SEGMENTATION BASED CLUSTERING OF HYPERSPECTRAL IMAGES

A project report submitted

in partial fulfilment of the requirements for award of the degree of

Bachelor of Technology

in

Civil Engineering

Submitted by

Hitenkumar Motiyani (Enrol. No. 191010011016)

Prashant Kumar Mali (Enrol. No. 191010012016)

Quazi Sameed (Enrol. No. 191010012019)

Department of Civil Engineering Institute of Infrastructure, Technology, Research And Management (IITRAM), Ahmedabad-380026, Gujarat, India

April 2023

SEGMENTATION BASED CLUSTERING OF HYPERSPECTRAL IMAGES.

A project report submitted
in partial fulfilment of the requirements for award
of the degree of

Bachelor of Technology
in
Civil Engineering

Submitted by
Hitenkumar Motiyani (Enroll. No. 191010011016)
Prashant Kumar Mali (Enroll. No. 191010012016)
Quazi Sameed (Enroll. No. 191010012019)

Supervisor(s)
Dr. Anand Mehta, Assistant Professor

Department of Civil Engineering
Institute of Infrastructure, Technology, Research And Management
(IITRAM), Ahmedabad-380026, Gujarat, India
April, 2023

Department of Civil Engineering
Institute of Infrastructure, Technology, Research And Management, Ahmedabad

DECLARATION

I/We declare that this written submission represents my/our ideas in my/our own words. Where others ideas or words have been included, I/We have adequately cited and referenced the original sources. I/We also declare that I/We have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I/We understand that any violation of the above will cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Hitenkumar Harish Motiyani
(191010011016)

Prashant Kumar Mali
(191010012016)

Quazi Sameed
(191010012019)

Date: 20/04/2023

Department of Civil Engineering
Institute of Infrastructure, Technology, Research And Management, Ahmedabad

CERTIFICATE

This is to certify that the thesis entitled, **"Segmentation based Clustering of Hyperspectral Images"** being submitted by **Mr. Hitenkumar Harish Motiyani (Enrol. No. 191010011016), Mr. Prashant Kumar Mali (Enrol. No. 191010012016)** and **Mr. Quazi Sameed (Enrol. No. 191010012019)** the Institute of Infrastructure, Technology, Research and Management (IITRAM), Ahmedabad for the award of the degree of **Bachelor of Technology in Civil Engineering** is a bonafide record of research work carried out by him under my/our supervision and guidance. The thesis work, in my/our opinion, has reached the requisite standard fulfilling the requirement for the degree of **Bachelor of Technology**.

The results contained in this thesis have not been submitted, in part or full, to any other University or Institute for the award of any degree or diploma.

Date:

Place:

.....	.....
Dr. Anand Mehta	**Dr. Manoj Langhi**
Assistant Professor,	Coordinator
Department of Civil Engineering.	Department of Civil Engineering.

Abstract

This research uses information in the form of an image obtained through remote sensing. Remote sensing refers to gaining knowledge about an object, area, or phenomenon via the analysis of data acquired by a device that is not in contact with the object, area, or phenomenon under investigation. The image employed in our project is hyperspectral image which constitutes bands that are continuous and regularly spaced with higher spectral resolution, thus giving the opportunity to push further the information extraction capability.

The main motivation of this work is information extraction from less interpretative image. The image obtained through remote sensing is unclassified which is further classified to generate data that contains land used land cover information in order to provide a better understanding of land utilization aspects.

In this study, feature reduction, segmentation, and clustering on hyperspectral imagery are performed, to extract information. The proposed methodologies are multi-stage frameworks. In first methodology, initially, *k*-means is utilized to perform feature reduction. In the next stage, *k*-means is again deployed to perform hyperspectral image segmentation, using new feature set obtained from the first stage. Finally, *k*-means clustering is carried out on segmented hyperspectral image by making use of reduced feature set. While, in second methodology, algorithm utilizes *k*-means for segmentation, further uses a local feature selection technique to obtain the top bands for each cluster and deploys clustering on segmented hyperspectral imagery. Here, *k*-means is initially utilized for image segmentation. From the obtained segments, significant segments are identified using Gini impurity. Finally, the cluster map is obtained by merging insignificant clusters with significant clusters. Moreover, in third methodology, which is similar to second methodology, algorithm applies entropy based local feature selection to choose the top bands for each cluster. A frame work with numerous steps constitutes the proposed methodology which follows by segmenting a hyperspectral image using *k*-means to obtain new segments. Then, identifying significant clusters using entropy and combining the remaining clusters with the significant ones according to the minimum distance from significant clusters, taking into account significant segments

with bands having least redundancy from the local band selection technique in the subsequent stage.

To assess the performances of all the proposed methodologies Purity, Normalized Mutual Information (NMI), Adjusted Mutual Information (AMI) and Overall Accuracy scores are used. The experimental results prove that the third proposed methodology has an edge over the other compared clustering methodologies. Results indicate that accuracy is enhanced to 84.31% by selecting multiple feature set through Entropy while performing segmentation-based clustering.

Keywords: Remote Sensing, Hyperspectral image, Clustering, Feature Reduction, Segmentation, Local Band Selection, Relevancy, Redundancy, Gini impurity, Entropy.

Table of contents

List of Tables

List of Figures

Chapter 1 Introduction

The multidimensional dataset that makes up hyperspectral data can be displayed as a collection of images. A sufficient amount of spectral information is provided by hyperspectral images to identify and distinguish various materials [1]. Hyperspectral imaging is one of the datasets from which we can find and recognize things in a variety of images. Red, green, and blue are the main colour ranges on which human vision is based, but spectroscopy breaks vision into numerous more bands. The electromagnetic spectrum's hundreds of adjacent spectral bands make up the visual data which hyperspectral sensors collects [2]. The most crucial techniques in hyperspectral imaging that aids in information extraction are cluster analysis, dimensionality reduction, and image segmentation [3].

One important data analysis method that is frequently used for numerous empirical applications in new areas is cluster analysis [4]. The process of identifying groups of things is called clustering. This ensures that the objects in a group will be similar to one another and distinct from the objects in other groups, and that they will generally share the same properties [5].

k-means is one of the widely used clustering methods [8]. Compared to a hierarchical clustering, it is simpler and faster in terms of computation. Additionally, it works well with a lot of different factors. For a different number of groups, however, it produces a different cluster outcome. Consequently, it is crucial to establish the proper amount of clusters. Moreover, a different set of initial centroid values will provide a different set.

In general, dimensionality of HSI can be decreased using feature extraction or feature selection (also known as band selection) techniques. A reduced data set is produced by projecting the original HSI onto a lower dimensional space for feature extraction [9]. To accurately represent the original data set, some discriminative bands are chosen while selecting bands. Feature extraction often generated better outcomes in tests. The smaller data sets are easier to understand because to band selection, which can preserve the physical information of the original data. [15].

Dimensions, or the number of bands, attributes, or variables that make up a dataset, are reduced by dimensionality reduction. The aim is to reduce the number of sizes, which are represented as bands [6]. Since these bands are frequently connected, there is some extra information that

adds to the dataset's noise. The learning outcomes of our model are negatively impacted by this extraneous input; hence it is essential to employ feature reduction approaches.

A specific image processing approach known as image segmentation is used to separate an image into two or more useful regions. The process of making boundaries between various semantic components in an image is known as image segmentation [7]. Image segmentation, from a more technical point of view, is the process of giving each pixel in the image a label so that pixels with the same label are related with respect to some visual or semantic attribute [14].

Based on the measurement, the image might have certain properties like that grey level, colour intensity, texture information, depth, or motion. The clustering method is used in image segmentation. Identifying groups of related images is the process of clustering in image segmentation [16]. There are numerous clustering algorithms that can be categorised in order to obtain the super pixel information. To achieve the right outcome with great efficiency, which affects storage picture, clustering approach is used [17].

In image segmentation, the image is divided into various regions so that the pixels have a high difference between spaces and a high similarity in each area [7]. It is a useful tool for a variety of tasks, including hyperspectral image processing, and there are various methods for image segmentation, including neural networks and thresholds. Utilizing the clustering method is one of the other techniques that is quick and simple [5].

Simple Linear Iterative Clustering (SLIC), a commonly used technique for image segmentation, is approached by making use of superpixels. A practical primitive for computing local image attributes is the superpixel. They take advantage of image redundancy. They have proven to be more and more beneficial for tasks including object localisation, skeletonization, depth estimation, and image segmentation [13].

Superpixels must be quick, simple to use, and able to create segmentations of excellent quality in order to be helpful. Unfortunately, the majority of modern superpixel techniques fall short of all of these specifications [13]. We shall show that they frequently have significant computational costs, subpar segmentation, irregular size and form, or various hard-to-tune properties.

In many image processing methods, segmentation is an essential component. It is possible to identify regions of interest and objects in the scene from the segmentation findings, which is super beneficial for the subsequent image analysis or annotation [7].

Recent work includes a variety of techniques: for example, recognition task like face recognition, traffic control system for video surveillance. Thus, it involves partitioning images into multiple segments or objects [14].

In this research three novel clustering methodologies are proposed. In the first methodology, the main contribution is that a clustering methodology is proposed in which clustering, feature reduction and segmentation are performed by utilizing k-means only. In the second `methodology, the scope of this research is a new clustering technique that employs Gini Impurity. Also, a novel local band selection strategy with relevancy and redundancy concept is proposed. Here, local band selection means we are using reduced bandset for each cluster.

Finally, the third methodology, main contribution is entropy based local feature selection technique and clustering method. The proposed local feature selection technique takes into account both relevancy and redundancy concept while identifying top features (bands). Local feature selection in this instance refers to selecting a reduced set of bands for each cluster.

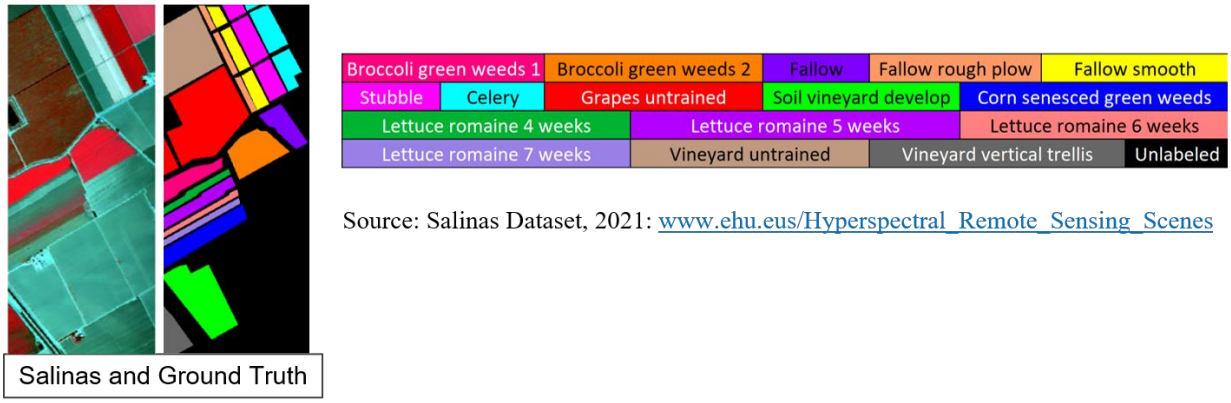
The rest of the paper is organised as follows. Chapter 2 represents Literature Review. Chapter 3 describes the methodology. Chapter 4 presents datasets considered and functions utilized. Chapter 5 presents the results of the study. Chapter 6 concludes this work.

1.1 Objective

To propose a novel segmentation-based clustering methodology for classification of hyperspectral image and to improve classification accuracy of the input dataset. The main motivation of this work is information extraction from less interpretative image and to convert this information into land use and land cover data. The need of Land Use Land Cover Data is that it provides a better understanding of land utilization aspects. Also, it may be used for planning, monitoring, and evaluation of development, industrial activity, or reclamation.

1.2 Datasets

The AVIRIS Airborne Visible Infrared Imaging Spectrometer sensor (224-band) was used to capture the Salinas dataset (contains 204 bands), over the Salinas Valley in California. A high spatial resolution of 3.7-m pixels sets it apart. The covered region has 217 samples and 512 lines. Salinas ground truth, however, contains 16 different types of land use and covers. This image was available only as at-sensor radiance data. It includes vegetables, bare soils, and vineyard fields.



Source: Salinas Dataset, 2021: www.ehu.eus/Hyperspectral_Remote_Sensing_Scenes

Fig.1.1 Salinas Dataset and its Ground truth Classes (Source: [12]).

Pavia Center and Pavia University are the second and third datasets taken into account in the trials. These two scenes were captured by the ROSIS (Reflective Optics System Imaging Spectrometer) sensor during a flight campaign over Pavia, a city in northern Italy. Pavia University has 103 spectral bands and contains nine classes on the other hand Pavia Center has 102 spectral bands and consists nine classes while it's subset has eight. Here, 400 x 400 pixels and 610 x 340 pixels are the taken as subset of Pavia Centre and Pavia University, from their original datasets, which are 1096 x 1096 pixels and 610 x 610 pixels, respectively. The two dataset's geometric resolution is 1.3 metres.

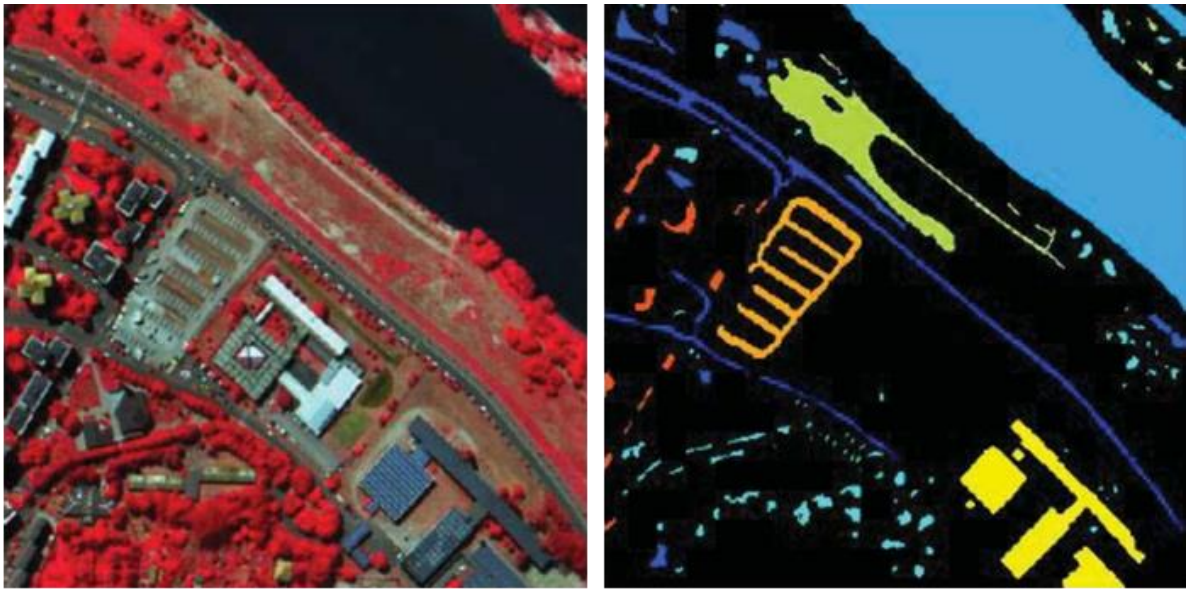


Fig.1.2 Pavia Center and its Ground truth Classes (Source: [12]).

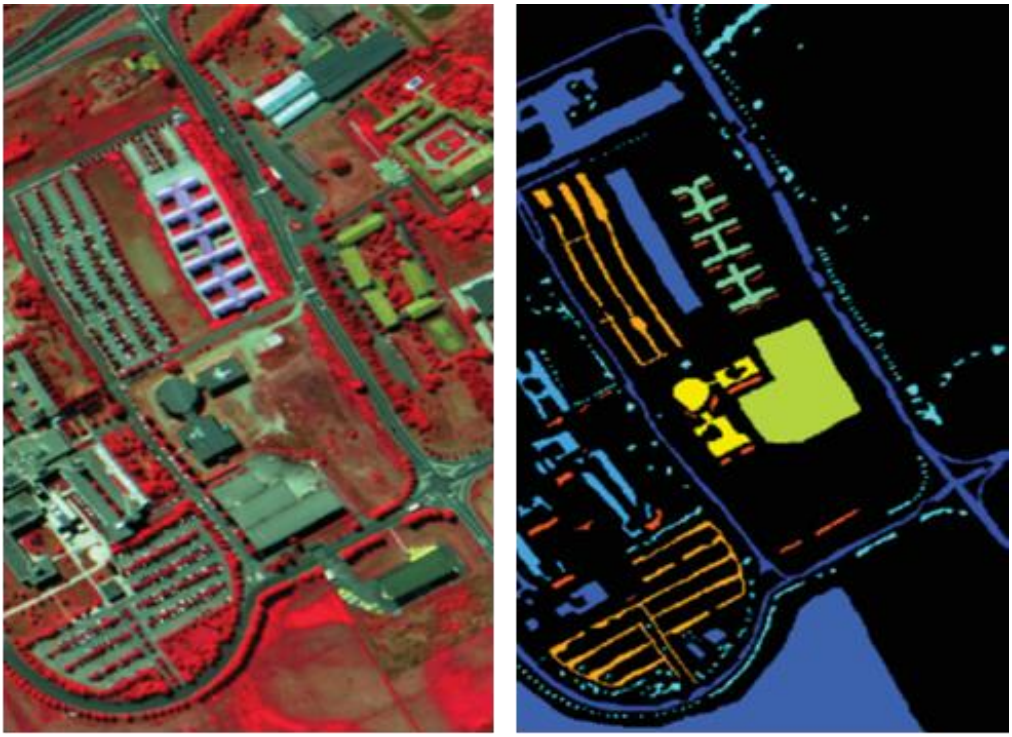


Fig.1.3 Pavia University and its Ground truth Classes (Source: [12]).

