N92-23366

P. 14

FEATURE DETECTION IN SATELLITE IMAGES USING NEURAL NETWORK TECHNOLOGY

Marijke F. Augusteijn and Arturo S. Dimalanta
Computer Science Department
University of Colorado at Colorado Springs

ABSTRACT

This report describes a study of the feasibility of automated classification of satellite images[1]. Satellite images were characterized by the textures they contain. In particular, the detection of cloud textures was investigated. The method of second-order gray level statistics, using co-occurrence matrices, was applied to extract feature vectors from image segments. Neural network technology was employed to classify these feature vectors. The Cascade-Correlation architecture was successfully used as a classifier. The use of a Kohonen network was also investigated but this architecture could not reliably classify the feature vectors due to the complicated structure of the classification problem. The best results were obtained when data from different spectral bands were fused.

Keywords: Image Classification, Texture Analysis, Neural Networks.

INTRODUCTION

The extremely large volume of satellite image data that has been produced to date is difficult to classify for users. As an example, it has been estimated that only 5% of the Landsat images have ever been viewed by humans. Therefore, the ability to automatically classify satellite images is of keen interest to all potential users. If a computer could sort images by topic and possibly associate them with a level of interest (given some objective) then a human user would only have to search through a preselected set. This project is a feasibility study with the main purpose to determine if a specified feature can reliably be detected in a satellite image by computer.

An important task is to determine an appropriate set of features. Although it is sometimes important to detect actual objects in satellite images, most features are mainly visible as textures. For example, the waves in the ocean are observed as a texture, various forms of land (urban, agricultural or forests) appear as different textures, and the clouds in the sky form yet another texture. Thus, texture identification seems a valid means to classify images. This feasibility study will focus on the identification and discrimination of a single, possibly noisy texture. The feature selected is the texture of clouds. Clouds are particularly interesting because they do not necessarily cover an area. Clouds can be dense or sparse. When the clouds are sparse it will be possible to partially see through them and observe the surface below. In this case, the cloud texture will be intermixed with other textures. Thus, an automated technique for cloud identification must be capable of dealing with a considerable level of noise caused by these other textures.

Cloud detection and classification have been studied by many researchers (Goodman and Henderson-Sellers, 1988, and Rossow, 1989). Satellite observations of clouds have been utilized in atmospheric research ever since the first satel-

[1] This project was funded by CTA, Incorporated.

122

PRECEDING PAGE BLANK NOT FILMED

lite images were returned. Satellite images showing cloud formations are characterized by high variability of texture, irregularity of shapes, and a high level of boundary ambiguity, complicating cloud detection. Some researchers (Lee *et al.*, 1990), have gone beyond the identification task and have classified cloud textures as stratocumulus, cumulus, or cirrus. Accurate cloud detection is important for weather forecasting and the study of global changes in climate. In addition, there are other phenomena that produce cloudlike textures. For example, the smoke produced by a forest fire may look like a cloud. Also, the vapors released by volcanic eruption will be cloudlike in appearance. If clouds could be successfully identified even when mixed with other textures, it is expected that the same techniques will be applicable to the detection of large fires and volcanic activities.

Texture Identification

Texture identification has long been recognized as an important means for image classification, and many techniques to measure texture are available (Weszka *et al.*, 1979). A fairly simple procedure that has been successfully used by many researchers is *second-order gray level statistics* (Haralick *et al.*, 1973). This method is defined in the spatial domain and takes the statistical nature of the texture into account. A set of *co-occurrence matrices* is calculated, which measures the frequency of the simultaneous occurrence of two specified gray levels at two designated relative positions in an image segment (displaying the texture). Generally, four different matrices are used, each computing the frequency of gray level co-occurrence at neighboring positions in four different directions (horizontal, vertical, and along the two diagonal directions of the image).

A variety of measures can be employed to extract useful textural information from these matrices. Haralick *et al.* (1973) define fourteen different measures but consider four of them most useful. They are the angular second moment (sometimes called energy or homogeneity), the contrast, the correlation, and the entropy of a texture.

Neural Networks

Neural networks have recently become popular as general classifiers. For example, they were used in a cloud classification study (Lee *et al.*, 1990). The appeal of neural networks as pattern recognition systems is based upon several considerations. They appear to perform as well or better than other classification techniques and require no assumptions about the nature of the distribution of the pattern data. A comparison of neural networks to classical methods like K-nearest neighbor and discriminant analysis has shown that neural networks can achieve equal performance using a much smaller set of training data (Lee *et al.*, 1990). They have the capability to learn extremely complex patterns and are also suitable for multi-channel data fusion.

