**A SPECTRAL - SPATIAL MULTICRITERIA ACTIVE LEARNING TECHNIQUE FOR HYPERSPECTRRAL IMAGE CLASSIFICATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **Page No** |
| 1 | Abstract | 3 |
| 2 | Introduction | 5 |
| 3 | Existing system | 13 |
| 4 | Disadvantages | 14 |
| 5 | Proposed system | 15 |
| 6 | Advantages | 16 |
| 7 | Flow Diagram | 17 |
| 8 | Use case Diagram | 18 |
| 9 | Sequential Diagram | 19 |
| 10 | System Architecture | 20 |
| 11 | Block Diagram | 21 |
| 12 | Modules | 22 |
| 13 | Modules Description | 23 |
| 14 | Literature Survey | 48 |
| 15 | Testing of Product | 68 |
| 16 | System Requirements | 75 |
| 17 | Software Description | 76 |
| 18 | System Design | 82 |
| 19 | Screenshot | 86 |
| 20 | Conclusion | 93 |
| 21 | Future Enhancement | 94 |
| 22 | References | 95 |

**TABLE OF CONTENT**

**ABSTRACT**

Hyperspectral image classification with limited labelled samples is a challenging task and still an open research issue. A novel technique is presented to address such an issue by exploiting dimensionality reduction, spectral-spatial information, and classification with active learning. a hierarchical deep framework called Spectral-Spatial Response (SSR)to jointly learn spectral and spatial features of Hyperspectral Images (HSIs) by iteratively abstractingneighboring regions. SSR forms a deep architecture and is able to learn discriminative spectral-spatialfeatures of the input HSI at different scales. It includes several existing spectral-spatial-basedmethods as special scenarios within a single unified framework. Based on SSR, we further proposethe Subspace Learning-based Networks (SLN) as an example of SSR for HSI classification. In SLN,the joint spectral and spatial features are learned using templates simply learned by Marginal FisherAnalysis (MFA) and Principal Component Analysis (PCA). An important contribution to the successof SLN is the exploitation of label information of training samples and the local spatial structureof HSI. Extensive experimental results on four challenging HSI datasets taken from the Airborne Visible-Infrared Imaging Spectrometer (AVIRIS) and Reflective Optics System Imaging Spectrometer(ROSIS) airborne sensors show the implementational simplicity of SLN and verify the superiority of SSR for HSI classification. Although the dimensionality reduction mitigates the curse of dimensionality problem, the classification results still rely on the quality of the available labeled samples. Due to the usually complex statistical distributions of the patterns belonging to different classes, informative labeled samples (i.e., the nonredundant samples that distinguish among different classes) are essential to train the classifier. Two recent approaches to HSI classification using limited labeled samples are semisupervised learning and active learning (AL). Semisupervised learning incorporates both the labeled and unlabeled data into the training phase of a classifier to obtain better decision boundaries. In contrast, AL is a paradigm to reduce the labeling effort and optimize the performance of a classifier by including only most informative patterns (which have highest training information for supervised learning) into the training set. AL techniques are usually based on iterative algorithms. At each iteration, one or multiple most informative unlabeled patterns are chosen for manual labeling and the classification model is retrained with the additional labeled samples. The step of training and the step of assigning labels are iterated alternately until a stable classification result is obtained, i.e., the classification accuracy does not increase further by increasing the number of training samples.

**INTRODUCTION**

Hyper spectral images (HSIs) are characterized by hundreds of bands acquired in contiguous spectral ranges and narrow spectrum intervals. They represent a very rich information source for a precise characterization and recognition of objects on the ground. In the past decades, researchers devoted great attention to the classification of HSIs for numerous applications, like the detailed classification of forest areas, the analysis of inland water and coastal zones, the analysis of natural risks, etc. Due to the existence of a large number of bands, classification of HSI requires a sufficiently large number of training (labelled) samples in order to mitigate curse of dimensionality (or Hughes phenomenon). However, in most of the hyper spectral applications, the numbers of available labelled samples are scarce and very costly to collect. To address such a problem, dimensionality reduction of HSIs is widely used in the literature. Dimensionality reduction decreases the number of the HSI spectral channels with the help of feature selection (extraction) techniques that select (extract) only no redundant informative features that preserve discriminative properties of the data. Although the dimensionality reduction mitigates the curse of dimensionality problem, the classification results still rely on the quality of the available labelled samples. Due to the usually complex statistical distributions of the patterns belonging to different classes, informative labelled samples (i.e., the no redundant samples that distinguish among different classes) are essential to train the classifier. Two recent approaches to HSI classification using limited labelled samples are semi supervised learning and active learning (AL). Semi supervised learning incorporates both the labelled and unlabelled data into the training phase of a classifier to obtain better decision boundaries. In contrast, AL is a paradigm to reduce the labeling effort and optimize the performance of a classifier by including only most informative patterns (which have highest training information for supervised learning) into the training set. AL techniques are usually based on iterative algorithms. At each iteration, one or multiple most informative unlabelled patterns are chosen for manual labeling and the classification model is retrained with the additional labelled samples. The step of training and the step of assigning labels are iterated alternately until a stable classification result is obtained, i.e., the classification accuracy does not increase further by increasing the number of training samples. Accordingly, the classifier is trained only with the most informative samples, thus, reducing the labeling cost. In the literature, many studies have shown that AL is a promising approach to classification of HSI with limited labelled samples. The fundamental component of AL is the design of a query function that should incorporate a set of criteria for selecting the most informative patterns to label from an unlabeled pool U. Depending on the number of samples to be selected at each iteration, two kinds of AL methods exist in the literature:

1) those that select the single most informative sample at each iteration,

2) those that select a batch of informative samples at each iteration.

To avoid retraining the classifier for each new labeled sample added to the training set, batch mode AL methods are preferred in the remote sensing community. AL has been widely studied in the pattern recognition literature. In the recent years, several AL techniques have been proposed for classification of multispectral and hyperspectral remote sensing images. Mitra et al. presented an AL technique by adopting a one-against-all (OAA) architecture of binary support vector machine (SVM) classifiers. They select batch of uncertain samples, one from each binary SVM, by considering that closest to the discriminating hyperplane. It selects the unlabeledsample that maximizes the information gain between thea posteriori probability distribution estimated from the currenttraining set and the training set obtained by including also thatsample. In, two batch mode AL techniques are proposed forclassification of remote sensing images. The first one extendsthe SVM margin sampling method by selecting the samplesthat are closest to the separating hyperplane and associated withdifferent closest support vectors. The second method is basedon a committee of classifiers. The samples that have maximumdisagreement among the committee of learners are selected. In[26], Demir et al. investigated several SVM-based batch modeALtechniques for the classification of remote sensing images. In, a batch mode AL technique based on multiple uncertaintyfor SVM classifiers is presented. Few cluster assumption basedAL techniques are presented in. A cost-sensitive ALmethod for the classification of remote sensing images is presentedin and extended in. This method includes inthe query function also the cost associated with the accessibilityof the unlabeled samples. An AL technique based on a Gaussianprocess classifier for HSI analysis is presented in. Allthe above-mentioned AL methods only exploit spectral information.There are few techniques existing in the literature thatexploit spectral and spatial information to achieve improvedclassification results .As mentioned before, feature selection (or extraction) playsan important role forHSI classificationwith limited labeled samples.Moreover, in practice, pixels are spatially related due tothe homogeneous spatial distribution of land covers. It is highlyprobable that two adjacent pixels belong to the same class. Thus,information captured in neighboring locations may provide usefulsupplementary knowledge for analysis of a pixel. Therefore,spectral information with the support of spatial information caneffectively reduce the uncertainty of class assignment and helpto find the most informative samples.In this paper, we propose a novel technique for the classificationof HSI with limited labeled samples. The proposedtechnique is divided into two phases. Considering the importanceof dimensionality reduction and spatial information forthe analysis of HSIs, Phase I extracts the features correspondingto each pixel of HSI using both spectral and spatial information.To this end, first principal component analysis (PCA) is used toreduce the dimensionality of HSI; then, extended morphologicalprofiles (EMP) are exploited.

**DOMAIN INTRODUCTION**

**Digital image processing**:

Digital image processing is the use of computer [algorithms](https://en.wikipedia.org/wiki/Algorithm) to perform [image processing](https://en.wikipedia.org/wiki/Image_processing) on [digital images](https://en.wikipedia.org/wiki/Digital_image). As a subcategory or field of [digital signal processing](https://en.wikipedia.org/wiki/Digital_signal_processing), digital image processing has many advantages over [analog image processing](https://en.wikipedia.org/wiki/Analog_image_processing). It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of [multidimensional systems](https://en.wikipedia.org/wiki/Multidimensional_systems).

Digital image processing deals with manipulation of digital images through a digital computer. It is a subfield of signals and systems but focus particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of that system is a digital image and the system process that image using efficient algorithms, and gives an image as an output.

**PROCESS INTRODUCTION**

* **Synthetic Aperture Radar** is a technique for computing high-resolution radar returns that exceed the traditional resolution limits imposed by the physical size, or aperture, of an antenna.
* SAR exploits antenna motion to synthesize a large “virtual’ aperture, as if the physical antenna were larger than it actually is. The SAR technique is used to form a high-resolution backscatter image of a distant area using an airborne radar platform.