Table 1.1 Datasets and their respective samples number

Groundtruth of Salinas scene and their respective tests samples			Groundtruth of Pavia University with respective test samples			Groundtruth of Pavia centre with respective test samples		
#	**Class**	**Samples**	**#**	**Class**	**Samples**	**#**	**Class**	**Samples**
1	Brocoli_green_weeds_1	2009	1	Asphalt	6631	1	Water	824
3	Fallow	1976	2	Meadows	18649	2	Trees	820
4	Fallow_rough_plow	1394	3	Gravel	2099	3	Asphalt	816
5	Fallow_smooth	2678	4	Trees	3064	4	Self-Blocking Bricks	808
6	Stubble	3959	5	Painted metal sheets	1345	5	Bitumen	808
7	Celery	3579	6	Bare Soil	5029	6	Shadows	476
8	Grapes_untrained	11271	7	Bitumen	1330	7	Meadows	824
9	Soil_vinyard_develop	6203	8	Self-Blocking Bricks	3682	8	Bare Soil	820
10	Corn_senesced_green_weeds	3278	9	Shadows	947			
11	Romaine_ lettuce_4_week	1068						
12	Romaine_ lettuce_5_week	1927						
13	Romaine_ lettuce_6_weeek	916						
14	Romaine_ lettuce_7_week	1070						
15	Vinyard_Untrained fields	7268						
16	Vinyard_vertical_trellis fields	1807						

1.3 Software and Machine

The aforementioned clustering methodology was coded in Spyder, Python 3.9.7 on a computer device with 8 GB of RAM and an Intel i5 10th generation processor running Windows 11 at 4.30 GHz.

1.4 Structure of Report

Chapter 1 represents Introduction and the rest of the paper is organised as follows. Chapter 2 represents Literature Review. Chapter 3 describes the methodology. Chapter 4 presents datasets considered and functions utilized. Chapter 5 presents the results of the study. Chapter 6 concludes this work.

Chapter 2 Literature Review

Some of the hyperspectral image clustering work is based on a well-liked algorithm called the Sequential Hierarchical method, which treats each entity as a separate cluster [2]. Two clusters are combined at each level of the clustering process, and the process is repeated until only one cluster, comprising the whole dataset, is left. However, partitioning techniques result in discrete, non-overlapping groups [9]. Due to the fact that only a single data division is formed, the technique is frequently referred to as nonhierarchical clustering [10].

Compared with supervised or semi-supervised methods, HSI clustering is often a fundamental but challenging task, due to prior knowledge deficiency, large spectral variability, and high dimension of HIS [19]. The abundant spectral information of HSIs comes at the cost of greatly reducing spatial resolution which hinders the widespread applications of HSIs [21]. Moreover, Principal Component Analysis (PCA) was introduced as a linear dimensionality reduction technique for HSI which projects the high-dimensional data to a low-dimensional subspace with principal components maximizing the variance of the projected data [20].

Describe the new superpixel algorithm known as simple linear iterative clustering (SLIC), which effectively creates superpixels by adapting the k-means clustering method. SLIC adheres to boundaries as well as or better than older techniques, despite its simplicity. Additionally, it increases segmentation performance while being quicker and more memory-efficient [22].

Although there are hundreds of segmentation techniques documented in the literature, neither one method nor all methods are equally effective for every kind of image. Additionally, algorithms created for one type of image, such as an ordinary intensity image, may not always be applicable to another type of image, such as an MRI or RI. This is especially true if the algorithm makes use of a particular model for image formation. For instance, some visual image segmentation algorithms are predicated on the idea that the illumination component and the reflectance component can be used to model the grey level function f(x,y). On the other hand, based on the theory of how visual images are formed, the grey level distributions in Pal and Pal tS) have been modelled as Poisson distributions. MRI/RIs should not be subjected to such 17's methods. However, the majority of segmentation techniques created for one class of images can be quickly extended to another. Although it was designed for range images, the variable order surface fitting method, tg), can be used for other images that can be modelled as a noisy version of piece-wise smooth surfaces.

Numerous nonlinear methods of dimensionality reduction have been put forth in the last ten years. For a summary, see, for instance, [10, 11, 12, 13]. The nonlinear techniques can handle

complex nonlinear data, in contrast to the conventional linear techniques. The nonlinear dimensionality reduction techniques may be advantageous for real world data. On challenging artificial tasks, nonlinear techniques perform better than their linear counterparts, according to prior research. For example, the Swiss roll dataset consists of a collection of points that are located on a 2D manifold with spiral-like attributes that is contained within a 3D space. In contrast to linear techniques, a wide variety of nonlinear methods successfully find this embedding. Beyond this observation, it is unclear how much the performances of the various dimensionality reduction techniques vary on artificial versus natural tasks (a comparison is made in [22], but its scope is very constrained in terms of the number of techniques and tasks it covers).

Chapter 3 Theoretical Background

3.1 Hyperspectral Imaging

Hyperspectral Imaging (HSI) has received a lot of interest in processing recently. Since, HSIs may offer rich band information from many wavelengths, they are frequently used in a variety of research fields. The reflectance of electromagnetic waves of various wavelengths are captured by HSIs, and each electromagnetic wave's reflectance is saved as a two dimensional image. Thus, an HSI is a data cube that includes a large number of two dimensional images. Even while HSI applications have seen tremendous success, dealing with huge dimensional data to extract information remains a difficult task. Hence, to tackle such crucial situation, techniques like cluster analysis, image segmentation and dimensionality reduction technique like principal component analysis are utilized

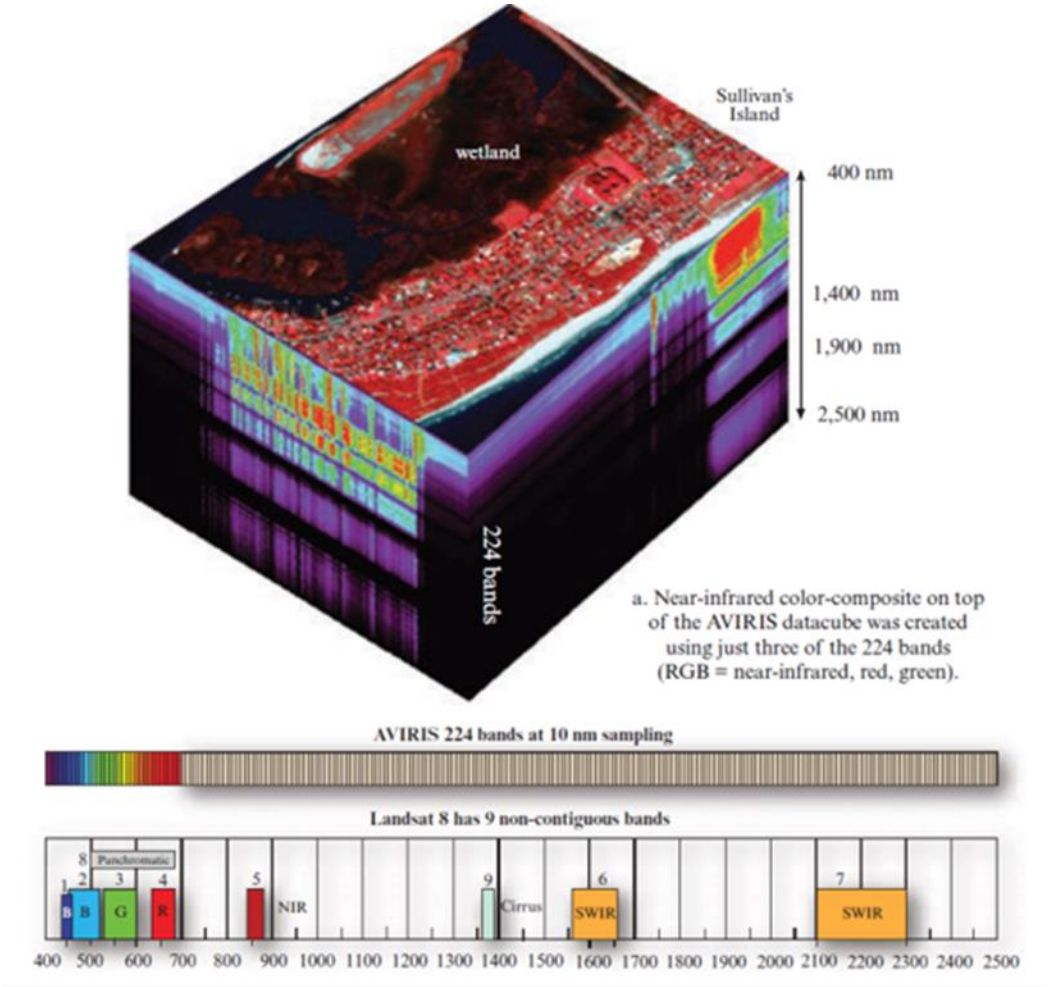


Fig.3.1 Hyperspectral Imaging (Source: [23]).

Hyperspectral image is considered in the form of a matrix which comprises information in rows and columns with number of pixels and bands. In this type of image, bands are continuous and regularly spaced with higher spectral resolution, thus giving the opportunity to push further the information extraction capability.

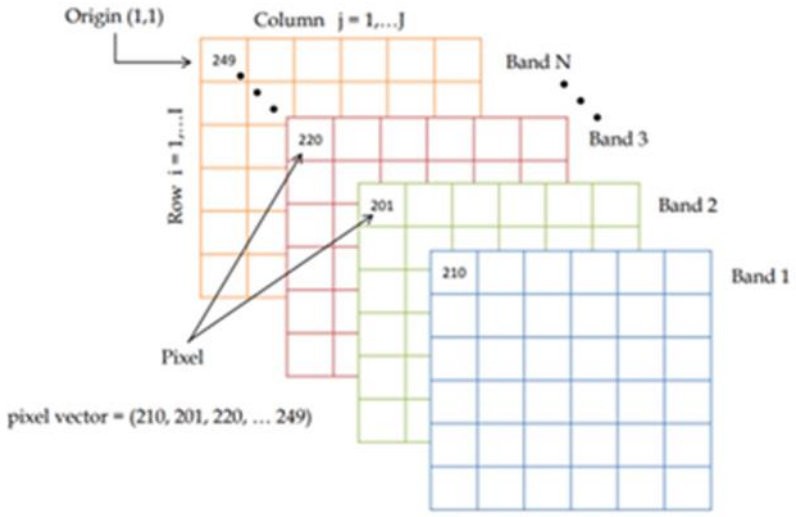


Fig.3.2 An example of Hyperspectral Data

This technology has a broad range of applications across multiple fields. HSI can be used to assess crop health and detect disease or nutrient deficiencies before they become visible to the naked eye. This can help farmers optimize fertilizer application and reduce crop loss. It can help detect pollution, track changes in land use, and monitor natural disasters such as wildfires, floods, and hurricanes. It can be used to diagnose and monitor skin conditions such as melanoma, as well as detect cancer cells in tissue samples. It can be used to collect data about the Earth's surface from space, allowing us to monitor environmental changes on a global scale. It can be used to analyze crime scene evidence, such as bloodstains, to identify substances that may not be visible under normal lighting conditions. Overall, hyperspectral imaging has numerous applications that are revolutionizing various industries and fields, and its potential for innovation and discovery is vast.

3.2 Image Classification

Image classification is the process of categorizing and labelling groups of pixels or vectors within an image based on specific rules. For example, an image classification algorithm can be trained to identify different types of animals, such as dogs, cats, and birds, based on their visual features.

The categorization law can be devised using one or more spectral or textural characteristics. Each pixel of the image is assigned to a particular class. Classification transforms the image data into

an information. Image classification has many applications, such as object recognition in computer vision, medical image analysis, and autonomous vehicles.

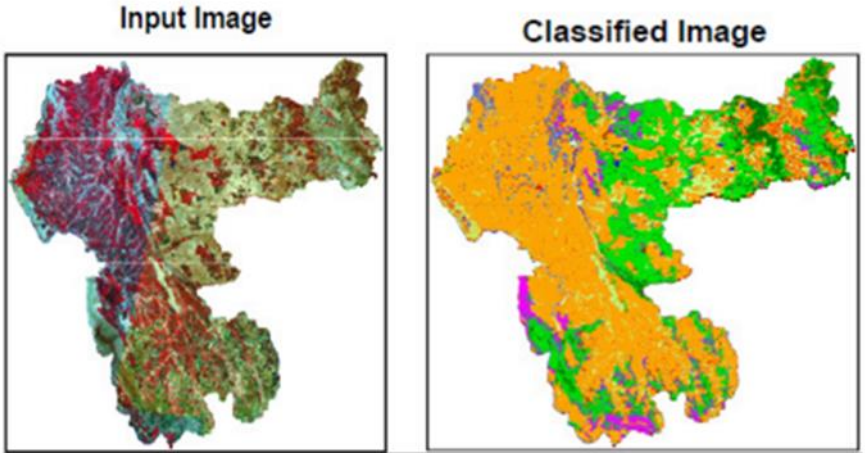


Fig.3.3 Image Classification

There are two types of image classification: Supervised Classification and Unsupervised Classification.

3.2.1 Supervised Classification

Supervised classification is a type of machine learning task where a model is trained to predict the class of new, unseen data based on labeled training examples. In other words, the model is presented with a set of input data and corresponding output labels, and it learns to map the inputs to the correct outputs. The labeled data used for training is typically divided into two sets: a training set used to train the model, and a validation or test set used to evaluate the model's performance on new, unseen data. Supervised classification can be used for a wide range of tasks, such as image classification, text classification, and speech recognition. Some common algorithms used for supervised classification include logistic regression, decision trees, random forests, support vector machines, and neural networks.

The steps for performing supervised classification starts with identifying training set (information class) followed by selection of sample pixels in an image representing specific class, and then by directing the image processing software to use training sets as references for the classification of all other pixels in the image.

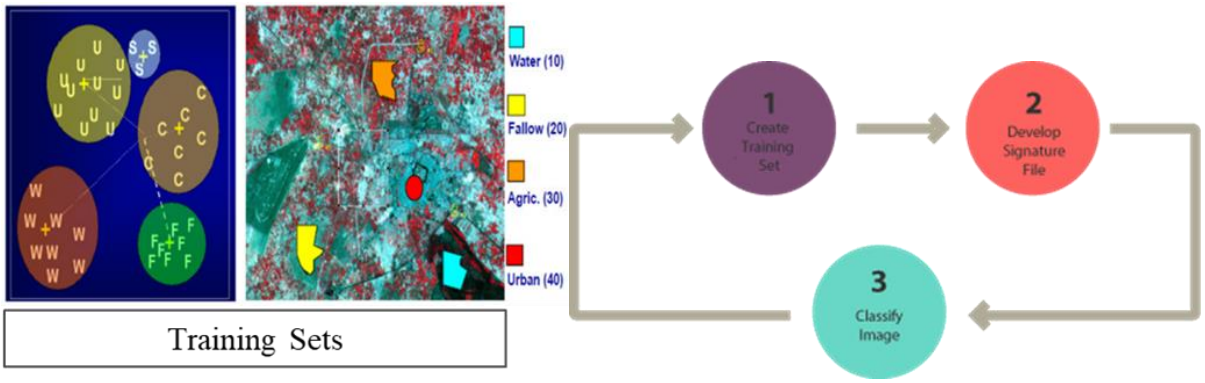


Fig.3.4 Supervised Classification algorithm

3.2.2 Unsupervised Classification

Unsupervised classification does not need to know in advance which classes are of interest. Instead, it analyses the data and divides it into the most common natural spectral groupings, or clusters, that are present.

In this work, the type of image classification method used is unsupervised classification. The following are the methods utilized in unsupervised classification:

3.2.2.1 *k*-means Algorithm

k-means is a partition-based clustering algorithm that seeks to locate for the user a predetermined number of *k* clusters, which are represented by their centroids, by lowering the function of square error [8]. Despite being straightforward, *k*-means can be used for a variety of data sets. One clustering technique based on division and non-hierarchical clustering techniques is the *k*-means algorithm. The *k*-means algorithm searches for a cluster between the set of numerical items X and the integer *k*, which lowers the total square errors within clusters.

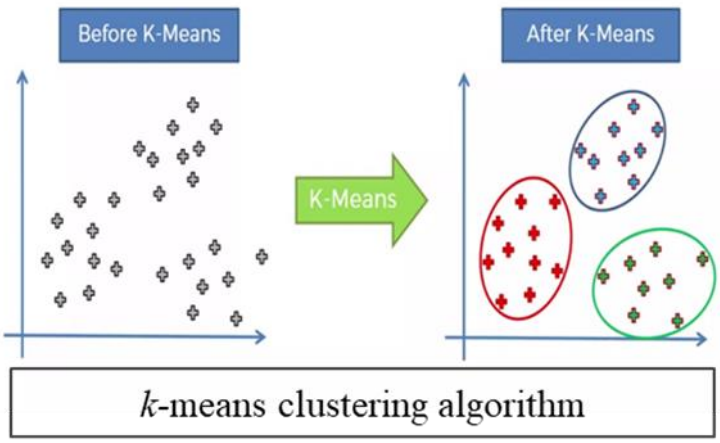


Fig.3.5 An example of *k*-means clustering algorithm

Starting with the assembly of *k* cluster centres, the *k*-means method is used. The input data points (pixels) are then dispersed among the clusters that already exist in accordance with their Euclidean boundaries, and the closest cluster is chosen. The centre of each cluster is then updated by computing the average (centroid) for each cluster [9]. The membership of each cluster has changed as a result of this update. In order to prevent additional changes in the value of any cluster centre, the pixel vectors are reset repeatedly, and the cluster centres are updated [10]. The steps for performing *k*-means clustering are shown below (Algorithm 1).

Algorithm 1 *k*-means based clustering

Input: $X = \{x, x_1, x_2, ..., \} x_n$ // Set of *n* pixel vectors.

Output: Set of *k* clusters.

- Select *k* pixel vectors randomly to configure the cluster.
- For each pixel vector, find the center of the nearest cluster and set that pixel vector to the appropriate cluster.
- Update the centers of each cluster by using the centroid for the pixel vectors assigned to that cluster.
- Repeat procedure until there is no more change in the value of the means.