An important task is the selection of a neural network architecture appropriate for the application. Pattern recognition is often accomplished by means of a feedforward architecture. This type of network has its processing elements organized in different layers. The bottom layer accepts an input pattern and calculates the activations and outputs of its processing elements. The output values are then passed to the next layer, which performs a similar task. This continues until the top layer is reached. The output of the top layer represents the classification of. the given pattern. The layers between top and bottom are often called hidden layers and are responsible for the correct mapping between the input patterns and their classifications. The most familiar architecture in this class consists of three layers in which consecutive layers are completely connected, as shown in Figure 1.

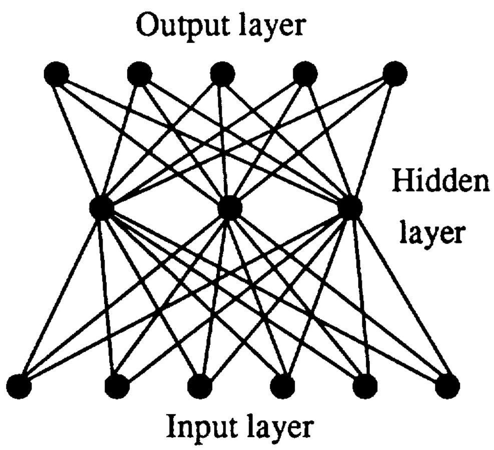

Figure 1. Processing elements and connections organized as a three layer neural network

The correct mapping is acquired during a training phase. In supervised training, the input patterns and the associated desired outputs are presented to the network. The network will update the connection strength between the processing units based on the difference between the desired and current outputs (the current measure of error). The most well-known updating scheme is backpropagation, which calculates an error measure at the output nodes and distributes this error back to the hidden nodes (Rumelhart *et al.*, 1986). However, although backpropagation has been used in numerous successful applications, it has several disadvantages. This learning method is extremely slow. The patterns that form the training set need to be presented many times, often thousands of times, before the network convergences to a solution. Sometimes, the correct solution will not be found. Although the algorithm attempts to find a global minimum of the total error, it may get trapped in a local minimum from which it cannot escape. Also, correct execution depends on the assignment of an appropriate number of nodes to the hidden layer(s). However, determining this number is more an art than a science. Many researchers have attempted to improve on backpropagation. One of these more recent architectures (Cascade-Correlation) is used in this study.

The Satellite Images

The set of satellite images used in this research consists of five scenes in both visible and IR spectral bands. They were obtained by an Advance Very High Resolution Radiometer (AVHRR) instrument. Images were available in five spectral bands. The wavelengths of each band are shown in Table 1.

The five scenes were obtained from the Great Lake area of the United States, the Atlantic Ocean, Barrow, Siberia and the Polar Cap. These scenes contain a variety of surface types, including clouds, water, sea ice, and land. Three of them show appreciable cloud cover with large variations in density. In areas containing sparse clouds, the underlying surface is clearly visible. Different types of surfaces appear through the cloud cover. Especially the Polar Cap scene, showing clouds against a background of ice, appears a challenging classification problem even for humans.

Table 1. Satellite Sensor Wavelength (μ m)

Satellite Band	Wavelength
Channel 1	0.58 - 0.68
2	0.725 - 1.1
3	3.55 - 3.93
4	10.5 - 11.3
5	11.5 - 12.5

ARCHITECTURES FOR TEXTURE ANALYSIS

A successful architecture developed to improve the slow learning characteristics of backpropagation is Cascade-Correlation (Fahlman and Lebiere, 1990). Like backpropagation, it incorporates supervised learning and has proved to be a powerful classifier. However, supervised learning generally does not reveal the underlying structure of the classification problem. In the simplest case, the various patterns will form distinct clusters with each cluster corresponding to a different class. However, it may happen that the clusters overlap. Then, the patterns belonging to the different classes are not well separated presenting a challenging problem to the classifier. In this case, a supervised architecture will experience more difficulty in learning the classification (and may even fail) but it will not show how the different classes relate. A self-organizing network like the one designed by Kohonen (1988) will show this underlying structure. This architecture employs unsupervised learning and organizes its units to reflect the relative configuration of the patterns.

