Fuzzy C Means Clustering of Hyperspectral Data for Mineral Identification

Professor Anigbogu, Sylvanus Okwudili and Olanloye, Dauda Odunayo.

***Abstract*— Advanced precision engineering has enabled the development of hyperspectral sensors. These sensors can be used in collection of highly detailed spectra information from required target using non-intrusive, zero-impact and even remote sensing methods. A particular usage is the use of hyperspectral sensors on satellites to obtain information about earth surfaces and minerals beneath the surface. However, the data so-obtained consist of high dimension (hundreds of bands) spectrum that needs to be processed to obtain classification data and thus identify the existing minerals. Thus hyperspectral data dimension reduction has become an active research topic. In this work, we used a new Metaheuristic method namely halfing method, to generate a characteristic map of a given spectrum. The method literally ratios one half of the spectrum against the other, thus attenuating defects due to measurement conditions. We then used fuzzy c-means clustering to obtain cluster centers from the characteristics map. The obtained cluster center is used to classify and thus identify the given spectra. Any variation or perturbation in the given spectrum, appears as fluctuation in the coordinates of the obtained spectral center. The method reduces the dimension of the hyperspectral data to manageable dimension without loss in information or precision. R programming Language is used for the implementation of the system.**

***Index Terms*— Hyperspectroscopy, Metaheuristic, Fuzzy c-means clustering, characterization map**

I. Introduction

Hyperspectral images can be defined as images whose pixels contain a fine sampling of the electromagnetic spectra. Therefore each pixel is a high dimensional vector, whose components are the measured reflectance values inside a fine wavelength band of the spectra. Most of the hyperspectral sensors cover the visible light spectrum and the near infrared (NIR) spectrum. Fig 1 shows the structure of a hyperspectral image from a computational point of view.

It consists of a 3D matrix, whose third dimension corresponds to the reflectance spectra sampled at the pixel. The first two dimensions correspond to the spatial coordinates in the image plane. Another view of the data in a hyperspectral image is given in Fig 2. Fig 3, illustrate how the hyperspectral image is captured in a remote sensing setting.

A high altitude device, either an airplane or a satellite, goes over the land picking the images. On board, sensors often capture one line of the image, so that the motion of the device gives the second spatial dimension. Fig3. shows that different land covers produce different spectra in the corresponding image pixels. This additional spectral information has the promise of allowing image automation and allows detection of materials to be highly efficient and robust without resorting to spatial processing.

The unique capabilities of hyperspectral images in expressing the properties of earth surface guide the researchers towards developing method that as much as possible, decrease the need of human interference in processing data. A fundamental step in the processing of hyperspectral images is the segmentation through a clustering process [3].

In fact, processing of hyperspectral images will produce a lot of useful information on the earth surface. A lot of research works with different methods or approaches are currently going on in this important area. Data clustering is one of the important techniques that have been introduced in the processing of hyperspectral images.

Clustering is a method of partitioning set of data into subset (clusters) in a way that the elements in each cluster are more similar to each other than the elements in the other cluster [5]. Venkateswara and Reddy [9] defined clustering as a process which partition a given data set into homogeneous group based on given features such that similar objects are kept in a group whereas dissimilar objects are kept in different group. This means that similar objects will belong to a particular cluster and dissimilar objects will belong to another cluster

Subhagata, Dilip and Sanjib [8] noted two important benefit of clustering to include:

(i) Easy tackling of noisy data and outlier

(ii) Ability to deal with data having various variables, such as continuous variables that require standardized data, binary variable, nominal variable e.t.c.

According to Venkateswara and Reddy [9], the advent of many data clustering algorithm in the recent few years and its intensive use in wide variety of application, including image processing, computational biology, mobile communication, medicine and economics has lead to the popularity of its algorithm. It is a very popular method due to its efficacy, simplicity and computational efficiency [10].

In clustering, if the cluster center has a well defined boundary, it is referred to as crisp clustering otherwise it is referred to as fuzzy clustering.

Fuzzy c means is a typical example of fuzzy clustering that has been applied to solve different problems.

Sikka, Sinha, singh and Mishra [7] were able to modify Fuzzy C Means(FCM) algorithm and produced a new algorithm which was capable of identifying the tissue and the tumor area in the brain. This is a good application of FCM in Biological computation.

Krinidis and Chatzis [6] developed an algorithm called fuzzy local Information C Means (FLICM) which is capable of removing the hindrances of the FCM algorithm.

Belhassen and Zaidi [1] developed a novel modified FCM algorithm which solve the problem encountered by FCM with noisy and low resolution on logical data .It was tested and

Manuscript received January 22, 2015.

Professor Anigbogu, Sylvanus Okwudili, Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria.+2348066184124.

Olanloye, Dauda Odunayo. Department of Computer Science, Emmanuel Alayande College of Education, Oyo. Nigeria. +2348038196016.

found to be less prone to error compared to conventional FCM.

Fu and Medico [4] developed a clustering algorithm described as Fuzzy Clustering by Local Approximation of Membership (FLAME). It is meant for DNA Microarray analysis. It was able to define neighbor –hood point and identify Cluster supporting objects.

Going by the literature, it was observed that enough efforts has not been made in the area of applying FCM in the area of mineral identification.

This research work therefore focused on fuzzy clustering. Specifically, an attempt is being made to use fuzzy c means to process hyperspectral data for the purpose of mineral identification. Different cluster centers are generated and the best ones are mapped into 3D for visualization with KSOM.