3.3 Feature Reduction

The amount of pixels and bands in a hyperspectral image give it an extremely high spectral resolution [1]. Feature reduction is used to reduce the number of features in a dataset without significantly lowering the quantity of information retained [9]. The analysis will proceed more quickly for the smaller dataset. The procedure for accomplishing feature reduction entails arranging the provided dataset into a 2D matrix, where the rows represent pixels and the columns represent bands. Here, *k*-means is performed on bands instead of pixels in order to obtain representative bands from each set of clusters.

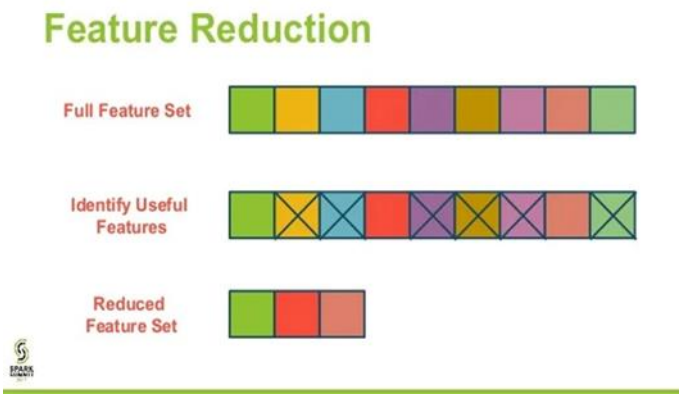


Fig.3.6 An example of Feature Reduction algorithm

These high numbers attributes (band) sets make training extremely slow.

- Harder to find a good solution.
- These problems are often referred to as the curse of dimensionality.

The two types of feature reduction techniques are feature extraction and feature reduction.

Principal Component Analysis (PCA) comes under feature extraction technique while local band selections are featured in feature reduction.

3.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique which generally transforms original attributes into new attributes of same numbers known as PC's but this time there is correlation between the attributes [11].

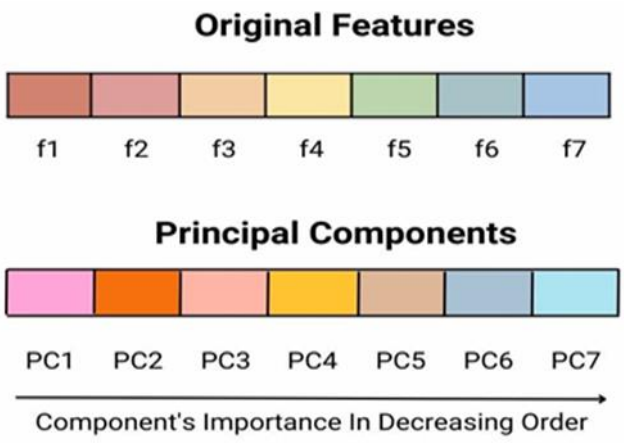


Fig.3.7 Principal Components (Source: [24]).

Firstly, *k* -means with PCA was advanced where the data was standardized and covariance was found to determine the correlation between them. Moreover, the principal components (PC) of the datapoints were obtained to know eigenvector and eigenvalues in order to find out the percentage of variance each PC constitutes.

The steps for performing PCA are shown below (Algorithm 2)

Algorithm 2 Principal Component Analysis (PCA)

- To standardize the data for comparison.
- To find the covariance in order to determine the correlation.
- To know the principal components of the datapoints by obtaining eigenvector and eigenvalues.
- Sorting the eigen values in decreasing order and determining the percentage of variance / information each PC constitutes.

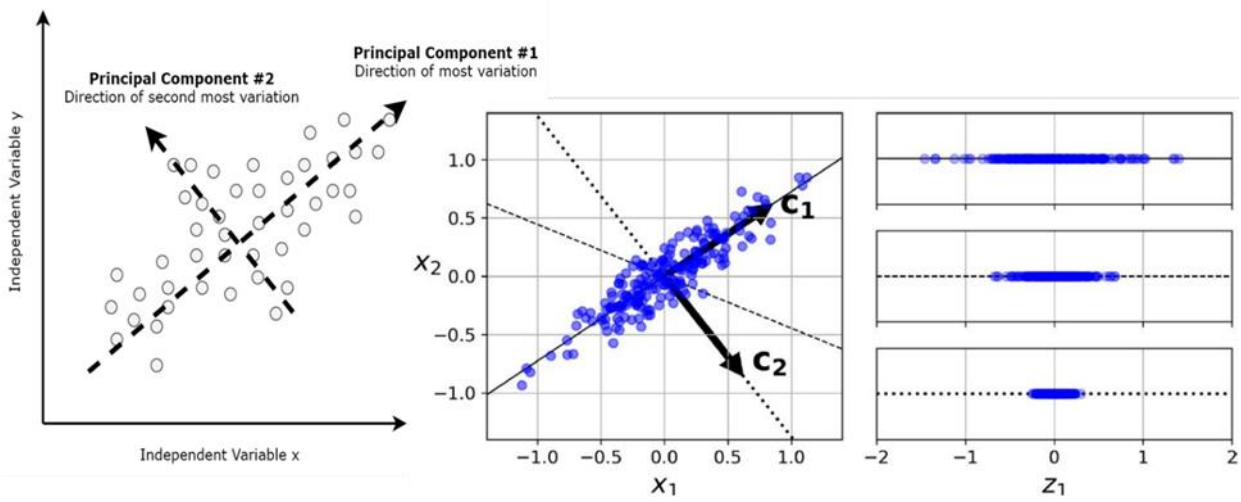


Fig.3.8 An example of formations of Principal Components, (Source: [24]).

3.4 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density- based clustering algorithm. The algorithm basically identifies core point, boundary point, noise point/outlier. The DBSCAN method counts how many points are situated close to it, or within an epsilon-sized distance from it. A point is regarded as a core point if there are at least minimal sampling instances of it in the surrounding area. The cluster to which all instances in a core point's neighbourhood belong to the same cluster. Any instance that is neither a core point nor has one nearby is regarded as a noise point. The outlier is finally eliminated. To determine the best accuracy, the epsilon and minimum sample values were constantly changed.

The steps for performing DBSCAN are shown below (Algorithm 2)

Algorithm 3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

- Randomly select a point from the dataset that has not yet been visited.
- Find all points in the dataset that are within a distance of epsilon from the selected point.
- If the number of nearby points is greater than or equal to min_points, a new cluster is formed, and all nearby points are assigned to the cluster.
- If the number of nearby points is less than min_points, the point is labeled as noise and is not assigned to any cluster.
- Repeat the process for all unvisited points until all points have been visited.

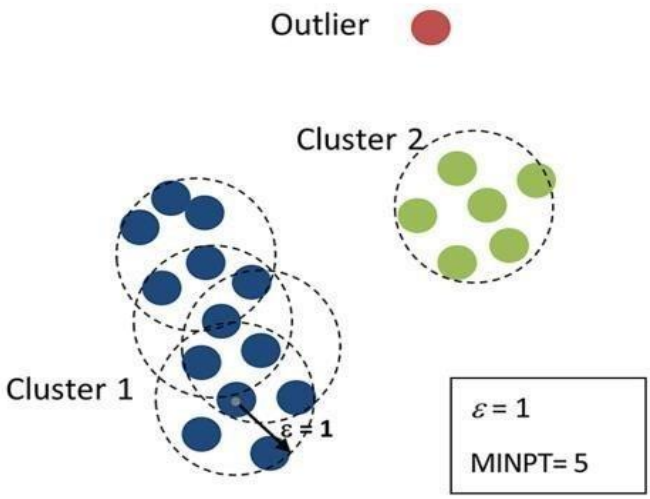


Fig.3.9 An example of DBSCAN algorithm (Source: [24]).

3.5 Image Segmentation

The method in which each pixel of an image is given a label using the image segmentation, and pixels with the same label share certain properties [7]. A series of sections that together make up the full image, or a set of features that are retrieved from the image, are the results of image segmentation. Each pixel in a given area has the same computed or defined attributes, such as texture, density, or colour. In relation to the same property, neighbouring communities can differ significantly.

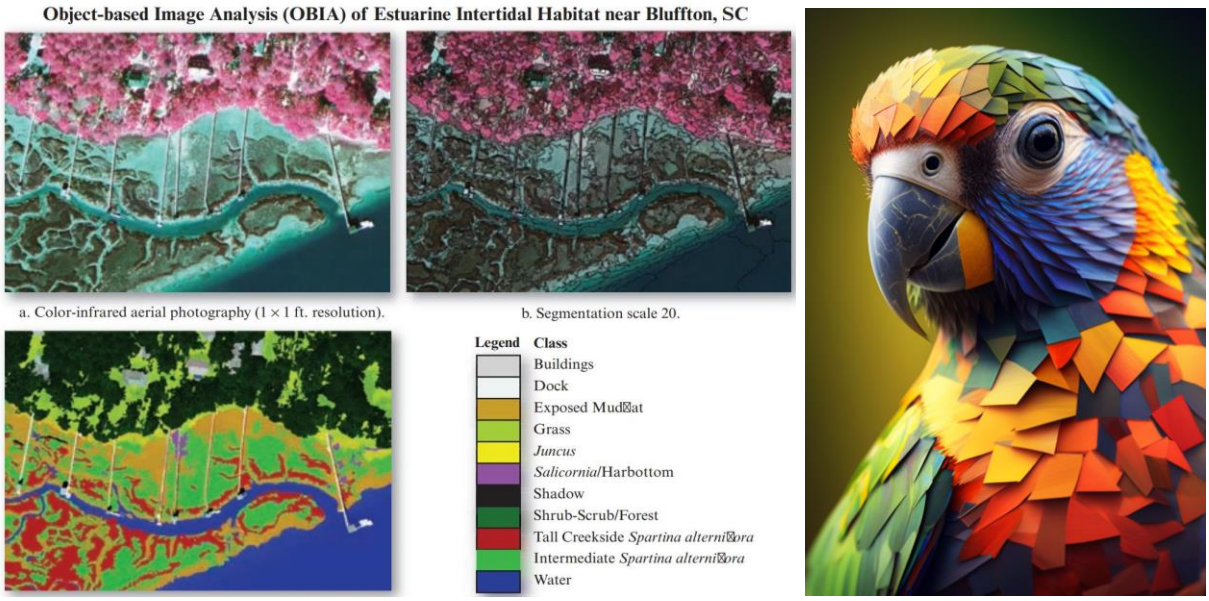


Fig.3.10. Image Segmentation (Source: [23]).

3.5.1 Simple Linear Iterative Clustering (SLIC)

The core idea of SLIC segmentation is based on the similarity clustering of pixel space position and pixel colour feature to achieve a segmented image. Firstly, segments are created in SLIC and then, gathered together by defining Gini index of each segment. Further, according to increasing gini index, each segment is arranged, as the segment having least gini value shows more information. Likely, best segments are selected and rest of them are accumulated with the selected ones by obtaining mean of individual segments and calculating distances between all of them, respectively.

It has a different distance measurement which enables compactness and regularity in the superpixel shapes. The CIELAB space is three-dimensional and covers the entire gamut (range) of human colour perception.

SLIC takes a desired number of nearly equally-sized superpixels K as input. Each superpixels will have N/K pixels.

Fig.3.11 An example of SLIC (Source: [25]).

The three coordinates of CIELAB represents:

- L* = the lightness of the colour (L*= 0 yields black and L* = 100 indicates diffuse white),
- a* = its position between red and green (where negative indicate green; positive indicate red),
- b* = its position between yellow and blue (where negative indicate blue; positive indicates yellow).

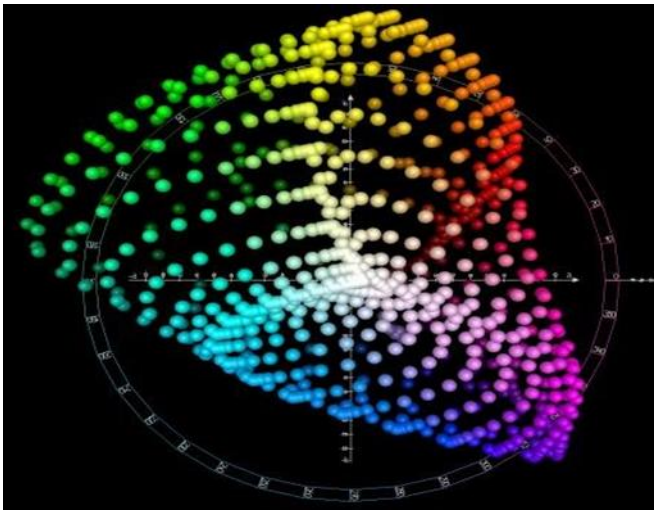


Fig.3.12 Top view of CIELAB Colourspace

The steps for performing SLIC are shown below (Algorithm 4):

Algorithm 4 Simple Linear Iterative Clustering (SLIC)

- Apply PCA on input dataset and create segmented image using SLIC.
- Out of n segments created, find best 16 (Salinas) segments by arranging all segments in ascending order of 'Gini Index'.
- Calculate Euclidian distances between remaining segments to each of best 16 segments and assign that segment to nearest best segment.
- Now take mean of all pixel under a segment.

3.6. Gini Impurity

Gini impurity is a measure of how often a randomly chosen element from a set would be incorrectly labelled if it were randomly labeled according to the distribution of labels in the subset. Gini Impurity is used to recognize significant segments by arranging all segments in ascending order, as here our assumption is that the lower Gini Impurity implies that for a segment most of the pixel values are similar. Further, Gini Impurity is utilized to identify the top bands in each cluster by arranging bands with greater Gini Impurity in descending order as here we assume that high value of Gini Impurity means more variability in the bands which translates to more information content.

Formally, GI is formulated in equation as:

$$\text{Gini(g)} = \sum_{i=1}^{J} p_i(1-p_i)$$

where,

p is the i^{th} pixel of cluster and J is the total number pixels in a cluster.

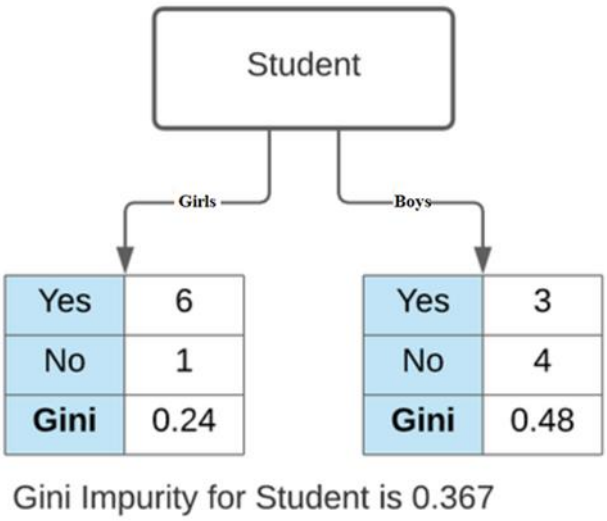
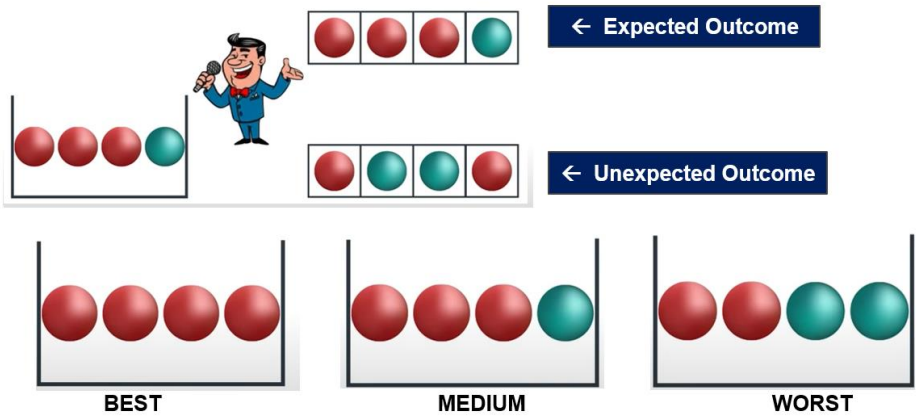


Fig.3.13 An example of Gini Impurity

3.7. Shannon Impurity

In essence, Shannon entropy is a measurement of the degree of uncertainty surrounding a random variable. Shannon entropy, in particular, measures the anticipated value of the data in a message.



log(ab) = log(a) + log(b)

	P(red)	P(blue)	P(winning)	-log2(P(winning))	Entropy
	1	0	1×1×1×1 = 1	0+0+0+0	0
	0.75	0.25	0.75×0.75×0.75×0.25 = 0.105	0.415+0.415+0.415+2	0.81
	0.5	0.5	0.5×0.5×0.5×0.5 = 0.0625	1+1+1+1	1

Fig 3.14 . Illustration of Shannon Entropy (Source: youtu.be/9r7FIXEAGvs).

Consider a dataset X that contains samples from k classes. The probability of samples belonging to class i at a given node can be denoted as pi.

Then the Shannon Entropy is defined as:

$$\text{Shannon Entropy(p)} = -\sum_{i=1}^{J} p_i \log_2(p_i)$$

where, p_{i-} is the pixel of i^{th} cluster and J is the total number pixels in a cluster.

