

A Virtual Approach to Image Segmentation and Taxonomy in Satellite Imagery from Space Video Surveillance

Dhanalakshmi V

Ph.D Research Scholar

Dept. of Computer Science and Engineering
National Institute of Technology Durgapur
West Bengal, India

Dr. Goutam Sanyal

Professor & Dean (SW)

Dept. of Computer Science and Engineering
National Institute of Technology Durgapur
West Bengal, India

Abstract - The resolution of space images incorporates an effective segmentation. The level of detail of heterogeneity of the images can be solved using various classification tools. Most of the tools available do not cater for the high resolution images, as Space images are defined as a picture of the earth taken from an earth orbital space by remote sensing. Remote sensing is a science of obtaining information about an object, area or phenomenon through the analysis of data acquired by a device that is not in contact with the object. The space image segmentation and classification is done using k-means and back propagation algorithm. Wide database of images has been used to test both the algorithms. The paper also portrays the comparative study of the results obtained by both the algorithm. The comparison of results shows good accuracy in both the methods.

Keywords - Segmentation, classification, Remote sensing, earth orbital space, K-means and back propagation algorithm.

I INTRODUCTION

The objective of this paper is to provide an effective segmentation and classification tool. In order to better explain the structure of this work, the preliminary information about the space_Image and Remote Sensing is discussed. Space images are picture of the earth taken from an earth orbital space. This image is a remotely sensed image. Remote sensing is a technology that deals with science of obtaining information about an object. [1,2] Commonly remote sensing is referred to the collection and analysis of data regarding the earth using electromagnetic sensors, which are operated from the space borne platform. Space image classification includes two steps namely Segmentation step and Classification step. In this paper, the space image segmentation and classification is done using k-means and back propagation algorithm.

A. K-MEANS AND BACK PROPAGATION ALGORITHMS

1) k - Means Algorithm

This algorithm starts with some clusters of pixels in the feature space, each of them defined by its center. The first step consists in allocating each pixel to the nearest cluster. In the second step the new centers are computed with the new clusters. These two steps are

repeated until convergence. The basic step of k-means clustering is simple. In the beginning determine number of clusters K and assume the centroid or center of these clusters. Take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids.

The K means algorithm will do the three steps below until convergence.

Iterate until **stable** (= no object move group)

- Determine the centroid coordinate.
- Determine the distance of each Object to the centroids.
- Group the object based on minimum distance.

The experimentation was done on the images downloaded from www.imageprocessing.com. After implementation of K - mean, the results of 'segmentation and classification' are stored and then same images are given to the neural network classifier for further study.

2) Back Propagation Algorithm of Ann

The back propagation algorithm is a generalization of the least mean square algorithm that modifies network weights to minimize the mean squared error between the desired and actual outputs of the network. Back propagation uses supervised learning in which the network is trained using data for which inputs as well as desired outputs are known. Once trained, the network weights are frozen and can be used to compute output values for new input samples.

Back propagation Algorithm. Start with randomly chosen weights. **While** MSE is unsatisfactory and computational bounds are not exceeded, **do for** each input pattern.

- Compute hidden node inputs $(net_{p,j}^{(1)})$
- Compute hidden node outputs $(x_{p,j}^{(1)})$
- Compute inputs to the outputs nodes $(net_{p,k}^{(2)})$
- Compute the network outputs $(o_{p,k})$
- Compute the error between $o_{p,k}$ and desired output $d_{p,k}$
- Modify the weights between hidden and output nodes:

$$\Delta w_{k,j}^{(2,1)} = \eta (d_{p,k} - o_{p,k}) \delta' (net_{p,k}^{(2)}) x_{p,j}^{(1)}$$

g) Modify the weights between input and hidden nodes:

$$\Delta w_{ji}^{(1,0)} = \eta \sum_k ((a_{p,k} - o_{p,k}) \delta'(net_{p,k}^{(2)}) \Delta w_{k,j}^{(2,1)}) \delta'(net_{p,i}^{(1)}) x_{p,i}$$

End-for

End-while

B. MODULE CATEGORIZATION

The following modules are categorized based upon the proposed model

- Authentication
- Image master
- Preview imaging
- Segmentation and classification
- Pixel proxy creation
- Graph comparison

1) Authentication

The authentication is the major part for any kind of software. Generally authentication is used for security purpose to protect from intruders. Here, two walls majorly acting for security named as authentication wall and authorization wall. Authentication wall filters the users by providing username and password. After successful entering into the authentication wall, the authorization wall will check the entered user having administrator rights or normal rights. Based upon the rights, the access permission will be allowed.