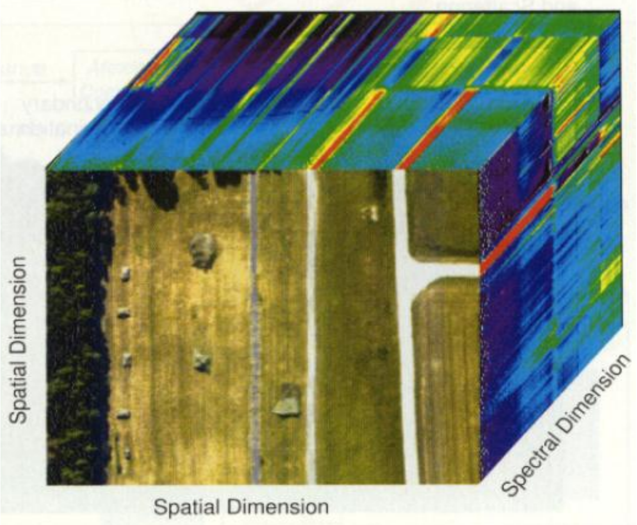


Fig 1: Structure of a hyperspectral image

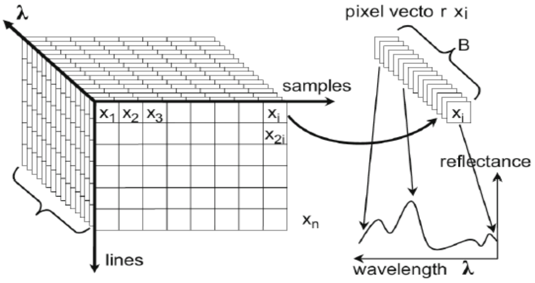


Fig 2: ***Imaging spectroscopy data cube illustrating the –Dspatial and spectral character of the data***

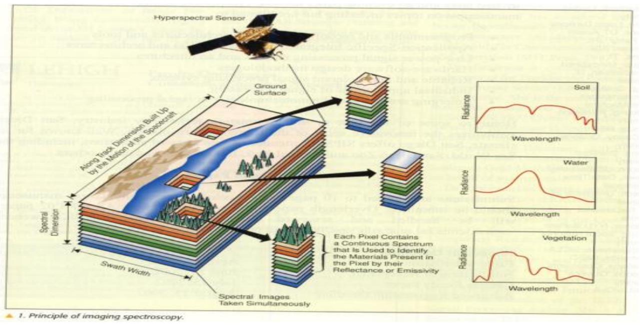

Hyperspectral Sensor
Ground Surface
Along Track Dimension Built Up by the Motion of the Spacecraft
Spectral Dimension
Swath Width
Spectral Images Taken Simultaneously
Each Pixel Contains a Continuous Spectrum that Is Used to Identify the Materials Present in the Pixel by their Reflectance or Emissivity
Soil
Radiance
Wavelength
Water
Radiance
Wavelength
Vegetation
Radiance
Wavelength
1. Principle of imaging spectroscopy.

Fig 3: An illustration of the remote sensing hyperspectral image capture

II. METHODOLOGY

A. Input module:

The data may be read in bulk or streams with appropriate staging, extraction, transform and load functions. The data should be read into an internal matrix data type for further processing. Also, the module should enable processing functions such as normalization, selection of bands, examination of spectrum for particular pixel and visualization of the data cube in general.

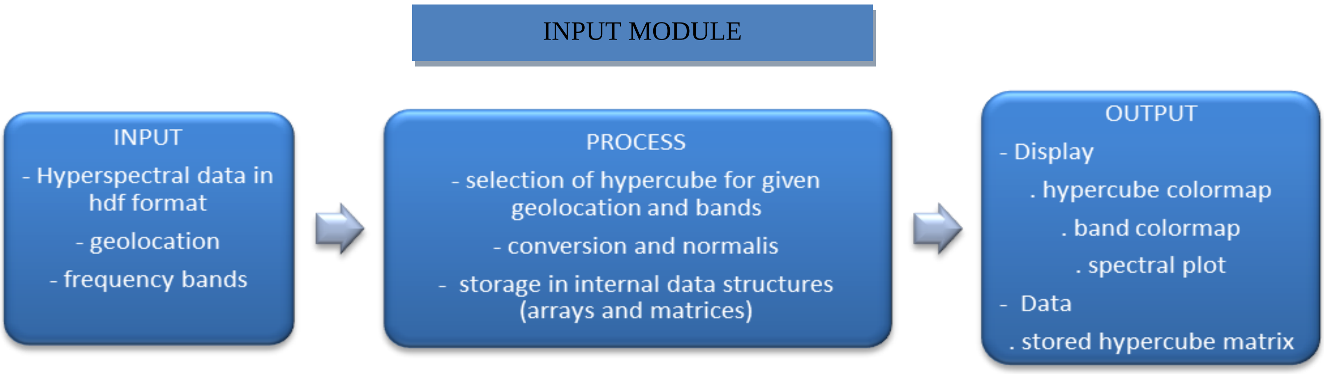

INPUT MODULE
INPUT
- Hyperspectral data in hdf format
- geolocation
- frequency bands
PROCESS
- selection of hypercube for given geolocation and bands
- conversion and normalis
- storage in internal data structures (arrays and matrices)
OUTPUT
- Display
. hypercube colormap
. band colormap
. spectral plot
- Data
. stored hypercube matrix

Fig 4: ***Module design***

Algorithms

Data normalization

We normalize the data set to keep the data range between 0 and 1 to ease subsequent computation.

Step1: Find the maximum in the selected data cube matrix: max (D_{ijb})

Step2: Divide each data point with the max: Dnorm=D/max (D_{ijb})

B. Characterization module

Module requirements

Essentially this module is responsible for conversion of the spectrum for a given pixel, and as such all pixels, in the data to a form that can be used to characterize the pixel. The output is a form of map plot, ‘character’ image of the spectrum for the pixel. It should provide a visual plot of the map and store the characterization map in a matrix data type for further processing.