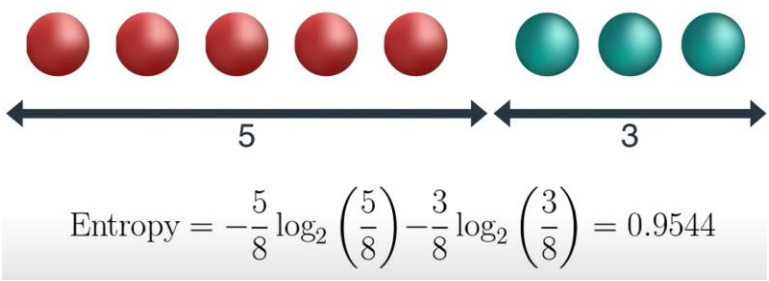


Fig 3.15. Illustration of Shannon Entropy

3.8 Local Band Selection

The number of images (bands) in hyperspectral data gives it a very high dimensionality [18]. It's crucial to acquire the bands in a way that retains as much information as is feasible as an outcome. The band selection approach uses some informative measurements like entropy to determine the relative relevance of each spectral band. It then uses a ranking criterion to choose the top-ranked bands from a sorted sequence of more relevant and fewer redundant bands [7]

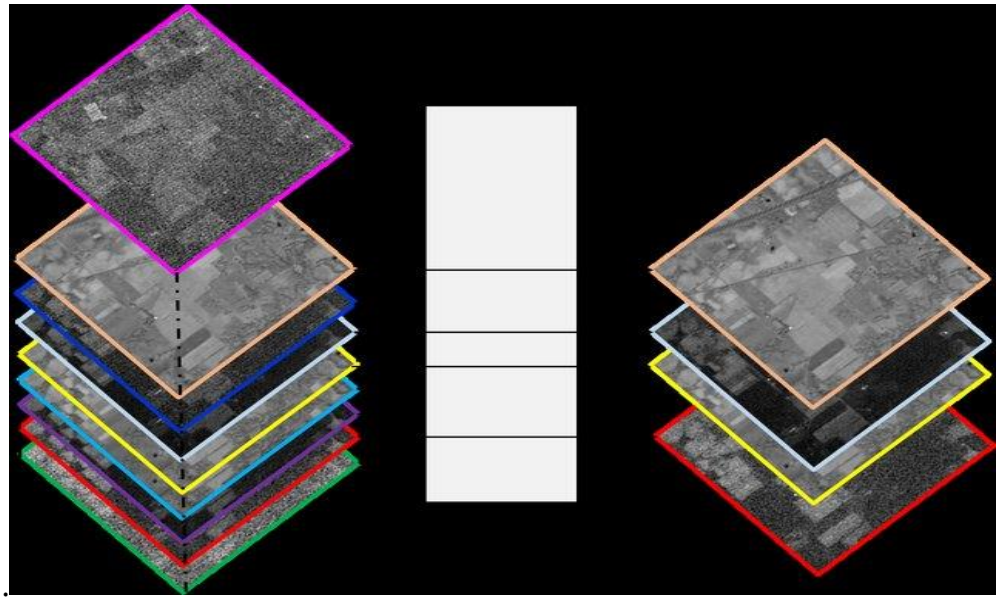


Fig 3.16. Local Band Selection

(Source: Researchgate publication_Hyperspectral-image-dimensionality-reduction_W640)

3.9 Accuracy Assessment

The clustering evaluation is done using the functions Normalized Mutual Information (NMI) and purity, respectively [6]. The degree to which clusters comprise a single class is referred to as purity. It is a real number between [0, 1] in the range. The performance of clustering will improve as purity increases.

$$Purity(\Omega, \mathbb{C}) = \frac{1}{N}\sum_{k}\max_{j}\left|\omega_k \cap c_j\right| \tag{3.1}$$

where,

$\Omega = \{\omega_1, \omega_2, \omega_3, ..., \omega_k\}$ is the set of clusters,

$\mathbb{C} = \{c_1, c_2, c_3, ..., c_j\}$ is the set of classes and

N is the total number of pixels.

The function known as normalised mutual information assesses how well the two assignments agree with one another. The NMI ranges from 0 to 1. NMI = 1 if clustering perfectly recreates the class and NMI = 0 if the clustering is random with respect to class membership.

Let C be the set of classes that are obtained from the ground reference information and **Ω** is the set of the clusters obtained from the algorithm. Their mutual information, MI (**Ω**, C) can be obtained as follows:

$$MI(\Omega,\mathbb{C}) = \sum_{k}\sum_{j} p(\omega_k \cap c_j)\log_2 \frac{p(\omega_k \cap c_j)}{p(\omega_k)\,p(c_j)} \qquad (3.2)$$

where p(ω_k), p (c_j) and p($\omega_k \cap c_j$) are the probabilities of an arbitrarily selected pixel belonging to cluster ω_k , class c_j and cluster ωk as well as class c_j at the same time, respectively. Then NMI can be obtained

$$NMI(\Omega,\mathbb{C}) = \frac{MI(\Omega,\mathbb{C})}{\max(H(\Omega),H(\mathbb{C}))} \qquad (3.3)$$

where,

$H(\Omega) = -\sum_{k} p(\omega_k)\log_2 p(\omega_k)$ and $H(C) = -\sum_{j} p(c_j)\log_2 p(\omega_j)$ are the entropies of Ω and C, respectively.

The functions Adjusted Mutual Information (AMI) and Overall Accuracy, respectively, are used to evaluate the clustering [6]. For comparing clustering in this context, adjusted mutual information, a kind of mutual information, may be employed. When the two partitions are identical, the AMI is 1, and when the MI between the two partitions is equal to the value that would be anticipated by chance alone, it is 0.

When the two partitions have more clusters (with a fixed number of set elements N), the baseline value of mutual information between them tends to be higher and does not take on a constant value. By adopting a hypergeometric model of randomness, it can be shown that the expected mutual information between two random clustering is:

Let C be the set of classes produced from the ground reference data, and let **Ω** be the set of clusters derived by the algorithm.

$$\mathrm{EMI}(\Omega,\mathrm{C}) = \sum_{k=1}\sum_{j=1}\sum_{n_{kj}=(a_k+b_j-N)^+}^{\min(a_k,b_j)} \frac{n_{kj}}{N}\log\left(\frac{N.n_{kj}}{a_k b_j}\right) \times \frac{a_k!\,b_j!\,(N-a_k)!\,(N-b_j)!}{N!\,n_{kj}!\,(a_k-n_{kj})!\,(b_j-n_{kj})!\,(N-a_k-b_j+n_{kj})!}$$

where, $(a_k + b_j - N)^+$ denotes max(0, ak + bj - N). The variables a_k and b_j are partial sums of the contingency table;

that is,

$$a_k = \sum_{j=1} n_{kj}$$

and

$$b_j = \sum_{k=1} n_{kj}$$

The adjusted measure for the mutual information may then be defined to be:

$$\mathrm{AMI}(\Omega, \mathrm{C}) = \frac{\mathrm{MI}(\Omega, \mathrm{C}) - \mathrm{EMI}(\Omega, \mathrm{C})\}}{\max\{\mathrm{H}(\Omega), \mathrm{H}(\mathrm{C})\} - \mathrm{EMI}(\Omega, \mathrm{C})} \tag{3.4}$$

$\mathrm{H}(\Omega) = -\sum_k p(\omega_k) \log_2 p(\omega_k)$ and $\mathrm{H}(\mathrm{C}) = -\sum_j p(c_j) \log_2 p(\omega_j)$ are the entropies of Ω and C, respectively.

While, the Overall Accuracy (OA), on the other hand, is determined by a contingency matrix that reports the intersection cardinality for every true/predicted cluster pair. When the samples are independent and have an identical distribution, it gives enough data for all clustering metrics, eliminating the need to take into consideration instances that are not clustered in all cases. OA presents its results in percentage.

Chapter 4 Methodology

Three methodologies are proposed in this research. The description of first proposed methodology is followed by deploying clustering on hyperspectral image initially by utilizing *k*-means with feature reduction. Secondly, performing image segmentation with *k*-means on the reduced feature dataset and lastly, approaching towards *k*-means clustering. While, methodology consists of PCA technique for feature reduction and SLIC technique for image segmentation method and finally, clustering approach which utilizes Euclidean distance by calculating mean of pixels under a segment. One important data analysis method that is frequently used for numerous empirical applications in new areas is cluster analysis [4]. The process of identifying groups of things is called clustering.

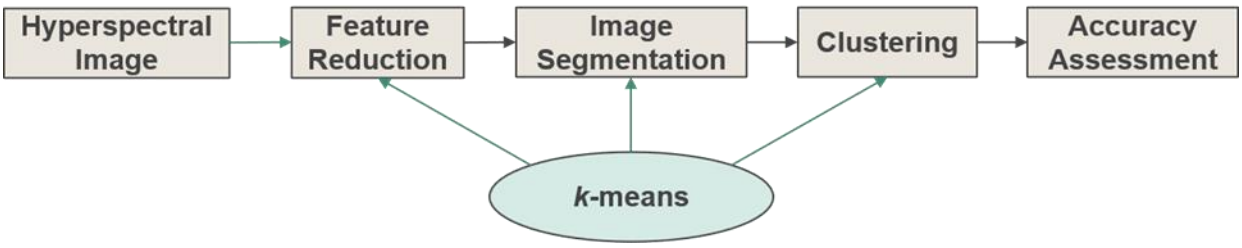


Fig 4.1. Proposed Methodology 1

Furthermore, in second proposed methodology Image segmentation is performed by *k*-means followed by Local Band Selection and Segment merging by using Gini impurity as a ranking criterion. Gini Impurity is used to recognize significant segments by arranging all segments in ascending order, as here our assumption is that the lower Gini Impurity implies that for a segment most of the pixel values are similar. Further, Gini Impurity is utilized to identify the top bands in each cluster by arranging bands with greater Gini Impurity in descending order as here we assume that high value of Gini Impurity means more variability in the bands which translates to more information content.

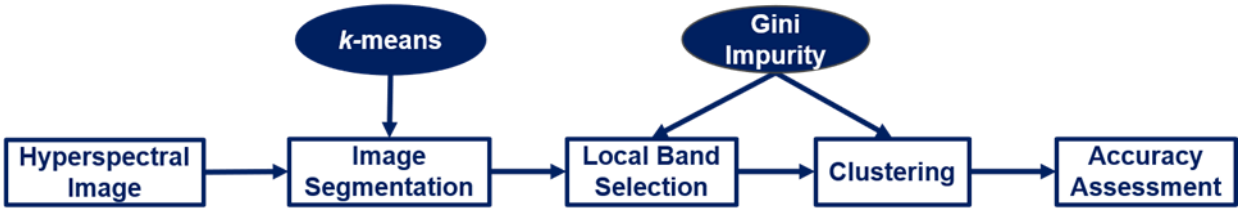


Fig.4.2. Proposed Methodology 2

Meanwhile in third methodology, Image segmentation is performed by k-means followed by Local Band Selection and Segment merging by using Shannon Entropy as a ranking criterion. Shannon entropy is employed which basically is a measure of the uncertainty associated with a random variable. Specifically, Shannon entropy quantifies the expected value of the information contained in a message.

In this study, Shannon entropy is employed to identify significant clusters by organising all segments in ascending order. This is because our hypothesis is that the lower the entropy, the more identical the pixel values are for a cluster. Additionally, bands with higher entropy values are arranged in descending order to identify the top bands in each cluster. This is done because we believe that higher entropy values indicate more band variability, which correlates to higher information content.

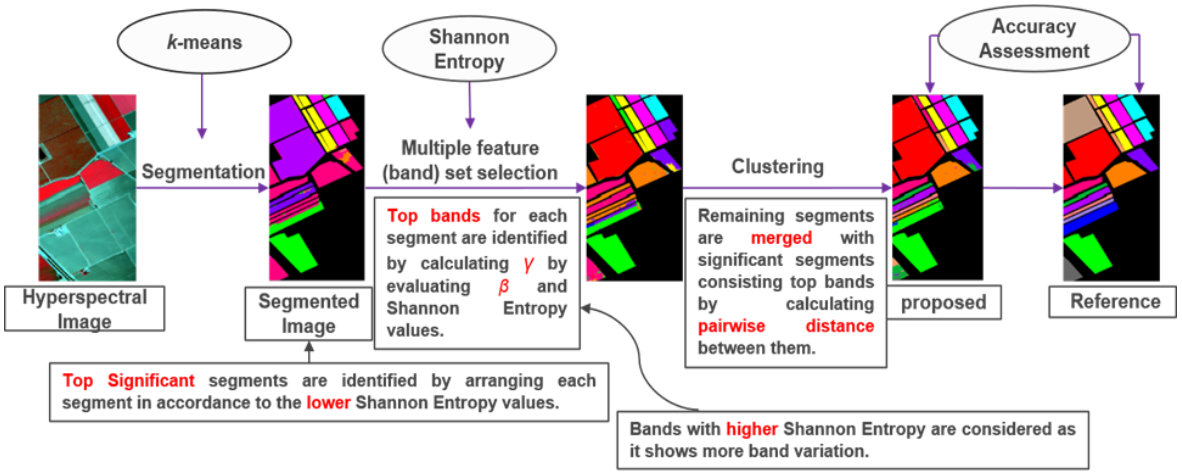


Fig 4.3. Proposed Methodology 3

4.1. Image Segmentation using *k*-means (Proposed 1)

The steps for performing image segmentation are shown as, by organizing the dataset in a matrix and applying *k* -means. Further, defining labels to each cluster and obtaining cluster map. Lastly, converting the cluster map into segmentation map using connected components labelling procedure.

The steps for image segmentation are shown below (Algorithm 3)

Algorithm 5 Image Segmentation using *k*-means

- Input dataset of Salinas.
- For feature reduction, perform *k*-means clustering on bands (features).
- Now to create segments on reduced dataset perform *k*-means clustering.
- Obtain cluster map.
- Convert the cluster map into segmentation map using connected component.
- Apply *k*-means clustering.

Cluster Map					
1	1	2	2	2	2
1	1	2	2	2	2
1	1	2	2	5	5
4	4	2	2	5	5
4	4	2	2	1	1
4	4	2	2	1	1

Segmentation Map					
1	1	2	2	2	2
1	1	2	2	2	2
1	1	2	2	5	5
4	4	2	2	5	5
4	4	2	2	6	6
4	4	2	2	6	6

Fig 4.4. Converting Cluster map to Segmentation map using Connected Components

4.2 Feature Selection using Gini Impurity (Proposed 2)

A hyperspectral image has a very high spectral resolution due to the number of pixels and bands it contains [1]. As a result, it's important to arrange number of bands in a dataset without drastically reducing the amount of information maintained. The band selection method measures the importance of each spectral band according to some informative measurements such as gini impurity which uses a ranking criterion to select the top-ranked bands in a sorted sequence containing relevancy and less redundancy [9]. Moreover, redundancy is produced by bands with comparable gini indices. So, a distance-weighted parameter score is introduced to prevent this. The proposed band selection strategy is inspired by the methodology discussed in [10] and [17].

Formally, score is formulated in equation as:

$$delta = \max_{GI_j < GI_i} d \tag{4.1}$$

$$score = (delta) * (GI_x) \tag{4.2}$$

where, delta is a function that store maximum distance for a test band from all bands whose gini index is higher than that of test band, d is distance between mean vectors of all pixels in a cluster for particular segment to remaining bands of the segment, GI is the gini index for corresponding bands, i and j are band indexes.

The algorithm for proposed methodology is depicted below:

Algorithm 6 Segmentation with local band selection

Input: Hyperspectral Image.

Output: Clustered Image.

- Segmentation using k-means clustering. Convert cluster map to segmentation map. It will contain number of segments (group of pixels).
- Now, each segment is considered as a cluster.
- Arrange all clusters in ascending order of their gini impurity.
- Choose top segments which contain number of pixels greater than five as we assume cluster less than five do not have much informative content.
- Similarly for each segment compute gini impurity and delta for all bands. A higher gini impurity signifies variability in a band, hence more significant.
- Calculate delta by detecting bands whose gini is greater than j th band and by finding max distance among band whose gini is greater than j th band. Evaluate score and arrange segments in descending order of score parameter.
- Now assess the pairwise distance between top significant and remaining segments considering pixel array for significant bands of significant clusters.
- Propagate labels to clusters from significant clusters according to minimum distance.
- Now rearrange these segments to form a cluster map.

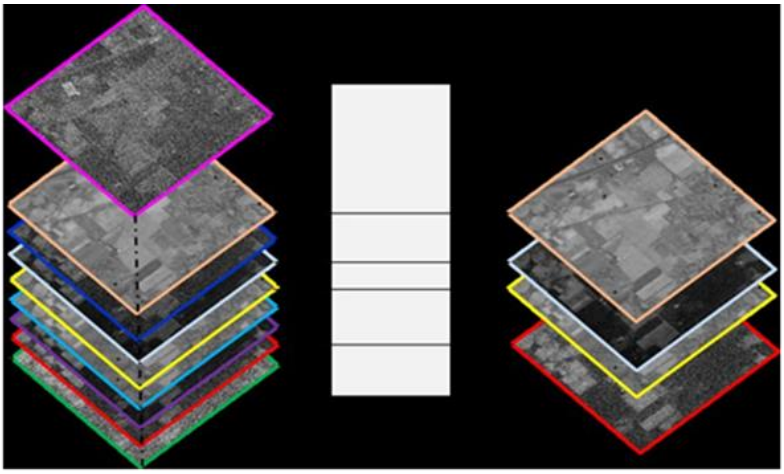
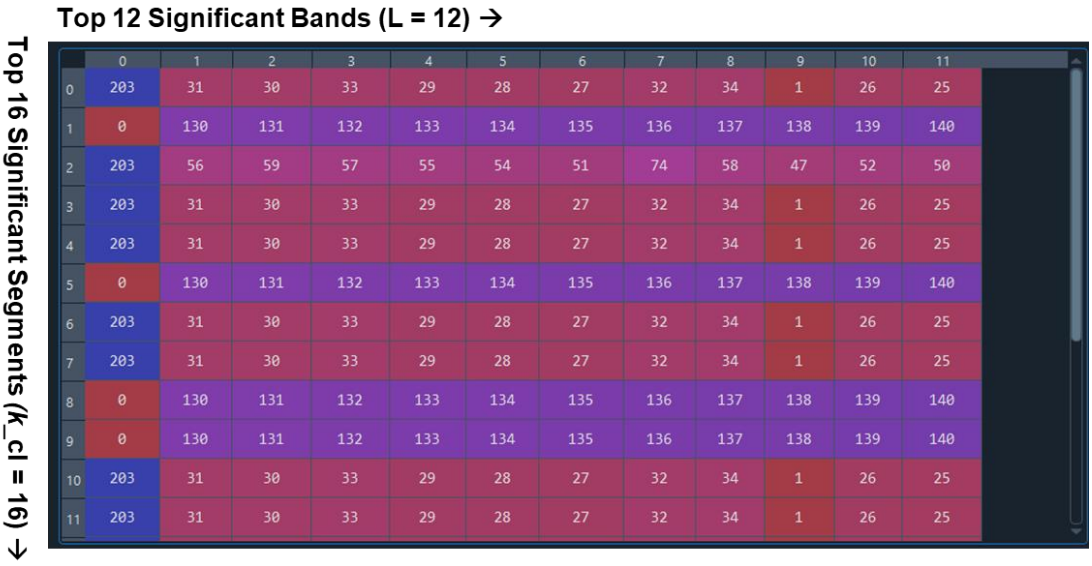