Synthetic-aperture radar (SAR) is a form of radar that is used to create two- or three-dimensional images of objects, such as landscapes. SAR uses the motion of the radar antenna over a target region to provide finer spatial resolution than conventional beam-scanning radars. SAR is typically mounted on a moving platform, such as an aircraft or spacecraft, and has its origins in an advanced form of side looking airborne radar (SLAR). The distance the SAR device travels over a target in the time taken for the radar pulses to return to the antenna creates the large synthetic antenna aperture (the size of the antenna). Typically, the larger the aperture, the higher the image resolution will be, regardless of whether the aperture is physical (a large antenna) or synthetic (a moving antenna) – this allows SAR to create high-resolution images with comparatively small physical antennas. To create a SAR image, successive pulses of radio waves are transmitted to "illuminate" a target scene, and the echo of each pulse is received and recorded.

The pulses are transmitted and the echoes received using a single beam-forming antenna, with wavelengths of a meter down to several millimeters. As the SAR device on board the aircraft or spacecraft moves, the antenna location relative to the target changes with time. Signal processing of the successive recorded radar echoes allows the combining of the recordings from these multiple antenna positions. This process forms the synthetic antenna aperture and allows the creation of higher-resolution images than would otherwise be possible with a given physical antenna. As of 2010, airborne systems provide resolutions of about 10 cm, ultra-wideband systems provide resolutions of a few millimeters, and experimental terahertz SAR has provided sub-millimeter resolution in the laboratory.

**Motivation and applications:**

The properties of SAR can be described as having high-resolution capability, which is independent of flight altitude, not being dependent on the weather, as SAR can select proper frequency range. SAR also have a great day and night imaging capability considering their own illumination. SAR images have wide applications in remote sensing and mapping of the surfaces of both the Earth and other planets. Some of the other important applications of SAR are topography, oceanography, glaciology, geology (for example, terrain discrimination and subsurface imaging), and forestry, which includes forest height, biomass, deforestation. Volcano and earthquake monitoring is a part of differential interferometry. It is also useful in environment monitoring like oil spills, flooding, urban growth, global change and military surveillance, which includes strategic policy and tactical assessment.[4] SAR can also be implemented as inverse SAR by observing a moving target over a substantial time with a stationary antenna.

**Basic Principle:**

A synthetic-aperture radar is an imaging radar mounted on a moving platform.[5] Electromagnetic waves are sequentially transmitted, and reflected echoes are collected, digitized and stored by the radar antenna for later processing. As transmission and reception occur at different time, they map to different positions. The well ordered combination of the received signals builds a virtual aperture that is much longer than the physical antenna length. This is why it is named "synthetic aperture", giving it the property of being an imaging radar. The range direction is parallel to flight track and perpendicular to azimuth direction, which is also known as along-track direction because it is in line with the position of the object within the antenna's field of view. The 3D processing is done in two steps: the azimuth and range direction are focused for the generation of 2D (azimuth-range) high-resolution images, after which a digital elevation model (DEM is used to measure the phase differences between complex images, which is determined from different look angles to recover the height information. This height information, along with the azimuth-range coordinates provided by 2-D SAR focusing, gives the third dimension, which is the elevation direction The first step requires only standard processing algorithms, for the second step, an additional pre-processing stage such as image co-registration and phase calibration is used.[2][8]In addition to this, multiple baselines can be used to extend 3D imaging to the time dimension. 4D and multi-D SAR imaging allows imaging of complex scenarios, such as urban areas, and has improved performances with respect to classical interferometric techniques such as persistent scatterers interferometry (PSI).

**Algorithm**

The SAR algorithm, as given here, applies to phased arrays generally. A three-dimensional array (a volume) of scene elements is defined, which will represent the volume of space within which targets exist. Each element of the array is a cubical voxel representing the probability (a "density") of a reflective surface being at that location in space. (Note that two-dimensional SARs are also possible—showing only a top-down view of the target area.)Initially, the SAR algorithm gives each voxel a density of zero. Then, for each captured waveform, the entire volume is iterated. For a given waveform and voxel, the distance from the position represented by that voxel to the antenna(e) used to capture that waveform is calculated. That distance represents a time delay into the waveform. The sample value at that position in the waveform is then added to the voxel's density value. This represents a possible echo from a target at that position. Note that there are several optional approaches here, depending on the precision of the waveform timing, among other things.

For example, if phase cannot be accurately known, then only the envelope magnitude (with the help of a Hilbert transform) of the waveform sample might be added to the voxel. If polarization and phase are known in the waveform and are accurate enough, then these values might be added to a more complex voxel that holds such measurements separately. After all waveforms have been iterated over all voxels, the basic SAR processing is complete. What remains, in the simplest approach, is to decide what voxel density value represents a solid object. Voxels whose density is below that threshold are ignored. Note that the threshold level chosen must at least be higher than the peak energy of any single wave, otherwise that wave peak would appear as a sphere (or ellipse, in the case of multistatic operation) of false "density" across the entire volume. Thus to detect a point on a target, there must be at least two different antenna echoes from that point. Consequently, there is a need for large numbers of antenna positions to properly characterize a target. The voxels that passed the threshold criteria are visualized in 2D or 3D. Optionally, added visual quality can sometimes be had by use of a surface detection algorithm like marching cubes.

**EXISTING SYSTEM**

For classification we have been used the Support Vector Machine and Relevance Vector Machine. A singularity-based criterion is also used to identify informative pixels by taking spatial information into account when selecting samples. The spatial features generated via 3D-RDWT not only provide sufficient views for multiview AL, but are also less sensitive to additive noise. Class values contains the most unlabeled pixels. They represent a very rich information source for a precise characterization and recognition of objects on the ground. In the past decades, researchers devoted great attention to the classification of HSIs for numerous applications, like the detailed classification of forest areas, the analysis of inland water and coastal zones, the analysis of natural risks. Due to the existence of a large number of bands, classification of HSI requires a sufficiently large number of training samples in order to mitigate curse of dimensionality. However, in most of the hyperspectral applications, the numbers of available labeled samples are scarce and very costly to collect. To address such a problem, dimensionality reduction of HSIs is widely used in the literature. Dimensionality reduction decreases the number of the HSI spectral channels with the help of feature selection techniques that select only nonredundant informative features that preserve discriminative properties of the data.

**DISADVANTAGES**

* Large number of bands
* Classification requires a large number of training samples.
* The numbers of available labelled samples are scarce and very costly to collect.
* It provides most uncertain pixels.

**PROPOSED SYSTEM**

We propose a technique for classification ofHSIs with limited labeled samples. The proposed technique is divided into two phases. Phase I generates the patterns corresponding to each pixel of the HSIs by extracting spectral-spatial features. Phase II performs the classification task by exploiting a novel AL technique. Fig. 1 shows the block diagram of the proposed framework. The detail steps of the proposed technique are given in next subsequent subsections.

Here we are divide a two patterns

* + Spectral Spatial Feature Extraction
  + Classification

**Spectral Spatial Feature Extraction**

The classification of an HSI when a limited number of labeled samples is available is a challenging task due to the curse of dimensionality problem. Moreover, due to the existence of large number of redundant and irrelevant bands, the distributions of different classes in the original feature space are complex and do not follow the cluster assumption property, i.e., the interclass differences between classes are not significant. Thus, cluster assumption criterion may fail to play a significant role for identifying informative samples. Both problems can be solved by reducing the dimensionality of the HSI data by selecting (or extracting) only discriminative features.

**Classification**

* Test features
* Target values
* Training set

These classify using SVM classification.

**ADVANTAGES**

* Detailed classification of forest areas.
* Analysis of inland water and coastal zones.
* Analysis of natural risks.
* Dimensionality reduction decreases the no of spectral channel with the help of feature selection.
* It select only no redundant informative features.

**DIAGRAMS**

**DATA FLOW DIAGRAM**

Load a Dataset

Pick a file

Preprocessing

Select a correct type of file

Red Band

Green Band

Blue Band

Segmentation

Clustering

Region-1

Region-2

Region-3

Feature Extraction

Classification

Measuring Performance

Form a chart

Yes

No

**USE CASE DIAGRAM**

Hyper spectral Image

Data Set

**SEQUENCE DIAGRAM**

Read a File

Preprocessing

Segmentation

Classification

Performance

Feature Extraction

imread ()

Red, Green, Blue

multisvm

Evaluate

Clustering

**SYSTEM ARCHITECTURE**

Load Data

set

Preprocessing

Red

Green

Blue

Segmentation

Clustering

Feature Extraction

Classification

Performance

Accuracy

Sensitivity

Specificity

Precision

Recall

F Measure

G Mean

Mutual Index

Correlation

SSIM

**BLOCK DIAGRAM**

Clustering

Feature Extraction

Load Dataset

Preprocessing

Segmentation

Classification

Performance

**MODULES**

* Read a File
* Preprocessing
* Segmentation
* Clustering
* Feature Extraction
* Classification
* Performance

**MODULE DESCRIPTION**

**1. READ A FILE**

In our process we have to load a hyper-spectral dataset to process. First, we group a hyperspectral images into dataset. We can select the any of the image from the dataset. It can be possible by the use of **uigetfile ()** function. It has two parameters, these are type of file and message. If we use ‘\*.\*’ for the type of file, we can select any type of file at runtime. If we use this type it displayed all type of files.

**Structures**

MATLAB has structure data types. Since all variables in MATLAB are arrays, a more adequate name is "structure array", where each element of the array has the same field names. In addition, MATLAB supports dynamic field names. Unfortunately, MATLAB JIT does not support MATLAB structures, therefore just a simple bundling of various variables into a structure will come at a cost

**Imread():**

A = imread(filename)

Reads the image from the file specified by filename, inferring the format of the file from its contents.

