

Searching for Patterns in Remote Sensing Image Databases Using Neural Networks

Justin D. Paola

Robert A. Schowengardt

CNASA-CR-199549)

~~(NIPS-95-05575)~~ SEARCHING FOR
PATTERNS IN REMOTE SENSING IMAGE
DATABASES USING NEURAL NETWORKS
(Research Inst. for Advanced
Computer Science) 12 p

N96-13235

Unclass

G3/63 0073243

RIACS Technical Report 95.17
August 16, 1995

Appeared in *Proceedings, 15th Annual IEEE International Geoscience and Remote Sensing Symposium*, Florence, Italy, July 20-14, 1995, pp. 443-445.

Searching for Patterns in Remote Sensing Image Databases Using Neural Networks

Justin D. Paola

Robert A. Schowengerdt

The Research Institute for Advanced Computer Science
is operated by Universities Space Research Association,
The American City Building, Suite 212, Columbia, MD 21044, (410) 730-2656

Work reported herein was partially supported by the National Aeronautics and Space Administration under Contract NAS 2-13721 to the Universities Space Research Association (USRA) and under Grant NAG 5-2198 to the University of Arizona Department of Electrical and Computer Engineering. Work performed at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffet Field, CA 94035-1000

ABSTRACT

We have investigated a method, based on a successful neural network multispectral image classification system, of searching for single patterns in remote sensing databases. While defining the pattern to search for and the feature to be used for that search (spectral, spatial, temporal, etc.) is challenging, a more difficult task is selecting competing patterns to train against the desired pattern. Schemes for competing pattern selection, including random selection and human interpreted selection, are discussed in the context of an example detection of dense urban areas in Landsat Thematic Mapper imagery. When applying the search to multiple images, a simple normalization method can alleviate the problem of inconsistent image calibration. Another potential problem, that of highly compressed data, was found to have a minimal effect on the ability to detect the desired pattern. The neural network algorithm has been implemented using the PVM (Parallel Virtual Machine) library and nearly-optimal speedups have been obtained that help alleviate the long process of searching through imagery.

INTRODUCTION

Neural networks have proven their worth as supervised multispectral classifiers in many previous experiments. With the advent of EOS and other remote sensing platforms, a major challenge in the near future will be the task of searching large remote sensing image databases for patterns of interest in particular applications. These patterns might be spectral, spatial, temporal, or any combination thereof.

There are several challenges in moving from multi-class training on a single image to a single-class search over many images. The first is that of defining training data for the neural network. Although training is simpler and faster with only one class, it is very important to provide adequate competing training sites so that the number of false alarms during searching will remain low. In a multi-class case this task is easier since the feature space is automatically partitioned into several segments, resulting in the area for any given class being relatively restricted.

Another challenge is presented by the variability of the images in the database. Factors that hinder pattern matching over multiple images are changing atmospheric conditions, changing sensor characteristics, changes on the ground over time, and changing sun angle.

A third challenge is the storage requirement of large image databases. Any pattern searching routine may have to handle data subject to lossy image compression.

Finally, the issue of processing time must be addressed. Searching through large databases of imagery, especially if large spatial windows are used, requires intensive processing.

FEATURES USED FOR DETECTION

In the example explored here, a 3x3 window was used in each of the six non-thermal Landsat Thematic Mapper (TM) bands. This is just one of many possibilities for the feature used in the search; any conceivable combination of spectral, spatial, and temporal features are possible using a neural network pattern detector. The network input layer can be adapted to accept whatever input is chosen. The size of the next layer, often called the hidden layer, must generally be specified experimentally, although we found that the exact number often made little difference if training was allowed to progress to a

large number of iterations. The example given here (detecting dense urban areas) was essentially a spectral problem. The 3x3 windows were added to provide a measure of texture into the process. Certainly, to detect a spatial pattern, larger windows would be used (e.g., see [1]).