The Cascade-Correlation Architecture

The Cascade-Correlation (Cascor) network is a dynamic architecture that incrementally builds its internal structure during training. Thus, the programmer need not be concerned with the appropriate number of units in the hidden layer(s) because the network itself will allocate the number of nodes required to solve the problem. The essence of the architecture is the following. Training in Cascor begins with the consideration of only two layers (input and output). They are fully connected and these connections are trained until no significant changes occur anymore. If, at that point, the total error is still unacceptably high, a hidden node will be positioned between these layers. The input connections to the new node are trained first. The algorithm attempts to maximize the correlation between the new node's activation and the output error of the network so that the new node may make up for the residual error to the greatest possible extent. The output connections are then trained by means of the quickprop algorithm, a second-order improvement to backpropagation (Fahlman, 1988). Hidden nodes are added, each one in its own separate layer, until the total error is below a preset threshold. Each hidden node is connected with all previously assigned hidden nodes, as well as with all input nodes, and is trained in isolation. Once trained, its input connections are frozen. Each hidden node is also connected with all output units. All output connections are trained after each addition of a hidden node. The basic architecture is shown in Figure 2. The resulting network is fast and capable of reliable classification.

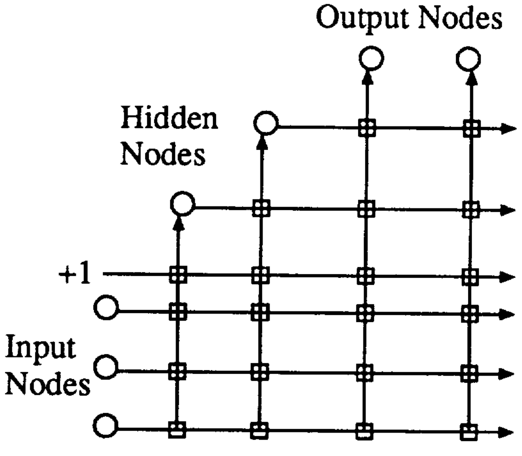

Figure 2. The Cascade Architecture
The vertical lines sum all incoming activations.

The initialization of the connection strengths is performed randomly between certain preset bounds. Thus, when Cascor is run several times on the same data set, a different number of hidden nodes may be generated. These different runs are referred to as trials. Different trials, although trained on the same data set, may show different performance when used to classify the test data.

The Kohonen Self-Organizing Map

The Kohonen self-organizing map facilitates a better understanding of the underlying structure of the classification problem. This method provides a means to project a high dimensional vector space onto a lower (usually two) dimensional space which is simple to represent graphically. It creates a topology preserving map in which units that are physically located next to each other will respond to input patterns that are likewise next to each other.

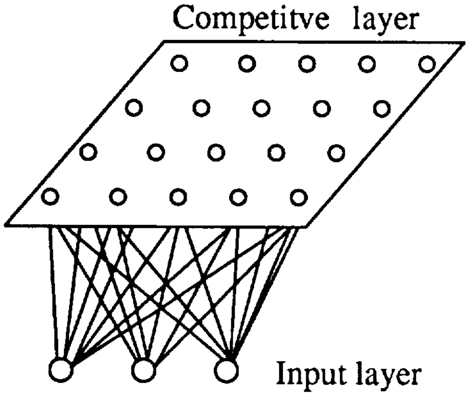

Figure 3. The architecture of the Kohonen Self-Organizing Map

The architecture consists of an input layer that is the size of the input pattern. This layer is completely connected to a (generally) two-dimensional organization of units as shown in Figure 3. The units in this second layer are competitive; that is, each one calculates an activation based on the input pattern and then enters into a competition with the other units in that layer. Each unit also represents a pattern, stored as the strengths (weights) of the connections leading to that unit from the input layer. The activation calculated by each one is proportional to the similarity between the input pattern and its stored pattern. The unit with the highest activation (whose stored pattern best approximates the current input) wins the competition. The winning unit as well as the units in its immediate neighborhood are selected for learning; that is, their weights are adjusted.

The architecture is initialized by assigning random weights (within certain preset limits) to all connections. Initially, it will be random which unit wins the competition. The winner and its neighbors will have their weights updated. The change is such that all weights move over a short distance towards the current input pattern which they begin to encode. Each presentation of an input pattern will move the weights of a set of units in the direction of that pattern. As training proceeds, the neighborhood affected will shrink. Thus, in the beginning a large group of units will be pulled towards a particular pattern while towards the end only a few will be moved. Eventually, after generally thousands of pattern presentations, the result of this kind of training is a topological organization of the units so that the ones encoding similar patterns will be geometrically grouped together. In this manner, the underlying structure of the clusters will become visible.