There are two closely interlinked concepts at the heart of security for distributed applications - authentication and authorization. Authentication is the process of obtaining some sort of credentials from the users and using those credentials to verify the user's identity. Authorization is the process of allowing an authenticated user access to resources.

An ASP.net application has two separate authentication layers. That is because ASP.net is not a standalone product. Rather it is a layer on top of Internet Information Services (IIS). All requests flow through IIS before they are handed to ASP.net. As a result, IIS can decide to deny access without the ASP.net process even knowing that someone requested a particular page. Here is an overview of the steps in the joint IIS and ASP.net authentication process.

- a. IIS first checks to make sure the incoming request comes from an IP address that is allowed access to the domain. If not it denies the request.
- b. Next IIS performs its own user authentication if it configured to do so. By default IIS allows anonymous access, so requests are automatically authenticated, but you can

change this default on a per - application basis with in IIS.

- c. If the request is passed to ASP.net with an authenticated user, ASP.net checks to see whether impersonation is enabled. If impersonation is enabled, ASP.net acts as though it were the authenticated user. If not ASP.net acts with its own configured account.
- d. Finally the identity from step 3 is used to request resources from the operating system. If ASP.net authentication can obtain all the necessary resources it grants the users request otherwise it is denied. Resources can include much more than just the ASP.net page itself you can also use Net's code access security features to extend this authorization step to disk files, Registry keys and other resources.

Authentication Providers. Assuming IIS passes a request to ASP.net, what happens next? The answer depends on the configuration of ASP.net itself. The ASP.net architecture includes the concept of and authentication provider a piece of code whose job is to verify credentials and decide whether a particular request should be considered authenticated. Out of the box ASP.net gives you a choice of three different authentication providers.

- The windows Authentication provider lets you authenticates users based on their windows accounts. This provider uses IIS to perform the authentication and then passes the authenticated identity to your code. This is the default provided for ASP.net.
- The passport authentication provider uses Microsoft's passport service to authenticate users.
- The forms authentication provider uses custom HTML forms to collect authentication information and lets you use your own logic to authenticate users. The user's credentials are stored in a cookie for use during the session.

Selecting an authentication provider is as simple as making an entry in the web.config file for the application. Also use one of these entries to select the corresponding built in authentication provider:

```
<authentication mode="windows">
authentication mode="passport">
<authentication mode="forms">
```

ASP.net also supports custom authentication providers. This simply means that you set the authentication mode for the application to none, then write your own custom code to perform authentication. For example, you might install an ISAPI filter in IIS that compares incoming requests to list of source IP addresses, and considers requests to be authenticated if they come from an acceptable address. In that case, you would set the authentication

mode to none to prevent any of the .net authentication providers from being triggered.

Authorization. After your application has authenticated users, you can proceed to authorize their access to resources. But there is a question to answer first: Just who is the user to whom your are grating access? It turns out that there are different answers to that question, depending on whether you implement impersonation. Impersonation is a technique that allows the ASP.NET process to act as the authenticated user, or as an arbitrary specified user.

ASP.NET impersonation is controlled by entries in the applications web.config file. The default setting is "no impersonation". You can explicitly specify that ASP.NET shouldn't use impersonation by including the following code in the file

```
<identity impersonate="false"/>
```

With this setting ASP.NET does not perform impersonation. It means that ASP.NET will runs with its own privileges. By default ASP.NET runs as an unprivileged account named ASPNET. You can change this by making a setting in the processModel section of the machine.config file. When you make this setting, it automatically applies to every site on the server. To user a high-privileged system account instead of a low-privileged, set the userName attribute of the processModel element to SYSTEM. Using this setting is a definite security risk, as it elevates the privileges of the ASP.NET process to a point where it can do bad things to the operating system.

When you disable impersonation, all the request will run in the context of the account running ASP.NET: either the ASPNET account or the system account. This is true when you are using anonymous access or authenticating users in some fashion. After the user has been authenticated, ASP.NET uses it own identity to request access to resources.