Modular design for characterization

Module design

The module selects the spectrum data for particular pixels and converts it into characterization map using halfing Metaheuristic approach

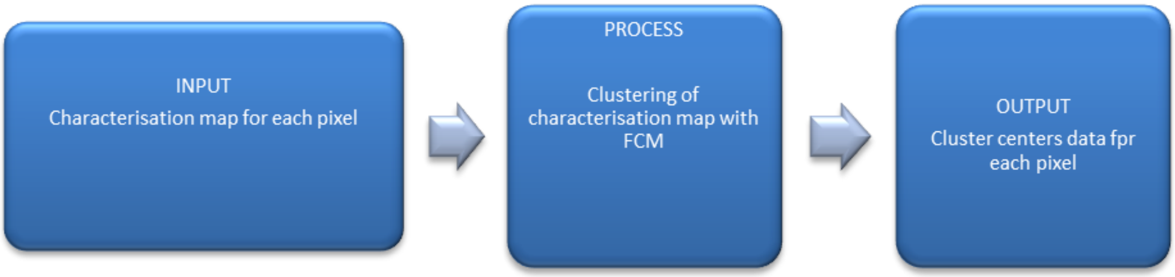

INPUT
Characterisation map for each pixel
PROCESS
Clustering of characterisation map with FCM
OUTPUT
Cluster centers data fpr each pixel

Fig 6: Clustering module design

C. *System Design*

Module design

Fuzzy c-means clustering algorithm was used to generate cluster centers from the characterization map for each pixel. The data for n-clusters consists of n pairs of coordinate data, i.e. 2n features for that particular spectrum.

Fuzzy c-means (FCM) is a data clustering technique where each data point belongs to a cluster to some degree that is specified by a membership grade(Fig7) . This technique was originally introduced by Jim Bezdek [2] as an improvement on earlier clustering methods. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters (Fig 8a and 8b)

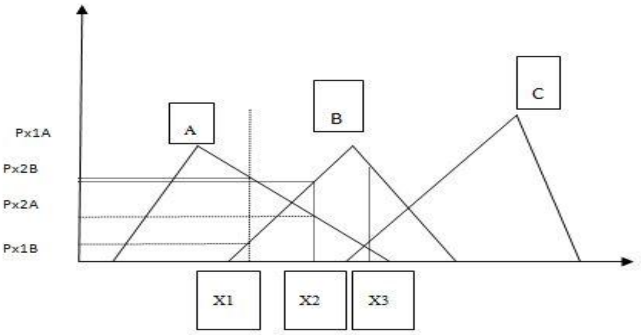


Fig 7: Membership function

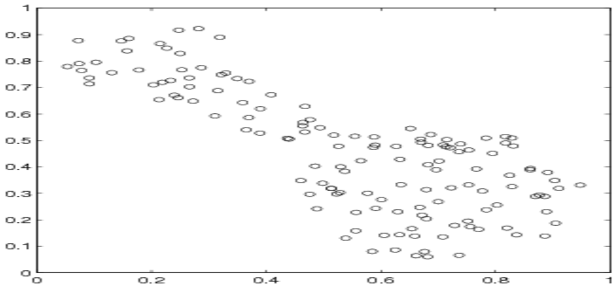


Fig 8a: Plots of data points to be clustered i.e.
2-D plots of data points to be clustered

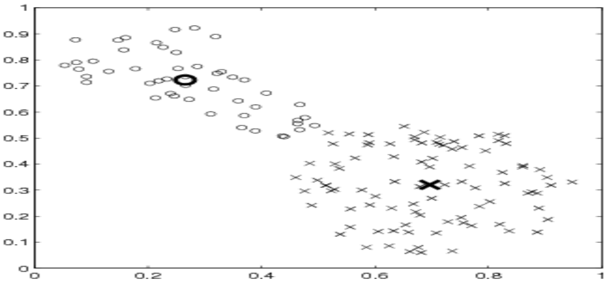


Fig 8b: Clustered data i.e. Clustered data, large characters indicate cluster centers

The FUZZY C-means Algorithm

FCM clustering algorithm was first proposed by Bezdek in 1973. For a given data set $D = \{X_j\}_{j=1}^{N}$ with $X_j \epsilon R^d$,the FCM clustering algorithm minimizes the objective function below:

$$f(D; U, C) = \sum_{i=1}^{n_i} \sum_{j=1}^{N} (\mu_{i,j})^m \|X_j - C_i\|_A^2 \qquad \dots\dots\dots (1)$$

where $U = (\mu_{i,j})_{\substack{1 \le i \le n_c \\ 1 \le j \le N}}$ is the fuzzy partition matrix with $\mu_{i,j}$ indicating the membership of x_j belonging to the ith cluster, while$C = (c_1, c_2, \dots, c_{n_c})^T$ is the matrix of cluster centers and its elements $c_i \in R^d$ $(i = 1, 2, \dots, n_c)$ are the centers of the n_c clusters. Moreover, m>1 in equation……1 above.
formula is the weighting exponent, and the distance $\|X_j - c_i\|_A^2$ is defined as

$$D_{ijA}^2 = \|X_j - c_i\|_A^2 = (X_j - c_i)^T A (X_j - c_i) \qquad \dots\dots (2)$$

where A = I with I being the identity matrix in this paper. The clustering procedure of the FCM algorithm can be summarized in Algorithm FCM. Once the FCM completes, the cluster label index k for the sample x_j in the data set D can be given by

$$k = \arg \max_{i=1}^{n_c} \mu_{i,j} \qquad \dots\dots (3)$$

where index j = 1, 2, . . . ,N.