Fig 4.5. Local Feature Selection

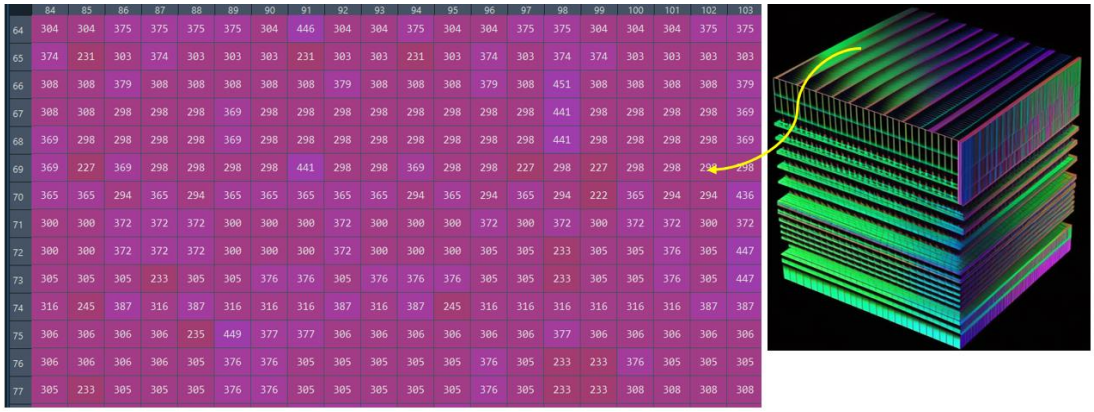
Bands and their Shannon Entropy→

	195	196	197	198	199	200	201	202	203
1	0.021692	0.0230723	0.0335522	0.0219146	0.0144928	0.020979	0.0384615	0.0880503	0.0229885
2	0.00789323	0.00562622	0.00534442	0.0134615	0.00415007	0.0141213	0.0131356	0.0331633	0.0212264
3	8.20811e-06	8.94757e-06	1.29875e-05	1.45632e-05	1.02886e-05	1.32118e-05	2.68666e-05	5.76632e-05	6.85619e-05
4	0.000737585	0.00070852	0.000668785	0.000666158	0.00067554	0.000713212	0.000872111	0.000294318	0.000585651



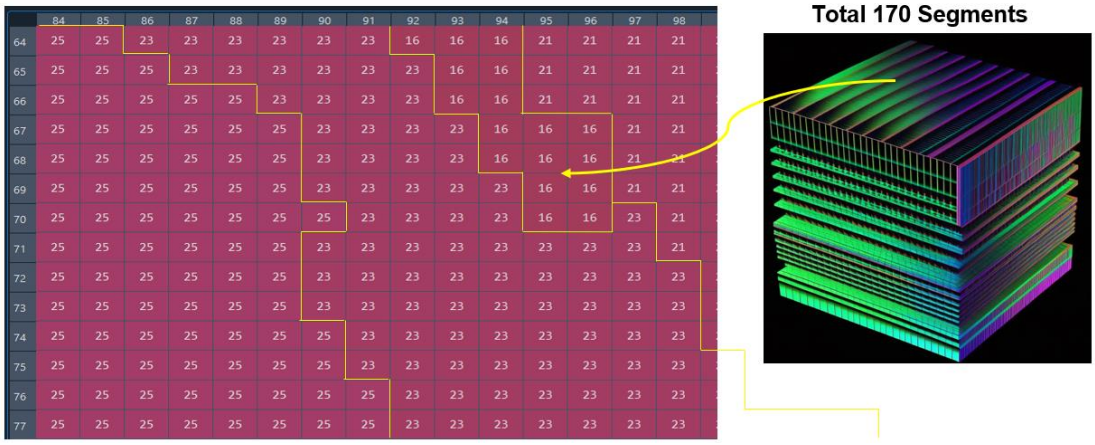
	0	1	2	3	4	5	6	7	8	9	10	11
0	203	31	30	33	29	28	27	32	34	1	26	25
1	0	130	131	132	133	134	135	136	137	138	139	140
2	203	56	59	57	55	54	51	74	58	47	52	50
3	203	31	30	33	29	28	27	32	34	1	26	25
4	203	31	30	33	29	28	27	32	34	1	26	25
5	0	130	131	132	133	134	135	136	137	138	139	140
6	203	31	30	33	29	28	27	32	34	1	26	25
7	203	31	30	33	29	28	27	32	34	1	26	25
8	0	130	131	132	133	134	135	136	137	138	139	140
9	0	130	131	132	133	134	135	136	137	138	139	140
10	203	31	30	33	29	28	27	32	34	1	26	25
11	203	31	30	33	29	28	27	32	34	1	26	25

Fig 4.6. An example of Local Feature Selection



	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
64	304	304	375	375	375	375	304	446	304	304	375	304	304	375	375	304	304	304	375	375
65	374	231	303	374	303	303	303	231	303	303	231	303	374	303	374	374	303	303	303	303
66	308	308	379	308	308	308	308	308	379	308	308	308	379	308	451	308	308	308	308	379
67	308	308	298	298	298	369	298	298	298	298	298	298	298	298	441	298	298	298	298	369
68	369	298	298	298	298	369	298	298	298	298	298	298	298	298	441	298	298	298	298	369
69	369	227	369	298	298	298	298	441	298	298	369	298	298	227	298	227	298	298	298	298
70	365	365	294	365	294	365	365	365	365	365	294	365	294	365	294	222	365	294	294	436
71	300	300	372	372	372	300	300	300	372	300	300	300	372	300	372	300	372	372	300	372
72	300	300	372	372	372	300	300	300	372	300	300	300	305	305	233	305	305	376	305	447
73	305	305	305	233	305	305	376	376	305	376	376	376	305	305	233	305	305	376	305	447
74	316	245	387	316	387	316	316	316	387	316	387	245	316	316	316	316	316	316	387	387
75	306	306	306	306	235	449	377	377	306	306	306	306	306	306	377	306	306	306	306	306
76	306	306	306	306	305	376	376	305	305	305	305	305	376	305	233	233	376	305	305	305
77	305	233	305	305	305	376	376	305	305	305	305	305	376	305	233	233	308	308	308	308

Fig 4.7. A small part of input image



	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98
64	25	25	23	23	23	23	23	23	16	16	16	21	21	21	21
65	25	25	25	23	23	23	23	23	23	16	16	21	21	21	21
66	25	25	25	25	25	23	23	23	23	16	16	21	21	21	21
67	25	25	25	25	25	25	23	23	23	23	16	16	16	21	21
68	25	25	25	25	25	25	23	23	23	23	16	16	16	21	21
69	25	25	25	25	25	25	23	23	23	23	23	16	16	21	21
70	25	25	25	25	25	25	25	23	23	23	23	16	16	23	21
71	25	25	25	25	25	25	23	23	23	23	23	23	23	23	21
72	25	25	25	25	25	25	23	23	23	23	23	23	23	23	23
73	25	25	25	25	25	25	23	23	23	23	23	23	23	23	23
74	25	25	25	25	25	25	25	23	23	23	23	23	23	23	23
75	25	25	25	25	25	25	25	23	23	23	23	23	23	23	23
76	25	25	25	25	25	25	25	25	23	23	23	23	23	23	23
77	25	25	25	25	25	25	25	25	23	23	23	23	23	23	23

Fig 4.8. A small part of input image after segmentation.

Pixel No.	Pixels Location		Pixel Values in different bands					
	x	y	1	2	3	4	5	6
1	64	92	304	308	306	312	298	305
2	64	93	306	307	303	309	299	305
3	64	94	312	317	315	308	313	308
4	65	93	298	296	299	294	302	300
5	65	94	295	300	295	308	299	296
6	66	93	408	412	409	405	407	406
7	66	94	350	356	354	348	352	349
8	67	94	360	360	362	369	358	365
9	67	95	298	294	296	289	292	293
10	68	94	320	322	322	325	324	320

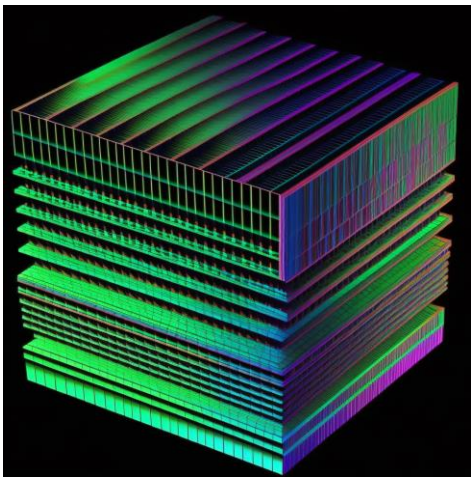


Fig 4.9. Sample Matrix of a segment containing pixel's location and corresponding band value in different bands.

4.3 Multiple feature (band) set selection (Proposed 3)

The number of images (bands) in hyperspectral data gives it a very high dimensionality [18]. It's crucial to acquire the bands in a way that retains as much information as is feasible as an outcome.

The band selection approach uses some informative measurements like entropy to determine the relative relevance of each spectral band. It then uses a ranking criterion to choose the top-ranked bands from a sorted sequence of more relevant and fewer redundant bands [7].

Moreover, bands with similar entropy values result in redundancy. In order to avoid this, a distance-weighted parameter score (β) is introduced.

Formally, score (β) is formulated in equation as:

$$\beta = \max_{ET_j < ET_i} d \tag{4.3}$$

$$\gamma = (\beta) * (EI_x) \tag{4.4}$$

where,

β is a function that store maximum distance for a test band from all bands whose entropy is higher than that of test band, d is distance between bands in a cluster to remaining bands of the same cluster. ET is the entropy for corresponding bands, i and j are band indexes.

In this research, ET is used to determine the relevant bands while β is used to decrease redundancy and γ comprises both relevance and redundancy criteria. The following algorithm shows the steps for the proposed methodology.

The algorithm for proposed methodology is depicted below:

Algorithm 7 Multiple feature (band) set selection

Input: Hyperspectral Image.

Output: Clustered Image.

- Using k-means clustering for segmentation. Generate the segmentation map from the cluster map. It will have multiple segments in it (group of pixels)
- Every segment is now regarded as a cluster. Sort every cluster according to increasing entropy.
- Select the top segments that have more pixels than five because we assume that clusters with fewer than five pixels do not contain much information
- Calculate the shannon entropy and β for all bands for each segment in a same manner. Higher entropy indicates more band variation, making it more significant.
- Find the maximum distance between bands whose entropy is greater than the j th band and use that information to calculate the β.
- Compute the γ and arrange the segments according to the descending order of γ values.
- Evaluate the pairwise distance between the most significant and remaining segments while taking the bands that have been found for each significant cluster into account.
- Labels should be distributed among clusters based on the shortest distance from significant clusters.
- Assemble these segments into a cluster map now.

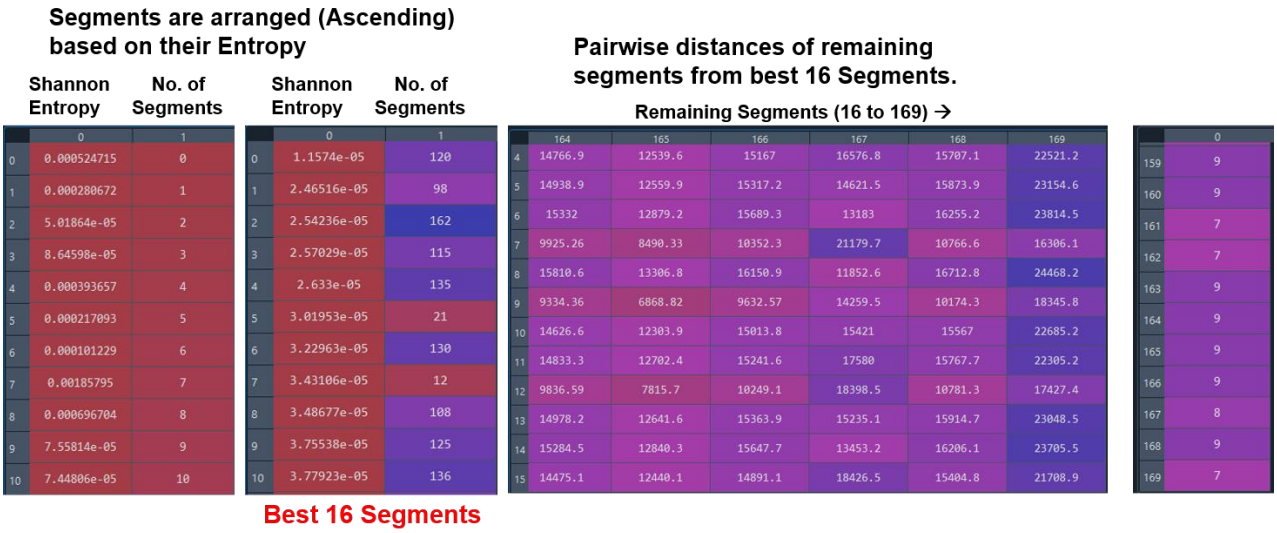


Fig 4.10. Segment merging based on Shannon Entropy.

Chapter 5 Results

For first proposed methodology, hyperspectral image classification is carried out using *k*-means with the help of feature reduction and image segmentation. Fig. 5.1 shows different values of *k* that are initialised while performing *k*-means clustering for Salinas Dataset. While, *k*_fr represents number of features in bands and *k*_seg shows number of regions.

Optimum number of clusters for Salinas was found to be 16. For feature reduction, a set of values of *k*_fr are taken. Starting with *k*_fr = 2 to *k*_fr = 20 multiple values of corresponding NMI are generated and maximum NMI value is marked. As there is no significant change in values of NMI so any value in above range can be considered.

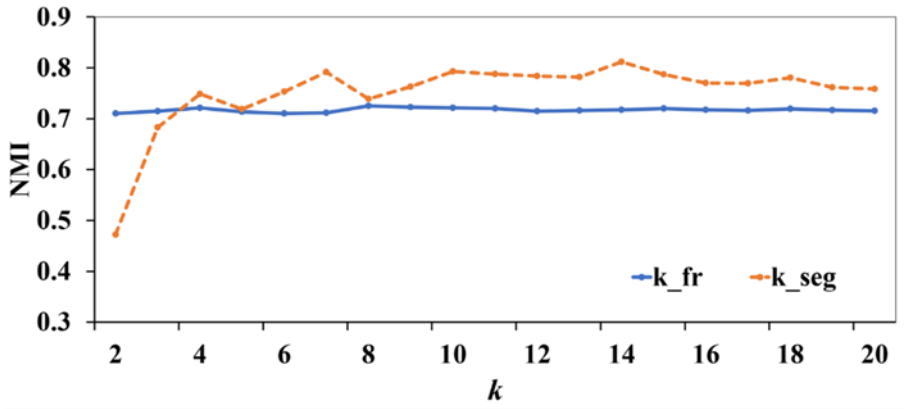


Fig.5.1. Optimum value of NMI for Salinas using feature reduction and image segmentation.

Similarly, for image segmentation a set of values of *k*_seg are taken. Initiating with *k*_seg = 2 to *k*_seg = 20 corresponding values of NMI are plotted and maximum NMI value is marked (as shown in Fig.5.1). From the optimum values of *k*_fr and *k*_seg NMI is calculated and it's analysed that NMI is significantly improved from 72.42 % to 81.15 %. Using feature reduction and image segmentation accuracy is increased by 01.20 % and 08.61 % respectively.

Furthermore**,** for Salinas dataset, the values for number of segments were altered in the range of 100 to 500, after applying PCA technique. After experimentation, at segment 100, maximum accuracy of 68.7597 % is achieved.

From Table 5.1, it can be observed that first proposed methodology has performed better than the compared clustering methodologies. Further it is noticed that the proposed approach produced greater NMI and Purity scores.

Table 5.1. Accuracy values for Salinas (Proposed 1).

Datasets	**Salinas**	
Methods	**NMI**	**Purity**
k-means	0.7242	0.6734
k-means + PCA	0.7656	0.6785
DBSCAN	0.7228	0.5846
PCA + SLIC + clustering	0.6876	0.5875
Proposed 1 (*k*-means + *k*_fr + *k*_seg)	**0.8115**	**0.6835**

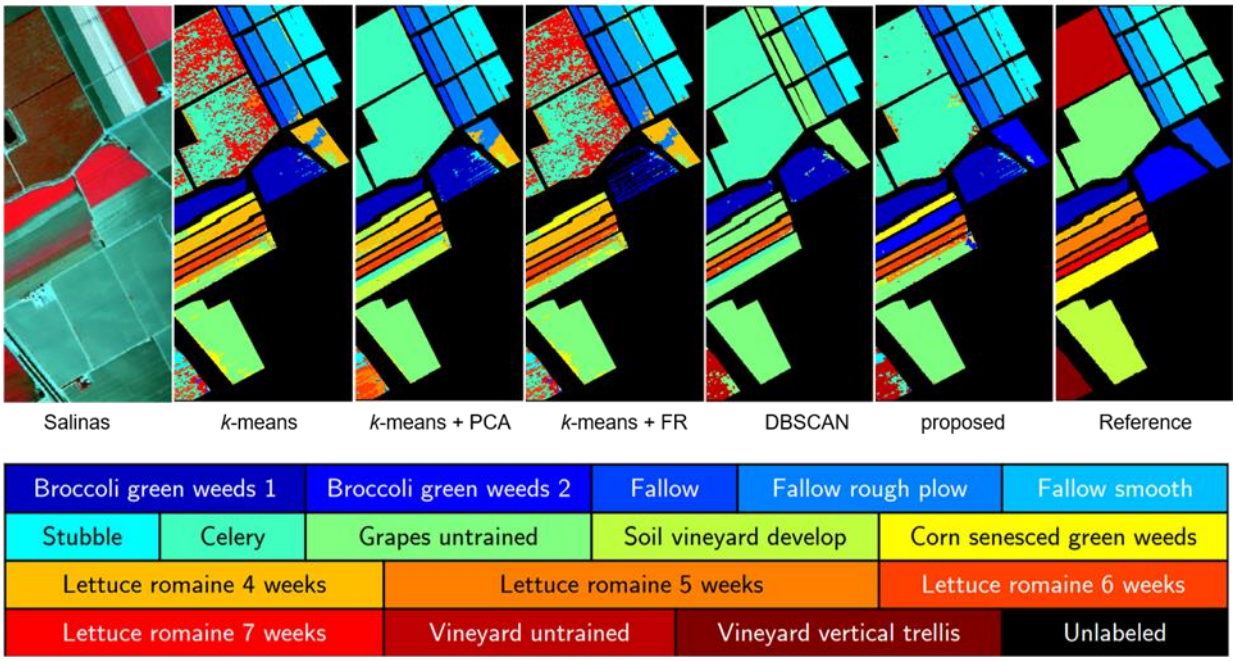


Fig. 5.2. Comparison of proposed methodology 1 with other existing methods.