The second possible setting is to turn on impersonation.

```
<identity impersonate="true"/>
```

In this case, ASP.NET takes on the identity IIS passes to it. If you are allowing anonymous access in IIS, this means ASP.NET will impersonate the IUSR_ComputerName account that IIS itself uses. If you aren't allowing anonymous access, ASP.NET will take on the credentials of the authenticated user and make requests for resources as if it were that user. Thus by turning impersonation on and using a non-anonymous method of authentication in IIS, you can let users log on and use their identities within your ASP.NET application.

Finally, you can specify a particular identity to use for all authenticated requests

```
<identity impersonate="true"  
username="DOMAIN\username"  
password="password"/>
```

With this setting, all the requests are made as the specified user (Assuming the password it correct in the configuration file). So, for example you could

designate a user for a single application, and use that user's identity every time someone authenticates to the application. The drawback to this technique is that you must embed the user's password in the web.config file in plain text. Although ASP.NET won't allow anyone to download this file, this is still a security risk if anyone can get the file by other means.

2) Image Master

Captured Remote sensing images will be allowed to store in the centralized database with the sufficient details related to that remote sensing images like image details, image taken time and date and so on. Here the image master acting as the gateway to store the images in the database.

3) Preview Imaging

The preview imaging will be very helpful to the user in order to preview a bunch of remote sensing images stored in the centralized database already, in a same time. It is mainly used to select the required image soon to allow it for segmentation and classification process.

4) Segmentation and Classification

The Segmentation and Classification process is a very important tool to segment the particular portion of a remote sensing image and to allow it to zoom the pixel areas based upon the user criteria. This particular portion will be allowed for different types of classifiers process to get a better accuracy one [5,6].

The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics

Clustering Methods. The K-means algorithm is an iterative technique that is used to partition an image into K clusters. The basic algorithm is:

1. Pick K cluster centers, either randomly or based on some heuristic
2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center

3. Re-compute the cluster centers by averaging all of the pixels in the cluster
4. Repeat steps 2 and 3 until convergence is attained (e.g. no pixels change clusters)

The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors. K can be selected manually, randomly, or by a heuristic.

This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K .

In statistics and machine learning, the k-means algorithm is clustering algorithm to partition n objects into k clusters, where $k < n$. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. The model requires that the object attributes correspond to elements of a vector space.

The most common form of the algorithm uses an iterative refinement heuristic known as Lloyd's algorithm. Lloyd's algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed). Lloyd's algorithm and k-means are often used synonymously, but in reality Lloyd's algorithm is a heuristic for solving the k-means problem, as with certain combinations of starting points and centroids, Lloyd's algorithm can in fact converge to the wrong answer. Other variations exist, but Lloyd's algorithm has remained popular, because it converges extremely quickly in practice.

In terms of performance the algorithm is not guaranteed to return a global optimum. The quality of the final solution depends largely on the initial set of clusters, and may, in practice, be much poorer than the global optimum. Since the algorithm is extremely fast, a common method is to run the algorithm several times and return the best clustering found. A drawback of the k-means algorithm is that the number of clusters k is an input parameter. An inappropriate choice of k may yield poor results. The algorithm also assumes that the variance is an appropriate measure of cluster scatter.

Histogram-Based Methods - Histogram-based methods are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the

histogram are used to locate the clusters in the image. Color or intensity can be used as the measure.

A refinement of this technique is to recursively apply the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This is repeated with smaller and smaller clusters until no more clusters are formed.^{[1][3]}

One disadvantage of the histogram-seeking method is that it may be difficult to identify significant peaks and valleys in the image. In this technique of image classification distance metric and integrated region matching are familiar.

Histogram-based approaches can also be quickly adapted to occur over multiple frames, while maintaining their single pass efficiency. The histogram can be done in multiple fashions when multiple frames are considered. The same approach that is taken with one frame can be applied to multiple, and after the results are merged, peaks and valleys that were previously difficult to identify are more likely to be distinguishable. The histogram can also be applied on a per pixel basis where the information results are used to determine the most frequent color for the pixel location. This approach segments based on active objects and a static environment, resulting in a different type of segmentation useful in Video tracking.

