PAPER • OPEN ACCESS

Hyperspectral image segmentation using discriminant independent component analysis and swarm optimization approaches

To cite this article: Murinto and D P Ismi 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **674** 012054

View the article online for updates and enhancements.

Hyperspectral image segmentation using discriminant independent component analysis and swarm optimization approaches

Murinto [*] **and D P Ismi**

Department of Informatics Engineering Universitas Ahmad Dahlan Yogyakarta

* Email : murintokusno@tif.uad.ac.id, dewi.ismi@tif.uad.ac.id

Abstract. Hyperspectral data consists of hundreds of bands. The high dimension of hyperspectral data is a challenge for the researcher to design efficiency and accuracy image segmentation algorithm. In this paper, a new approach to extraction features and independence of hyperspectral image proposes using Discriminant independent component analysis (DICA) and multilevel thresholding techniques based on Otsu problems for image segmentation are introduced. Image segmentation is initial process of image analysis and recognition. Otsu's problem with multilevel thresholding is solved in each band from hyperspectral data that has been reduced dimensionality using DICA. The main purpose using multilevel thresholding is to get an optimal threshold that maximizes variance between classes. The swarm optimization approach used in this study is Darwinian particle swarm optimization (DPSO). The result of experiment showed that DPSO was better compared to other swarm optimization approaches. DPSO shows a statistically significant increase, both from CPU time processing and fitness value. DPSO technique is able to find optimal threshold with greater variance between classes and smaller search times compared to particle swarm optimization (PSO).

Keywords: Discriminant Independent Component Analysis, Darwinian Particle Swarm Optimization, Image Segmentation, Particle Swarm Optimization

1. Introduction

Multispectral and hyperspectral are two groups in spectral imaging. Multispectral and Hyperspectral are two groups in spectral imaging. Hyperspectral imagery (HSI) has more bands, usually more than one hundred bands and its spectral resolution is higher when compared to multispectral image. A hyperspectral image is often obtained through imaging spectrometers, such as HYDICE (Hyperspectral Digital Imagery Collection Experiment) and AVIRIS (Airborne Visible InfraRed Imaging Spectrometer). Each spectral vector corresponds to reflectance value of certain pixels in all spectral wavelengths. The hyperspectral image consists of hundreds data channels. The high dimensions of hyperspectral data make it difficult for researcher to design accurate and efficient image segmentation algorithms on such types of images. A large number of spectral bands show high-dimensional data and present challenges to segmentation, classification and image analysis. Most of the methods used are designed for gray-level analysis, color image or multispectral image not for hyperspectral images.

Applying this method to HSI will be a big challenge because of the increasing dimensions of data. Feature reduction is searching for a vector set that represents reduction dimensions of the observation

data, a process that produces a smaller number of features through a combination of existing channels. Each pixel of a hyperspectral image is a vector containing all spectral information provided through different spectral channels at different wavelengths. The dimensions of vector depend on number of spectral channels obtained through sensor and, in the case of hyperspectral data often number of spectral channels reaches hundreds.

In fact, feature selection or feature extraction techniques are often displayed as pre-processing on hyperspectral data analysis [1]. Such processing is often carried out in multispectral imagery to improve class to separate or to eliminate certain types of noise. One technique for reducing data dimensions is to use principal component analysis (PCA) [2], which estimates eigenvalues data to display a projection into a new feature space where maximum variance of data is maintained. PCA only involves statistics up to the second-order; this is probably not effective when used for hyperspectral data [3]. To overcome problems associated with dimensional reduction of data using PCA, several algorithms has been proposed such as kernel principal component analysis (KPCA) [4]. Other data dimension reduction techniques that have been widely used in recent years are Independent Component Analysis (ICA) [5]. This technique projects data into a new space where components are not independent (independent), estimated from a statistical point of view. Multivariate data with lower dimensions and independent features in discriminant independent component analysis (DICA) were obtained using entropy maximization [6] [7].