COMPETING PATTERNS

The nature of the neural network pattern detection method is one of competition for decision regions in the feature space. These regions are adjusted by the network training algorithm iteratively until a minimal mean square error is achieved between the desired and actual output of the net. For a single pattern detection, there is usually a single output for the network, and this output has a high value for the desired pattern and a low value for all other patterns. It is the combined error over all patterns that is minimized.

The challenge is in how one defines 'all other patterns' and provides these to the network for competitive training against the desired pattern. The easiest way, from a training standpoint, is to provide several specific competing patterns – for example, the other classes from a supervised classification. Other studies have shown how, in a multi-class case, the network partitions the feature space to provide an accurate classification [2,3].

Another method is to select random, instead of specific user-selected, competing patterns from the image(s) containing the desired pattern. This is easier to implement, as all other signatures do not have to be accounted for by a human image interpreter. A variation of this is to use a grid of competing patterns. In either case, the competing patterns might by chance pick up some samples of the desired pattern. Since the network minimizes the error over all patterns, this, in general, does not cause problems.

A third possibility is to create synthetic patterns by setting the value of each feature (e.g., the value of the pixel in each band) randomly. This has the advantage of requiring no image from which to extract patterns.

These methods were attempted for a relatively simple pattern detection problem – that of finding dense urban areas in Landsat TM imagery. The search results for this pattern are easy to verify visually. A 3x3 window in each of the 6 non-thermal TM bands was used as the feature for training. The 'dense urban' area was defined by a 9x9 region of downtown Tucson, Arizona. Thus there were a total of 49 3x3 patterns defining the search pattern. Fig. 1a shows the results of using the training regions for the other classes of a multispectral classification to compete against the desired pattern. The urban area was highlighted, but there were also some other areas with high values, that, depending on the threshold used for detection, may produce false alarms.

The second example (Fig. 1b) shows the results of using a combination of the second and third competing pattern methods discussed above. The competing patterns consisted of patterns from the image on an evenly spaced grid, and an equal number of uniformly generated random patterns. In this case the desired pattern was much better separated from the background. However, a network trained only with uniformly generated competing patterns did not perform very well. The feature space is simply too large to be adequately covered by a reasonable amount of random patterns.

IMAGE CALIBRATION

Ideally, an image database would consist of, for example, reflectance values, with all atmosphere, sensor, and ambient light differences removed. Unfortunately, such datasets are uncommon. Thus, we have tried a simple mean and standard deviation matching of the images.

The first test was to try the network, as trained on the Tucson image, on other imagery. The network used to produce Fig. 1a was implemented on a TM image of Oakland, California. Although the urban area was detected accurately, the bay and other water bodies created false detections. This was due to the Tucson image lacking a water class, and is not necessarily a calibration problem. This was shown to be the case when the network of Fig. 1b, which was trained using randomly generated patterns in addition to those from the Tucson image itself, was run on the Oakland image. The results were very good (Fig. 2a).

The Tucson and Oakland images were level 1 Landsat data, with no conversion to reflectance. So, in addition to the intrinsic differences between the two areas, there were other differences due to atmospheric conditions, sun angle, etc. These external differences were not as great, however, as they were between the Tucson image and an image of the Washington, D.C. area. To provide a simple correction, the Washington image was adjusted so that each band matched that of the Tucson image in mean and standard deviation for the area shown. The results of the detection on this adjusted image are shown in Fig. 2b.

COMPRESSED IMAGERY

The calibration problems presented above are common to all remote sensing imagery. Another problem that might be encountered in a large image database that has nothing to do with the quality or calibration of the data is lossy compression. Lossy compression schemes such as the industry standard JPEG, can provide compression ratios of up to 30:1 and still maintain visual integrity. The effect of this compression on pattern detection will depend on the specific problem. Fig. 2c shows the results from the Tucson-trained network as applied to a 29:1 JPEG compressed version of the Oakland image. While some of the pixel-to-pixel detail was eliminated, the image is practically identical to that of Fig. 2a for the purpose of the pattern detection.