THE CLASSIFICATION EXPERIMENTS

The set of satellite images used in the experiments consists of 23 photographs showing 5 different scenes. Their distribution over the spectral bands is as follows. Bands 2 through 4 each contain 5 images (one of each scene), and bands 1 and 5 each have 4 images (with the Great Lakes scene missing). All photographs are of size 512 x 512, contain 256 gray levels and have a resolution of 1100 meters per pixel. The images in bands 1 and 2 look most natural to the human eye since the corresponding wavelengths are in the visible or nearinfrared range, as shown by Table 1. The ones in bands 3 to 5 appear slightly unfamiliar since these are infrared photographs.

Several classification experiments were performed. All of them employed the same set of segments extracted from the images. All segments were selected using the Channel 2 photographs and had a size of 25 by 25 pixels. These segments were classified depending on the prevalent cloud pattern present. Not all cloud patterns appear the same. As mentioned before, a major cause for the differences in these patterns is cloud density. Assigning all the different densities to a single class did not seem reasonable. It was decided to define three classes of cloud patterns in the following way. When a segment is completely filled by clouds it will be labeled as *dense clouds*. Different patterns of dense clouds do occur, but these will all be assigned to the same class. When the cloud density is such that clouds fill the segment for at least two thirds of the area, this segment will be labelled as *medium clouds*. Finally, a segment showing light cloud cover such that less than one third of the area is actually covered by clouds is labelled as *sparse clouds*. All segments were selected to show as uniform a cloud pattern as possible. They do not cross texture boundaries, showing dense clouds in one part and possibly no clouds in another part. Thus, the medium and sparse cloud segments show clouds interspersed with land, water, ice, or a combination of these surfaces. All segments without any cloud cover are labelled as *no clouds*. These segments are filled with land, water, and ice, in various combinations.

Once the segments were selected in Channel 2, corresponding segments with the same coordinates were extracted from all other channels of the same scene. A feature vector was then formed for each segment in the following way. A set of four directional co-occurrence matrices was calculated for each one. The four prevalent measures, angular second momentum, contrast, correlation, and entropy were computed from each matrix. In order to measure a rotationally invariant texture, the feature values derived from the four directional matrices were averaged. The four values thus obtained were combined with the average gray level (which had to be scaled) and the standard deviation of the gray levels in each segment. The resulting six-dimensional feature vector was then normalized.

Classification with Cascade-Correlation

Cascor was used in three different experiments. In the first one, the feature vectors used for training the network and those that test the net were all taken from the same image. Thus, this experiment consisted of 23 independent tests, one for each of the 23 images. These tests were performed to get an initial impression of the classification capabilities but were not considered to be of major importance. The second experiment combined all images of a particular channel. It consisted of 5 independent tests, one for each channel. The third experiment combined information from different channels. Out of the many possible combinations five were selected that appeared most promising.

Experiment 1: Classification within a single image

All feature vectors generated from a single image were collected. In most cases, the image contained all four classes and provided 32 vectors, 8 for each class. One vector out of each group of 8 was randomly chosen as the test case. All others were used for training. All training vectors were randomly ordered so that the network would be exposed to all four classes simultaneously. Cascor always converged to a solution in a short time interval. The number of hidden nodes allocated varied from 3 to 11 with an average of about 6 when all 4 classes were present in the image. Images containing fewer classes generated fewer nodes. Each test consisted of a single trial. Classification in these tests scored over 90% on the average.

Experiment 2: Classification within a single channel

This experiment involved all feature vectors generated from images belonging to the same channel. These vectors were partitioned in a training set and in a test set. Four tests were performed in each channel. Each test used a different set of test items. Test items were obtained by random selection from each class and each image of a channel. The remaining vectors were used for training. A typical training set consisted of about 100 vectors, and about 16 vectors were used for testing. (The test and training sets in Channels 1 and 5 were somewhat smaller because one of the scenes was not available in these channels.)