Segmentation is an important part of image processing that attempts to identify pixel intensity corresponding to a predefined class. Thresholding is the simplest segmentation method because it works by taking a certain threshold value (T). Pixels with higher intensity values than T are labeled as first-class while the rest are labeled second class. When an image is segmented into two classes, this is called the bi-level thresholding (BT) and requires only one T value. On the other hand, when pixels are separated into more than two classes, this task is called multilevel thresholding (MT) and requires more than one T value [8].

Image segmentation plays an important role in the field of remote sensing hyperspectral analysis. As an example to improve the classification results, the integration of the segmentation process and classification in the process of hyperspectral image processing has been carried out [9]. In certain cases to decide whether a pixel is inserted into a particular class simultaneously, it will be based on the feature vector and some other additional information obtained from the segmentation step. An accuracy of mind segmentation is needed in this approach to be more effective. A number of multispectral and hyperspectral image segmentation methods have been introduced in some literature. Some of these methods are mostly based on region-merging techniques, where the nearest image segments are combined with each other based on their homogeneity. Segmentation can be classified into four types: methods based on histogram thresholding, methods based on texture analysis, methods based on clustering and methods based on split-merging regions. The thresholding technique can be divided into two different groups, first, the optimal thresholding method [10] [11], where threshold optimal search makes the threshold classes on the histogram achieve the desired characteristics. Usually the threshold is chosen by optimizing the objective function (fitness function). Second, the thresholding method is based on the characteristics, detecting the threshold limit through some characteristic measurements of the histogram. Although this method is fast so that it is suitable for multilevel thresholding, the threshold number for this approach is difficult to determine and requires further special things.

Several algorithms have been proposed in several literatures relating to optimal thresholding. Some of these studies are related to bi-level thresholding, while others relate to multilevel thresholding problems. Bi-level thresholding reduces an optimization problem to determine the threshold T that maximizes the between class-variance and minimizes class variance (within-class variance) [12]. At bi-level thresholding, the problem is solved through the T value search. This approach is easy to implement but has drawbacks that are generally the old computation. The search for n-1 optimal threshold involves fitness evaluation $n(L - n + 1)^{n-1}$ threshold combinations. This means that this method is not suitable from the point of view of execution time calculation. The task of determining n

- 1 optimal threshold on n-level imagery can be formulated as a multidimensional problem. Bio-inspired algorithms have been used in situations where conventional optimization techniques cannot find satisfactory solutions or the time needed to solve them too much, for example, when a function that is optimized is discontinuous and cannot be derived differentiated and / or presents too many nonlinear parameters.

One of the known bio-inspired algorithms is particle swarm optimization (PSO) [13]. The PSO consists of a number of particles that collectively move in the search space (for example the pixels of the image) to find global optimality (e.g. maximizing inter-class variance from the distribution of intensity levels in a given image). However, a common problem with PSO and similar optimization algorithms are that the algorithm can get stuck at a locally optimal point, and the algorithm might work with some existing problems but it might also fail on other problems. Another swarm optimization approach is the Darwinian Particle Swarm Optimization (DPSO) introduced by Tillet [14] to overcome the vulnerability in local optima.

This research proposed a method for discriminant analysis based on the extraction of independent features obtained by entropy maximization namely discriminant independent component analysis (DICA). Fisher's criteria and the sum of marginal negentropy of extracted independent features are maximized simultaneously. While the segmentation method used is swarm optimization approach namely particle swarm optimization (PSO) and Darwinian particle swarm optimization (DPSO).