Neural Networks Segmentation - Neural Network segmentation relies on processing small areas of an image using an artificial neural network^[2,3] or a set of neural networks. After such processing the decision-making mechanism marks the areas of an image accordingly to the category recognized by the neural network. Image classification relies on the spectral distinctness of classes or spectra-temporal variability.

Image Classification - Artificial Neural Network and Image Classification - In general, a neural network is referred to as a mathematical model of theorized mind and brain activity (Simpson, 1990). Neural network features corresponding to the synapses, neurons, and axons of the brain are input weights, processing elements, and output paths. In an artificial neural network, the processing element (PE) is the analog to the human brain's biological neuron. A processing element has many input paths, analogous to the brain's dendrites, and the information transferred along these paths is combined by one of a variety of mathematical functions, most commonly simple summation. The result of these combined inputs is some level of internal activity for the receiving processing element. The combined input contained within the processing element is modified by a transfer function before being passed to other connected processing elements, whose input paths are usually weighted by the perceived synaptic strength of neural connections.

A neural network consists of organized topological interconnections among the learning rules, and knowledge recall. The topological structure establishes the frame of the network, the learning paradigm trains the network by presenting example input data pattern and the corresponding desired output, and the recall applies the pattern recognition knowledge learned in the training step to process and in this case classify the raw data. The most popular forms of neural networks typically consist of three or more layers--an input layer, an output layer, and one or more hidden layers. The input layer consists of one or more processing elements which present the training data, and the output layer consists of one or more processing elements which store the results of the network.

In the case of remote sensing data classification, the inputs often represent the vector of brightness values for the multispectral data. Hence, for single-date Landsat data, there would be seven input nodes, each corresponding to a band of the Thematic Mapper sensor. The input patterns could consist also of ancillary data (e.g., multitemporal spectral patterns, image texture, elevation and its derivatives, etc.). Since the learning and recall depend on the linear and nonlinear combination of data patterns instead of the statistical parameters of the input data, neural networks offer the opportunity to analyze spatial data with different origins and properties simultaneously, without a priori assumptions about the distribution for each data type.

In fact, neural networks have the ability to learn those distributions, if they exist, in the input data. Therefore, a neural network can be trained by data in different types. The one, two, or perhaps more hidden layers consist of a number of processing elements which enable the translation of input data into output information, which, in the present context, is the land cover classification corresponding to an input pattern. Ideally, each data type will make a unique contribution to the discrimination of land cover class patterns, therefore, enabling the neural network to learn the spectral, spatial, and temporal signature of each class.

5) Pixel Proxy Creation

Once the image allowed for classifier process, the pixel of the original image will be analyzed and it dynamically creates a pixel proxy where it stores each values of finite number of classes. After Classifier process completes, the pixel proxy creates for 'n' number of classifier technology. All this process history is allowed to store in the centralized database.

6) Graph Comparison

Graph Comparison will show the accuracy of the output of selected image after segmentation and

classification process through k-means classifier and back propagation of artificial neural network.

III SYSTEM ANALYSIS AND DESIGN

Color based segmentation and classification using K-MEANS and Back Propagation algorithms.

- 1) **Step 1:** Fig. 1 shows the original image as read and it is displayed.

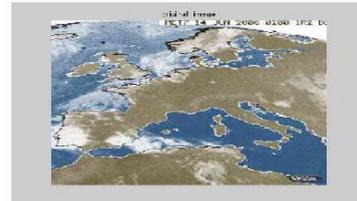


Fig. 1 Original Image

- 2) **Step 2:** In Fig. 2, the image is converted from RGB color space to YCBCR color space.

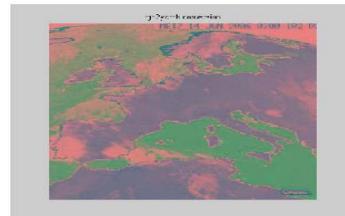


Fig. 2 RGB to YCBCR Conversion

- 3) **Step 3:** Above image is given as an input for both the classifiers i.e., K-Means classifiers and Neural Network Classifiers. Hereafter we have compared the results obtained from both the classifiers.