Cascor was run five times on each training set. It always converged to a solution with a varying number of hidden nodes. Each group of five trials that were tested on the same data forms a test case. The test cases were labeled 1 through 4. It was observed that performance within a test case could vary considerably. This may be caused by the relatively small set of test data. Table 2 lists the average percentage of misclassifications of each test case and the misclassification percentage of the trial in each case that performed best. This table also shows the average number of hidden nodes generated during training and the overall average percentage of misclassifications observed in the channel. It is seen that the misclassification percentages are rather high and increase in the infrared channels.

More precise classification data can be obtained if the nature of the misclassification is taken into account. Three of the four classes correspond to different levels of cloud cover and are therefore quite similar. It may be considered less serious if a vector is misclassified within the group of cloud cover classes than when clouds are not recognized at all. Thus, it may also be important to make a distinction between segments showing some level of cloud cover and those containing no clouds at all. Tables 3 and 4 provide examples of the nature of the misclassifications obtained from the set of "best trials" of each test in Channels 2 and 3. These tables show the actual classifications horizontally and the classifications assigned by Cascor vertically. The numbers indicate fractions. Thus, the numbers along the diagonals indicate the fraction of correct classifications by the network, and the numbers off the diagonals show the fraction of misclassifications in each category.

Table 3 shows that most mis-classifications in Channel 2 were made between the different cloud cover categories. There are relatively few cases where a segment showing some cloud cover was taken for a segment that contained no clouds at all, or conversely.

Table 2. The number of hidden nodes and percentage of misclassifications in each test performed in each of the five channels

Channel	Test ID	Average Number of Hidden Nodes	Average Number of Misclassifications in %	Average in Channel	Misclassifications in best trial in %
1	1	21	25		17
	2	21	28		17
	3	22	22		17
	4	23	17	23	6
2	1	28	22		12.5
	2	25	28		25
	3	25	25		19
	4	26	22	24	12.5
3	1	28	40		31
	2	28	32		19
	3	30	31		12.5
	4	29	40	36	31
4	1	29	45		31
	2	29	46		37
	3	29	34		25
	4	30	45	42	25
5	1	23	42		33
	2	25	28		17
	3	24	52		50
	4	24	32	38	25

Table 3. The classification in Channel 2 of the best trials given as fractions

classification assigned by Cascor	actual classification →			
↓	dense clouds	medium clouds	sparse clouds	no clouds
dense clouds	0.67			0.04
medium clouds	0.17	1.00	0.13	
sparse clouds	0.08		0.81	0.04
no clouds	0.08		0.06	0.92

Table 4. The classification in Channel 3 of the best trials given as fractions

classification assigned by Cascor	actual classification →			
↓	dense clouds	medium clouds	sparse clouds	no clouds
dense clouds	0.50	0.17	0.06	
medium clouds	0.33	0.75	0.19	
sparse clouds	0.17	0.08	0.75	0.08
no clouds				0.92

When judging this result it should be taken into account that many of the *no clouds* segments show ice cover which, at least to the human eye, appears similar to a dense cloud cover. However, the neural network generally had no problem distinguishing between these similar textures. Table 4 shows the results in Channel 3. These are particularly interesting because all cloud segments in this channel were classified as containing some cloud cover. The classification in terms of *clouds* or *no clouds* was generally found to be above 80% in all channels.

Experiment 3: Classification using fused channel data

The previous experiment showed that the classification results differed for the various channels. Also, the kind of misclassifications seemed to vary slightly between the channels. In particular, in Channel 3 all clouds were classified as clouds, although misclassifications occurred between the different types. On the other hand, the detailed classification as different types of cloud cover in Channels 1 and 2 surpassed that of Channel 3. If information obtained from different channels were to be combined, better classification results could be expected. It was decided not to investigate all possible combinations but to select the more promising ones.

Channels 1 and 2 show the fewest misclassifications. Therefore, the data of these two channels were combined in an expectation of improved classification. Channel 3 is of interest because of its ability to distinguish between segments containing some cloud cover and those containing no clouds at all. The data of this channel were combined with those of Channel 2. Also, in order to make optimal use of the available data, it was decided to combine all five channels. Initially, these three types of test were performed. After it was observed that the Channel 2, 3 combination led to significantly improved results it was decided to also investigate the combined data of Channels 1, 2, 3 and Channels 2, 3, 4.