2. Multilevel Thresholding for Optimization Problem

Multilevel segmentation techniques provide an efficient way to image analysis. The problem faced is usually the automatic selection of an optimal n-level threshold. A threshold can be found using Otsu's method (Otsu, 1979) [12], which can maximize variance between classes. Pixels of an image is represented in the L gray level. Number of pixels at level i as n_i , the total number of pixels denoted through $N = n_1 + n_2 + + \cdots + n_L$. The gray level histogram is normalized and is considered probability distribution equation is written in the equation (1).

$$p_i = \frac{n_i}{N}$$
$$p_i \geq 0 \ , \ \textstyle\sum_{i=1}^{L} p_i = 1 \tag{1}$$

Images are divided into classes K_0 and K_1 by a threshold at level k in bi-level thresholding problem. K_0 is pixels with levels $[1,2,..,L]$ and K_1 is pixels with levels $[k+1,..,L]$. Probabilities of class occurrences and mean class levels can be written as equations (2) and (3) respectively. While the average total level of the image μ_T can be written as equation (4).

$$\omega_0 = \textstyle\sum_{i=1}^{k} p_i \quad \omega_1 = \textstyle\sum_{i=k+1}^{L} p_i \tag{2}$$
$$\mu_0 = \textstyle\sum_{i=1}^{k} i p_i / \omega_0 \quad \omega_1 = \textstyle\sum_{i=k+1}^{L} i p_i / \omega_1 \tag{3}$$
$$\mu_T = \textstyle\sum_{i=1}^{L} i p_i \tag{4}$$

Two relations can be written as equation (5) for several choices of k.

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T, \ \omega_0 + \omega_1 = 1 \tag{5}$$

Furthermore, the objective function of the Otsu method can be defined as equation (6).

$$\text{Maximizes } \sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 \tag{6}$$

Otsu's method can be extended to multilevel thresholding problems. It is assumed that there is a threshold, which divides the image into m + 1 class : K_0 for $[1,2,..,k]$, K_1 for $[k_1 + 1, ..., k_2]$ and K_m for $[k_m + 1, ..., L$. Then from the Otsu's function, it can be written as (7).

$$\text{Maximizes } \sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2 + \cdots + \omega_m (\mu_m - \mu_T)^2 \tag{7}$$

One method used to perform segmentation processes is to use the thresholding method. Threshold image thresholding T partitioning an image I into parts of a particular region based on the value of T itself. Suppose there is an L level of intensity in each RGB component of an image, and these levels are within the range of values (0,1,2, ... L-1), then it can be defined in equation (8).

$$P_i^K = \frac{h_i^K}{N} \ , \ \textstyle\sum_{i=1}^{N} P_i^K, \ K = \{R, G, B\} \tag{8}$$

Where is i intensity, $0 \leq i \leq L - 1$. K is image components, $K = \{R, G, B\}$. N is the total numbers pixels in image. h_i^K is number of pixels for intensity level i correspondence with component K. While total average (mean) is mean that combined from every image component as an equation (9).

$$\mu_T^K = \sum_{i=1}^{L} i \, P_i^K, \ K = \{R, G, B\} \tag{9}$$

from 2-*level thresholding* can be extended to find n-level *thresholding* where is *n-1* level *threshold* $m_j^k, j = 1, 2, .., n - 1$. The image can be written as an equation function (10).

$$F^k(x, y) = \begin{cases} 0, f^k(x, y) \leq t_1^k \\ \frac{1}{2}(t_1^k + t_2^k), \ t_1^k < f^k(x, y) \leq t_2^k \\ \vdots \qquad\qquad \vdots \\ \frac{1}{2}(t_{n-2}^k + t_{n-1}^k), \ t_{n-2}^k < f^k(x, y) \leq t_{n-1}^k \\ L, \qquad f^k(x, y) > t_{n-1}^k \end{cases} \tag{10}$$

Where si x and y are widht (W) and length (H) image pixels write as $f^k(x, y)$ dwith intensity level Lin each RGB component. In this situation, pixels of the image will be divided into n classes $D_1^k, ..., D_n^k$ which represents multiple objects or certain features on a particular object. The method to get the optimal threshold is by maximizing the variance between classes through equation (11).

$$\sigma_B^{k^2} = \sum_{j=1}^{n} w_j^k (\mu_j^k - \mu_T^k)^2, \ k = \{R, G, B\} \tag{11}$$