PVM IMPLEMENTATION

Searching for patterns in large image databases, particularly when windows of pixels are involved, requires intensive computing. Furthermore, the neural network backpropagation training algorithm is often computationally intensive. In the case of single pattern detection, however, the network has only one output node and the computation is not excessive. The second phase of using the trained net to search through images, on the other hand, can be quite time consuming, both in data I/O and computation, particularly when windows of pixels are involved. Table 1 shows timings for the example discussed here, as well as for a simple spectral search (no window). The training times are given for 10,000 iterations. It should be noted that adequate detections are capable with far less training. It is clear, however, that the large number of network inputs used in windowing the data results in significantly increased processing time.

A PVM implementation of the searching phase has been implemented on a cluster of SUN workstations. Since different images, or different parts of an image, can be

processed independently, this cluster approach works quite well for this application. It does not work as well for the training phase, where there is a great deal of communication relative to the processing each workstation would do. An efficient image classification method uses the "bag of tasks" paradigm. A central "administrator" process, preferably run on the machine that has fastest access to the image data, sends out the neural network configuration and interconnecting weight values to each "worker" process. The network representation is usually quite compact – e.g., the network used in the dense urban search required only 330 floating point values to store the weights. Then the administrator sends one line (or 3 lines for a 3x3 input net) of an image to each worker. When the worker is done it sends the search result (or classification) back to the administrator and requests another line. A worker task that is running on a more powerful, or less loaded, machine will take more image lines and the procedure will be done in an optimal fashion for the set of workstations at hand. In our experiments, we have found the speedup on three equally-equipped machines to be just under the optimal value of three.

Table 1: Neural net timings on a SUN SPARC 10. The first two columns show the results from a simple spectral pattern search (6 bands). The next two columns are for the net that produced Fig. 1b. In both cases the network had 6 hidden nodes and 1 output node, and was trained to 10,000 iterations with a 9x9 pixel training area and 1144 competing patterns.

	Training time for 6 input net (no window)	Search of 900x900 image	Training time for 54 input net (3x3 window)	Search of 900x900 image
Time	2,690 sec	130 sec	15,660 sec	829 sec

