1995

Image Classification using Textural Neural Networks

Edit J. Kaminsky
University of New Orleans, ejbourge@uno.edu

Herb S. Barad
Tulane University of Louisiana

Recommended Citation
Kaminsky, E. J., and H. Barad, "Image Classification using Textural Neural Networks," in lASTED Signal and Image Processing Cont. Proc. (SIP'95), (N. M. Namazi, editor), Las Vegas, NV, Nov. 20-23,1995, pp. 276-279.

Image Classification using Textural Neural Networks

Edit J. Kaminsky
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148
tel: (504) 286-5616 fax: (504) 286-3950
ejbee@uno.edu

Herb S. Barad
Department of Electrical Engineering
Tulane University
New Orleans, LA 70118
tel: (504) 865-5785

Abstract

This paper presents backpropagation neural networks that utilize texture information to accurately classify photographic images. Training with minimum sets is shown to yield excellent results.

1 Introduction

Vast amounts of remotely sensed data are collected to gather information about natural resources by scanning the surface of the earth. Low and high resolution sensors are currently in use and airborne photographic data are also collected. Identification, classification, and interpretation of information contained in these data are performed by experienced and skilled personnel for tasks such as environmental monitoring and disaster relief. The research reported here focuses on the utilization of novel textural neural networks (NN) to classify remotely sensed data according to land use/cover type.

The performance of combined spectral and spatial features for land use classifications was reported in [1]. The idea of incorporating textural information into a neural classifier was investigated by the current author in [2] and [3], where it was shown that incorporation of textural information improved the neural classification of Multispectral Scanner (MSS) data.

Two types of textural neural networks are studied here: the textural net with pre computed texture information, and the sliding-window textural net. The former computes texture measures from the sensor data prior to the classification process; these measures, together with the spectral data, are fed to a feed-forward neural network which classifies the input data as belonging to one of several types of land use/cover. The sliding-window network tries to incorporate texture in "real-time" by allowing the network to feed in parallel from a neighborhood of pixels; the pixel in the center of the sliding window is classified. These classifiers are applied to a filtered photographic image from Southeastern Louisiana. The use of small training sets is also investigated.

2 Texture Analysis

Image texture describes the primitives that compose the texture and the spatial dependence or interaction between them. Haralick [1] presented the Spatial Gray Level Dependence (SGLD) procedure for extracting 14 features from blocks of digital image data. This method was selected for our study because it has been shown to yield very good results that do not seriously deteriorate with noise. The texture information in the SGLD method is specified by the co-occurrence matrix $[P(d,\theta)]_{i,j}$. Element P_{ij} of $[P(d,\theta)]_{i,j}$ is the frequency of occurrence of one gray tone i and one gray tone j, separated by a distance d with angle θ between them. The distance and angles typically used are $d = \{1,2\}$ and $\theta = \{0^0, 45^0, 90^0, 135^0\}$. A principal components transformation is performed on the set of SGLD features and the most prominent component from each of the three bands is used for the network with pre-computed features.

The selection of the window size must be dependent on the resolution of the sensor. Larger windows will provide accurate results only if the window covers areas of relatively even texture. Also, since we would like to use the same sliding window to precompute textures as we do for the sliding window network, using a larger window size would cause the number of inputs to the latter network to increase, increasing the network's complexity. We tested our systems with 3x3 and 5x5 pixel^2 windows, but the two didn't provide significantly different results in most cases.

3 The Photographic Data

We use an aerial photograph taken with a Zeiss color infrared camera and digitized to about one-meter (1 m) resolution using an Iconic digital scanner which filtered the original into three bands for false color generation. The data corresponds to an area of about 1 Km by 1 Km from Southeastern Louisiana that includes a distilling, storage, and pumping facility for natural gas, and for which some ground truth is available. The location is across from Pilot Town in the Garden Isle district of the Mississippi River. The five classes in the photographic image are: grass, forest, stagnant water, flowing water, and urban. The photographic data is normalized to the range $(0.0, 1.0)$ before feeding the values to the neural network.