Where is j represented specific class so that w_j^k and μ_j^k are the probability of occurrence (event) and mean from class j. Probability of occurrence w_j^k and classes $D_1^k, ..., D_n^k$ given as equations (12).

$$w_j^k = \begin{cases} \sum_{\substack{i=1 \\ C=\{R,G,B\}}}^{t_j^k} P_i^c, \quad j = 1 \\ \sum_{\substack{=t_{j-1}^k+1 \\ C=\{R,G,B\}}}^{t_j^k} P_i^c, \quad 1 < j < n \\ \sum_{i=t_{j-1}^k+1}^{L} P_i^k, \quad j = n \end{cases} \tag{12}$$

While the mean of each class μ_j^k calculated using equation (13).

$$\mu_j^k = \begin{cases} \sum_{\substack{i=1 \\ C=\{R,G,B\}}}^{t_j^k} \frac{P_i^k}{w_j^k q}, \quad j = 1 \\ \sum_{\substack{=t_{j-1}^k+1 \\ C=\{R,G,B\}}}^{t_j^k} \frac{P_i^k}{w_i^k}, \quad 1 < j < n \\ \sum_{i=t_{j-1}^k+1}^{L} \frac{P_i^k}{w_i^k}, \quad j = n \end{cases} \tag{13}$$

This n-level thresholding optimization problem is reduced to a search optimization problem for the threshold t_j^k which maximizes 3 objective functions (i.e. fitness function) of each RGB component written in the form of equation (14).

$$\varphi^k = \max_{\substack{1 < t_j^k < \cdots < t_{n-1}^k < L \\ C=\{R,G,B\}}} \sigma_B^{k^2}(t_j^k) \tag{14}$$

3. Swarm Optimization Approach

Particle swarm optimization (PSO) technique first introduced by Kennedy and Eberhart (1995), is a stochastic optimization technique similar to the behavior of a flock of birds or sociological behavior of group humans. The basic idea of PSO is to involve a scenario in which a flock of birds searches for food sources in an area. All birds do not know exactly where the food, but with each iteration they will know how far the food will be found. The best strategy will be followed by birds that are close to food and also from the best previous position achieved. PSO is built with the concept of optimization through a swarm particle. The original PSO algorithm is written in the form of velocity updated equations and position updates as shown in equation (15) and equation (16) sequentially.

$$v_{ij}^{t+1} = v_{ij}^t + c_1 r_{1j}^t * \left(P_{best,i}^t - x_{ij}^t\right) + c_1 r_{2j}^t * \left(G_{best} - x_{ij}^t\right) \tag{15}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \tag{16}$$

The algorithm is controlled through individual experience (pbest), and overall experience (gbest) and the current movement of the particles to determine their next position in search space. Experiences are accelerated through two acceleration constant factors c_1 and c_2, two random numbers r_1, r_2 which is generated with a range value between 0 and 1. N-size population and the D dimension are denoted as $x = [x_1, x_2, .., x_N]^T$. Every particle $x_i(i = 1,2,3, ..., N)$ given as $x_i = [x_{i1}, x_{i2}, ..., x_{iD}]$, initial velocity was written as $v = [v_1, v_2, ..., v_N]^T$. Then, next velocity $x_i(i = 1,2,3, ..., N)$ given as $v_i = [v_{i1}, v_{i2}, ..., v_{iD}]$. Whereas index i varies from 1 to N and index j varies from 1 to D.

The basic modification of PSO algorithm which is usually carried out includes how to increase convergence speed, control exploration, and exploitation trade-offs, overcome convergence or premature stagnation problems, clamping techniques, problem of boundary value, initial and final conditions. In PSO, particle speed is very important. At each step of the PSO process, all particles are processed through speed adjustments for each particle movement in each dimension of the search space. There are two characteristics in PSO namely exploration and exploitation. Exploration is the ability to explore different areas of the search space to get optimal good, while exploitation is the ability to focus search in the search area to improve the expected solution. When the speed increases, the position of the particles will be updated quickly. And as a result, particles will leave the search space boundaries and there is a possibility that they will drift away from the search space. Therefore, to control this difference, the particle speed will be reduced to remain within the search space boundaries. Several techniques were developed to increase the speed of convergence, in order to balance the exploration and exploitation trade-offs and find a good quality of completion for PSO, namely the speed of flanking techniques (velocity clamping), inertial weight (w), constriction coefficient (s). A common problem with optimization algorithms is being trapped into optimal locality. Certain algorithms can work well on one problem but may fail in another. If an algorithm can be designed to adapt to the fitness function, adjust itself to the fitness landscape, a stronger algorithm with wider application, without the need for special engineering problems will result.