Table 5.2. Parameters setting for Salinas.

Methods	**Variables**	**Salinas**
k-means	*k*_cl	16
k-means + PCA	*k*_cl	16
	% of Variance Data	99
DBSCAN	Minimum Samples	40
	Epsilon	500
SLIC	No. of Segments	100
	Compactness	10

On the other hand, Table 5.2 shows the parameters setting used during the classification. Fig. 5.2 shows the comparison between the original image and improvised images of Salinas.

In second proposed methodology, hyperspectral image classification is carried out using image segmentation and band selection techniques considering redundancy and relevancy. First, we have compared the result of the proposed method i^{th} clustering techniques. Then, we have carried out following experiments: k_seg, k_seg + LBS_R and proposed.

Table 5.3 shows accuracy values corresponding to various parameters and Table 5.4 illuatrates parameters setting. Nonetheless, apart from proposed method, approaches such as local band selection technique with redundancy is conveyed. Let say that, kM stands for k-means clustering while LBSR portrays band selection technique with redundancy. However, k_seg represents number of segments, k_cl shows number of significant clusters and L demonstrates number of top bands taken into consideration. Moreover, PCA + kM is performed, where the proportion of principal components (PC) are selected such that, PC are able to account for at least 99% of the variance in the dataset. While, in PCA + SLIC + kM, the optimum segments value obtained for Salinas is 50.

Table 5.3. Accuracy values for Salinas (Proposed 2).

Datasets	**Salinas**	
Methods	**NMI**	**Purity**
k-means	0.7242	0.6734
*k*_seg	0.7921	0.6738
*k*_seg + LBS_R	0.7932	0.6426
Proposed 2 (*k*_seg + LBS)	**0.8130**	**0.7011**

Table 5.3 accuracy values obtained for Salinas dataset corresponding to methods such as k-means, k_seg, k_seg + LBSR and the third proposed methodology. Table 5.4 contains different values of k seg that are initialised while performing k-means segmentation based clustering for Salinas Dataset. Optimum number of regions for Salinas was found to be six for proposed, by implementing a set of values of k seg, starting with k_seg = 2 to k _seg = 15, multiple value of corresponding NMI are generated and maximum NMI value is marked. Similarly, for clustering a set of values of k_cl are taken and likewise, for band selection values of L are altered. Initiating with k cl = 10 to k cl = 30 and L was too varied from 10 to 100, corresponding values of NMI are observed and maximum NMI value is considered. From the optimum values of k cl and L, NMI is calculated and it's analysed that NMI is significantly improved from 79.21% to 81.3%. Using band selection technique with inclusion of redundancy removal score on bands, accuracy is increased by 0.14% and 2.57% respectively.

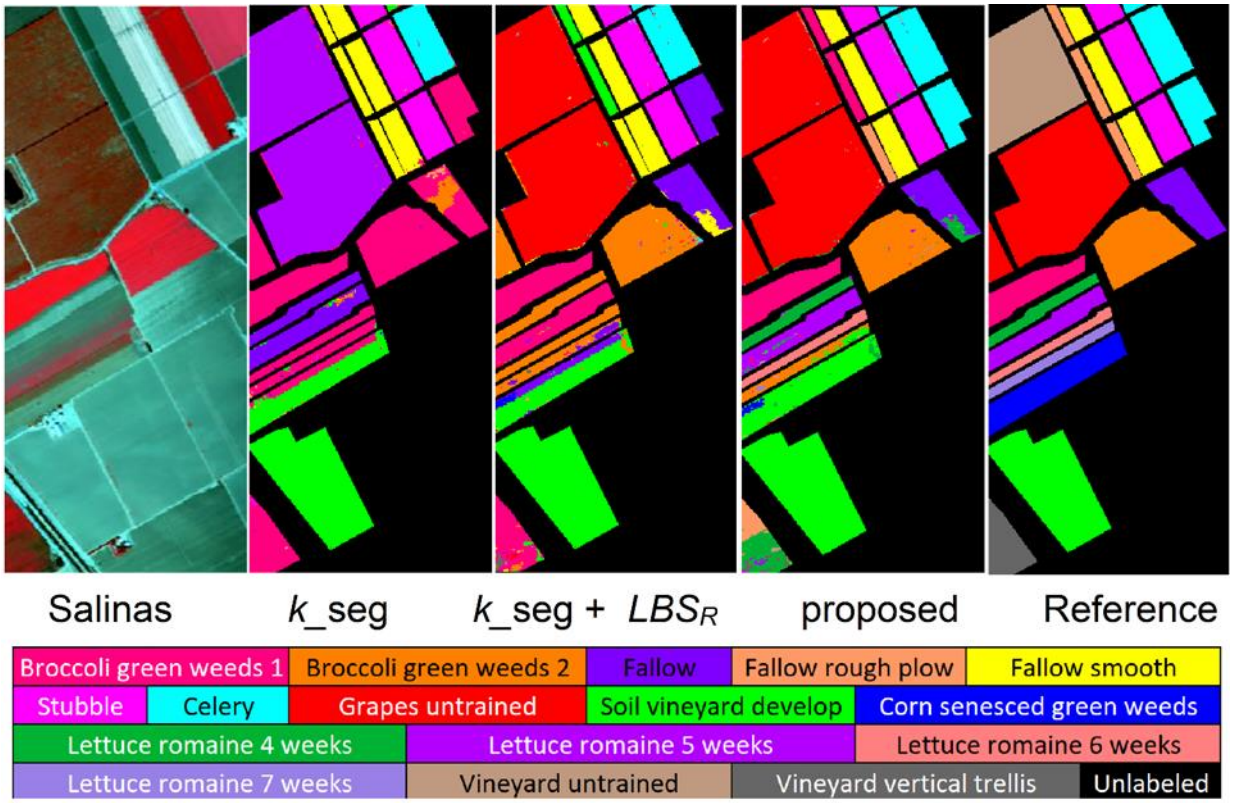


Fig. 5.3. Comparison of proposed methodology 2 with other existing methods.

Table 5.4 Parameters settings for Salinas (Proposed 2).

Variables →	***k_seg***	***k*_cl**	**L**
k-means	-	16	All bands
*k*_seg	5	28	All bands
*k*_seg + LBS_R	7	18	12
Proposed 2 (*k*_seg+ LBS)	**6**	**26**	**12**

On the other hand, Table 5.4 shows the parameters setting used during the classification. Fig. 5.3 shows the comparison between the original image and improvised images of Salinas.

In third methodology, hyperspectral images are classified using image segmentation and band selection approaches while accounting for relevance and redundancy. The results of the proposed methodology were first examined with following clustering strategies: *k*_seg, *k*_seg + LBS_R and Proposed 3 (*k*_seg+ LBS)

However, in addition to the proposed approach, techniques like the multiple feature (band) set selection technique with redundancy are discussed. Let's say *k*_seg denotes for segmentation using *k*-means, LBS_R represents multiple feature (band) set selection technique with redundancy factor not included and that *k*M stands for *k*-means clustering. Nonetheless, *k*_seg denotes the

quantity of segments, *k*_cl denotes the quantity of significant clusters, and L denotes the quantity of top bands taken into account.

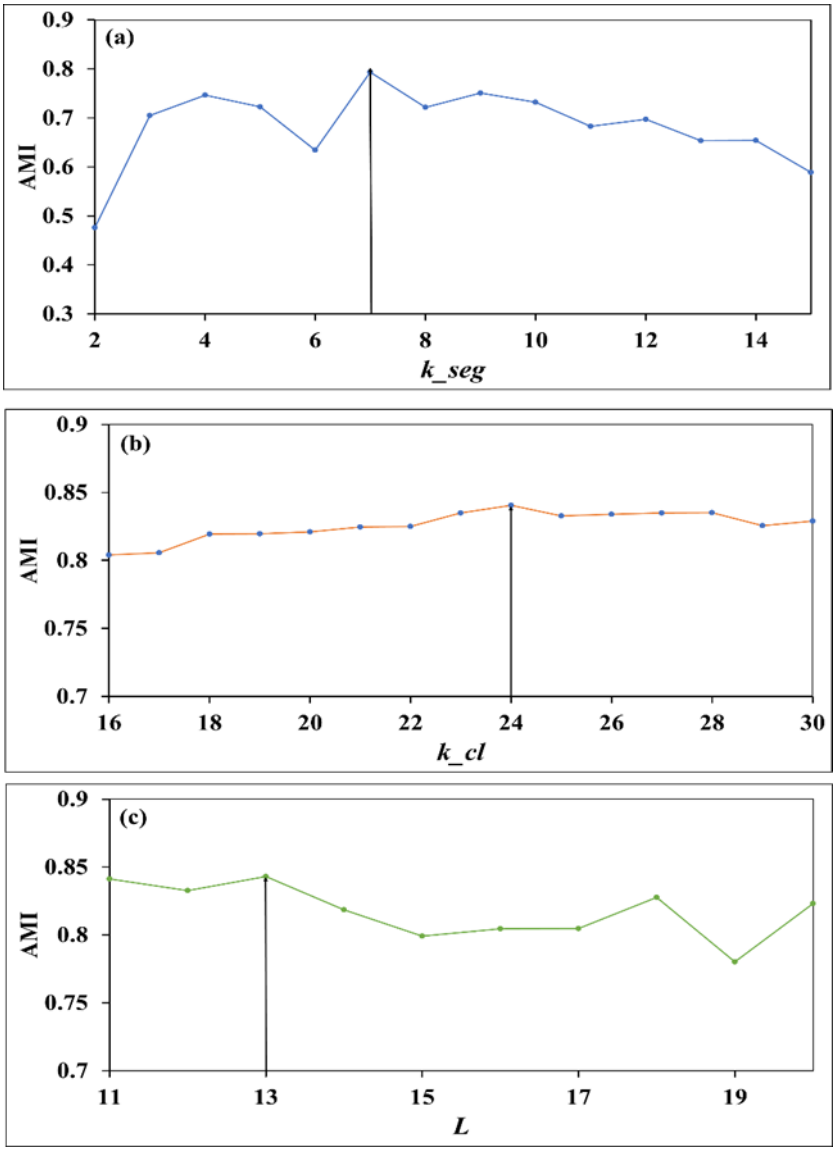


Fig. 5.4. Proposed strategy of Salinas Dataset.

Table 5.5. Accuracy values for Salinas (Proposed 3).

Datasets	**Salinas**	
Methods	**AMI**	**OA(%)**
Seg_*k*	0.7048	67.95
*k*_seg + MFS_R	0.8210	69.05
Proposed 3 (SegETMFS)	**0.8431**	**72.85**

For proposed strategy, many initialised k seg values are implemented in the range of 2 to 15 (as shown in Fig. 2(a)) and after fixing the value of k seg i.e 7, according to the maximum AMI values, further, k cl values were altered in a set of 16 to 30 and generated as 24 (as shown in Fig.

2(b)). Finally, after fixing k seg and k cl values, L was changed from 11 to 20 and was found out to be 13 (as shown in Fig. 2(c)). Thus, the accuracy attained is 84.31%.

Table 5.6. Parameters settings for Salinas (Proposed 3).

Variables →	***k_seg***	***k*_cl**	**L**
k-means	-	16	All bands
*k*_seg	8	16	All bands
*k*_seg + LBS_R	6	25	10
Proposed 3 (*k*_seg+ LBS)	**7**	**24**	**13**

The above Table 5.6 illustrates the parameter settings utilized while performing clustering of Salinas dataset by the aid of proposed methodology and the other methodologies mentioned in the table.

According to the Table 5.5, carrying out similar procedure for other experiments as carried for proposed strategy, it is examined that accuracy is improved by 9.76% and 2.61% respectively, using the multiple feature (band) set selection technique with the redundancy removal score on bands.

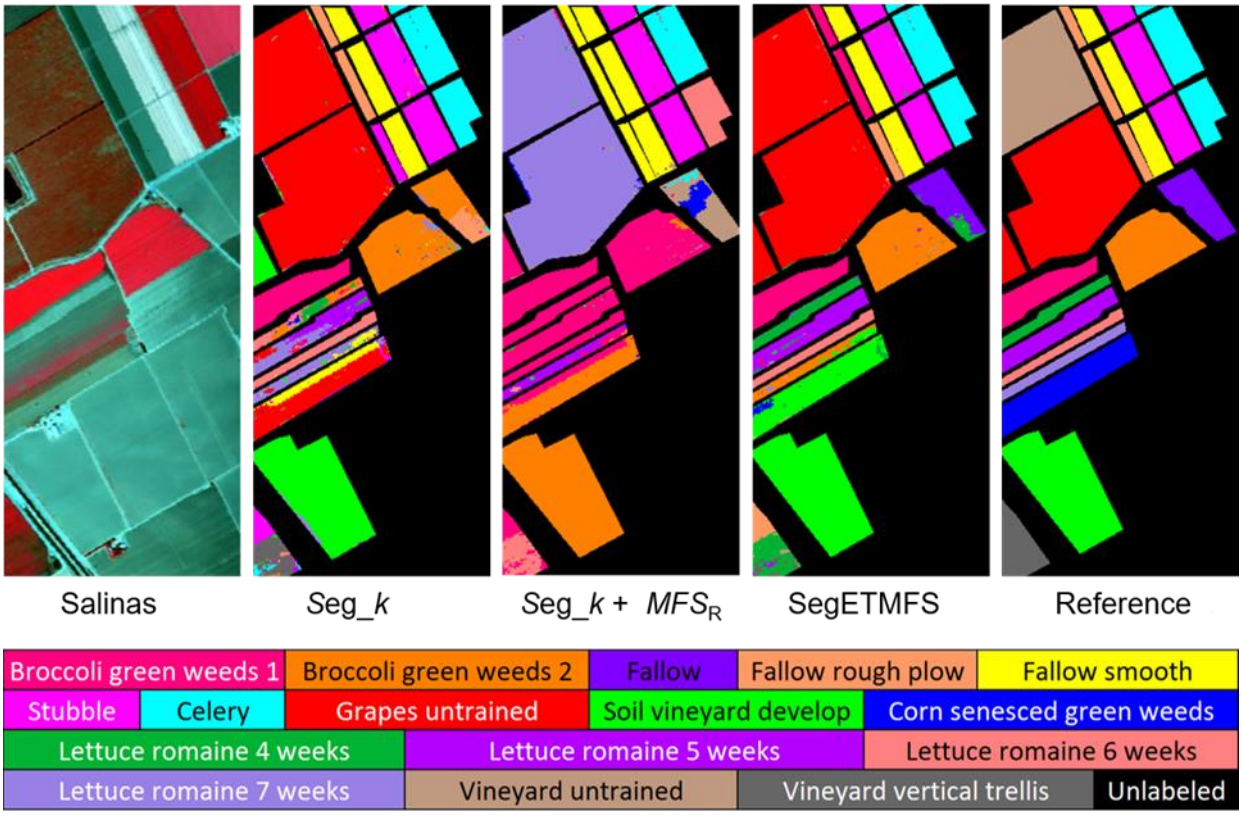


Fig. 5.5. Comparison among third proposed methodology with other existing methods.

Fig. 5.5 shows the comparison between the original image and improvised images of Salinas after classification by the aid of different methodologies.

Table 5.7. Accuracy values for Salinas, Pavia Center and Pavia University.

Dataset	**Salinas**			**Pavia Center**			**Pavia University**		
	*k*_cl	NMI	Purity	*k*_cl	NMI	Purity	*k*_cl	NMI	Purity
*k*M	16	0.7742	0.6734	8	0.7694	0.7961	9	0.5203	0.6696
PCA + *k*M	16	0.7656	0.6785	7	0.7750	0.8048	8	0.5663	0.7150
*k*_seg	28	0.7921	0.6738	17	0.7388	0.8641	17	0.5042	0.7087
*k*_seg + LBS_R	18	0.7932	0.6426	19	0.7538	0.8208	19	0.5242	0.7009
H2NMF	26	0.6372	0.5877	8	0.7596	0.8592	9	0.4641	0.6261
Proposed 1	**16**	**0.8115**	**0.6835**	**8**	**0.8097**	**0.8155**	**9**	**0.6166**	**0.7362**
Proposed 2	**26**	**0.8130**	**0.7011**	**18**	**0.7976**	**0.9001**	**19**	**0.5303**	**0.7141**
Proposed 3	**24**	**0.8430**	**0.7288**	**20**	**0.7903**	**0.8956**	**13**	**0.5407**	**0.6832**

Table 5.7 shows accuracy values for Salinas, Pavia Center and Pavia University with comparison of few other existing methods.

Followings are the acronyms for abovementioned methods:

- Proposed 1 (*k*-means + *k*_fr + *k*_seg)
- Proposed 2 (*k*_seg+ LBS using Gini Impurity)
- Proposed 3 (*k*_seg+ LBS using Shannon Entropy)
- SLIC is Simple Linear Iterative Clustering,
- PCA is Principle Component Analysis,
- H2NMF Hierarchical rank 2 non-negative matrix factorization,
- LBS_R is Local Band Selection with Redundant bands.