If filename is a multi-image file, then imread reads the first image in the file.

A = imread(filename,fmt) additionally specifies the format of the file with the standard file extension indicated by fmt.

If imread cannot find a file with the name specified by filename, it looks for a file named filename.fmt.

**READ A FILE - FLOW**

Load a Dataset

Pick a file

Select a correct type

Go to Preprocessing

No

Yes

**2. PREPROCESSING**

Pre-processing is a common name for operations with images at the lowest level of abstraction. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. The pre-processing is the size adjusting of the considered image, luminance normalization, statistical normalization, filtering noise with specified filter, conversion to certain class.

A color image is usually stored in memory as a **raster map**, a two-dimensional array of small integer triplets; or (rarely) as three separate raster maps, one for each channel.

Separate R, G, and B image layers Eight bits per sample (24 bits per pixel) seem adequate for most uses, but faint banding artifacts may still be visible in some smoothly varying images, especially those subject to processing. Particularly demanding applications may use 10 bits per sample or more. On the other hand, some widely used image file formats and graphics cards may use only **8 bits** per pixel, i.e., only **256 different colors**, or 2–3 bits per channel. Converting continuous-tone images like photographs to such formats requires dithering and yields rather grainy and fuzzy results. Graphics cards that support **16 bits** per pixel provide **65536 distinct colors**, or 5–6 bits per channel. This resolution seems satisfactory for non-professional uses, even without dithering.

Preprocessing gives a 3 band images to us.

These are

* Red band
* Green band
* Blue band

**Red Band:**

To display the original image into Red band image, the value is 1. **Red=img(:,:,1);**

**Green Band:**

To display the original image into Red band image, the value is 2.

**green=img(:,:,2);**

**Blue Band:**

To display the original image into Red band image, the value is 1.

**Red=img(:,:,3);**

**PREPROCESSING - FLOW**

Select a file

Preprocessing

Red Band

Green Band

Blue Band

**3. SEGMENTATION**

Image segmentation is the process of dividing an image into multiple parts. This is typically used to identify objects or other relevant information in digital images. For example, one way to find regions in an image is to look for abrupt discontinuities in pixel values, which typically indicate edges. These edges can define regions. Other methods divide the image into regions based on colour values or texture.

* + **Thresholding methods such as** [**Otsu’s method**](https://in.mathworks.com/help/images/examples/correcting-nonuniform-illumination.html)

**Correcting Nonuniform Illumination**

It shows how to correct nonuniform illumination in an image to make it easy to identify individual grains of rice in the image. You can then learn about the characteristics of the grains and easily compute statistics for all the grains in the image.

* Read Image

### Use Morphological Opening to Estimate the Background

Notice that the background illumination is brighter in the center of the image than at the bottom. Use imopen to estimate the background illumination.

### Subtract the Background Image from the Original Image

### Increase the Image Contrast

I3 = imadjust(I2);

imshow(I3);

### Threshold the Image

Create a new binary image by thresholding the adjusted image. Remove background noise with bwareaopen.

### Identify Objects in the Image

The function bwconncomp finds all the connected components (objects) in the binary image. The accuracy of your results depend on the size of the objects, the connectivity parameter ,and whether or not any objects are touching (in which case they may be labeled as one object). Some of the rice grains in bw are touching.

### Examine One Object

Each distinct object is labeled with the same integer value. Show the grain that is the 50th connected component.

### View All Objects

One way to visualize connected components is to create a label matrix and then display it as a pseudo-color indexed image.

Use labelmatrix to create a label matrix from the output of bwconncomp. Note that labelmatrix stores the label matrix in the smallest numeric class necessary for the number of objects.

### Compute Area of Each Object

Each rice grain is one connected component in the cc structure. Use regionprops on cc to get the area.

### Compute Area-based Statistics

Create a new vector grain\_areas, which holds the area measurement for each grain.

### Create Histogram of the Area

histogram(grain\_areas)

* + **Colour-based Segmentation such as** [**K-means clustering**](https://in.mathworks.com/help/images/examples/color-based-segmentation-using-k-means-clustering.html?prodcode=IP&language=en)

Convert Image from RGB Color Space to L\*a\*b\* Color Space How many colors do you see in the image if you ignore variations in brightness? There are three colors: white, blue, and pink. Notice how easily you can visually distinguish these colors from one another. The L\*a\*b\* color space (also known as CIELAB or CIE L\*a\*b\*) enables you to quantify these visual differences. The L\*a\*b\* color space is derived from the CIE XYZ tristimulus values. The L\*a\*b\* space consists of a luminosity layer 'L\*', chromaticity-layer 'a\*' indicating where color falls along the red-green axis, and chromaticity-layer 'b\*' indicating where the color falls along the blue-yellow axis. All of the color information is in the 'a\*' and 'b\*' layers. You can measure the difference between two colors using the Euclidean distance metric. Convert the image to L\*a\*b\* color space using makecform and applycform.

**cform = makecform('srgb2lab');**

**lab\_he = applycform(he,cform);**

* + **Transform methods such as** [**watershed segmentation**](https://in.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html?prodcode=IP&language=en)

It shows how to use watershed segmentation to separate touching objects in an image. The watershed transform is often applied to this problem. The watershed transform finds "catchment basins" and "watershed ridge lines" in an image by treating it as a surface where light pixels are high and dark pixels are low. Segmentation using the watershed transform works better if you can identify, or "mark," foreground objects and background locations. Marker-controlled watershed segmentation follows this basic procedure:

* Compute a segmentation function. This is an image whose dark regions are the objects you are trying to segment.
* Compute foreground markers. These are connected blobs of pixels within each of the objects.
* Compute background markers. These are pixels that are not part of any object.
* Modify the segmentation function so that it only has minima at the foreground and background marker locations.
* Compute the watershed transform of the modified segmentation function.

This example highlights many different Image Processing Toolbox functions, including **fspecial, imfilter, watershed, label2rgb, imopen, imclose, imreconstruct, imcomplement, imregionalmax, bwareaopen, graythresh, and imimposemin.**

* + **Texture methods such as** [**texture filters**](https://in.mathworks.com/help/images/examples/texture-segmentation-using-texture-filters.html)

It shows how to use texture segmentation to identify regions based on their texture. Your goal is to segment two kinds of fabric in an image using texture filters.

### Read Image

### Create Texture Image

### Create Rough Mask for the Bottom Texture

### Use Rough Mask to Segment the Top Texture

### Display Segmentation Results

### **Using Other Texture Filters in Segmentation**

Instead of entropyfilt, you can use stdfilt and rangefilt with other morphological functions to achieve similar segmentation results.

S = stdfilt(I,nhood);

imshow(rescale(S))

R = rangefilt(I,ones(5));

imshow(R)

**Clustering:**

Clustering can be considered the most important unsupervised learning problem. A cluster is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

Cluster analysis is an [unsupervised learning](https://in.mathworks.com/discovery/unsupervised-learning.html) method and an important task in exploratory data analysis. Popular clustering algorithms include:

* **Hierarchical clustering**: builds a multilevel hierarchy of clusters by creating a cluster tree

hierarchical clustering generally fall into two types:

**Agglomerative:** This is a **"bottom up"** approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

**Divisive:** This is a **"top down"** approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

**Algorithm Description**:

To perform agglomerative hierarchical cluster analysis on a data set using Statistics and Machine Learning Toolbox functions, follow this procedure: Find the similarity or dissimilarity between every pair of objects in the data set. In this step, you calculate the distance between objects using the pdist function. The pdist function supports many different ways to compute this measurement. See Similarity Measures for more information. Group the objects into a binary, hierarchical cluster tree. In this step, you link pairs of objects that are in close proximity using the linkage function. The linkage function uses the distance information generated in

* To determine the proximity of objects to each other. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed. See Linkages for more information.
* Determine where to cut the hierarchical tree into clusters. In this step, you use the cluster function to prune branches off the bottom of the hierarchical tree, and assign all the objects below each cut to a single cluster.
* This creates a partition of the data. The cluster function can create these clusters by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.
* **k-Means clustering**: partitions data into k distinct clusters based on distance to the centroid of a cluster. k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes. The algorithm has a loose relationship to the k-nearest neighbor classifier, a popular machine learning technique for classification that is often confused with k-means because of the k in the name. One can apply the 1-nearest neighbor classifier on the cluster centers obtained by k-means to classify new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.
* **Gaussian mixture models**: models clusters as a mixture of multivariate normal density components.

A **mixture model** is a probabilistic model for representing the presence of subpopulations within an overall population, without requiring that an observed data set should identify the sub-population to which an individual observation belongs. Formally a mixture model corresponds to the mixture distribution that represents the probability distribution of observations in the overall population.

However, while problems associated with **"mixture distributions"** relate to deriving the properties of the overall population from those of the sub-populations, **"mixture models"** are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population identity information.

Some ways of implementing mixture models involve steps that attribute postulated sub-population-identities to individual observations (or weights towards such sub-populations), in which case these can be regarded as types of unsupervised learning or clustering procedures.

However, not all inference procedures involve such steps. Mixture models should not be confused with models for compositional data, i.e., data whose components are constrained to sum to a constant value (1, 100%, etc.).

However, compositional models can be thought of as mixture models, where members of the population are sampled at random. Conversely, mixture models can be thought of as compositional models, where the total size reading population has been normalized to 1.

* **Self-organizing maps**: uses neural networks that learn the topology and distribution of the data.