Algorithm FCM. The fuzzy c-means clustering algorithm

```
    Input:      Data set $D = \{X_j\}_{j=1}^{N}$
    Initialization:    Number of clusters $n_c$ $(1 \le n_c \le N)$,
     weighting exponent $m$,
termination tolerance $\varepsilon$,
fuzzy partition matrix $U = (\mu_{i,j})_{n_c \times N}$ $(1 \le n_c \le N)$
Repeat
   for t =1, 2, . . . , do
Step 1: Compute the cluster centers
$c_i^{(t)} = \frac{\sum_{j=1}^{N} (\mu_{i,j}^{(t-1)})^m X_j}{\sum_{j=1}^{N} (\mu_{i,j}^{(t-1)})^m}$
Step 2: Calculate the distances $D_{ijA}^2$ as
$D_{ijA}^2 = \|X_j - c_i\|_A^2 = (X_j - c_i)^T A (X_j - c_i)$
$1 \le i \le n_c, 1 \le j \le N$
Step 3: Update the fuzzy partition matrix:
$\mu_{i,j}^{(t)} = \frac{1}{\sum_{k=1}^{n_c} (D_{ijA} / D_{kjA})^{2/(m-1)}}$
   end for
  until $\|U^{(t)} - U^{(t-1)}\| < \varepsilon$
```

To determine an appropriate number of clusters n_c for a given data set, a cluster validity function (PBMF) needs to be selected and this was proposed by Pakhira et al. It achieved more satisfying results when compared with the extended Xie–Beni index. The PBMF-index is used for choosing the optimal number of clusters in the paper, which is defined as follows:

$$PBMF(n_c) = \left(\frac{1}{n_c} \times \frac{E_1}{J_m} \times D_{n_c}\right)^2 \qquad \text{...... (4)}$$

where n_c is the number of clusters, $E_1 = \sum_{j=1}^{N} \mu_{i,j} \|X_j - w_1\|$ w_1 is the center of the unique cluster for the whole data set, and $D_{n_c} = max_{k,l=1, k\neq l}^{n_c} \|c_k - c_l\|$. $J_m = \sum_{j=1}^{N} \sum_{i=1}^{n_c} \mu_{i,j}^m \|X_j - C_i\|$ is the optimal number of clusters that maximize the value of the PBMF-index which can ensure the formation of a small number of compact clusters with large separation between at least two clusters

D. Unsupervised learning module

Module requirements

The task of this module Fig (9) is to classify each pixel in the input matrix into certain number of classes that signifies the presence of certain type of mineral in the given area. This module will learn to classify the cluster center data obtained from the previous module into appropriate classes. After appropriate unsupervised training, the trained network should be able to recall the class to which a particular given pixel belongs from the cluster center data for such pixel.

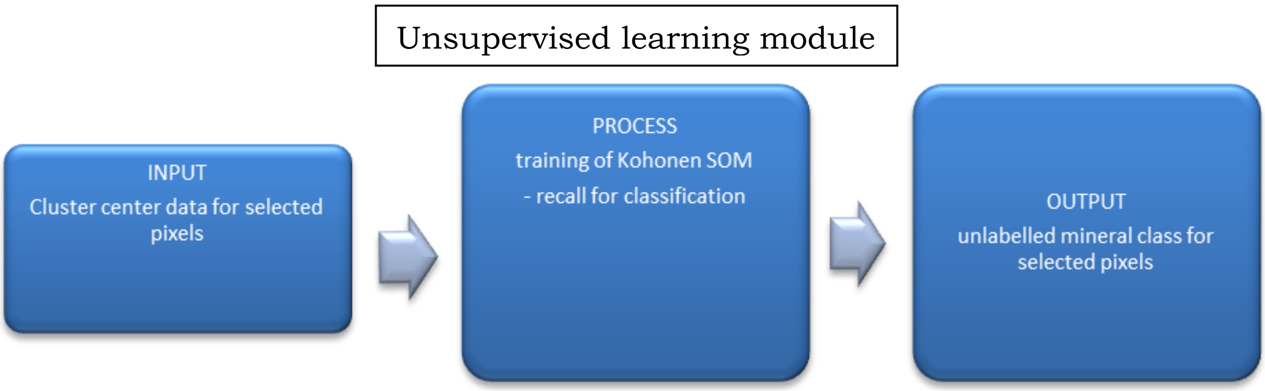

Unsupervised learning module
INPUT
Cluster center data for selected pixels
PROCESS
training of Kohonen SOM
- recall for classification
OUTPUT
unlabelled mineral class for selected pixels

Fig 9: Module design

The module uses unsupervised learning algorithm, namely Kohonen self-organizing maps, to classify the cluster center data for pixels. It takes the cluster center data for selected pixel samples as input and generates pixel classes as output.

E. Features of Kohonen's Self-Organizing Map (KSOM):

Fig 10 is a flowchart that explains the process of updating the weight of clusters .Fig 11 and 12 illustrate the adaptation of weight in KSOM. Fig 13 illustrate the cluster processing in KSOM. The training of the neurons is further explained by Fig 14.