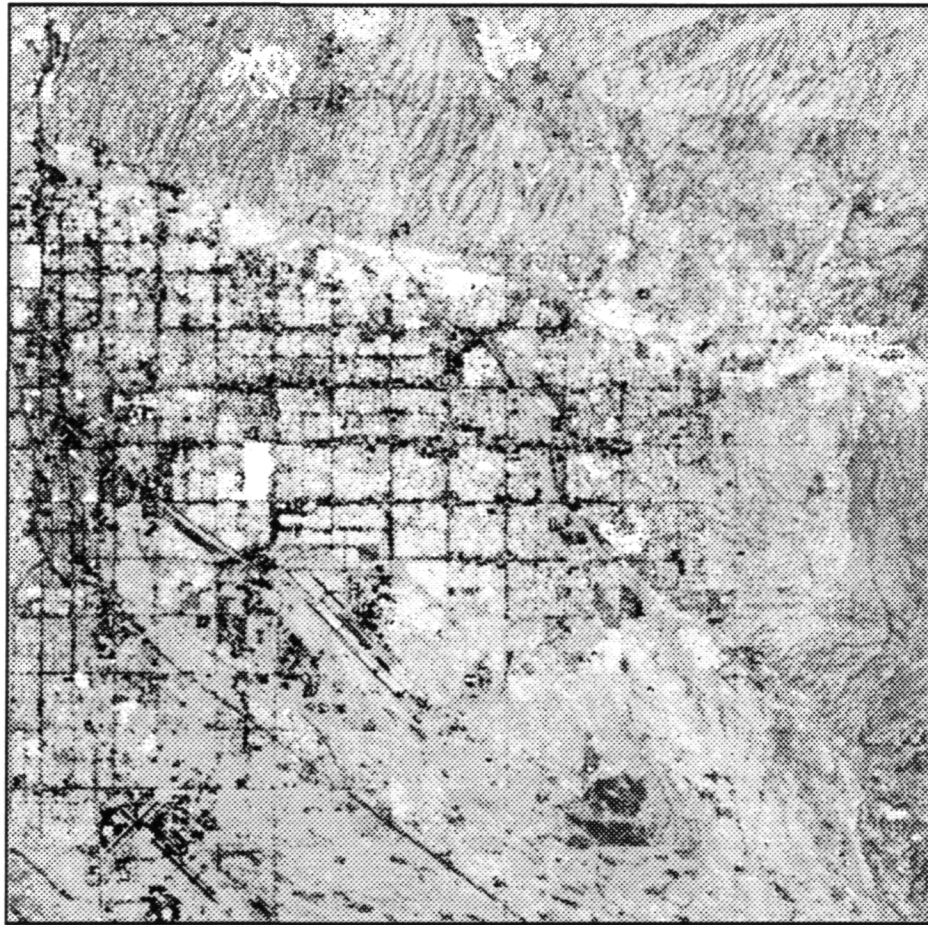
CONCLUSION

A general pattern matching algorithm is not expected to achieve consistently high accuracy for all the varied imagery used in remote sensing applications. Fortunately, for this application, the goal is not to achieve the highest possible accuracy, but to provide a good estimate of candidate matches that can be used to guide further investigation. The flexibility of the neural network allows for adaptation to many different types of imagery and pattern signatures, while providing moderate accuracy in pattern matching.

In addition to the dense urban area detection discussed in this paper, we have attempted other searches, including more subtle TM classes such as 'grassland' and 'pine-oak woodland', as well as other land-cover classes using temporal NDVI data. These patterns are more difficult to detect, particularly in imagery not used for training. This result stresses the need for a well-calibrated dataset of global imagery in the EOS era in order to achieve widely applicable content-based browsing of the type investigated here.

REFERENCES

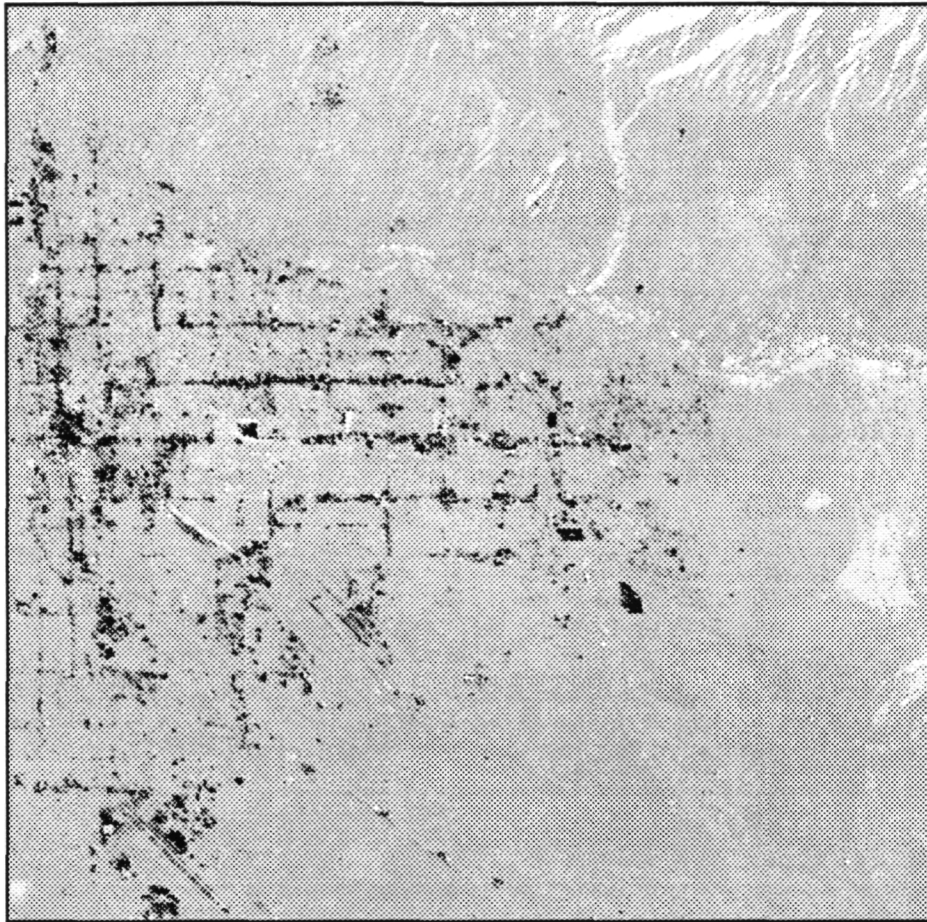
- [1] J. D. Paola and R. A. Schowengerdt, "Comparisons of Neural Networks to Standard Techniques for Image Classification and Correlation", *Proceedings, 14th Annual IEEE International Geoscience and Remote Sensing Symposium*, pp. 1404-1406, Pasadena, Ca., Aug. 1994.
- [2] R. A. Schowengerdt and J. D. Paola, "Parallel Computing and Data Compression for Pattern Matching in Remote Sensing Image Databases", *Proceedings, Conference on Recent Advances in Remote Sensing, The European Symposium on Satellite Remote Sensing*, Rome, Italy, Sept. 1994.
- [3] F. Fierens, I. Kanellopoulos, G. G. Wilkinson, and J. Mégier, "Comparison and Visualization of Feature Space Behavior of Statistical and Neural Classifiers of Satellite Imagery", *Proceedings, 14th Annual IEEE International Geoscience and Remote Sensing Symposium*, pp. 1880-1882, Pasadena, Ca., Aug. 1994.