In specific implementation of PSO, a group of test settlers was utilized. For applying natural selection with a single flock, an algorithm must detect when stagnation has occurred. In finding a better natural selection model using the PSO algorithm, Darwinian Particle Swarm Optimization (DPSO) was introduced by Tillet et al [14]. To analyze the general state of each flock, the suitability of all particles is evaluated and the environment and the best individual position of each particle are updated. If a new global solution is found, new particles will appear. The particles are removed if the flock fails to find a more suitable state in the specified number of steps (Algorithm 1). Some simple rules are followed to remove flocks, remove particles, and spawn new flocks and new particles: i) when the herd population is below the minimum limit, the herd is removed, and ii) the worst performing particles in the herd are removed when the number of steps is the maximum limit (SC_c^{max}) without increasing the fitness function achieved. After deletion of particles, instead of being set to zero, the counter is reset to the value close to the threshold number, according to the equation (17).

$$SC_c(N_{kill}) = SC_c^{max}\left[1 - \frac{1}{N_{kill}+1}\right] \tag{17}$$

With N_{kill} is number of particles removed from swarm during periods where there is no improvement in fitness. To spawn a new flock, a swarm of particles must never be removed and the maximum number of flocks must not be exceeded. However, the new flock is only made with probability p = f / NS, where f is a random number in [0, 1] and NS is the number of flocks. The algorithm of DPSO is shown in Algorithm 1.

Algorithm 1: DPSO [14]

Main Program Loop (1 Step)	*Evolve Swarm Algorithm*
	For each particle in the swarm
For each swarm in the collection	*Update Particles' Fitness*
Evolve the swarm (Evolve Swarm Algorithm: Right)	*Update Particles' Best*
	Move Particle
For each swarm in the collection	*If swarm gets better*
Allow the swarm to spawn	*Reward swarm: spawn particle: extend swarm life*
Delete 'failed' swarms	*If the swarm has not improved*
	Punish swarm: possibly delete particle: reduce swarm life

4. Experiment Results

Data reduction and image segmentation based on the swarm optimization approach in this paper was implemented using Matlab R.2016a in a computer Intel® Core ™ i5-4210U CPU@1.70 GHz. The data used in this study are AVIRIS Indian Pines hyperspectral image data obtained from AVIRIS sensors above the Northwestern Indiana region in 1992. Images size is 145 x 145 pixels. The number of image bands is 220 spectral bands (some bands contain noise and water absorption) [15]. Figure 1 shows the image of Aviris Indian Pines and its histogram.
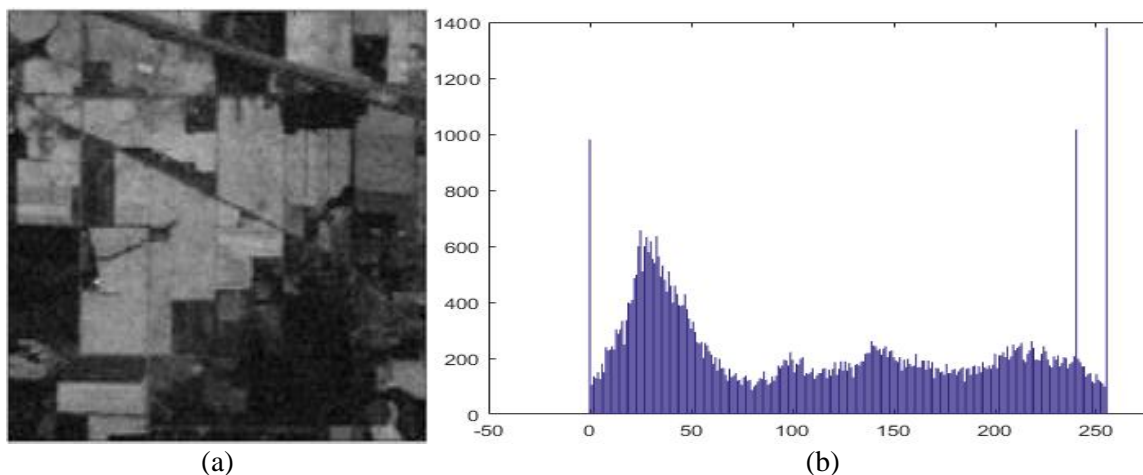