- 4) **Step 4:** Every pixel in the image is labeled using the results from K-Means. For every object in our input, K-means returns an index corresponding to a cluster. Every pixel in the image is labeled with its cluster-index. Output obtained under K-Means Classifier is shown in Fig.3 (a) and similar procedure is employed on ANN classifier and the output is shown in Fig 3(b).

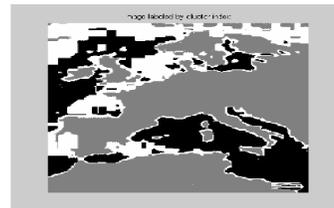


Fig 3(a) Output from K-Means Classifier



Fig 3(b) Output from ANN Classifier

5) Step 5: The above images are used to separate out three different color objects and shown in Fig 4(a) and Fig 4(b).

OBJECTS IN CLUSTER 1



Fig 4(a) Objects in cluster 1 by K-means



Fig 4(b) Objects in cluster 1 ANN classifier

6) Step 6: Objects in Cluster 2 are shown in Fig. 5.

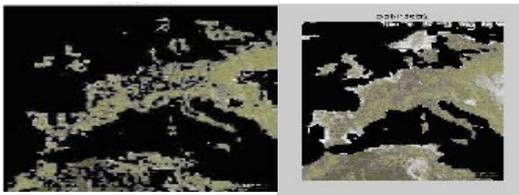


Fig. 5 Objects in cluster 2 by K-means and ANN classifier

7) Step 7: Objects in Cluster 3 is shown in Fig. 6.

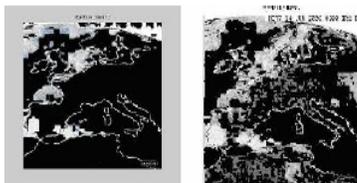
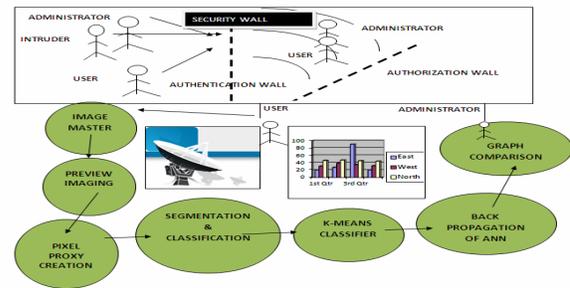


Fig. 6 Objects in cluster 3 by K-Means and ANN classifier

IV SYSTEM ARCHITECTURE



V CONCLUSION

K-Means algorithm and back propagation are studied in detail and implemented on various database for the segmentation and classification purpose. It is found that K-means algorithm gives very high accuracy, but it is useful for single database at a time. Whereas neural network is useful for multiple databases, once it is trained for it. Neural network also provides good accuracy. In future different neural network algorithms can be used to classify the space images and the classification results of those images will be compared with results of existing classification methods.

REFERENCES

- [1] Flayed Sabine and W.H.Freeman, "Remote sensing, Principles and Interpretation".
- [2] S.N. Sivanandam & Sumathi & S. N. Deepa, "Introduction to Neural Networks using Matlab 6.0".
- [3] Milan Sonka, Vaclav Lavaca & Roger Bole, "Image Processing, analysis and Machine Vision".
- [4] William Pratt, "Digital Image Processing". [5] Rafael C. Gonzalez & Richard E.Woods, "Digital Image Processing".
- [5] Rui Xu and Donald Wunsch, "Survey of Clustering Algorithms", IEEE, Vol. 16, No.3, pp.645-678, May 2005.
- [6] Y. Zhang, J. Weng & W.S. Hwang, "Biological Aspects of Neurocomputing. Auditory Learning: A Development method", IEEE, Vol. 16, No.3 pp.601-616, May 2005.
- [7] G. Dong & M. Xie, "Color Clustering & Learning for Image Segmentation Based on Neural Networks", IEEE, Vol. 16, No. 4, Pp.925- 936.
- [8] Xiao yang Tan, Song can Chen & Zhizhou," Recognizing partially. Occluded Expression Variant Faces from Single Training Image per Person with SOM & soft K-NN Ensemble," IEEE Trans. Pattern Anal. Vol.16, No. k4, pp.875-855, July 2005.
- [9] P.D.Deshpande, "Survey of Satellite Image Segmentation and Classification", NCETE2006, MAE, Alandi(D), Pune.