a

Figure 1: Search results for dense urban areas in a Landsat Thematic Mapper (TM) image of Tucson, Arizona, after training on the same image using a 9x9 region to define the 'dense urban' pattern. The network used a 3x3 window in each of the six non-thermal bands, and had 6 hidden layer nodes. Darker values represent more likely matches.

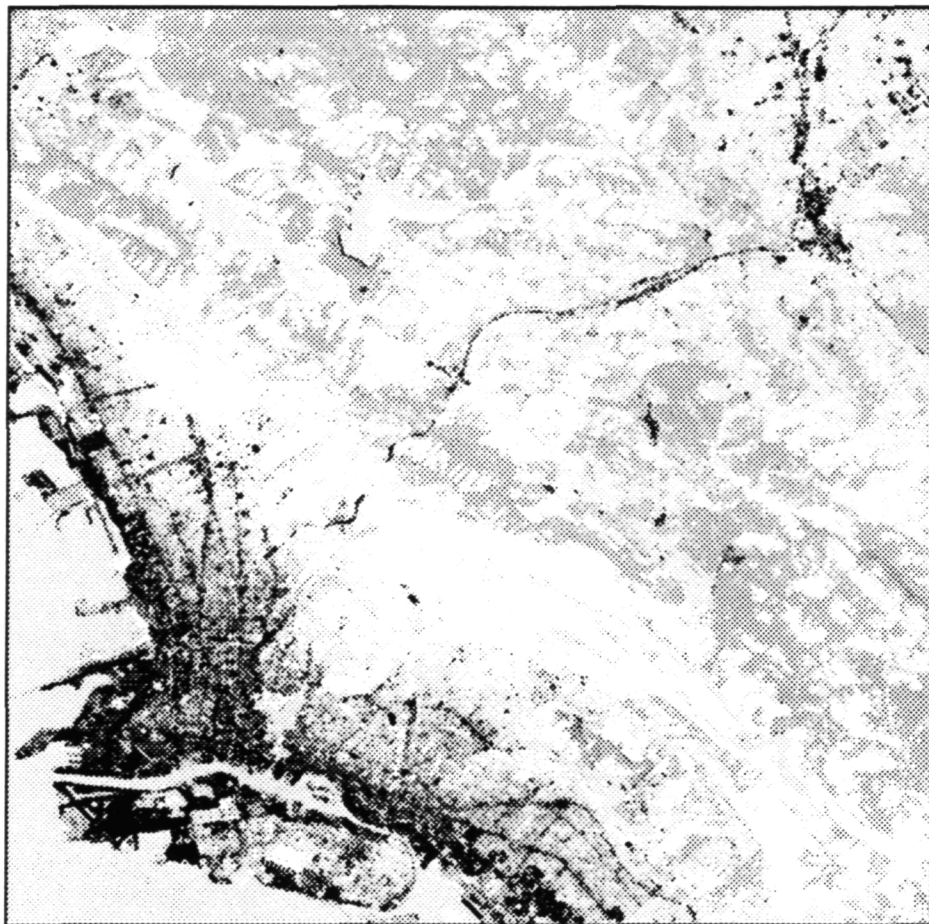
- a) In this case, the competing patterns were the training data from all the other classes of a supervised classification (a total of 394 competing patterns).