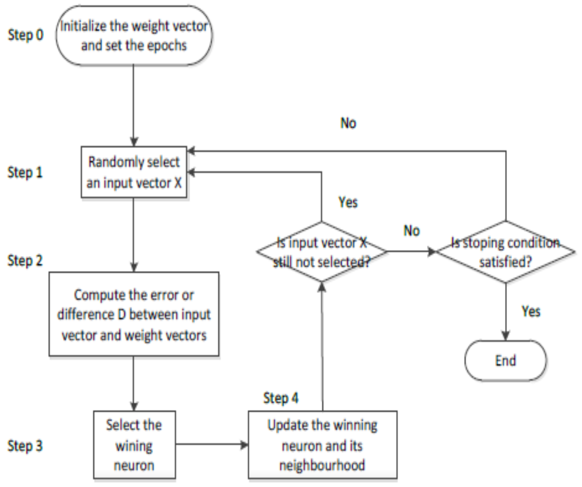

Step 0
Initialize the weight vector and set the epochs
Step 1
Randomly select an input vector X
Step 2
Compute the error or difference D between input vector and weight vectors
Step 3
Select the wining neuron
Step 4
Update the winning neuron and its neighbourhood
Is input vector X still not selected?
Yes
No
Is stoping condition satisfied?
No
Yes
End

Fig 10: Flowchart for the process of updating weights for cluster units

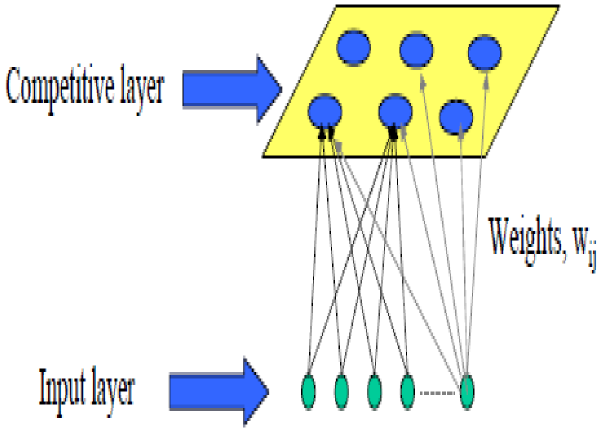


Fig 11: Segment of unsupervisory learning system (KSOM)

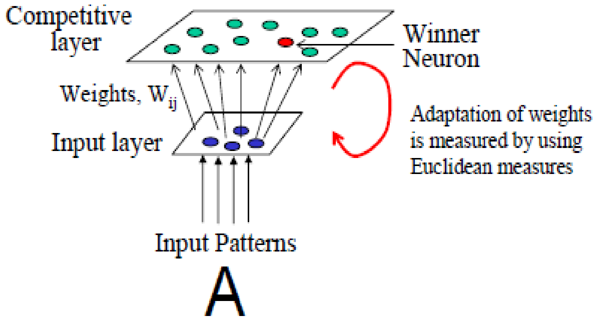


Fig 12: Adaptation of weight in KSOM

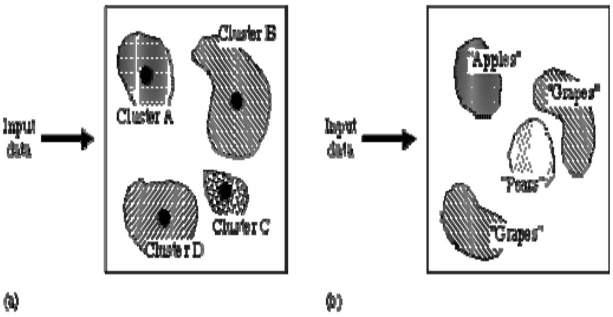


Fig 13(a, b): Cluster processing using KMOS

(a, b) By observing the input patterns, KSOM reorganizes them by clustering similar patterns into groups

F. Basic Principles of the KSOM

- The KSOM neural network is basically a single-layer feedforward network.
- When an input pattern is presented, each unit in the 1st layer takes on the value of the corresponding entry in the input pattern.
- The 2nd layer units then sum their inputs and compete to find a single winning unit.

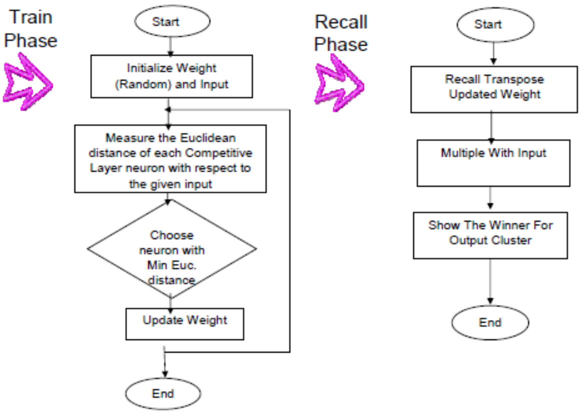

Train Phase
Start
Initialize Weight (Random) and Input
Measure the Euclidean distance of each Competitive Layer neuron with respect to the given input
Choose neuron with Min Euc. distance
Update Weight
End
Recall Phase
Start
Recall Transpose Updated Weight
Multiple With Input
Show The Winner For Output Cluster
End

Fig 14: Simple KSOM Flowcharts

H.The KSOM Algorithm

Step 1: Set up input neuron matrix, In _ X and In _Y.
Thus, total number of input neurons,
I = In _ X * In _Y.
We used i = 0 to I-1 for numbering the neurons in this layer.
Eg., Xj is the label for the input neurons i.e. X0 to X I-1

Step 2: Set up competitive layer matrix, Out _ X and Out _ Y.
For simplicity Out _ X = Out _ Y (such that we have a square map).
Therefore, total number of competitive neurons,
J = Out _ X * Out _ Y
We used j = 0 to J-1 for numbering the neurons in this layer

Step 3: Initialize connection weights (randomize) between input layer neurons and competitive layer neurons, Wij.
Set initial topological neighborhood parameters, *d*0.
Usually,

$$do \le \frac{Out\_X}{2}$$

Set initial learning rate parameter, α0
(Usually between 0.2 to 0.5)
Set total number of iterations, T (usually 10,000).
Start with iteration t = 0.
Apply the first pattern to the input of the KSOM.