4 Neural Network Classifiers

The output generated by the neural networks is coded as winner-takes-all. The generalized delta rule was selected for all our networks, and the error for the output layer is the difference between the desired output and the net's output. The error is then transformed by the derivative of the transfer function and backpropagated to prior layers where it is accumulated. Both the net with pre-computed features and the sliding window classifier consist of fully-connected, three-layer, feed-forward, backpropagation networks with sigmoid transfer function. The varying parameters are the number of inputs, the type and dimension of the input data, and the number of neurons in the hidden layer.

The weight update equation for the generalized delta rule is: $w'_{ij} = w_{ij} + C_1 e_i x_{ij} + C_2 m_{ij}$, where C_1 and C_2 are the learning and momentum coefficients, respectively, x_{ij} is the j^{th} input to the i^{th} neuron in the current layer, w_{ij} is the connecting weight from the j^{th} input to the i^{th} neuron in the current layer, w'_{ij} is the weight after it has been updated, e_i is the error of the i^{th} neuron, and m_{ij} is the momentum term updated with: $m'_{ij} = w'_{ij} - w_{ij}$,which smooths out the weight changes, such that general trends in weight change are reinforced whereas oscillatory changes tend to be canceled. As the learning process progresses, decreasing the values of C_1 and C_2 causes the network to converge better.

Several training strategies were pursued : 1) A training set consisting of 5,000 randomly chosen examples per class (total of 25,000 training pixels) was used for most training. 2) To test the training ability of the networks with minimum training sets, a set of only 500 randomly selected pixels was generated; this training set was representative of the actual distribution of classes within the image, and therefore the number of training pixels varied much from class to class. 3) A new set of 500 training examples was generated by choosing 100 random pixels from within each of the five classes. Successful training of networks was accomplished with as few as 100 examples per class of the photographic data. Each example in the training set must be presented to the network many times in order for the system to learn. We trained our nets for 110,000, 220,000, 330,000, and 500,000 iterations, with the training data randomly chosen from the already random training set. Testing was performed by sequentially classifying each pixel in the complete images.

4.1 Textural NN with Pre-computed Features

Ten textural features—(1), (2), (4), (6), and (8)–(13) of [1]—are computed for each pixel for a distance of 1 pixel and angles in the set $\{0^0, 45^0, 90^0, 135^0\}$ for each of the three bands. The average over the angles is then computed to yield a ten-dimensional feature vector per image pixel per band. The principal component of this space is then computed to reduce the dimensionality of the space to only one dimension per band. The neural systems were trained and tested using principal component textural features alone (three inputs) and principal components textural features along with the original RGB values (six inputs). Pure spectral (RGB) values (three inputs) were used as a baseline.

The baseline neural network classified based only on the RGB values of the pixel to be classified, and consisted of 3 input neurons, 5 hidden neurons, and 5 neurons in the output layer (i.e., configuration 3–5–5); the textural neural classifier with pre-computed features had configuration 6–10–5.

4.2 The Sliding-Window Classifier

Feature vectors fed to the sliding-window networks consist purely of remotely sensed spectral data. For each pixel being classified, a set of N^2 pixels are fed to the network, where N is the length (and width) of the window in pixels, and is odd. The pixel being classified is always in the center of the window. Each pixel is three-dimensional so that a total of $3N^2$ components form the input feature vector to the classifier. The idea behind the sliding-window net was to provide the neural network with pixel values in the neighborhood of the pixel being classified for the system to extract texture-like information on its own. The sliding window neural classifier had configuration 27–40–5 for the 3-by-3 sliding window.

5 Results

Our evaluation of classifier performance is based primarily on classification accuracy. Confusion tables provide information about the accuracy of the textural neural classifiers and also provide insight into the confusion between classes. Results from the analysis of training set size are also given.

5.1 Accuracy

Results of the neural network classification of the high resolution photographic data were obtained for the full training set after training had stopped, and for the complete image from which the training pixels were selected. Only a slight deterioration was noticed when testing over the complete data set, indicating that training results are very representative of the actual network performance.

Table 1: Accuracy of the 3–5–5 spectral net(%).

Class	Grass	FW	Urban	SW	Forest
Grass	**98.94**	0.55	0.00	0.00	0.19
FW	0.04	**99.01**	0.96	0.00	0.00
Urban	0.00	0.07	**99.93**	0.00	0.00
SW	0.18	2.01	0.00	**97.80**	0.02
Forest	3.78	0.00	0.00	0.96	**95.26**

1.81% incorrect overall

Table 2: Accuracy of the 6–10–5 textural net (%).