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction.

Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

A self-organizing map showing U.S. Congress voting patterns. The input data was a table with a row for each member of Congress, and columns for certain votes containing each member's yes/no/abstain vote.

The SOM algorithm arranged these members in a two-dimensional grid placing similar members closer together.

The **first plot** shows the grouping when the data are split into two clusters.

The **second plot** shows average distance to neighbours: larger distances are darker.

The **third plot** predicts Republican (red) or Democratic (blue) party membership.

The other plots each overlay the resulting map with predicted values on an input dimension: **red** means a predicted '**yes**' vote on that bill, **blue** means a '**no**' vote. The plot was created in Synapse. This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling.

The artificial neural network introduced by the Finnish professor Teuvo Kohonen in the 1980s is sometimes called a Kohonen map or network.

The Kohonen net is a computationally convenient abstraction building on biological models of neural systems from the 1970s and morphogenesis models dating back to Alan Turing in the 1950s.

Like most artificial neural networks, SOMs operate in **two modes**: **training** and **mapping**.

"Training" builds the map using input examples (a competitive process, also called vector).

The distinguishing feature of each of these algorithms is the metric to measure similarity.

Cluster analysis is used in bioinformatics for sequence analysis and genetic clustering; in data mining for sequence and [pattern mining](https://in.mathworks.com/discovery/pattern-recognition.html); in medical imaging for image segmentation; and in computer vision for object recognition.

**Clustering – Flow:**

Segmentation file

Clustering

Cluster Index Image

Region-1

Region-2

Region-3

**5. Feature Extraction:**

Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector.

This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector).

Determining a subset of the initial features is called feature selection.

The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations.

Feature extraction is related to dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features. Determining a subset of the initial features is called feature selection.

The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

**Feature Selection:**

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

Feature selection techniques are used for **reasons**: simplification of models to make them easier to interpret by researchers/users,

* shorter training times, to avoid the curse of dimensionality, enhanced generalization by reducing overfitting (formally, reduction of variance
* The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information.
* Redundant or irrelevant features are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.
* Feature selection techniques should be distinguished from feature extraction.
* Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features.
* Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points).
* Archetypal cases for the application of feature selection include the analysis of written texts and DNA microarray data, where there are many thousands of features, and a few tens to hundreds of samples.

**Training dataset**

A training dataset is a dataset of examples used for learning, that is to fit the parameters (e.g., weights) of, for example, a classifier.

Most approaches that search through training data for empirical relationships tend to overfit the data, meaning that they can identify apparent relationships in the training data that do not hold in general.

**Test dataset**

A test dataset is a dataset that is independent of the training dataset, but that follows the same probability distribution as the training dataset. If a model fit to the training dataset also fits the test dataset well, minimal overfitting has taken place (see figure below).

A better fitting of the training dataset as opposed to the test dataset usually points to overfitting. **A test set** is therefore a set of examples used only to assess the performance (i.e. generalization) of a fully specified classifier.

**Validation dataset**

A validation dataset is a set of examples used to tune the hyperparameters of a classifier. In artificial neural networks, an hyperparameter is, for example, the number of hidden units.

It, as well as the testing set (as mentioned above), should follow the same probability distribution as the training dataset. In order to avoid overfitting, when any classification parameter needs to be adjusted, it is necessary to have a validation dataset in addition to the training and test datasets.

For example, if the most suitable classifier for the problem is sought, the training dataset is used to train the candidate algorithms, the validation dataset is used to compare their performances and decide which one to take and, finally, the test dataset is used to obtain the performance characteristics such as accuracy, sensitivity, specificity, F-measure, and so on.

The validation dataset functions as a hybrid: it is training data used by testing, but neither as part of the low-level training nor as part of the final testing.

The basic process of using a validation dataset for model selection (as part of training dataset, validation dataset, and test dataset) is: Since our goal is to find the network having the best performance on new data, the simplest approach to the comparison of different networks is to evaluate the error function using data which is independent of that used for training. Various networks are trained by minimization of an appropriate error function defined with respect to a training data set.

The performance of the networks is then compared by evaluating the error function using an independent validation set, and the network having the smallest error with respect to the validation set is selected.

**Selection of a validation dataset**

It is part of the training dataset can be set aside and used as a validation set: this is known as the holdout method[citation needed]. Common proportions are 70%/30% training/validation[citation needed].

Cross-validation Alternatively, the hold out process can be repeated, repeatedly partitioning the original training dataset into a training dataset and a validation dataset: this is known as cross-validation.

These repeated partitions can be done in various ways, such as dividing into 2 equal datasets and using them as training/validation, and then validation/training, or repeatedly selecting a random subset as a validation dataset.

**Feature Extraction – Flow:**

Cluster Regions

Feature Extraction

Train Features

Test Feature

**6. Classification:**

Classification is a type of supervised machine learning in which an algorithm “learns” to classify new observations from examples of labeled data. To explore classification models interactively, use the Classification Learner app. For greater flexibility, you can pass predictor or feature data with corresponding responses or labels to an algorithm-fitting function in the command-line interface.To train regression models, such as logistic regression, regression trees, Gaussian process regression, and support vector regression.

**SVM Classification:**

* A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors.
* SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

Simple generic function that takes two labelled classes and trains binary SVM classifier. Has very basic example code to call **SVM classifier** and **train SVM** on labelled data.

It has two basic types of classification:

* + Binary classifier
  + Multi class classifier

**Binary Classification:**

**Binary or binomial classification** is the task of classifying the elements of a given set into two groups (predicting which group each one belongs to) on the basis of a classification rule. Contexts requiring a decision as to whether or not an item has some qualitative property, some specified characteristic, or some typical binary classification include:

* **Medical testing** to determine if a patient has certain disease or not – the classification property is the presence of the disease.
* A "pass or fail" test method or quality control in factories, i.e. deciding if a specification has or has not been met – a Go/no go classification.
* **Information retrieval**, namely deciding whether a page or an article should be in the result set of a search or not – the classification property is the relevance of the article, or the usefulness to the user.

Binary classification is dichotomization applied to practical purposes, and in many practical binary classification problems, the two groups are not symmetric – rather than overall accuracy, the relative proportion of different types of errors is of interest. For example, in medical testing, a **false positive** (detecting a disease when it is not present) is considered differently from a **false negative.**

Some of the methods commonly used for binary classification are:

* Decision trees
* Random forests
* Bayesian networks
* Support vector machines
* Neural networks
* Logistic regression

Each classifier is best in only a select domain based upon the number of observations, the dimensionality of the feature vector, the noise in the data and many other factors. For example random forests perform better than SVM classifiers for 3D point clouds.

**Multiclass classification:**

In machine learning, multiclass or multinomial classification is the problem of classifying instances into one of three or more classes. (Classifying instances into one of the two classes is called binary classification.) While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies. Multiclass classification should not be confused with multi-label classification, where multiple labels are to be predicted for each instance.

**Transformation to Binary:**

It follws two methods,

* One-vs-rest
* One-vs-one

**Extension from Binary:**

* Neural networks

##### Extreme learning machines

#### k-nearest neighbours

#### Naive Bayes

#### Decision trees

#### Support vector machines

### Hierarchical classification

**7. Performance:**

Classification models in machine learning are evaluated for their performance by common performance measures.

This function calculates the following performance measures:

* Accuracy
* Sensitivity
* Specificity
* Precision
* Recall
* F-Measure
* G-mean.

The function and description of the arguments are:

**EVAL = Evaluate(ACTUAL,PREDICTED)**

Input:   
ACTUAL = Column matrix with actual class labels of the training examples   
PREDICTED = Column matrix with predicted class labels by the classification model   
Output:   
EVAL = Row matrix with all the performance measures

**Accuracy:**

Accuracy can be defined as the amount of uncertainty in a measurement with respect to an absolute standard. Accuracy specifications usually contain the effect of errors due to gain and offset parameters. Offset errors can be given as a unit of measurement such as volts or ohms and are independent of the magnitude of the input signal being measured. An example might be given as ±1.0 millivolt (mV) offset error, regardless of the range or gain settings. In contrast, gain errors do depend on the magnitude of the input signal and are expressed as a percentage of the reading, such as ±0.1%. Total accuracy is therefore equal to the sum of the two: ±(0.1% of input +1.0 mV).

conditions: input 0-10V, Accuracy = ±(0.1% of input + 1mV)

**Precision:**

Precision describes the reproducibility of the measurement. For example, measure a steady state signal many times. In this case if the values are close together then it has a high degree of precision or repeatability. The values do not have to be the true values just grouped together. Take the average of the measurements and the difference is between it and the true value is accuracy.

**Sensitivity:**

**Sensitivity** (also called the true positive rate, the recall, or probability of detection[1] in some fields) measures the proportion of positives that are correctly identified as such (e.g. the percentage of sick people who are correctly identified as having the condition).

Sensitivity -- a measure of the smallest signal the instrument can measure. Usually, this is defined at the lowest range setting of the instrument.

Sensitivity is an absolute quantity, the smallest absolute amount of change that can be detected by a measurement. Consider a measurement device that has a ±1.0 volt input range and ±4 counts of noise, if the A/D converter resolution is 212 the peak-to-peak sensitivity will be ±4 counts x (2 ÷ 4096) or ±1.9mV p-p. This will dictate how the sensor responds. For example, take a sensor that is rated for 1000 units with an output voltage of 0-1 volts (V). This means that at 1 volt the equivalent measurement is 1000 units or 1mV equals one unit. However the sensitivity is 1.9mV p-p so it will take two units before the input detects a change.