Chapter 6 Conclusion

In this study we have discussed several approaches of image classification based on *k*-means clustering, DBSCAN clustering, feature reduction using *k*-means and Principal Component Analysis. Also, image segmentation using *k*-means and SLIC is utilized. Experiments were performed on Salinas dataset and the results were compared on the basis of accuracy assessment. It can be observed from Table 5.1 that classification accuracies (NMI) are significantly improved. Among the evaluated methods, *k*-means incorporation with feature reduction and segmentation has given the best accuracy. The accuracy measurements in terms of NMI for *k*-means + *k*_fr + *k*_seg for dataset Salinas is 81.15 % respectively. While, the other methods have comparatively less accuracy than the first proposed methodology.

Furthermore, in order to achieve more accuracy values second methodology has been proposed, several approaches of image classification based on segmentation clustering with local band selection techniques through gini impurity are performed in order to enhance the accuracy.

Moreover, in third methodology a novel clustering method and local feature selection is proposed in this paper. After performing image segmentation, the segments are subsequently clustered according to Entropy. Also, a multiple feature (band) set selection based on entropy is suggested and used in the clustering strategy's final stage. Multiple feature (band) set selection takes into consideration both relevance and redundancy criteria. Following that, the proposed technique ($SegLBS_R$) was compared to several clustering strategies. The experiments revealed that using the multiple feature (band) set selection method produced higher accuracy. Therefore, it can be said that the proposed method has achieved greater performance than other clustering algorithms that were compared relevancy and remove redundancy. Experiments were performed on Salinas dataset by organizing the data into a matrix and finally applying unsupervised classification via gini impurity in second methodology and via *k*-means in first methodology. Lastly, the results gathered were compared. Higher accuracy is achieved by segments consisting less redundant bands. Thus, it can be concluded that the third proposed method has shown significant improvement over other segmentation based clustering methods including the first and second proposed methodologies.

References

[1] A. Femenias and S. Marın, “Hyperspectral imaging,” Electromagnetic Technologies in Food Science, 2021.

[2] M. Mateen, J. Wen, D. Nasrullah, and M. Azeem Akbar, “The role of hyperspectral imaging: A literature review,” International Journal of Advanced Computer Science and Applications, vol. 9, p. 51, 09 2018.

[3] A. Mehta and O. Dikshit, “Comparative study on projected clustering methods for hyperspectral imagery classification,” Geocarto International, vol. 31, pp. 296–307.

[4] A. F. Alkarkhi, “Cluster analysis, chapter 11,” in Easy Statistics for Food Science with R, A. F. Alkarkhi and W. A. Alqaraghuli, Eds. Academic Press, pp. 177–186.

[5] A. Mehta and O. Dikshit, “Projected clustering of hyperspectral imagery using region merging,” Remote Sensing Letters, vol. 7, no. 8, pp. 721–730, 2016.

[6] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval. Cambridge University Press, 2008, cambridge Books Online.

[7] A. Mehta and O. Dikshit, “Segmentation-based projected clustering of hyperspectral images using mutual nearest neighbour,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 12, pp. 5237–5244, 2017.

[8] Y. Li and H. Wu, “A clustering method based on k-means algorithm,” Physics Procedia, vol. 25, pp. 1104–1109, 12 2012.

[9] S. Lloyd, “Least squares quantization in pcm,” IEEE Transactions on Information Theory, vol. 28, pp. 1104–1109, 1982.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn., Data Clustering: A Review, 1999, pp. 264–323.

[11] J. A. Richards, Remote Sensing Digital Image Analysis: An Introduction. SpringerVerlag, Berlin, Heidelberg, 5th Edition, 2013, pp. 403–446.

[12] Ehu, “Salinas, pavia university and pavia center datasets,”. Online Available: https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[13] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[14] H. Motiyani, P. K. Mali, and A. Mehta, “Hyperspectral image segmentation, feature reduction and clustering using k-means,” in 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2022, pp. 389–393

[15] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” Science, vol. 344, no. 6191, pp. 1492–1496, 2014. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1242072

[16] S. L. Polk and J. M. Murphy, “Multiscale clustering of hyperspectral images through spectral-spatial diffusion geometry,” in 2021 IEEE international Geoscience and Remote Sensing Symposium IGARSS, 2021, pp. 4688-4691.

[17] C. Hinojosa, E. Vera, and H. Arguello, “A fast and accurate similarity- constrained subspace clustering algorithm for hyperspectral image,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 10 773–10 783, 2021.

[18] A. Femenias and S. Mar´ın, “Hyperspectral imaging, chapter 15,” in Electromagnetic Technologies in Food Science. John Wiley Sons, Ltd, 2021, pp. 363–390.

[19] Yao Ding; Zhili Zhang; Xiaofeng Zhao; Yaoming Cai; Siye Li, "Self-Supervised Locality Preserving Low-Pass Graph Convolutional Embedding for Large-Scale Hyperspectral Image Clustering”, IEEE Transactions on Geoscience and Remote Sensing, Vol 60, August, 2022.

[20] Sen Jia; Yue Yuan; Nanying Li; Jianhui Liao; Qiang Huang; Xiuping Jia, "A Multiscale Superpixel-Level Group Clustering Framework for Hyperspectral Band Selection", IEEE Transactions on Geoscience and Remote Sensing, Vol 60, February, 2022.

[21] Y. Li, L. Zhang, C. Tian, C. Ding, Y. Zhang and W. Wei, "Hyperspectral image superresolution extending: An effective fusion based method without knowing the spatial transformation matrix", Proc. IEEE Int. Conf. Multimedia Expo (ICME), pp. 11171122, July, 2017.

[22] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282.

[23] John R. Jensen, Remote Sensing of the Environment- An Earth Resource Perspective.

[24] Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition.

[25] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

Appendix A: Python Scripts

1. Proposed 1 (Segmentation, feature reduction and clustering using *k*-means.**)**

```
# -*- coding: utf-8 -*-
# Importing the Libraries
import scipy.io
import numpy as np
from sklearn import metrics
from skimage.color import label2rgb
from skimage.measure import label, regionprops
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances


# Cluster Purity function
def purity_score(y_true, y_pred):
    # compute contingency matrix (also called confusion matrix)
    contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
    # return purity
    return np.sum(np.amax(contingency_matrix, axis=0)) /
    np.sum(contingency_matrix)

for m in range(10,11):
    for n in range(1, 2):

        # Read HSI
        mat = scipy.io.loadmat('Salinas_corrected.mat');
        mat = mat['salinas_corrected']
        # convert data into float
        mat = np.asfarray(mat, dtype='float')

        # Read Ground truth
        gt_mat = scipy.io.loadmat('Salinas_gt.mat');
        gt_mat = gt_mat['salinas_gt']
        w, h = (gt_mat.shape)
        gt_array = np.reshape(gt_mat, (w * h),)
        idx = gt_array!=0
        lbl_true = gt_array[idx]


        # Plot image
        from matplotlib import pyplot as plt
        plt.imshow(mat[:,:,15])
        #plt.show()

        # Reshape image
```

```
w, h, d = original_shape = tuple(mat.shape)
image_array = np.reshape(mat, (w * h, d))
mat_01  = image_array

#feature Reduction
mat_02 = np.transpose(mat_01)

#apply KMeans

k_fr = 8
kmeans = KMeans(n_clusters = k_fr)

# predict value for whole image
kmeans_labels = kmeans.fit_predict(mat_02)


# reshape whole image for display purpose
Fr = kmeans.cluster_centers_
mat_03 = np.transpose(Fr)

# Segmentation

k_seg = 14
# Segmentation using KMeans
kmeans = KMeans(n_clusters=k_seg).fit(mat_03)
lbl = kmeans.labels_

#Taking care of 0 labels
lbl = lbl + 1

# Obtain Cluster Map
lbl_img = np.reshape(lbl, (w, h))

# Cluster Map to Segmentation Map
labeled, numRegns = label(lbl_img,connectivity=1,return_num=True)

#reshaping mat_03
w, h, d = original_shape = tuple(mat.shape)
mat_04 = np.reshape(mat_03, (w,h, k_fr))

#Arrange segments in nxk form

regions = regionprops(labeled)
seg_nxk = np.zeros((numRegns,k_fr));
i = 0
for prop in regions:
    pxIdLst = prop.coords
    for j in range(k_fr):
        band = mat_04[pxIdLst[:,0],pxIdLst[:,1],j];
        seg_nxk[i,j] = np.mean(band)
    i = i + 1
```

```
#--------------Clustering_----------------
#apply KMeans

k_cl = 16
kmeans = KMeans(n_clusters = k_cl)

# predict value for whole image
kmeans_labels = kmeans.fit_predict(seg_nxk)
lbl = kmeans.labels_

#Rearrange to form cluster map
LbldImg = labeled;
i = 0
for prop in regions:
    pxIdLst = prop.coords
    LbldImg[pxIdLst[:,0],pxIdLst[:,1]] = lbl[i]
    i = i + 1


# display image
plt.imshow(label2rgb(LbldImg))
plt.show()

# write to text file
filename = "SA_Pre_FR_post_k_seg_"+ str(m) + 'run_' + str(n), '.txt'
np.savetxt(filename, LbldImg, delimiter=',',fmt='%d')


# Calculate the Overall accurracy
from sklearn.metrics.cluster import normalized_mutual_info_score

Lbl = np.reshape(LbldImg, (w * h))
lbl_pred = Lbl[idx]
NMI = normalized_mutual_info_score(lbl_true, lbl_pred)

print("K SEG = ", m, " run ",n)
print('NMI = ',NMI)
purity = purity_score(lbl_true, lbl_pred)
print('Purity =', purity)

# Save thematic Map \ cluster map - Matlab format
mdic = {"lbl_img": LbldImg, "label": "thematic map"}
scipy.io.savemat("SAl_KM_16_Pre_FR_8_Post_Seg_"+ str(m) +"_run_"+ str(n)
+ '.mat', mdic)
```

2. Proposed 2 (Segmentation using *k*-means, Local Band Selection and Segment Merging using Gini Impurity)

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 17 19:35:08 2023

@author: rajpr
"""

# Importing the Libraries
import scipy.io
import numpy as np
from sklearn import metrics
from skimage.color import label2rgb
from skimage.measure import label, regionprops
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from scipy.optimize import linear_sum_assignment



for ab in range(1,11):
    # Cluster Purity function
    def purity_score(y_true, y_pred):
        # compute contingency matrix (also called confusion matrix)
        contingency_matrix = metrics.cluster.contingency_matrix(y_true, y_pred)
        # return purity
        return np.sum(np.amax(contingency_matrix, axis=0)) /
        np.sum(contingency_matrix)


    #define function to calculate Gini coefficient
    def gini(x):
        total = 0
        for i, xi in enumerate(x[:-1], 1):
            total += np.sum(np.abs(xi - x[i:]))
            return total / (len(x)**2 * np.mean(x))


    # Read HSI
    mat = scipy.io.loadmat('Salinas_corrected.mat');
    mat = mat['salinas_corrected']
    # convert data into float
    mat = np.asfarray(mat, dtype='float')

    # Read Ground truth
    gt_mat = scipy.io.loadmat('Salinas_gt.mat');
    gt_mat = gt_mat['salinas_gt']
```

```
w, h = (gt_mat.shape)
gt_array = np.reshape(gt_mat, (w * h),)
idx = gt_array!=0
navaibleidx = gt_array ==0
lbl_true = gt_array[idx]

# Plot image
from matplotlib import pyplot as plt
plt.imshow(mat[:,:,15])
#plt.show()

# Reshape image
w, h, d = original_shape = tuple(mat.shape)
image_array = np.reshape(mat, (w * h, d))
mat_01  = image_array

# convert data into float
mat_01 = np.asfarray(image_array, dtype='float')

# Segmentation
k_seg = 6
# Segmentation using KMeans
kmeans = KMeans(n_clusters=k_seg).fit(mat_01)
lbl = kmeans.labels_

#Taking care of 0 labels
lbl = lbl + 1

# Obtain Cluster Map
lbl_img = np.reshape(lbl, (w, h))

# Cluster Map to Segmentation Map
labeled, numRegns = label(lbl_img,connectivity=2,return_num=True)

regions = regionprops(labeled)
seg_nxk = np.zeros((numRegns,d));
i = 0
for prop in regions:
    pxIdLst = prop.coords
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        seg_nxk[i,j] = np.mean(band)
    i = i + 1

#Arrange segments in nxk form
#For each segment, mean of all the pixels in each band, is the new pixel
 vector
regions = regionprops(labeled)

#Matrix of segment
gni_mat = np.zeros((numRegns,1)) #gini index for all segment
```

```
gini_band = np.zeros((numRegns,d))
n_pix = np.zeros((numRegns,1))
k_cl = 26
i = 0
L = 12 # Top bands
L_gini_band = np.zeros((k_cl,L))


temp_array = np.zeros((k_cl,d)) #temp matrix to store band number
#for significant segments
for prop in regions:
    pxIdLst = prop.coords
    matrix = np.zeros((len(pxIdLst),d))
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        for aa in range(len(band)):
            matrix[aa,j] = band[aa] #matrix for each segment
    gni_mat[i] = gini(matrix) #gini for each segment
    n_pix[i] = len(pxIdLst)

    i = i + 1

sig_gini = np.zeros(numRegns)
for i in range(numRegns):
    sig_gini[i] = 50
    if(n_pix[i]> 5):
        sig_gini[i] = gni_mat[i]

# sig_gini = np.argsort(sig_gini)

# for band selection
i = 0
for prop in regions:
    pxIdLst = prop.coords
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        gini_band[i,j] = gini(band) # matrix for gini of segment and band


    i = i + 1


for i in range(k_cl):
    t = int(sig_gini[i])
    b_dist = np.zeros((d,d))  # to store euclidean distance among all band
    for a particular segment
    for x in range(d):
        for y in range(d):
            b_dist[x,y] = np.linalg.norm(seg_nxk[t,x] - seg_nxk[t,y])

    for j in range(d):
```

```
        max = 0
        for k in range(d):
            if gini_band[t,k] > gini_band[t,j] :    #first condition for
                                    band whose gini greater than jth band
                if b_dist[j,k] > max :   #finding max distance among band
                                    whose gini is greater than jth band
                    max = b_dist[j,k]
        temp_array[i,j] = max

scaler = MinMaxScaler()
m = scaler.fit(temp_array)
new_dist = m.transform(temp_array)

score = np.zeros((k_cl,d))     #creating a parameter for sorting by
                                multiplying gini and max distance.
for i in range(k_cl):
    t = int(sig_gini[i])
    for j in range(d):
        score[i,j] = gini_band[t,j] * new_dist[i,j]

score = np.argsort(-score) #sorting in decreasing order

for i in range(k_cl):
    for q in range(L):
        L_gini_band[i,q] = score[i,q]  # Extracting top L bands

#Dividing matrices
split_1 = np.zeros((k_cl,L+1))
split_2 = np.zeros((numRegns,1))

for x in range(numRegns):
    if (x<k_cl):
        split_1[x,0]=sig_gini[x]
        for y in range(1,L+1):
            c = int(split_1[x,0])
            split_1[x,y] = L_gini_band[x,y-1]
    else:
        split_2[x,0]=sig_gini[x]

a = np.zeros((2,L))
new_mat = np.zeros((k_cl,numRegns))
for i in range(k_cl):
    for j in range(k_cl,numRegns):
        for y in range(L):
            a[0,y] = int(split_1[i,y+1]) #assigning band number of
                                        significant cluster
            ff = int(a[0,y])
            u = int(split_1[i,0])  #extracting segment number of significant
                                    segments
            a[0,y] = seg_nxk[u,ff] #assigning band value corresponding to
                                    band numbers
```

```
            d = int(split_2[j,0]) #extracting segment number of non
                                  signifigant segment
            a[1,y] = seg_nxk[d,ff] #assigning band value of nonsignificant
                                   segment corresponding significants
                                   segment's band number
        new_mat[i,j] = np.linalg.norm(a[0,:] - a[1,:])
         #calculating and assigning multidimensional euclidian distance
          between significant and non-significant segments

        #  #-------------Clustering_---------------

#Calculating distance between centroid of 16 segments and other segments
lbl= np.zeros((numRegns,1))
for i in range(k_cl):
    lbl[i]=i

for z in range(k_cl,numRegns):
    min=new_mat[0,z]
    for x in range(k_cl):
        if(new_mat[x,z]<min):
            min=new_mat[x,z]
            lbl[z]=x

#Rearrange to form cluster map
LbldImg = labeled;
i = 0
for prop in regions:
    pxIdLst = prop.coords
    LbldImg[pxIdLst[:,0],pxIdLst[:,1]] = lbl[i]
    i = i + 1

# display image
plt.imshow(label2rgb(LbldImg))
plt.show()

# write to text file
filename = "SA_Pre_FR_post_k_seg_"+ str(ab) + 'run_' + str(cd), '.txt'
np.savetxt(filename, LbldImg, delimiter=',',fmt='%d')

# Calculate the Overall accurracy
from sklearn.metrics.cluster import normalized_mutual_info_score

Lbl = np.reshape(LbldImg, (w * h))
lbl_pred = Lbl[idx]
nmi=normalized_mutual_info_score(lbl_true, lbl_pred)
print("k_seg = 6", "k_cl= 26 run ",ab)
print('NMI = ', nmi)
purity = purity_score(lbl_true, lbl_pred)

print('Purity =', purity)
# Save thematic Map \ cluster map - Matlab format
```

```
    mdic = {"lbl_img": LbldImg, "label": "thematic map"}
    scipy.io.savemat("Proposed gini run "+ str(ab) + '.mat', mdic)
```

3. Proposed 3 (Segmentation using *k*-means, Local Band Selection and Segment Merging using Shannon Entropy)

```
# -*- coding: utf-8 -*-

# Importing the Libraries
import scipy.io
import numpy as np
from sklearn import metrics
from skimage.color import label2rgb
from skimage.measure import label, regionprops
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
from skimage.measure import shannon_entropy

for ab in range(19,20):
    for cd in range(1,5):
        # Cluster Purity function
        def purity_score(y_true, y_pred):
            # compute contingency matrix (also called confusion matrix)
            contingency_matrix = metrics.cluster.contingency_matrix(y_true,
            y_pred)
            # return purity
            return np.sum(np.amax(contingency_matrix, axis=0)) /
            np.sum(contingency_matrix)