The channel data were combined by means of concatenating the appropriate feature vectors. Each feature vector used in the previous experiments has six components. As an example of how vectors were combined, consider the two sets used for the classification tests in Channels 1 and 2. Each vector in the Channel 1 set is generated from a specific segment in a Channel 1 image. Each one has a corresponding vector in the Channel 2 set generated from the analogous segment of the same scene in Channel 2. The information in the two channels was combined through concatenating each pair of corresponding vectors. Thus, the training set used for the Channel 1 and 2 combined test was of the same size as the training set used for testing Channel 1. However, each vector in the combined test had twelve components. The feature vectors of Channels 2 and 3 were combined in the same manner. The feature vectors of the combined Channels 1, 2 and 3 and Channels 2, 3 and 4 tests each had eighteen components. Finally, the corresponding feature vectors of all five channels were combined to form a thirty component vector for the classification experiment combining all channels.

The tests were conducted in the same manner as in the second experiment. Again, the feature vectors were partitioned into a training and a test set in four different ways. For each training set, Cascor was trained in five separate trials. All of them converged and were tested. Table 5 shows the results for the fused data. The nature of the misclassification of the "best trials" in each test is shown in

Table 5. The number of hidden nodes and percentage of misclassifications in each test case performed in the combined channels

Channels	Test ID	Average Number of Hidden Nodes	Average Number of Misclassifications in %	Average in Channels	Misclassifications in best trial in %
1 and 2	1	17	23		17
	2	18	10		0
	3	19	33		17
	4	17	32	24	17
2 and 3	1	19	9		0
	2	19	7		6
	3	18	15		0
	4	17	5	9	0
1, 2 and 3	1	12	12		0
	2	14	18		17
	3	13	10		8
	4	12	29	16	25
2, 3 and 4	1	17	15		0
	2	17	7		0
	3	18	9		0
	4	19	4	9	0
1, 2, 3, 4, and 5	1	12	15		0
	2	12	8		8
	3	12	10		8
	4	10	29	15	25

Table 6. The classification in Channels 2 and 3 combined of the best trials

classification assigned by Cascor ↓	actual classification →			
	dense clouds	medium clouds	sparse clouds	no clouds
dense clouds	0.92			
medium clouds	0.08	1.00		
sparse clouds			1.00	
no clouds				1.00

Table 7. The classification of all five channels combined of the best trials

classification assigned by Cascor ↓	actual classification →			
	dense clouds	medium clouds	sparse clouds	no clouds
dense clouds	0.88		0.08	
medium clouds	0.12	0.75		
sparse clouds		0.25	0.92	0.05
no clouds				0.95

Tables 6 and 7 for Channels 2 and 3 and all five channels combined, respectively.

Comparing the classification results of the fused Channel 1 and 2 data with the classification in Channels 1 and 2 separately, it is seen that the combined result gives the same overall classification performance. However, the fused data of Channel 2 and 3 showed significant improvement. Correct classification reached over 90% and matched the performance of the single image tests. In particular, the precise misclassification results displayed in Table 6 show that the "best trials" in each test had almost no misclassifications at all. When all five channels were combined, the classification performance dropped somewhat but is still better than classification in each of the channels separately. In particular, Table 7 shows that the separation between segments containing some level of cloud cover and those containing no clouds at all is quite good for these tests. The experiments in the three channel combinations showed similar results. It is remarkable, though, that in the Channel 2, 3 and 4 combination, there always was at least one trial that showed perfect classification.

Kohonen's self-organizing maps

The various sets of feature vectors were also used to produce the topological selforganizing maps. A topological map was generated for each channel separately as well as for the channel combinations discussed before.

When producing a map showing the organization of the feature vectors within a channel, the input layer must consist of six units since the single channel feature vectors have six components. In order to show the results of the combined channels, this input layer needs to be enlarged according to the increased size of the feature vectors. The competitive layer had 100 units organized as a 10 by 10 grid. The total number of feature vector presentations was 100,000. Convergence to a stable configuration was achieved. The size of the initial neighborhood was 5 by 5 and the initial learning rate was 0.2.

Figure 4 shows the topological map obtained from the Channel 1 data as an example. It is seen that the *no clouds* vectors are spread out most and show up in almost any segment of the plane. This is to be expected because these vectors represent many different textures. However, the different cloud types do not cluster very well either. Some clusters can be distinguished; for example, there is a *dense clouds* cluster consisting of five units in the top left quadrant of the plane. But some smaller clusters and isolated units representing the *dense clouds* texture are found in other locations. The *medium clouds* and *sparse clouds* patterns are distributed too. Similar distributions were observed in the other channels. It may be concluded that none of the five channels show strong clustering of feature vectors belonging to any of the four classes distinguished in the experiments. Thus, many of the feature vectors belonging to the same class are quite dissimilar. The clustering patterns of the larger vectors combining the results of more than a single channel were not significantly different.