(a)   (b)

Figure 1. Aviris Indian Pines data set: (a) Aviris Indian Pines 3 band (b) Image Histogram

The data dimension reduction method used in this paper is discriminant independent component analysis (DICA). The number of independent components (IC) is the first, second and third independent components. Aviris Indian Pines Image which consists of 3 independent components is then subject to a segmentation process. Figure 2 shows the original image as well as the first, second and third independent components (IC's) of Aviris Indian Pines image.
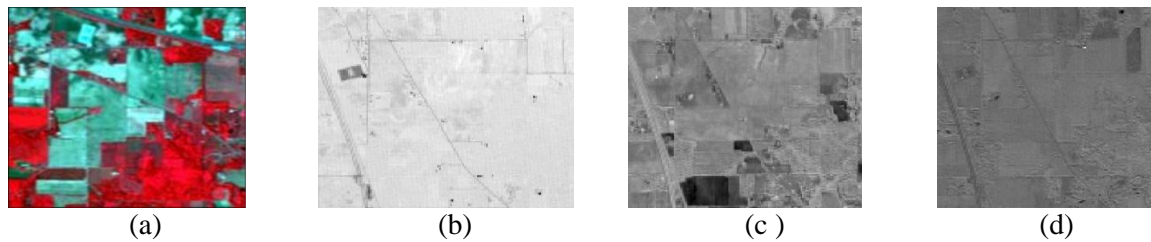
| (a) | (b) | (c) | (d) |

Figure 2. Aviris Indian Pines data set: (a) RGB Image (b) - (d) the first, second and third independent components

The image segmentation technique used in this research is based on swarm optimization approach. Two swarm optimization approaches are used namely particle swarm optimization (PSO) and Darwinian Particle Swarm Optimization (DPSO). While the quality of segmentation images produced from these two techniques is measured by using a measure of segmentation quality through the PSNR, SSIM and PSE values produced [16] [17]. Initializing PSO and DPSO parameters is shown in Table 1.

Table 1. Parameter Initialization of DPSO and PSO

Parameter	DPSO	PSO
Iteration numbers	200	200
Population number	100	100
C1	0.5	0.5
C2	0.5	0.5
Vmax	4	4
Vmin	-4	-4
Xmax	255	255
Xmin	0	0
Min Population	-	15
Max Population	-	60
Numbers of Swarm	-	4
Min Swarm	-	2
Max Swarm	-	6
Stagnancy	-	8

CPU processing time of the data set used in this paper was tested on the PSO and DPSO algorithms respectively for thresholding levels of 8, 10 and 12. The average CPU processing time was obtained from 20 runs. The results are shown in Table 2.