Step 4: Compute the winning neuron (jc) in the competitive layer which is the minimum Euclidean distance from input layer to competitive layer such that
First, for each j (from j = 0 to J-1), compute the Euclidean distance as follows:

$$\|E(j)\| = \sqrt{\sum_{i=0}^{i=I-1}(w_{ij}-x_i)^2} \quad .....(5)$$

Then compare all these distances i.e. from $\|E(0)\|\ to\ \|E(j)\|$
and find the minimum distance $\|E(j_c)\|$which is the winner neuron, j_c.

$$\|E(j_c)\| = min\|E(j)\|$$

Calculate Euclidean distance for each Competitive layer neuron

$$\|E(j1)\| = \sqrt{(w_{11}-x_1)^2+(w_{21}-x_2)^2+\cdots+(w_{ij}-x_i)^2} \quad ......(6)$$

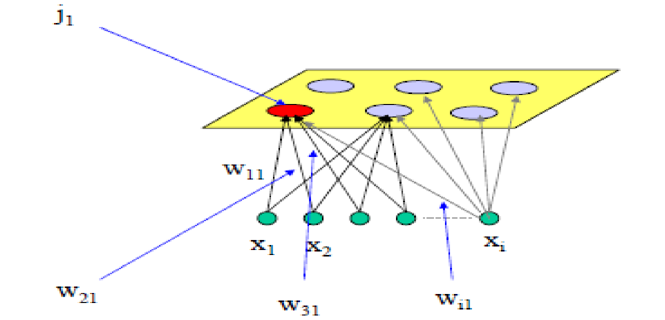


Fig 15: Diagrammatic illustration of the KSOM Algorithm

Step 5: Update weight for each connections i.e. For all neurons j within a specified neighbourhood of J, and for all i :

$$W_{ij}(new) = w_{ij}(old) + \Delta w_{ij}(new) \quad ......(7)$$

Where

$$\Delta W_{ij} = \begin{cases} \alpha_t(x_i - w_{ij}(old)) & \text{if unit is in neighbourhood } d_t \\ 0 & \text{otherwise} \end{cases}$$

Step 6: Update learning rate αt such that:

$$\alpha_t = \alpha_0\left(1-\frac{t}{T}\right)$$

Step 7: Reduce radius of topological neighborhood at specified times:

$$d_t = \mathrm{int}\left[d_0\left(1-\frac{t}{T}\right)\right] \quad ....(8)$$

Step 8: Increase iteration t: t=t+1
Repeat Steps 5 to 8 until t=T
Step 9: Repeat with next pattern chosen randomly
(Do Steps 4-9)

III. System Implementation

Computer simulations of self-organization in the Kohonen SOM.

1. Initialize weights to 0.5 + 10 % randomized value.
2. 2 input vectors, X1 and X2 with several scores of entries between the range of 0 and 1.
3. The Fig15 shows a plot of initial weights, wij.
4. Each unit in the competitive layer shows a point on this graph
5. The coordinate values of this point are the values of the incoming weights for the unit, thus wi1, wi2 are plotted for each competitive unit j.
6. All pairs of units in the competitive layer that are adjacent are connected.
7. This will allow us to see how the pattern of weights changes as the network organization evolves during training.

IV. PROGRAM TESTING

A. Results and Discussions

The test data consists of specter data for Cuprite, Nevada. The data consists of 600 by 320 pixels with 357 band spectrum ranging from 0.4 µm to 2.5 µm. Fig 16 shows the data cube visualization while Fig 17 shows a colormap slice (a band) of the given data . Figure 18 show the main interface window of the system. Fig 19 shows 3D plot of some bands of the input data.

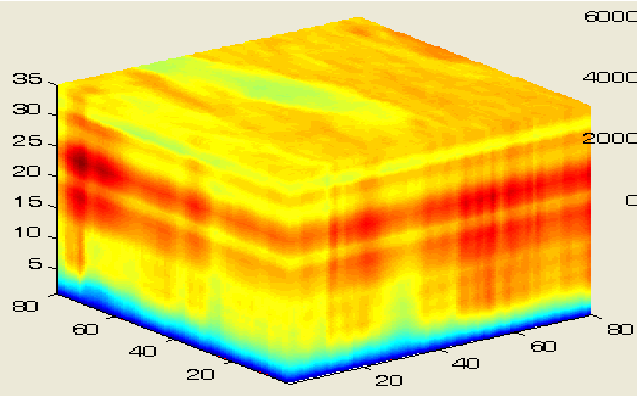


Fig 16: Data cube image of hyperspectral data.

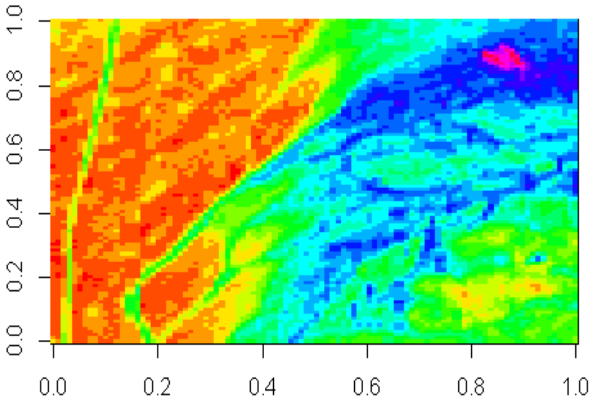
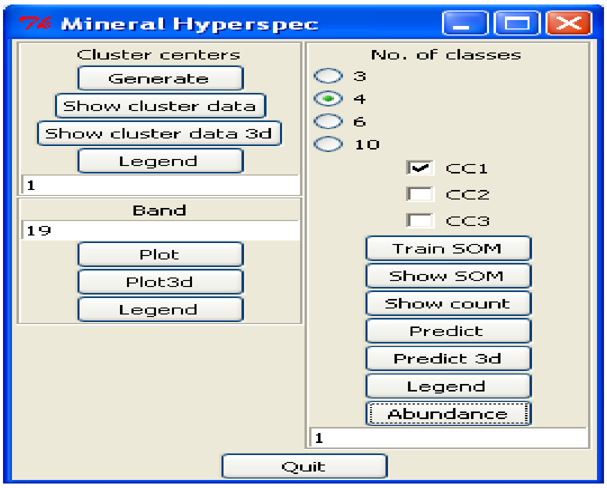