Class	Grass	FW	Urban	SW	Forest
Grass	**95.48**	0.85	0.10	0.00	3.57
FW	1.47	**96.05**	2.13	0.35	0.01
Urban	4.65	1.53	**93.83**	0.00	0.00
SW	0.07	2.53	0.00	**95.05**	2.35
Forest	2.37	0.00	0.00	2.80	**94.83**

4.96% incorrect overall

The 3–5–5 spectral network was first trained for 200,000 iterations using 25,000 RGB samples (5,000 from each of class). The average classification accuracy for this spectral net was over 98%. Clearly, no ancillary data is required to achieve acceptable classification performance for clean (low-noise), high-resolution data. Networks with only the principal component of each band of the SGLD texture space as inputs were trained and tested, and proved to be able to somewhat discriminate between classes (with a 60.54% accuracy). Confusion between the two classes of water and the two classes of vegetation occurred often, as expected due to the similar textural characteristics within macro-classes. The textural network with configuration 6–10–5 performed very well (97.40% training accuracy), with training results comparable to those obtained with the spectral neural network. The sliding window textural network did not perform as well as expected, with a 23.78 % training error rate. Since the input to the sliding window network includes the pure RGB values for the pixel being classified, along with this information for the surrounding pixels, it was expected that a performance at least as good as that of the spectral network would be achieved.

Table 1 shows the results of classifying the photographic image using the spectral network trained with the large set. Confusion occurs mainly between water types (stagnant/running) and vegetation types (forest/grass). The testing confusion table for the textural 6–10–5 network trained with the small uniform set is given in Table 2, with average accuracy is 95.04%. These results are encouraging because although the performance of the classifier has not been enhanced by adding the textural features, it became clear that doing so with clean data does not deteriorate the performance much; i.e., the neural network itself determines that the RGB spectral data is "best" and somewhat ignores the ancillary data. The sliding window textural networks did not perform as well with the high resolution data as they did with the low-resolution MSS data [4, 3]. The performance of the 27–40–5 sliding window network averaged 66.51% correct classification for testing of the entire set. Most misclassifications occur around the borders between classes. The performance of the textural network is slightly worse overall, but better in the interface between water classes in the channel.

5.2 Influence of Training Set

It is desirable to use minimum training sets to reduce the cost of gathering ground truth samples and to minimize the involvement of human experts. An important result of the work presented in this paper is that we were able to train neural networks with a very small subset of the photographic data. The smallest training set used was constituted by only 500 samples out of more than one million (0.05%); accurate training may be possible with even fewer samples. The classification results on the entire testing set using the small training set are surprisingly good, with less than 3% error rate, and all classes being extremely accurate. This is a very important result since one of the problems of supervised NN (and many other supervised algorithms) is the need for large training sets. The smallest the number of ground truth samples needed, the less expensive it is to collect them, the less interaction needed from human experts, and the more autonomous the AI system can be made. Results from two small training sets, both of the same size but containing different number of examples per class, indicate that the neural network performs much better when all classes are equally represented during training.

6 Conclusions

All the configurations tested worked very well proving that neural networks are a powerful, accurate, and fast method for classifying vast amounts of high resolution data for which a small number of ground truth samples is available. The introduction of texture measures does not significantly alter the classification accuracy for high resolution data. Very few ground truth samples are needed to train neural nets to classify photographic data with very small error rates.

References

[1] R. M. Haralick and K. Shanmugam, "Combined spectral and spatial processing of ERTS imagery data," *Remote Sensing of Environment*, vol. 3, no. 13, pp. 3–13, 1974.

[2] E. J. Kaminsky, D. Miller, and S. Rana, "Neural network classification of remote sensing data," *Computers and Geosciences*, vol. 21, pp. 377–386, April 1995.

[3] E. J. Kaminsky, H. Barad, W. Brown, and S. Rana, "Textural neural networks for classification of remotely sensed data," Center's Director Discretionary Fund Report CA09-9401, NASA, Computer Systems Branch/Sverdrup Technology, Data Services Dept., Stennis Space Center, MS, Aug. 1994.

[4] E. J. Kaminsky, H. Barad, and W. Brown, "Textural neural network and version space classifiers for remote sensing," *Internat. J. Remote Sensing*, 1995.