Table 2. Average CPU Processing Time

Level	DPSO	PSO
10	5.5383	10.7495
11	5.8246	11.7717
12	5.9267	14.1271

Table 2 shows the optimal threshold value, fitness, and the quality of segmentation results from the swarm optimization approach using Particle Swarm Optimization (PSO) and Darwinian Particle Swarm Optimization (DPSO). The quality parameters of the segmentation results used are peak signal to noise ratio (PSNR), mean square error (MSE) and structural similarity index of SSIM). Fitness value (between-class variance) at each level of image segmentation using multilevel thresholding is sought at 10.11 and 12. Levels of optimal threshold and fitness showed in Table3 are calculated from Aviris Indian Pines images which have been reduced to 3 bands. Table III shows that DPSO is better

than PSO in terms of fitness value, PSNR, SSIM, and MSE. The fitness value generated by using the DPSO method is better because PSO may be trapped at optimal locality while the DPSO uses natural selection to avoid stagnation. From this it can be seen that DPSO can find the optimal threshold value with a faster time compared to PSO. From this it can be concluded that the DPSO method is more recommended to be used as an image segmentation method, especially in high-dimensional images such as hyperspectral images.

Table 3. The threshold value, fitness, PSNR, SSIM and MSE Segmentation of Aviris Indian Pines

Methods (level)	Threshold Value	Fitness	PSNR	SSIM	MSE
DPSO(10)	24,49,77,103,134,165,195,219,239	7260.8	29.7830	0.9338	203.82
DPSO(11)	19,40,63,87,112,139,167,193,218,237	7270.2	30.6581	0.9392	187.44
DPSO(12)	18,38,61,84,108,131,156,180,204,225,242	7277.1	31.4600	0.9391	180.99
PSO(10)	18,38,62,87,113,143,175,202,232	7258.6	29.6198	0.8828	354.19
PSO(11)	16,36,60,84,109,136,165,194,217,237	7269.9	30.5942	0.9126	251.62
PSO(12)	14,30,50,73,95,121,148,173,198,220,229	7276.1	31.3749	0.9287	218.98

The results of Aviris Indian Pines image segmentation using the Darwinian Particle Swarm Optimization (DPSO) and Particle Swarm Optimization (PSO) approaches are shown in Table IV. The segmentation results along with the histogram are shown at the threshold level T = 10.11,12.

Table 4. Results after Application of DPSO and PSO Approach to Aviris Indian Pines

Method	T=10	T=11	T=12
DPSO			
PSO			
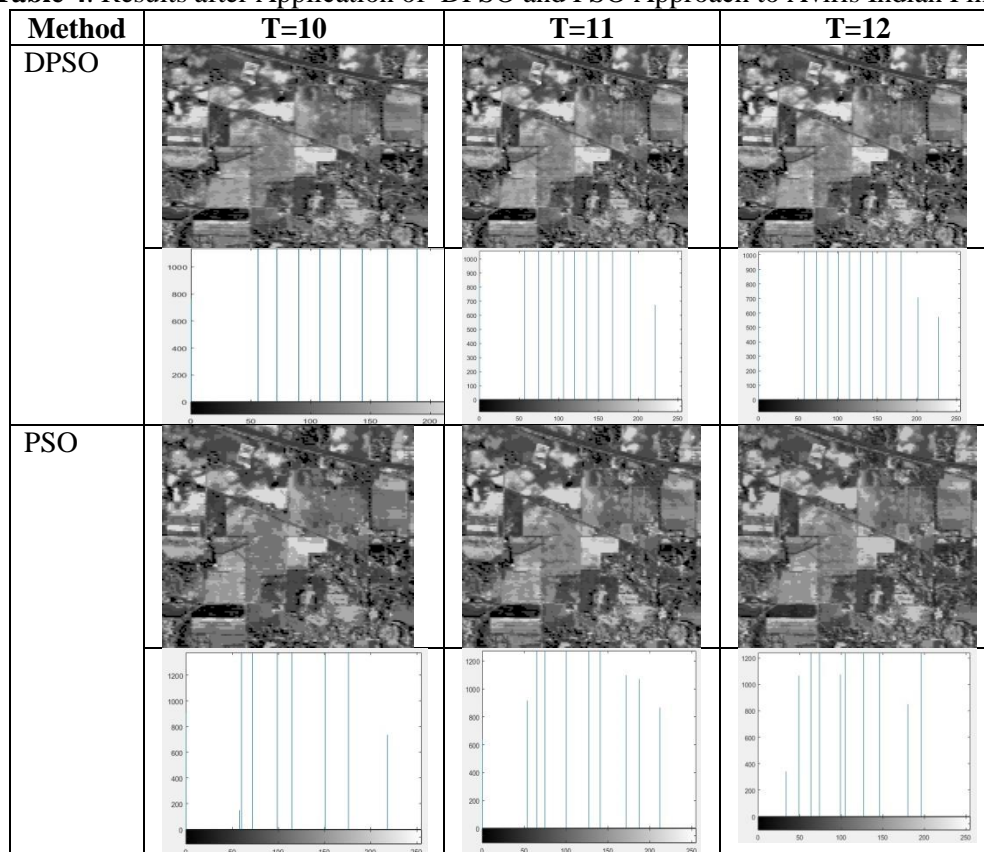
5. Conclusion