**Formula:**  A/(A + C) × 100 10/15 × 100 = 67%

**Specificity**

**Specificity** (also called the true negative rate) measures the proportion of negatives that are correctly identified as such (e.g. the percentage of healthy people who are correctly identified as not having the condition).

**Formula:**  D/(D + B) × 100 45/85 × 100 = 53%

**Resolution**

Resolution can be expressed in two ways:

* It is the ratio between the maximum signal measured to the smallest part that can be resolved - usually with an analog-to-digital (A/D) converter.
* It is the degree to which a change can be theoretically detected, usually expressed as a number of bits. This relates the number of bits of resolution to the actual voltage measurements.

**LITERATURE SURVEY**

**Title 1: Advances in hyper spectral image classification: Earth monitoring with statistical learning methods**

**Author: G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, Year: Jan 2014.**

The technological evolution of optical sensors over the last few decades has provided remote sensing analysts with rich spatial, spectral, and temporal information. In particular, the increase in spectral resolution of hyper spectral images (HSIs) and infrared sounders opens the doors to new application domains and poses new methodological challenges in data analysis. HSIs allow the characterization of objects of interest (e.g., land-cover classes) with unprecedented accuracy, and keeps inventories up to date. Improvements in spectral resolution have called for advances in signal processing and exploitation algorithms. This article focuses on the challenging problem of hyper spectral image classification, which has recently gained in popularity and attracted the interest of other scientific disciplines such as machine learning, image processing, and computer vision. In the remote sensing community, the term classification is used to denote the process that assigns single pixels to a set of classes, while the term segmentation is used for methods aggregating pixels into objects and then assigned to a class.

**Advanced Regularized Image Classification**

Before HSI, most of the classifiers used in remote sensing were parametric, such as Gaussian maximum likelihood or linear discriminant analysis. These methods, based on the estimate of the covariance matrix, were successful when dealing with early multi-spectral images, whose dimensionality was usually comprised between four and ten bands. HSI changed the rules, as the increased dimensionality of pixels raised to hundreds. Standard parametric methods became either unfeasible or unreliable, since estimating the class-covariance matrices requires many labeled samples, which are usually not available. For that reason, research turned to include regularization, either explicitly through Tikhonov's terms in the involved covariance matrices, or by perform classification in a subspace of reduced dimensionality

Terrain mapping, computer vision, geophysical image processing, image classification, image resolution, image segmentation, learning (artificial intelligence), optical sensors, statistical analysis

Land-cover classes, hyperspectral image classification, earth monitoring, statistical learning methods, optical sensors, remote sensing analysts, spatial information, spectral information, temporal information, spectral resolution, infrared sounders, HSI, signal processing algorithm, signal exploitation algorithm, machine learning, image processing, computer vision, data analysis

**Advantage:**

Classification used to assigns pixels into objects.

**Disadvantage:**

Use signal processing and exploitation algorithms

**Title 2: A batch-mode active learning technique based on multiple**

**uncertainty for SVM classifier**

**Author: S. Patra and L. Bruzzone**

**Year: May 2012.**

We present a novel batch-mode active learning technique for solving multiclass classification problems by using the support vector machine classifier with the one-against-all architecture. The uncertainty of each unlabelled sample is measured by defining a criterion which not only considers the smallest distance to the decision hyper planes but also takes into account the distances to other hyper planes if the sample is within the margin of their decision boundaries. To select batch of most uncertain samples from all over the decision region, the uncertain regions of the classifiers are partitioned into multiple parts depending on the number of geometrical margins of binary classifiers passing on them. Then, a balanced number of most uncertain samples are selected from each part. To minimize the redundancy and keep the diversity among these samples, the kernel *k*-means clustering algorithm is applied to the set of uncertain samples, and the representative sample (medoid) from each cluster is selected for labelling. The effectiveness of the proposed method is evaluated by comparing it with other batch-mode active learning techniques existing in the literature. Experimental results on two different remote sensing data sets confirmed the effectiveness of the proposed technique.

IN THE literature, many supervised methods have been proposed for classification of remotely sensed data. The classification results obtained by these methods rely on the quality of the labeled samples used for learning. To obtain proper labeled samples is usually expensive and time consuming. Moreover, the manual selection of the training samples often introduces redundancy into the training set of the classifier, thus slowing the training phase considerably without adding relevant information. In order to reduce the cost of labeling and optimize the performance of the classifier, the training set should be as small as possible by avoiding redundant samples and including only most informative patterns (which have highest training utility). Active learning is an approach that addresses this problem. The learning process repeatedly queries unlabeled samples to select the most informative patterns and updates the training set on the basis of a supervisor who attributes the labels to the selected unlabeled samples.

Support vector machines, geophysical image processing, geophysical techniques, image classification, learning (artificial intelligence), pattern clustering, remote sensing

Remote sensing data set, batch-mode active learning technique, SVM classifier, multiclass classification problem, support vector machine, one-against-all architecture, unlabeled sample uncertainty measurement, decision hyperplane, decision boundary, decision region, geometrical margin, binary classifiers, redundancy minimization, kernel k-means clustering algorithm, representative sample, medoid.

**Advantage:**

Minimum number of redundant data.

**Disadvantage:**

Balanced no of data only selected.

**Title 3: Dimension reduction using spatial and spectral regularized**

**local discriminant embedding for hyper spectral image classification**

**Author: Y. Zhou, J. Peng, and C. Chen**

**Year: Feb. 2015.**

Dimension reduction (DR) is a necessary and helpful pre-processing for hyperspectral image (HSI) classification. We propose a spatial and spectral regularized local discriminant embedding (SSRLDE) method for DR of hyperspectral data. In SSRLDE, hyperspectral pixels are first smoothed by the multiscale spatial weighted mean filtering. Then, the local similarity information is described by integrating a spectral-domain regularized local preserving scatter matrix and a spatial-domain local pixel neighbourhood preserving scatter matrix. Finally, the optimal discriminative projection is learned by minimizing a local spatial-spectral scatter and maximizing a modified total data scatter. Experimental results on benchmark hyperspectral data sets show that the proposed SSRLDE significantly outperforms the state-of-the-art DR methods for HSI classification.

Hyperspectral images (HSIs) are widely used in environmental mapping, geological research, crop analysis, and mineral identification .

These applications often require to classify each pixel in the scene. Because there are a huge number of features (or spectral bands) with only limited training samples available, HSI classification becomes a challenging task. A large number of spectral bands provide rich information for classifying various materials in the scene. However, with limited training samples, the performance of classifiers deteriorates as the dimensionality increases. High-dimensionality data processing also requires huge computational resources and storage capacity .Meanwhile, the spectral bands are often correlated, and not all of them are useful for the specific classification task. Therefore, to achieve an excellent classification performance, a dimension reduction (DR) procedure is required before training the classifiers.

Geophysical image processing, geophysical techniques, hyperspectral imaging, image classification

Spatial regularized local discriminant embedding dimension reduction, hyperspectral image classification preprocessing, HSI classification preprocessing, spectral regularized local discriminant embedding dimension reduction, SSRLDE method, hyperspectral data DR, hyperspectral pixel, multiscale spatial weighted mean filtering, local similarity information, spectral-domain regularized local preserving scatter matrix, spatial-domain local pixel neighborhood preserving scatter matrix, optimal discriminative projection, local spatial-spectral scatter minimization, modified total data scatter maximization, benchmark hyperspectral data set, state-of-the-art DR method

**Advantage:**

Maximum number of modified data scatters.

**Disadvantage:**

Minimum number of local spatial-spectral scatter.

**Title 4: Dimensionality reduction of hyper spectral images with sparse**

**Discriminant embedding**

**Author: H. Huang and M. Yang**

**Year: Sep. 2015.**

Sparse manifold learning has drawn more and more attentions recently, and sparsity preserving projections (SPP) has been proposed, which inherits the advantages of sparse reconstruction. However, SPP only focuses on the sparse structure, ignoring the discriminant information of labelled samples. We proposed a new supervised dimensionality reduction method, which is called sparse discriminant embedding (SDE), for hyperspectral image (HSI) classification. SDE utilizes the merits of both intermanifold structure and sparsity property. It not only preserves the sparse reconstructive relations through l1-graph but also enhances the intermanifold separability of data, and the discriminating power of SDE is further improved than SPP. Experiments on two real HSIs collected by the Airborne Visible/Infrared Imaging Spectrometer and Reflective Optics System Imaging Spectrometer sensors are performed to demonstrate the effectiveness of the proposed SDE method.

The development of hyperspectral remote sensing imaging technology, the use of hyperspectral images (HSIs) is becoming more and more widespread, such as target detection and land cover investigation. A pixel in HSI is typically a high-dimensional vector of intensities as a function of wavelength with a large spectral range and a high spectral resolution, which facilitates the superior discrimination of land cover types. However, classification of HSI still faces some challenges, among which are the following

* the high number of spectral channels;
* the spatial variability of spectral signature; and
* The high cost of true sample labelling. In particular, the high number of spectral channels and the low number of training samples pose the problem of the curse of dimensionality.

Therefore, the most important and urgent issue is how to reduce the number of bands largely, but without loss of information

Data reduction, hyperspectral imaging, image classification, image processing

Hyperspectral image dimensionality reduction, sparse discriminant embedding, sparse manifold learning, sparsity preserving projection, sparse reconstruction, sparse structure, supervised dimensionality reduction method, hyperspectral image classification, sparsity property, data intermanifold separability, airborne visible-infrared imaging spectrometer sensor, reflective optics system imaging spectrometer sensor

**Advantage:**

Ignore discriminant information**.**

**Disadvantage:**

Contains unlabelled samples.