CONCLUSIONS

The project researched the possibility of automated discrimination of a specified texture in AVHRR satellite images. The texture of cloud formations was selected and three different classes were defined based on the cloud density. Only a small set of satellite images was avail-

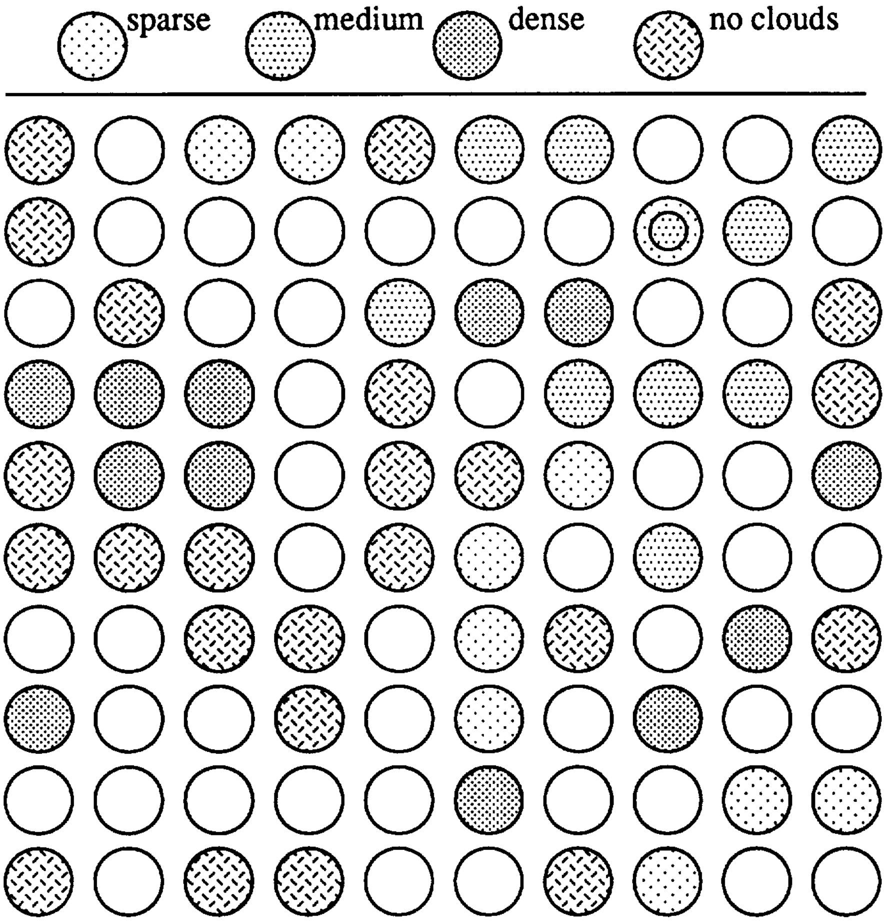

Figure 4. The clustering of the dense, medium, and no clouds feature vectors in Channel 1

able. Taking the difficulty of the classification into account, it may be concluded that this project was successful in the sense that it was found possible to discriminate cloud textures from all other textures with reasonable accuracy. Segments showing the various levels of cloud cover were extracted. In many segments, the cloud textures were mixed with various levels of noise due to small gaps in the cloud cover. The method of second-order gray level statistics was used to obtain feature vectors from these segments. The clustering properties of these vectors was studied by means of the Kohonen self-organizing maps.

The vectors generated by the *no clouds* class did not cluster very well as should be expected. These vectors represent many different textures and will show large variability. It should also be anticipated that the *sparse clouds* vectors would not cluster well. This turned out to be generally the case (although the largest cluster observed in any of the maps belonged to the *sparse clouds* class). The *medium clouds* and *dense clouds* feature vectors were expected to cluster better as compared to the other two classes, but this was found not to be the case. In order to obtain better clustering properties of feature vectors, different preprocessing methods could be studied. Possible candidates are the two-dimensional Fast Fourier transform, the Gabor transform and wavelets expansions. However, it should be realized that cloud textures show large variability and the classification problem may be inherently difficult, independent of which preprocessing technique is used.