In this study an approach has been proposed for the segmentation of hyperspectral images using data dimension reduction Discriminant Independent Component Analysis (DICA). The method used to overcome the Otsu optimization problem is done by finding the threshold value using multilevel

thresholding based on Darwinian particle swarm optimization (DPSO). The experimental results show that the DPSO method is better compared to the Particle Swarm Optimization (PSO) method in the context of CPU processing time and the fitness value generated.

Acknowledgment

References

[1]   Landgrebe, D.A. 2003 Hyperspectral image data analysis as a high dimensional signal processing problems. *IEEE Signal Processing Magazine*, 19:17–28.

[2]   Jolliffe, I. T. 1986. Principal Component Analysis and Factor Analysis. In *Principal component analysis* (pp. 115-128). Springer New York.

[3]   Richards, J. A.(2005 Analysis of remotely sensed data: The formative decades and the future. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 422-432.

[4]   Fauvel, M., Chanussot, J., and Benediktsson, J. A. 2012 A spatial-spectral kernel-based approach for the classification of remote-sensing images. *Pattern Recognition*, 45(1), 381-392.

[5]   Hyvärinen, A., Karhunen, J., & Oja, E. 2001 *Independent component analysis* (Vol. 46). John Wiley & Sons.

[6]   Dhir, Chandra Shekhar, Jahyung Lee, and Soo-Young Lee.2012 Extraction of independent discriminant features for data with asymmetric distribution, *Knowledge and information systems* 30.2. 359-375

[7]   Murinto and Nur Rochmah Dyah PA. 2017. Dimensionality Reduction using Hybrid Support Vector Machine and Discriminant Independent Component Analysis for Hyperspectral Image. International Journal of Advanced Computer Science and Applications(ijacsa), 8(11).

[8]   Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Perez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, *2013*.

[9]   Ghamisi, P., M.S. Couceiro, N.M.F. Ferreira and L. Kumar. 2012 Use of Darwinian particle swarm optimization technique for the segmentation of Remote Sensing images. *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Jul. 22-27, IEEE Xplore Press, Munich.

[10] Pun, T. (1980). A new method for grey-level picture thresholding using the entropy of the histogram. *Signal processing*, *2*(3), 223-237.

[11] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.

[12] Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, *13*(6), 3066-3091.

[13] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39-43). IEEE.

[14] Tillett, J., Rao, T., Sahin, F., & Rao, R. (2005). Darwinian particle swarm optimization.

[15] AVIRIS Dataset. (2002).[cited 2016 june 1].Available from ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C .tif.zip

[16] A.M. Eskicioglu, P.S. Fisher, "Image Quality Measures and Their Performance", *IEEE Transactions on Communication*, vol. 43, no. 12, pp. 2959-2965, December 1995.

[17] Murinto, Winiarti, S., Ismi, D. P., & Prahara, A. 2017 Image enhancement using piecewise linear contrast stretch methods based on unsharp masking algorithms for leather image processing. In *Science in Information Technology (ICSITech), 2017 3rd International Conference on* (pp. 669-673). IEEE.