**Title 5: Active learning with Gaussian process classifier for hyper spectral**

**Image classification**

**Author: S. Sun, P. Zhong, H. Xiao, and R. Wang**

**Year: Apr. 2015**

Gaussian process (GP) classifiers represent a powerful and interesting theoretical framework for the Bayesian classification of hyper spectral images. However, the collection of labelled samples is time consuming and costly for hyper spectral data, and the training samples available are often not enough for an adequate learning of the GP classifier. Moreover, the computational cost of performing inference using GP classifiers scales cubically with the size of the training set. To address the limitations of GP classifiers for hyper spectral image classification, reducing the label cost and keeping the training set in a moderate size, this paper introduces an active learning (AL) strategy to collect the most informative training samples for manual labelling. First, we propose three new AL heuristics based on the probabilistic output of GP classifiers aimed at actively selecting the most uncertain and confusing candidate samples from the unlabelled data. Moreover, we develop an incremental model updating scheme to avoid the repeated training of the GP classifiers during the AL process. The proposed approaches are tested on the classification of two real world hyper spectral data. Comparison with random sampling method reveals a better accuracy gain and faster convergence with the number of queries, and comparison with recent active learning approaches shows a competitive performance. Experimental results also verified the efficiency of the incremental model updating scheme.

The recent development of remote sensing instruments, the analysis of hyperspectral images is an active research area in remote sensing community. Exhibiting different spectral signatures, hyperspectral imagery is a well-suited technology for accurate image classification, which is an important task in many application domains (monitoring and management of the environment, precision agriculture, etc.). Given a set of observations (i.e., pixel vectors in a hyperspectral image), the goal of classification is to assign a unique label to each pixel vector so that it is well defined by a given class . Two difficulties should be addressed in hyperspectral image classification. First, the intrinsic properties of hyperspectral images need to be addressed specifically because conventional classification algorithms made for multispectral images do not adapt well to the analysis of hyperspectral images. Obstacles, such as the Hughes phenomenon , come out as the data dimensionality increases. Another well-known difficulty is the limited availability of training data, and the collection of labeled samples is generally difficult, expensive, and time consuming.

Bayes methods, Gaussian processes, geophysical image processing, hyperspectral imaging, image classification, learning (artificial intelligence), remote sensing

Gaussian process classifier, hyperspectral image classification, hyperspectral image Bayesian classification, label cost reduction, active learning strategy, informative training samples, manual labeling, active learning heuristics, GP classifier probabilistic output, incremental model updating scheme, real world hyperspectral data

**Advantage:**

Better accuracy and faster convergence.

**Disadvantage:**

Most uncertain and confusing samples from unlabelled data.

**Title 6: Wavelet-domain multiview active learning for spatial-spectral**

**hyper spectral image classification**

**Author: X. Zhou, S. Prasad, and M. M. Crawford**

**Year: Sep. 2016**

Active learning (AL) has been shown to be effective for strategic selection of training samples to support classification of hyper spectral imagery. It is well understood that the performance of classification can further be improved by utilizing the spatial information in hyper spectral images. In this paper, we propose a new wavelet-based Multiview AL approach for hyper spectral image classification. Specifically, a three-dimensional redundant wavelet transform (3D-RDWT) is used to generate multiple views that are then integrated in a Multiview AL framework. The spatial features generated via 3D-RDWT not only provide sufficient views for Multiview AL, but are also less sensitive to additive noise. Within this framework, we also propose new query criteria that result in effective AL. An intersection-based query criterion is proposed to reduce the redundancy within the contention pool. A singularity-based criterion is also used to identify informative pixels by taking spatial information into account when selecting samples. The proposed method is evaluated on four real-world hyper spectral datasets, and the experimental results demonstrate the efficacy of the proposed method compared with traditional AL methods.

Geophysical techniques, hyperspectral imaging, image classification, wavelet transforms

Wavelet-domain multiview active learning, spatial-spectral hyperspectral image classification, active learning, wavelet-based multiview active learning approach, three-dimensional redundant wavelet transform, 3D-RDWT, additive noise, intersection-based query criterion, singularity-based criterion, informative pixels, real-world hyperspectral datasets

**Advantages:**

Reduce redundant samples.

Generate Multiple views of samples.

**Disadvantage:**

Redundancy with contention pool of samples.

**Title 7: Local-manifold-learning based graph construction for**

**Semi supervised hyper spectral image classification**

**Author: L. Ma, M. Crawford, X. Yang, and Y. Guo**

**Year: May 2015**

Active learning (AL) has been shown to be effective for strategic selection of training samples to support classification of hyperspectral imagery. It is well understood that the performance of classification can further be improved by utilizing the spatial information in hyperspectral images. In this paper, we propose a new wavelet-based multiview AL approach for hyperspectral image classification. Specifically, a three-dimensional redundant wavelet transform (3D-RDWT) is used to generate multiple views that are then integrated in a multiview AL framework. The spatial features generated via 3D-RDWT not only provide sufficient views for multiview AL, but are also less sensitive to additive noise. Within this framework, we also propose new query criteria that result in effective AL. An intersection-based query criterion is proposed to reduce the redundancy within the contention pool. A singularity-based criterion is also used to identify informative pixels by taking spatial information into account when selecting samples. The proposed method is evaluated on four real-world hyperspectral datasets, and the experimental results demonstrate the efficacy of the proposed method compared with traditional AL methods.

FOR hyperspectral image classification, supervised classifiers require a large quantity of labeled data due to the high-dimensional spectral data vector. However, labeled instances are often difficult, costly, or time consuming to obtain. Since unlabeled data are relatively easy to collect, semisupervised learning (SSL) that utilizes both labeled and unlabeled data is widely employed to solve the small-size sample problem.

Computational geometry, geophysical image processing, graph theory, hyperspectral imaging, image classification, learning (artificial intelligence)

LTSA, graph construction method, local tangent space alignment, LML method, AVIRIS hyperspectral data, Hyperion, multiple submanifolds, complex geometry, local MLmethod-based sparse graph, SSL algorithms, graph-based semisupervised learning, semisupervised hyperspectral image classification, local manifold learning-based graph construction

**Advantage:**

It has four real world hyperspectral datasets.

**Disadvantage:**

It has redundant wavelet transform samples.

**Title 8: A survey of active learning algorithms for supervised remote sensing**

**image classification**

**Author: D. Tuia, M. Volpi, L. Copa, M. F. Kanevski, and J. Munoz-Mari**

**Year: Jun. 2011**

Defining an efficient training set is one of the most delicate phases for the success of remote sensing image classification routines. The complexity of the problem, the limited temporal and financial resources, as well as the high intraclass variance can make an algorithm fail if it is trained with a suboptimal dataset. Active learning aims at building efficient training sets by iteratively improving the model performance through sampling. A user-defined heuristic ranks the unlabelled pixels according to a function of the uncertainty of their class membership and then the user is asked to provide labels for the most uncertain pixels. This reviews and tests the main families of active learning algorithms: committee, large margin, and posterior probability-based. For each of them, the most recent advances in the remote sensing community are discussed and some heuristics are detailed and tested. Several challenging remote sensing scenarios are considered, including very high spatial resolution and hyperspectral image classification. Finally, guidelines for choosing the good architecture are provided for new and/or unexperienced user.

**Introduction**

Nowadays, the recourse to statistical learning models is a common practice for remote sensing data users; models such as support vector machines or neural networks are considered as state of the art algorithms for the classification of landuse using new generation satellite imagery . Applications of such models to very high spatial or spectral resolution have proven their efficiency for handling remote sensing data.

Remote sensing, geophysical image processing, image classification, learning (artificial intelligence)

User-defined heuristic, active learning algorithms, supervised remote sensing image classification, suboptimal dataset, remote sensing community, hyperspectral image classification

**Advantage:**

It includes high spatial resolution and hyperspectral image classification.

**Disadvantage:**

It provides most uncertain pixels.

**Title 9: A novel semi supervised active-learning algorithm for hyper spectral**

**Image classification**

**Author: Z.Wang, B. Du, L. Zhang, L. Zhang, and X. Jia**

**Year: Jun. 2017**

Less training samples are a challenging problem in hyperspectral image classification. Active learning and semisupervised learning are two promising techniques to address the problem. Active learning solves the problem by improving the quality of the training samples, while semisupervised learning solves the problem by increasing the quantity of the training samples. However, they pay too much attention to the discriminative information in the unlabelled data, leading to information bias to train supervised models, and much more effort to label samples. Therefore, a method to discover representativeness and discriminativeness by semi supervised active learning is proposed. It takes advantages of both active learning and semi supervised learning. The representativeness and discriminativeness are discovered with a labelling process based on a supervised clustering technique and classification results. Specifically, the supervised clustering results can discover important structural information in the unlabelled data, and the classification results are also highly confidential in the active-learning process. With these clustering results and classification results, we can assign pseudo labels to the unlabelled data. Meanwhile, the unlabelled samples that cannot be assigned with pseudo labels with high confidence at each iteration are regarded as candidates in active learning. The methodology is validated on four hyperspectral data sets.