Given this large variability in feature vectors within a class, it does not seem advisable to use a self-organizing neural network architecture for classification. The topological maps generated by the Kohonen network were of interest because they revealed the complexity of the classification problem. However, if this architecture had been used as a classifier, it would have generated many misclassifications. A neural network architecture employing supervised learning is better suited for this type of classification as demonstrated by this project. The Cascade-Correlation network performed well. The best results were obtained when data from Channels 2 and 3 or Channels 2, 3 and 4 were fused. In these cases, the four classes could be distinguished with an average accuracy of 91%. Moreover, several tests in these channel combinations showed no misclassification at all. If these better performing trained networks could be recognized in advance, much better classification results could be obtained.

We recently became aware of a similar study performed by Slawinski *et al.*, 1991. These researchers used the backpropagation architecture to classify different levels of cloud cover against an ocean background in AVHRR images. They used the pixel gray levels of small image segments together with first-order statistics measures as inputs to the neural networks. Their best results (93% correct classification) are similar to the best classifications obtained in our project. However, the ocean provides a rather homogeneous background and the variability in their images is essentially introduced by the cloud textures. When the background itself shows large variability, as in the majority of the images used for our project, classification methods that are largely based on the actual values of the pixels may not be successful.

This feasibility study has proved the possibility of automated satellite image classification. Future research could focus on the distinction of many different textures in these images. Eventually, a software package could be implemented that partitions an image into a set of overlapping segments and then scans each segment in an attempt to classify it according to its dominant texture. A set of identified textures could be used as an index in a data base through which images could be stored and retrieved. Research will be required to specify an appropriate set of textures. Various preprocesing techniques need to be investigated with respect to the clustering properties of the generated feature vectors. Additional research may be required to select the most appropriate neural network architecture. Based on the current results, Cascade-Correlation seems a good candidate for the expanded classification task. However, there is some evidence that the generalization characteristics of Cascade-Correlation are not as good as those of the backpropagation network (Crowder, 1990). Thus, it may be useful to consider additional architectures.

Acknowledgements: The satellite images were made available by CTA, Incorporated. The code for the Cascade-Correlation neural network was made available by R. Scott Crowder, III of Carnegie Mellon University. The code for the Kohonen self-organizing maps was written by Juan C. Soto, Jr. of the University of Colorado at Colorado Springs. The training of the Kohonen network was performed by Tammy L. Skufca, also of the University of Colorado at Colorado Springs.

REFERENCES

Crowder, R. S. (1990). Predicting the Mackey-Glass Timeseries with Cascade-Correlation Learning. In D. S. Touretzky (Ed.), *Proceedings of the 1990 Connectionist Models Summer School* (pp. 117-123). San Mateo: Morgan Kaufmann.

Fahlman, S. E., and Lebiere, C. (1990). The Cascade-Correlation Learning Architecture. In D. Touretzky (Ed.), *Advances in Neural Information Processing Sytems 2* (pp. 524-532). San Mateo: Morgan Kaufmann.

Fahlman, S. E. (1988). Faster-Learning Variations on Back-Propagation: An Empirical Study. In D. Touretzky (Ed.), *Proceedings of the 1988 Connectionist Models Summer School.* San Mateo: Morgan Kaufmann.

Goodman, A. H., and Henderson-Sellers, A. (1988). Cloud detection and analysis: A review of recent progress. *Atmos. Res.*, 21, 203-228.

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 610-621.

Kohonen, T. (1988). The Neural Phonetic Typewriter. *Computer*, 21, 11-22.

Lee, J., Weger, R. C., Sengupta, S. K., and Welch, R. M. (1990). A Neural Network Approach to Cloud Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 5, 846-855.

Rossow, W. B. (1989). Measuring cloud properties from space: A review. *J. Climate*, 2, 201-213.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Internal Representations by Error Propagation In D. E. Rumelhart and J. L. McClelland (Eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1: Foundations (pp. 318-364). Cambridge, Mass.: MIT Press.

Slawinski, O., Kowalski, J., and Cornillon, P. (1991). A Neural Network Approach to Cloud Detection in AVHRR Images, *Proceedings of the International Joint Conference on Neural Networks, Vol. I*, 283-288.

Weszka, J. S., Dyer, C. R., and Rosenfeld, A. (1979). A Comparative Study of Texture Measures for Terraine Classification. *IEEE Transactions on Systems, Man, and Cybernetics,* 6, 269.