        # Read HSI
        mat = scipy.io.loadmat('Salinas_corrected.mat');
        mat = mat['salinas_corrected']
        # convert data into float
        mat = np.asfarray(mat, dtype='float')

        # Read Ground truth
        gt_mat = scipy.io.loadmat('Salinas_gt.mat');
        gt_mat = gt_mat['salinas_gt']
        w, h = (gt_mat.shape)
        gt_array = np.reshape(gt_mat, (w * h),)
        idx = gt_array!=0
        lbl_true = gt_array[idx]

        # Plot image
        from matplotlib import pyplot as plt
        plt.imshow(mat[:,:,15])
        #plt.show()

        # Reshape image
        w, h, d = original_shape = tuple(mat.shape)
```

```
image_array = np.reshape(mat, (w * h, d))
mat_01  = image_array

# convert data into float
mat_01 = np.asfarray(image_array, dtype='float')

# Segmentation
k_seg = 7
# Segmentation using KMeans
kmeans = KMeans(n_clusters=k_seg).fit(mat_01)
lbl = kmeans.labels_

#Taking care of 0 labels
lbl = lbl + 1

# Obtain Cluster Map
lbl_img = np.reshape(lbl, (w, h))

# Cluster Map to Segmentation Map
labeled, numRegns = label(lbl_img,connectivity=2,return_num=True)


regions = regionprops(labeled)
seg_nxk = np.zeros((numRegns,d));
i = 0
for prop in regions:
    pxIdLst = prop.coords
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        seg_nxk[i,j] = np.mean(band)
    i = i + 1

#Arrange segments in nxk form
#For each segment, mean of all the pixels in each band, is the new pixel
 vector
regions = regionprops(labeled)

#Matrix of segment
entropy_band = np.zeros((numRegns,d))

entropy_mat = np.zeros((numRegns,1)) #Shannon Entropy for all segment

n_pix = np.zeros((numRegns,1))
k_cl = 24
i = 0
L = 13 # Top bands
L_entropy_band = np.zeros((k_cl,L))


temp_array = np.zeros((k_cl,d)) #temp matrix to store band number
#for significant segments
```

```
for prop in regions:
    pxIdLst = prop.coords
    matrix = np.zeros((len(pxIdLst),d))
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        for aa in range(len(band)):
            matrix[aa,j] = band[aa] #matrix for each segment
    entropy_mat[i] = shannon_entropy(matrix) #Shannon entropy for each
    segment
    n_pix[i] = len(pxIdLst)

    i = i + 1

sig_entropy = np.zeros(numRegns)
for i in range(numRegns):
    sig_entropy[i] = 50
    if(n_pix[i]> 5):
        sig_entropy[i] = entropy_mat[i]

sig_entropy = np.argsort(sig_entropy)

# for band selection
i = 0
for prop in regions:
    pxIdLst = prop.coords
    for j in range(d):
        band = mat[pxIdLst[:,0],pxIdLst[:,1],j];
        entropy_band[i,j] = shannon_entropy(band) # matrix for entropy
        of segment and band

    i = i + 1


for i in range(k_cl):
    t = int(sig_entropy[i])
    b_dist = np.zeros((d,d))  # to store euclidean distance among all
                                band for a particular segment
    for x in range(d):
        for y in range(d):
            b_dist[x,y] = np.linalg.norm(seg_nxk[t,x] - seg_nxk[t,y])

    for j in range(d):
        max = 0
        for k in range(d):
            if entropy_band[t,k] > entropy_band[t,j] :
                            #band whose entropy greater than jth band
                if b_dist[j,k] > max :
     #max distance among band whose entropy is greater than jth band
                    max = b_dist[j,k]
        temp_array[i,j] = max
```

```
scaler = MinMaxScaler()
m = scaler.fit(temp_array)
new_dist = m.transform(temp_array)

score = np.zeros((k_cl,d))     #creating a parameter for sorting by
                              multiplying entropy and max distance.
for i in range(k_cl):
    t = int(sig_entropy[i])
    for j in range(d):
        score[i,j] = entropy_band[t,j] * new_dist[i,j]

score = np.argsort(-score) #sorting in decreasing order

for i in range(k_cl):
    for q in range(L):
        L_entropy_band[i,q] = score[i,q]  # Extracting top L bands

#Dividing matrices
split_1 = np.zeros((k_cl,L+1))
split_2 = np.zeros((numRegns,1))

for x in range(numRegns):
    if (x<k_cl):
        split_1[x,0]=sig_entropy[x]
        for y in range(1,L+1):
            c = int(split_1[x,0])
            split_1[x,y] = L_entropy_band[x,y-1]
    else:
        split_2[x,0]=sig_entropy[x]

a = np.zeros((2,L))
new_mat = np.zeros((k_cl,numRegns))
for i in range(k_cl):
    for j in range(k_cl,numRegns):
        for y in range(L):
            a[0,y] = int(split_1[i,y+1])
            #assigning band number of significant cluster
            ff = int(a[0,y])
            u = int(split_1[i,0])
            #Extracting segment number of significant segments
            a[0,y] = seg_nxk[u,ff]
            #assigning band value corresponding to band numbers
            d = int(split_2[j,0])
            #extracting segment number of non signifigant segment
            a[1,y] = seg_nxk[d,ff]
            #assigning band value of nonsignificant segment
            #corresponding significants segment's band number
        new_mat[i,j] = np.linalg.norm(a[0,:] - a[1,:])
        #calculating and assigning multidimensional euclidian distance
        between significant and non-significant segments
```

```
        #  #--------------Clustering_----------------
#Calculating distance between centroid of 16 segments and other segments

lbl= np.zeros((numRegns,1))
for i in range(k_cl):
    lbl[i]=i

for z in range(k_cl,numRegns):
    min=new_mat[0,z]
    for x in range(k_cl):
        if(new_mat[x,z]<min):
            min=new_mat[x,z]
            lbl[z]=x

#Rearrange to form cluster map
LbldImg = labeled;
i = 0
for prop in regions:
    pxIdLst = prop.coords
    LbldImg[pxIdLst[:,0],pxIdLst[:,1]] = lbl[i]
    i = i + 1

# display image
plt.imshow(label2rgb(LbldImg))
plt.show()

# Calculate the Overall accurracy
from sklearn.metrics.cluster import normalized_mutual_info_score
from sklearn.metrics.cluster import adjusted_mutual_info_score

Lbl = np.reshape(LbldImg, (w * h))
lbl_pred = Lbl[idx]
nmi=normalized_mutual_info_score(lbl_true, lbl_pred)
ami = adjusted_mutual_info_score(lbl_true, lbl_pred)
print("k_seg = 7 ", "k_cl= 24  L = 13  run ",cd)
purity = purity_score(lbl_true, lbl_pred)
print('Purity =', purity)

# Save thematic Map \ cluster map - Matlab format
mdic = {"lbl_img": LbldImg, "label": "thematic map"}
scipy.io.savemat("SAl_K seg 7 kcl 24 L 13  run"+ str(cd) + '.mat', mdic)
```

Appendix B: Publications from this Project Work

1. P. K. Mali, H. Motiyani, and A. Mehta, “Hyperspectral Image Clustering, Feature Reduction and Segmentation using k-means”, presented at International Conference on Advances in Mechanics, Modelling, Computing and Statistics (**ICAMMCS-2022**).

2. H. Motiyani, P. K. Mali and A. Mehta, "Hyperspectral Image Segmentation, Feature Reduction and Clustering using k-means," 2022 International Conference on Computing, Communication, and Intelligent Systems (**ICCCIS-2022**), Greater Noida, India, 2022, pp. 389-393. (Publisher: **IEEE**). https://doi.org/10.1109/ICCCIS56430.2022.10037590

3. P. K. Mali, H. Motiyani, Q. Sameed and A. Mehta, “Hyperspectral Image clustering and local feature selection using Gini Impurity”, 2023 7th International Conference on Trends in Electronics and Informatics (**ICOEI-2023**), Tirunelveli, India, May 2023 (Publisher: **IEEE**) http://dx.doi.org/10.1109/ICOEI56765.2023.10125605

4. H. Motiyani, Q. Sameed, P. K. Mali and A. Mehta, "Clustering of Hyperspectral Images using Entropy based Multiple Features (Bands) Set Selection," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (**CISES-2023**), Greater Noida, India, 2023, pp. 849-854. (Publisher: **IEEE**) https://doi.org/10.1109/CISES58720.2023.10183495

Final_3

ORIGINALITY REPORT

17%	10%	13%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Pal, N.R.. "A review on image segmentation techniques", Pattern Recognition, 199309 Publication	1%
2	en.wikipedia.org Internet Source	1%
3	link.springer.com Internet Source	1%
4	Dety Nurfadilah, Sudarmawan Samidi. "HOW THE COVID-19 CRISIS IS AFFECTING CUSTOMERS' INTENTION TO USE ISLAMIC FINTECH SERVICES: EVIDENCE FROM INDONESIA", Journal of Islamic Monetary Economics and Finance, 2021 Publication	<1%
5	patents.google.com Internet Source	<1%
6	ntnuopen.ntnu.no Internet Source	<1%
7	Sen Jia, Yue Yuan, Nanying Li, Jianhui Liao, Qiang Huang, Xiuping Jia, Meng Xu. "A	<1%

Multiscale Superpixel-Level Group Clustering Framework for Hyperspectral Band Selection", IEEE Transactions on Geoscience and Remote Sensing, 2022
Publication

8	library.isical.ac.in:8080 Internet Source	<1 %
9	Xiaochun Wang, Xiali Wang, Don Mitchell Wilkes. "Machine Learning-based Natural Scene Recognition for Mobile Robot Localization in An Unknown Environment", Springer Science and Business Media LLC, 2020 Publication	<1 %
10	Yao Li, Liyi Zhang, Lei Chen. "Spectral-spatial hyperspectral image classification based on capsule network with limited training samples", International Journal of Remote Sensing, 2022 Publication	<1 %
11	www.jayrambhia.com Internet Source	<1 %
12	H. Shin. "Color image segmentation", Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat No PR00149) CVPR-99, 1999 Publication	<1 %

13	Yao Ding, Zhili Zhang, Xiaofeng Zhao, Wei Cai, Nengjun Yang, Haojie Hu, Xianxiang Huang, Yuan Cao, Weiwei Cai. "Unsupervised Self-correlated Learning Smoothy Enhanced Locality Preserving Graph Convolution Embedding Clustering for Hyperspectral Images", IEEE Transactions on Geoscience and Remote Sensing, 2022 Publication	<1%
14	Submitted to Dubai International Academy Student Paper	<1%
15	Wang Yongdong, Xu Dongwei, Peng Peng, Zhang Guijun. "Analysis of road travel behaviour based on big trajectory data", IET Intelligent Transport Systems, 2020 Publication	<1%
16	pdf.usaid.gov Internet Source	<1%
17	Anand Mehta, Onkar Dikshit. "Segmentation-Based Projected Clustering of Hyperspectral Images Using Mutual Nearest Neighbour", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2017 Publication	<1%
18	Submitted to University of Witwatersrand Student Paper	<1%

19	Andrea Paoli. "Swarm intelligence for unsupervised classification of hyperspectral images", 2009 IEEE International Geoscience and Remote Sensing Symposium, 07/2009 Publication	<1 %
20	Submitted to INTI Universal Holdings SDM BHD Student Paper	<1 %
21	cs.uef.fi Internet Source	<1 %
22	Munish Kumar, Surbhi Gupta, Neeraj Mohan. "A computational approach for printed document forensics using SURF and ORB features", Soft Computing, 2020 Publication	<1 %
23	Submitted to Eastern Institute of Technology Student Paper	<1 %
24	Melisa Mollaian, Gyula Dörgő, Ahmet Palazoglu. "Performing Multi-Objective Optimization Alongside Dimension Reduction to Determine Number of Clusters", Processes, 2022 Publication	<1 %
25	Banit', Ibtissam, N.A. ouagua, Mounir Ait Kerroum, Ahmed Hammouch, and Driss Aboutajdine. "Band selection by mutual information for hyper-spectral image	<1 %

classification", International Journal of Advanced Intelligence Paradigms, 2016.
Publication

26	Submitted to Unicaf University Student Paper	<1 %
27	Zhanyang Zhang, Jiaqi Chen, Zhiwei Liu. "SLIC segmentation method for full-polarised remote-sensing image", The Journal of Engineering, 2019 Publication	<1 %
28	Yang Xu, Fei Ye, Bo Ren, Liangfu Lu, Xudong Cui, Jocelyn Chanussot, Zebin Wu. "Tensor representation for remote sensing images", Elsevier BV, 2022 Publication	<1 %
29	Hind R.M, Farah Abbas, Ali Abdulkarem. "Performance Evaluation of K-Mean and Fuzzy C-Mean Image Segmentation Based Clustering Classifier", International Journal of Advanced Computer Science and Applications, 2015 Publication	<1 %
30	mdpi-res.com Internet Source	<1 %
31	"Hybrid Artificial Intelligence Systems", Springer Science and Business Media LLC, 2014 Publication	<1 %

32	ictactjournals.in Internet Source	<1 %
33	www.researchgate.net Internet Source	<1 %
34	Divyesh Varade, Ajay K. Maurya, Onkar Dikshit. "Unsupervised hyperspectral band selection using ranking based on a denoising error matching approach", International Journal of Remote Sensing, 2019 Publication	<1 %
35	akkio.gitbook.io Internet Source	<1 %
36	Yongbo Zhang, Miaomiao Wen, Ying Sun, Hui Chen, Yunkai Cai. "Black Carbon Emission Prediction of Diesel Engine Using Stacked Generalization", Atmosphere, 2022 Publication	<1 %
37	aj.tubitak.gov.tr Internet Source	<1 %
38	Hang Gong, Qiuxia Li, Chunlai Li, Haishan Dai, Zhiping He, Wenjing Wang, Haoyang Li, Feng Han, Abudusalamu Tuniyazi, Tingkui Mu. "Multiscale Information Fusion for Hyperspectral Image Classification Based on Hybrid 2D-3D CNN", Remote Sensing, 2021 Publication	<1 %

39	Submitted to University of Essex Student Paper	<1 %
40	B. Uma Shankar. "Novel Classification and Segmentation Techniques with Application to Remotely Sensed Images", Lecture Notes in Computer Science, 2007 Publication	<1 %
41	Submitted to Kenyatta University Student Paper	<1 %
42	origin-production.wikiwand.com Internet Source	<1 %
43	Ghamisi, Pedram, Jon Atli Benediktsson, and Johannes R. Sveinsson. "Automatic Spectral–Spatial Classification Framework Based on Attribute Profiles and Supervised Feature Extraction", IEEE Transactions on Geoscience and Remote Sensing, 2014. Publication	<1 %
44	core.ac.uk Internet Source	<1 %
45	Ozan Arslan, Özer Akyürek, Şinasi Kaya, Dursun Z. Şeker. "Dimension Reduction Methods Applied to Coastline Extraction on Hyperspectral Imagery", Geocarto International, 2018 Publication	<1 %

46	Submitted to University of Macau Student Paper	<1%
47	Submitted to University of Reading Student Paper	<1%
48	www.lib.ncsu.edu Internet Source	<1%
49	www.rangevoting.org Internet Source	<1%
50	Submitted to University of Iceland Student Paper	<1%
51	Wei Wei, Jiangtao Nie, Lei Zhang, Yanning Zhang. "Unsupervised Recurrent Hyperspectral Imagery Super-Resolution Using Pixel-Aware Refinement", IEEE Transactions on Geoscience and Remote Sensing, 2020 Publication	<1%
52	pstorage-loughborough-53465.s3.amazonaws.com Internet Source	<1%
53	uis.brage.unit.no Internet Source	<1%
54	zf.co-aol.com Internet Source	<1%

55	Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. "SLIC Superpixels Compared to State-of-the-art Superpixel Methods", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012. Publication	<1 %
56	arxiv.org Internet Source	<1 %
57	cyber.felk.cvut.cz Internet Source	<1 %
58	winnspace.uwinnipeg.ca Internet Source	<1 %
59	Bing-Yu Sun, , Chao-Yong Wang, Hai-Lei Wang, and Wen-Bo Li. "A two stage method for hyperspectral image classification", Proceeding of the 11th World Congress on Intelligent Control and Automation, 2014. Publication	<1 %
60	hdl.handle.net Internet Source	<1 %
61	moam.info Internet Source	<1 %
62	www.hindawi.com Internet Source	<1 %
	www.journaltocs.ac.uk	

63	Internet Source	<1%
64	www.mdpi.com Internet Source	<1%
65	Azam Peyvandipour, Adib Shafi, Nafiseh Saberian, Sorin Draghici. "Identification of cell types from single cell data using stable clustering", Scientific Reports, 2020 Publication	<1%
66	Lecture Notes in Computer Science, 2010. Publication	<1%
67	Lokman, Gurcan, and Guray Yilmaz. "Hyperspectral image classification using Support Vector Neural Network algorithm", 2015 7th International Conference on Recent Advances in Space Technologies (RAST), 2015. Publication	<1%
68	Submitted to Middle East College of Information Technology Student Paper	<1%
69	Yuliya Tarabalka, James C. Tilton. "Best merge region growing with integrated probabilistic classification for hyperspectral imagery", 2011 IEEE International Geoscience and Remote Sensing Symposium, 2011 Publication	<1%
	cnrt.nc	

70	Internet Source	<1%
71	deepai.org Internet Source	<1%
72	dokumen.pub Internet Source	<1%
73	iajit.org Internet Source	<1%
74	ischolar.informaticsglobal.com Internet Source	<1%
75	keep.lib.asu.edu Internet Source	<1%
76	www.arxiv-vanity.com Internet Source	<1%
77	www.science.gov Internet Source	<1%
78	www.scilit.net Internet Source	<1%
79	www2.umbc.edu Internet Source	<1%
80	W. Zhong, G. Altun, R. Harrison, P.C. Tai, Y. Pan. "Improved$hboxK$-Means Clustering Algorithm for Exploring Local Protein Sequence Motifs Representing Common	<1%

Structural Property", IEEE Transactions on Nanobioscience, 2005
Publication

81 Alexander Lavin, Diego Klabjan. "Clustering time-series energy data from smart meters", Energy Efficiency, 2014 <1%
Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		