To the development of spectral imaging techniques in recent years, hyperspectral remote-sensing images, which contain a lot of information in hundreds of continuous and narrow spectral bands, have been widely applied in the geosciences, in application, such as target detection, change detection, image fusion, as well as classification. Hyperspectral remote-sensing image classification is the crucial processing step in many applications Hence, it has drawn great interests in remote-sensing fields. As to hyperspectral image classification, the foremost task is that: given a hyperspectral image data set, how can we label only limited samples to train an effective classifier and quickly interpret the image. To train a good supervised classification model,

There are two key factors:

* The quality and
* The quantity of the training samples.

However, labelling a large number of samples is time-consuming and expensive

Geophysical techniques, hyperspectral imaging, image classification

Semi supervised active-learning algorithm, hyperspectral image classification, supervised models, representativeness method, discriminativeness method, labelling process, supervised clustering technique, supervised clustering classification, structural information

**Advantage:**

It takes both semi-supervised and active learning.

**Disadvantage:**

Unlabelled samples cannot be assigned with pseudo labels.

**Title 10:**  **Definition of effective training sets for super vised classification of**

**remote sensing images by a novel cost-sensitive active learning method**

**Author: B. Demir, L. Minello, and L. Bruzzone**

**Year: Feb. 2014**

The proposes of a novel cost-sensitive active learning (CSAL) method to the definition of reliable training sets for the classification of remote sensing images with support vector machines. Unlike standard active learning (AL) methods, the proposed CSAL method redefines AL by assuming that the labelling cost of samples during ground survey is not identical, but depends on both the samples accessibility and the traveling time to the considered locations. The proposed CSAL method selects the most informative samples on the basis of three criteria: 1) uncertainty; 2) diversity; and 3) labelling cost. The labelling cost of the samples is modelled by a novel cost function that exploits ancillary data such as the road network map and the digital elevation model of the considered area. In the proposed method, the three criteria are applied in two consecutive steps. In the first step, the most uncertain samples are selected, whereas in the second step the uncertain samples that are diverse and have low labelling cost are chosen. In order to select the uncertain samples that optimize the diversity and cost criteria, we propose two different optimization algorithms. The first algorithm is defined on the basis of a sequential forward selection optimization strategy, whereas the second one relies on a genetic algorithm.

The generation of land-cover maps is one of the most common applications in remote sensing image analysis and is usually achieved by supervised classification techniques. Such techniques require the availability of reliable ground reference samples to be used in the learning phase of the classification algorithm. Reliability of the labeled training samples depends on both the quantity and quality of the available samples. In remote sensing, the quality of samples is affected by two main issues:

1) the capability to model the spatial variability of the spectral signatures of the land-cover classes and

2) the high spatial correlation among the training samples collected in neighboring locations of the same area (that reduces the information conveyed by training samples with respect to the case of independent samples) . Thus, the collection of a sufficient number of reliable labeled samples is time-consuming, complex, and costly in operational scenarios, and can significantly affect the final accuracy of produced land-cover maps.

Terrain mapping, digital elevation models, genetic algorithms, geophysical image processing, image classification, land cover, learning (artificial intelligence), support vector machines

Cost-sensitive active learning method, land-cover maps, effective training sets, CSAL method, genetic algorithm, sequential forward selection optimization strategy, cost criteria, uncertain samples, digital elevation model, road network map, ancillary data, cost function, labeling cost, sample accessibility, traveling time, ground survey, standard active learning methods, support vector machines, supervised remote sensing image classification, reliable training sets

**Advantage:**

It proposes reliable training sets.

**Disadvantage:**

Uncertain samples optimize the diversity and cost criteria.

**TESTING OF PRODUCT**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error. Testing is vital to the success of the system.  System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.  The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways.  Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.  Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done.  Unit testing becomes verification efforts on the smallest unit of software design in the module.  This is also known as ‘module testing’.  The modules of the system are tested separately.  This testing is carried out during the programming itself.  In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module.  There are some validation checks for the fields.  For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included.  It is very easy to find error and debug the system.

**INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function.  Integrated testing is systematic testing that can be done with sample data.  The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.  Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**BLACK BOX TESTING:**

* + Black box testing is done to find incorrect or missing function
  + Interface error
  + Errors in external database access
  + Performance errors
  + Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behavior of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

# **USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

# **OUTPUT TESTING:**

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.  The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**System Implementation:**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. The people are not sure that the software is meant to make their job easier.

* The active user must be aware of the benefits of using the system
* Their confidence in the software built up
* Proper guidance is impaired to the user so that he is comfortable in using the application

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not running on the server, the actual processes will not take place.

**User Training:**

To achieve the objectives and benefits expected from the proposed system it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for education and training is more and more important.

Education is complementary to training. It brings life to formal training by explaining the background to the resources for them. Education involves creating the right atmosphere and motivating user staff. Education information can make training more interesting and more understandable.

**Training on the Application Software:**

After providing the necessary basic training on the computer awareness, the users will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design, type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. This training may be different across different user groups and across different levels of hierarchy.

**Operational Documentation:**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. A documentation providing the whole operations of the system is being developed. Useful tips and guidance is given inside the application itself to the user. The system is developed user friendly so that the user can work the system from the tips given in the application itself.

**System Maintenance:**

The maintenance phase of the software cycle is the time in which software performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affect a system which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system.

**Corrective Maintenance:**

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called Corrective Maintenance.

**Adaptive Maintenance:**

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore Adaptive maintenance termed as an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

**Perceptive Maintenance:**

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancement are received from users. To satisfy requests in this category, Perceptive maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

**Preventive Maintenance:**

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

**SYSTEM REQUIREMENTS**

**Hardware Requirements:**

* System : Intel Core
* Hard Disk : 160 GB
* Ram : 2GB

**Software Requirements:**

* O/S : Windows 7.
* IDE : MATLAB R2013a

**SOFTWARE DESCRIPTION**

**MATLAB:**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is a [fourth-generation programming language](http://whatis.techtarget.com/definition/programming-language-generations) and numerical analysis environment.

Typical uses include:

* Math and computation
* Algorithm development
* Modeling, simulation, and prototyping
* Data analysis, exploration, and visualization
* Scientific and engineering graphics
* Application development, including Graphical User Interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

**MATLAB features**

* A family of application-specific solutions called toolboxes.
* Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology.
* Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems.
* Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

**The MATLAB System:**

The MATLAB system consists of five main parts:

1. **The MATLAB language.**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

1. **The MATLAB working environment.**

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

1. **Handle Graphics.**

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

1. **The MATLAB mathematical function library.**

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

1. **The MATLAB Application Program Interface (API).**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Uses for MATLAB include [matrix](http://whatis.techtarget.com/definition/matrix) calculations, developing and running [algorithms](http://whatis.techtarget.com/definition/algorithm), creating user interfaces ([UI](http://searchsoa.techtarget.com/definition/user-interface)) and [data visualization](http://searchbusinessanalytics.techtarget.com/definition/data-visualization). The multi-[paradigm](http://whatis.techtarget.com/definition/paradigm) numerical computing environment allows developers to interface with programs developed in different languages, which makes it possible to harness the unique strengths of each language for various purposes.

MATLAB is used by engineers and scientists in many fields such as image and signal processing, communications, control systems for industry, [smart grid](http://whatis.techtarget.com/definition/smart-grid) design, [robotics](http://whatis.techtarget.com/definition/robotics) as well as computational finance.

Cleve Moler, a professor of Computer Science at the University of New Mexico, created MATLAB in the 1970s to help his students. MATLAB's commercial potential was identified by visiting engineer Jack little in 1983. Moler, Little and Steve Bangart founded MathWorks and rewrote MATLAB in [C](http://searchwindowsserver.techtarget.com/definition/C) under the auspices of their new company in 1984.

## **MATLAB Programming Language**

The MATLAB programming language is simpler than most programming languages and easier to learn. It is known as a high-level language because it is closer to the human language than the computer or machine language.

* The **semi-colon** in MATLAB indicates the end of statement. It can also be used to stop a statement from executing. For example, if you type in x=5+3 without the semicolon and click the Execute button, MATLAB will display the result as x=8. If you type in x=5+3; with the semicolon, and click Execute, MATLAB will not display the result of the computation.
* The **% sign** is used to indicate that the text following is a comment and not to be interpreted by MATLAB. Programmers use comments to provide explanations about the code they are writing. For example, in MATLAB you can write a=b+5 and the use the % sign to explain that 'a' is the length of a room and 'b' is the width.
* **Variable names** in MATLAB are case sensitive. For example, if you create a variable 'TempEveryHour' to represent the temperatures every hour, and need it to use this variable in a mathematical formula, you would need to call the variable by its exact same name: TempEveryHour and not TemperatureEveryHour or tempeveryhour.

MATLAB has several **advantages** over other methods or languages:

 Its basic data element is the matrix. A simple integer is considered a matrix of one row and one column. Several mathematical operations that work on arrays or matrices are built-in to the Matlab environment. For example, cross-products, dot-products, determinants, inverse matrices.

* Vectorized operations. Adding two arrays together needs only one command, instead of a for or while loop.
* The graphical output is optimized for interaction. You can plot your data very easily, and then change colors, sizes, scales, etc, by using the graphical interactive tools.
* Matlab’s functionality can be greatly expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality. Ex: Excel link allows data to be written in a format recognized by Excel, Statistics Toolbox allows more specialized statistical manipulation of data (Anova, Basic Fits, etc)

There are also **Disadvantages**:

* It uses a large amount of memory and on slow computers it is very hard to use.
* It sits “on top” of Windows, getting as much CPU time as Windows allows it to have. This makes real-time applications very complicated.