Fig 17: A band (slice) of hyperspectral data



Cluster centers: 1

Band: 19

No. of classes
○ 3
● 4
○ 6
○ 10

☑ CC1
☐ CC2
☐ CC3

1

Fig 18: The main interface window of the system

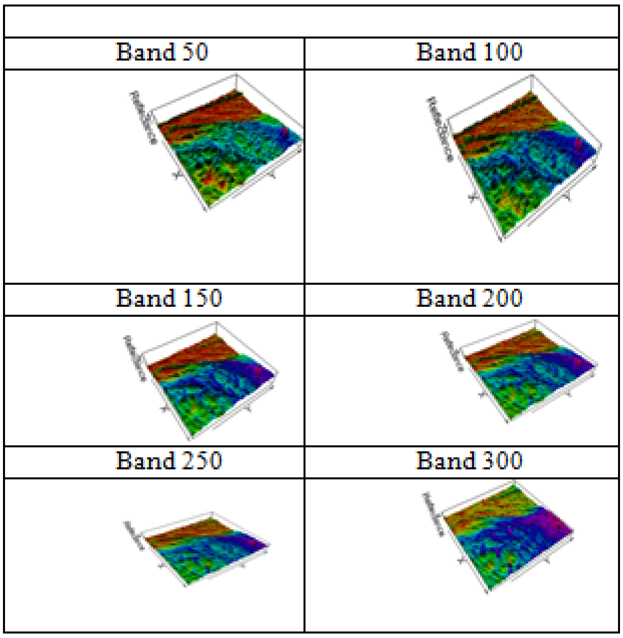


Fig 19: 3D Plot of some bands of the input data

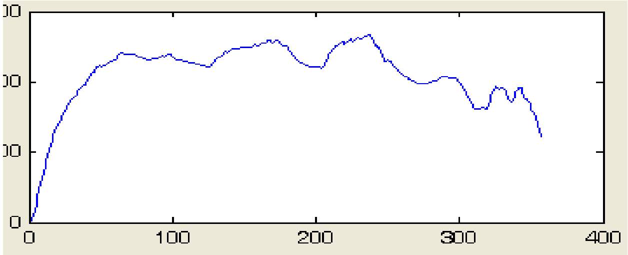


Fig 20: The Spectrum of a pixel

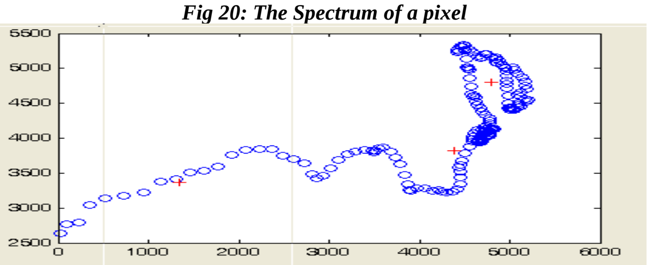


Fig 21: Clustered image of a pixel with 3 cluster centers shown in red

Processing of spectrum for each pixel.

The spectrum for each pixel is selected in turn, displayed in Fig 20 and turned into characterization map. The characterization map for a particular pixel is shown in fig 21. The characterization map is then clustered to obtain 3 cluster centers marked in red in Fig 21. The cluster center data for each pixel is thus calculated in turn and stored in a cluster center data structure for file storage or further processing. The cluster center data distill the essential features for classification and recognition of mineral classes in the given data.

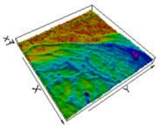
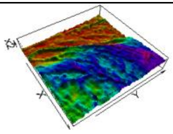
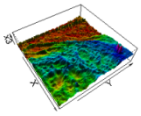
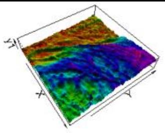
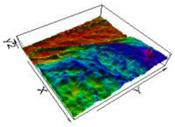
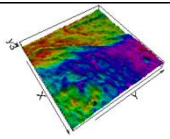
	CLUSTER CENTER		
	1	2	3
X			
Y			

Fig 22: 3D Plots of the cluster center data

Result of clustering, the x and y-coordinate data for each of the 3 cluster center for each of the pixel is shown above Fig (22)

The maps show clearly the correctness of the method in generating mineral predictor maps, fig 23.

The obtained predicted mineral maps now form the basis for mineral classification with intelligent system. In absence of a prior knowledge about the kinds of minerals available in the given area, the researchers used self-organizing kohonen maps to obtain classes of minerals in the given data. We selected some samples from the cluster center data and use it to train Kohonen network which predicts the classes of minerals as shown in Fig 23. The program is run for different numbers of output neurons and epochs and the results are shown variously in Figs 23-25.

