figure 10 respectively with the quantization error and inter cluster & intra cluster measures shown in the Table 1. For running the PSO we have chosen parameters swarm size as 10, maximum no. of iterations are 30, c1 & c2 are 1.042 both equal [8]. The w value is varied as per [8] in every iteration.

In our experiment w_1 (initial weight), w_2 (final weight), $v_{\max}$ are chosen to be 0.9,0.4 and 10 respectively for best results. The clustered images for PSO MRI brain is shown in figure 3. It can be seen that K-means trapped in local optimum and could not classify the clusters correctly. PSO in other hand is not trapped in this local minimum. This can be verified from the quantization error measure given in the Table 1. The quantization error is 0.13327 which is less than the value for K-means. The value of F and C_r are chosen to be 0.5 and 0.9 respectively as in [5]. The results shown in the table 1 clearly indicates the superiority of DE over other two approaches such as K-means and PSO. In all datasets the quality measures like quantization error (Q_e), intre and inter cluster distances are found to be better for DE over other two algorithms. The better results obtained in DE is due to the fact that it is not being biased towards any prior defined distribution for sampling mutational step sizes and its selection operators follows a hill-climbing process . Better exploitation is achieved in DE due to the determination of mutual steps sizes as the difference between individuals in the current population.

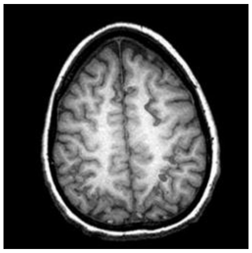
Fig 1

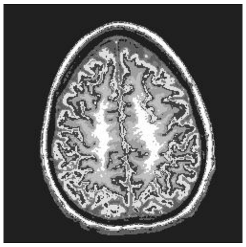
Fig 2

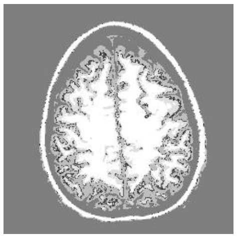
Fig 3

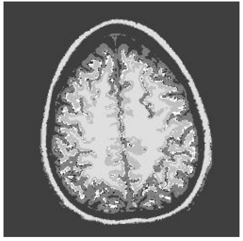
Fig 4

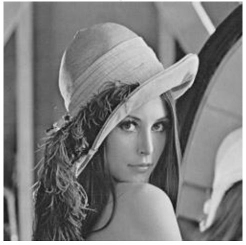
Fig 5

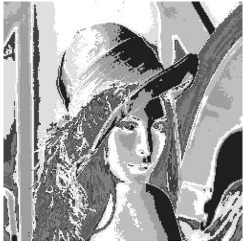
Fig 6

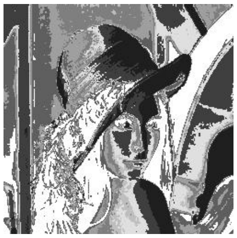
Fig 7

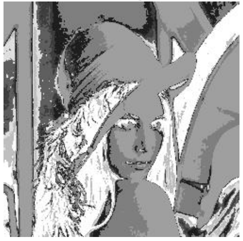
Fig 8

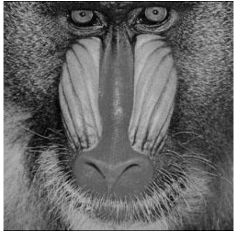
Fig 9

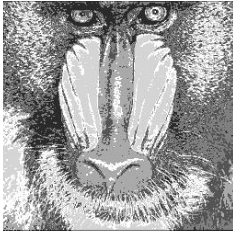
Fig 10

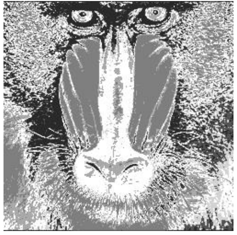
Fig 11

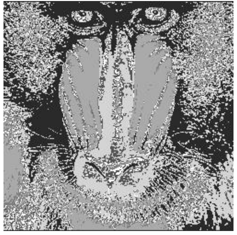
Fig 12

IV.CONCLUSION AND FURTEHR IMPROVEMENT

This paper presented a novel approach of clustering image dataset with DE. The DE clustering results are compared with well known K-means and PSO clustering for all investigated dataset. It was shown that PSO and DE produced better result compared to K-means with respect to the quantization error, inter- and intra-cluster distances. The local optima problem of K-means was alleviated using PSO and DE further improved the results. The main advantage of using DE is found to be having almost no parameter tuning. As further research we will investigate some adaptive approach for tuning F the mutation scale factor used in DE.

Table 1

Image	No. of clusters(User chosen)	Algorithm	Quantization error (Q_e)	Intra- cluster distance	Inter- cluster distance
MRI Brain	8	k-means	0.13819	0.18285	30.6242
		PSO	0.13327	0.12553	31.8400
		DE	**0.1130**	**0.10235**	**32.4646**
Lena	8	k-means	0.07748	0.11933	14.4542
		PSO	0.074451	0.10290	16.2662
		DE	**0.07345**	**0.0923**	**17.1212**
Mandrill	6	k-means	0.085077	0.13067	23.3382
		PSO	0.083897	0.13844	25.4293
		DE	**0.08917**	**0.10124**	**29.4243**

References

[1] J.Goldberger,S.Gordon,Greenspan. "Unsupervised image-set clustering using an information theoreticframework." *IEEE Trans. on Image Processing*, 15(2):449-458, 2006

[2] H.Frigui,R.Krishnapuram," A Roubust Competitive Clustering Algorithm with Applications in Computer Vision," IEEE Transcations on Pattern Analysis and Machine Intelligence, vol.21,no.5,1999.

[3] Y.Leung,J.Zhang,Z.Xu," Clustering by Space-Space filtering," IEEE Trans on Pattern Analysis and Machine Intelligence, vol.22,no.12,pp.1396-1410,2000

[4] J.T.Tou, R.C.Gonzalez "Pattern Recognition Principles", Addision-Wesley, Reading 1974

[5] Storn R and Price K (1997), Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces, Journal of Global Optimization, 11(4), 341–359..

[6] J Kennedy, RC Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942-1948, 1995.

[7] Kennedy, J., Mendes, R.: Population Structure and Particle Swarm Performance, International Proceedings of the 2002 Congress on Evolutionary Computation. Piscatawat, NJ: IEEE service Center (2002) 1671–1675

[8] Suganthan, P.N., "Particle Swarm Optimizer with Neighborhood Operator, In: Proc. Congress On Evolutionary Computation, Washington D.C. USA, July. Piscataway, NJ: IEEE Service Center, (1999) 1958-1961.

[9] M.Omran, AP Engelberecht, A.Salman, " Image Classification using Particle Swarm Optimization," Proceedings of the 4th Asia-Pacific Conference on Simultaed Evolution and Learning, Singapore,2002.