**SYSTEM DESIGN**

**Introduction:**

System design is the process or art of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. Design is the first phase in development phase for any engineer’s product system. Design is the creative process. It deals with the creative ability of the programmer. A good design is the key to effective system. The term “Design” is defined as “The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient details to permit its physical realization”.

**Input design**

The user interface design is very important for any application. The interface design describes how the software communicated within itself, to system that interpreted with it and with humans who use it. The interface is a packing for computer software if the interface is easy to learn, simple to use. If the interface design is very good, the user will fall into an interactive software application.

The input design is the process of converting the user-oriented inputs into the computer-based format. Errors entered by data entry operations can be controlled by input design. The data is fed into the system using simple interactive forms. The forms have been supplied with messages so that user can enter data without facing any difficulty.

The data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into the system. The goal for designing input data is to make data entry as easy, logical and free from errors.

The objectives of input design are:

* To produce a cost effective method of input
* To make the input forms understandable to the user
* To ensure the validation of data input
* To achieve the highest position level of accuracy

The various activities to be performed for the overall input processors are:

* Data recording at its source.
* Data transfer to input form.
* Data conversation to computer acceptable mode.
* Data validation.
* Data flow control.
* Data correction if necessary.

**Output Design**

The system output is the most important and direct source of information to the user. So intelligible output design improves the relationship with the user and helps in decision-making. Outputs from the computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these results for later consultation.

A major form of output is a hard copy obtained from the printer. These printouts are designed to include the exact requirements of the user. The outputs required by the end-user are defined during the logical design stages.

Two phases of the output design are:

* Output definition.
* Output specification

Computer outputs are the most important and direct source of information to the user. A quality output is one which meets the requirements of the end user and which presents information in a way which is clear, easy to read and visually attractive. The screens are designed in such a way that the outputs are provided to the user in an understandable form.

The objectives of output design are:

* Design output to serve the indented purpose.
* Provide output on time.
* Assume that output is where it is needed.
* Design output to fit the user

**FEASIBILITY STUDY**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

**Technical Feasibility**

This study center around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

**Behavioural Feasibility**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

**CONCLUSION**

The preprocessing technique is used to split the three different bands of image. Clustering technique used to find the similar patterns of samples. And the classification technique predict the class value by using test features values. Finally, we evaluate the performance using measuring performance parameters.

**FUTURE ENHANCEMENTS**

For clustering, we use k-means clustering algorithm. k-means clustering aims to partition n observations into k clusters. Instead of k-means we can use Hierarchical clustering. Hierarchical clustering groups data over a variety of scales by creating a cluster tree. This allows you to decide the level or scale of clustering that is most appropriate for your application.

**REFERENCES**

* G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, “Advances in hyperspectral image classification: Earth monitoring with statistical learning methods,” IEEE Signal Process. Mag., vol. 31, no. 1, pp. 45–54, Jan. 2014.
* G. Hughes, “On the mean accuracy of statistical pattern recognizers,” IEEE Trans. Inform. Theory, vol. 14, no. 1, pp. 55–63, Jan. 1968.
* S. B. Serpico and L. Bruzzone, “A new search algorithm for feature selection in hyperspectral remote sensing images,” IEEE Trans.Geosci.Remote Sens., vol. 39, no. 7, pp. 1360–1367, Jul. 2001.
* C.-I. Chang and S. Wang, “Constrained band selection for hyperspectral imagery,” IEEE Trans. Geosci. Remote Sens., vol. 44, no. 6, pp. 1575– 1585, Jun. 2006.
* W. Li, S. Prasad, J. Fowler, and L. Bruce, “Locality-preserving dimensionality reduction and classification for hyperspectral image nalysis,” IEEE Trans. Geosci. Remote Sens., vol. 50, no. 4, pp. 1185–1198, Apr.2012.
* Y. Zhou, J. Peng, and C. Chen, “Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification,” IEEE Trans.Geosci. Remote Sens., vol. 53, no. 2, pp. 1082– 1095, Feb. 2015.
* H. Huang and M. Yang, “Dimensionality reduction of hyperspectral images with sparse discriminant embedding,” IEEE Trans. Geosci. Emote Sens., vol. 53, no. 9, pp. 5160–5169, Sep. 2015.
* S. Patra, P. Modi, and L. Bruzzone, “Hyperspectral band selection based on rough set,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 10, pp. 5495–5503, Oct. 2015.
* Z. Wang, N. M. Nasrabadi, and T. S. Huang, “Semisupervised hyperspectral classification using task-driven dictionary learning with laplacian regularization,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 3, pp. 1161–1173, Mar. 2015.
* S. Patra and L. Bruzzone, “A novel SOM-SVM-based active learning technique for remote sensing image classification,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 11, pp. 6899–6910, Nov. 2014.
* B. Demir, L. Minello, and L. Bruzzone, “Definition of effective training sets for supervised classification of remote sensing images by a novel cost-sensitive active learning method,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 2, pp. 1272–1284, Feb. 2014.
* Z.Wang, B. Du, L. Zhang, L. Zhang, and X. Jia, “A novel semisupervised active-learning algorithm for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 6, pp. 3071–3083, Jun. 2017.
* J. Serra, Image Analysis and Mathematical Morphology. London, U.K.: Academic Press, 1982.
* J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, “Classification of hyperspectral data from urban areas based on extended morphological profiles,” IEEE Trans. Geosci. Remote Sens., vol. 43, no. 3, pp. 480–491, Mar. 2005
* M. Dalla Mura, A. Villa, J. A. Benediktsson, J. Chanussot, and L. Bruzzone, “Classification of hyperspectral images by using extendedmorphological attribute profiles and independent component analysis,” IEEE Geosci. Remote Sens. Lett., vol. 8, no. 3, pp. 542–546, May 2011.
* N. Falco, J. A. Benediktsson, and L. Bruzzone, “Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 12, pp. 6223–6240, Nov. 2015.
* X. Huang, X. Han, L. Zhang, J. Gong, W. Liao, and J. A. Benediktsson, “Generalized differential morphological profiles for remote sensing image classification,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 4, pp. 1736–1751, Apr. 2016.
* W. Liao et al., “Promoting partial reconstruction for the morphological analysis of very high resolution urban remote sensing images,” IEEE Geosci. Remote Sens. Mag., vol. 5, no. 2, pp. 8–28, Jun. 2017.
* F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” IEEE Trans. Geosci. Remote Sens., vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
* K. Brinker, “Incorporating diversity in active learning with support vector machines,” in Proc. 20th Int. Conf. Mach. Learn., 2003
* R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. Singapore: Wiley, 2001.
* D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. New York, NY, USA: Addison-Wesley, 1989.
* S. Holzwarth et al., “HySens - DAIS 7915/ROSIS imaging spectrometers at DLR,” in Proc. 3rd EARSeL Workshop Imag. Spectroscopy, 2003
* C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machine,” 2001. [Online]. Available: http://csie.ntu.tw/˜cjlin/libsvm
* G. M. Foody, “Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy,” Photogramm. Eng. Remote Sens., vol. 70, pp. 627–634, 2004.
* J. Li, J. Bioucas-Dias, and A. Plaza, “Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 2, pp. 844–856, Feb. 2013.
* S. Sun, P. Zhong, H. Xiao, and R. Wang, “An MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery,” IEEE J. Sel. Topics Signal Process., vol. 9, no. 6, pp. 1074–1088, Sep. 2015.
* Z. Zhang, E. Pasolli, M. M. Crawford, and J. C. Tilton, “An active learning framework for hyperspectral image classification using hierarchical segmentation,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 2, pp. 640–654, Feb. 2016.
* X. Zhou, S. Prasad, and M. M. Crawford, “Wavelet-domain multiview active learning for spatial-spectral hyperspectral image classification,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 9, pp. 4047–4059, Sep. 2016.
* P. Rigollet, “Generalization error bounds in semi-supervised classification under the cluster assumption,” J. Mach. Learn. Res., vol. 8, pp. 1369–1392, 2007.
* K. Pearson, “On lines and planes of closest fit to systems of points in space,” Philosoph. Mag., vol. 2, no. 11, pp. 559–572, 1901.
* P. Mitra, B. U. Shankar, and S. K. Pal, “Segmentation of multispectral remote sensing images using active support vector machines,” Pattern Recognit. Lett., vol. 25, no. 9, pp. 1067–1074, 2004.
* S. Rajan, J. Ghosh, and M. M. Crawford, “An active learning approach to hyperspectral data classification,” IEEE Trans. Geosci. Remote Sens., vol. 46, no. 4, pp. 1231–1242, Apr. 2008.
* D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery, “Active learning methods for remote sensing image classification,” IEEE Trans. Geosci. Remote Sens., vol. 47, no. 7, pp. 2218–2232, Jul. 2009.
* B. Demir, C. Persello, and L. Bruzzone, “Batch-mode active-learning methods for the interactive classification of remote sensing images,” IEEE Trans. Geosci. Remote Sens., vol. 49, no. 3, pp. 1014–1031, Mar. 2011.
* S. Patra and L. Bruzzone, “A batch-mode active learning technique based on multiple uncertainty for SVM classifier,” IEEE Geosci. Remote Sens. Lett., vol. 9, no. 3, pp. 497–501, May 2012.
* S. Patra and L. Bruzzone, “A fast cluster-assumption based active learning technique for classification of remote sensing images,” IEEE Trans. Geosci. Remote Sens., no. 5, pp. 1617–1626, May 2011.