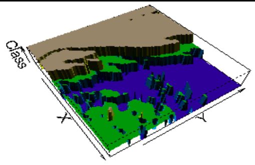
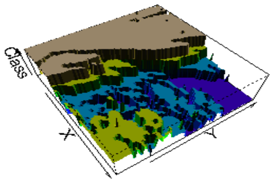
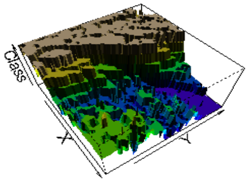
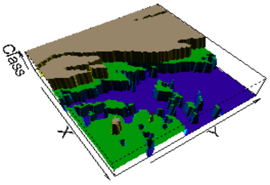
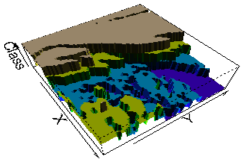
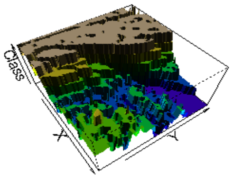
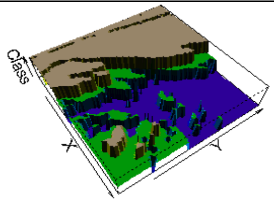
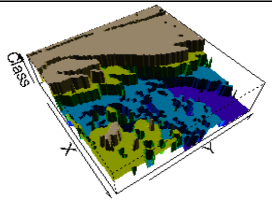
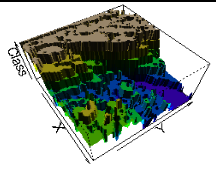
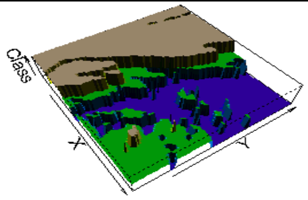
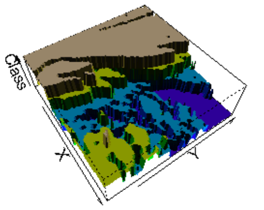
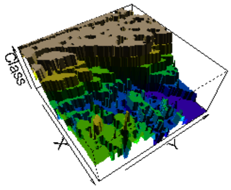
	PREDICTED MINERAL MAP		
	3 classes	4 classes	6 classes
CC1			
CC2			
CC3			
JJoint CC123			

Fig 23: Predicted Mineral Map

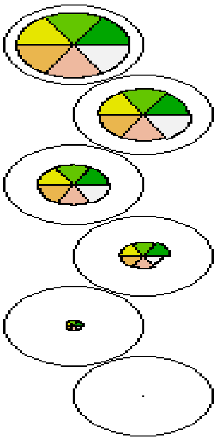


Fig 24: Plot of learning SOM

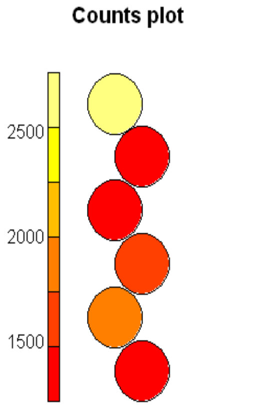


Fig 25: Class count plot of trained SOM

V. CONCLUSION

In this research work, an attempt was made to cluster hyperspectral data using fuzzy C means. The cluster centers generated were used to train Kohonen Self Organizing Map to identify various types of minerals available in a particular location

The KSOM is trained without teaching signals or target. Based on a series of input patterns, KSOM learns by itself to cluster the patterns according to their similar features. Hence, it was able to classify different types of minerals that were present in the given location.

REFERENCES

[1]. Belhassen, S.—Zaidi, H. (2010). A Novel Fuzzy C-Means Algorithm for Unsupervised Heterogeneous Tumor Quantification in PET. Medical Physics, Vol. 37, No. 3, pp. 1309–1324.

[2]. Bezdek, J., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer, Academic Publishers, Norwell, Massachusetts. ISBN 0-306-40671-3.

[3]. Farhad Samadzadegan, Amin Alizadeh Naeini, Shahin Rahmatollahi Namin, Mahmud Haghshenas (2011). Evaluating the potential of particle swarm optimization in clustering of hyperspectral imagery using fuzzy c-means. *International Conference on Asia Agriculture and Animal IPCBEE vol.13, IACSIT Press, Singapoore.*

[4]. Fu, L.—Medico, E.: FLAME: A Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data. BMC Bioinformatics. Vol. 8, 2007, No. 3, doi:10.1186/1471-2105-8-3.

[5]. HöPPNER, F., KLAWONN, F., KRUSE, R. & RUNKLER, T. (1999). *Fuzzy cluster analysis: methods for classification, data analysis and* image *recognition*, John Wiley.

[6]. Krinidis, S.—Chatzis, V. (2010). A Robust Fuzzy Local Information C-Means Clustering Algorithm. IEEE Trans. on Image Processing, Vol. 19, No. 5, pp. 1328–1337.

[7]. Sikka, K.—Sinha, N.—Singh, P.K.—Mishra, A.K. (2009). A Fully Automated Algorithm Under Modified FCM Framework fr Imroved Brain MR Image Segmentation. Magnetic Resonance Imaging, Vol. 27, No. 7, pp. 994–1004.

[8]. Subhagata Chattopadhyay, Dilip Kumar Pratihar and Sanjib Chandra De Sarkar (2011). A Comparative Study of Fuzzy C-Means Algorithm and Entropy-Based Fuzzy Clustering Algorithms. Computing and Informatics, Vol. 30, pp 701–720.

[9]. Venkateswara Reddy Eluri, Dr.E.S.Reddy (2013). A Comparative Study of Color Image Segmentation Using Hard, Fuzzy,Rough Set Based Clustering Techniques. International Journal of Computers & Technology. Vol 11, No.8, ISSN 2277-3061. www.cirworld.com.

[10]. YANG, F., ZHANG, C. & SUN, T. (2009). Particle Swarm Optimization and Differential Evolution in Fuzzy Clustering. Advances *in Neuro-Information Processing,* 501-508.