b

Fig. 1 (cont.): Search results for dense urban areas in a TM image of Tucson, Arizona, after training on the same image using a 9x9 region to define the 'dense urban' pattern. The network used a 3x3 window in each of the six non-thermal bands, and had 6 hidden layer nodes. Darker values represent more likely matches.

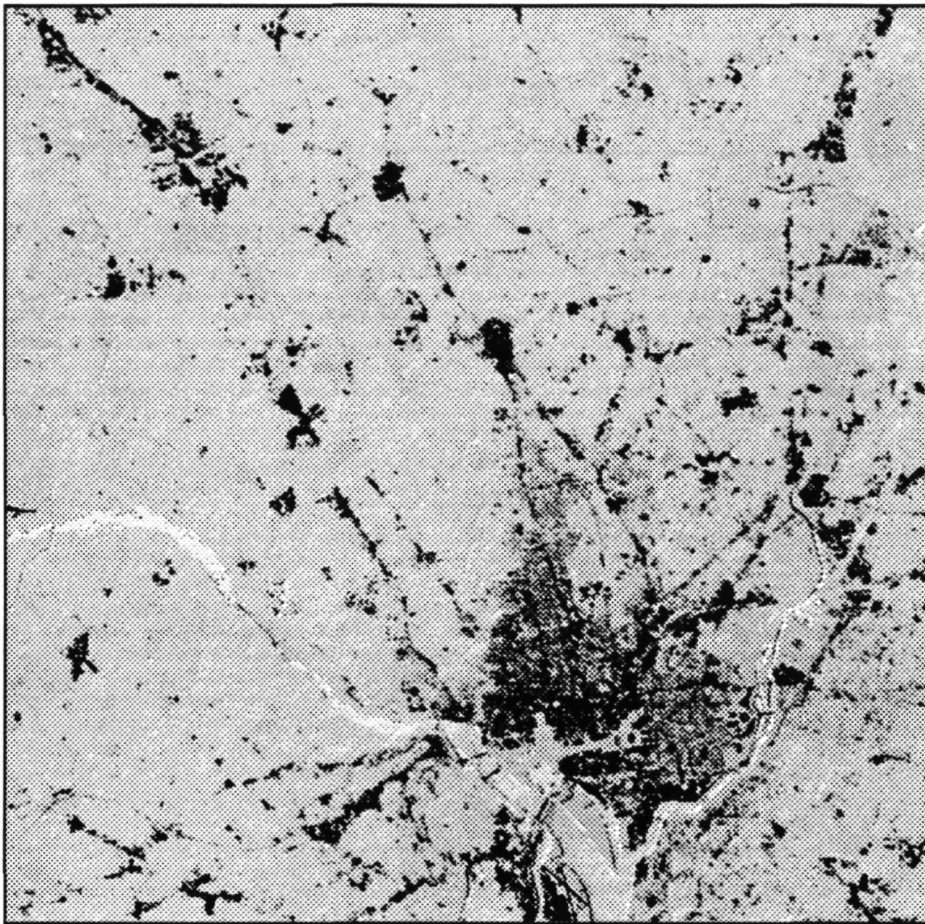
b) The competing patterns were a grid of 572 patterns from the image supplemented by 572 uniformly generated random patterns (for a total of 1144 competing patterns).



a

Figure 2: The network as trained in Fig. 1b, applied to:

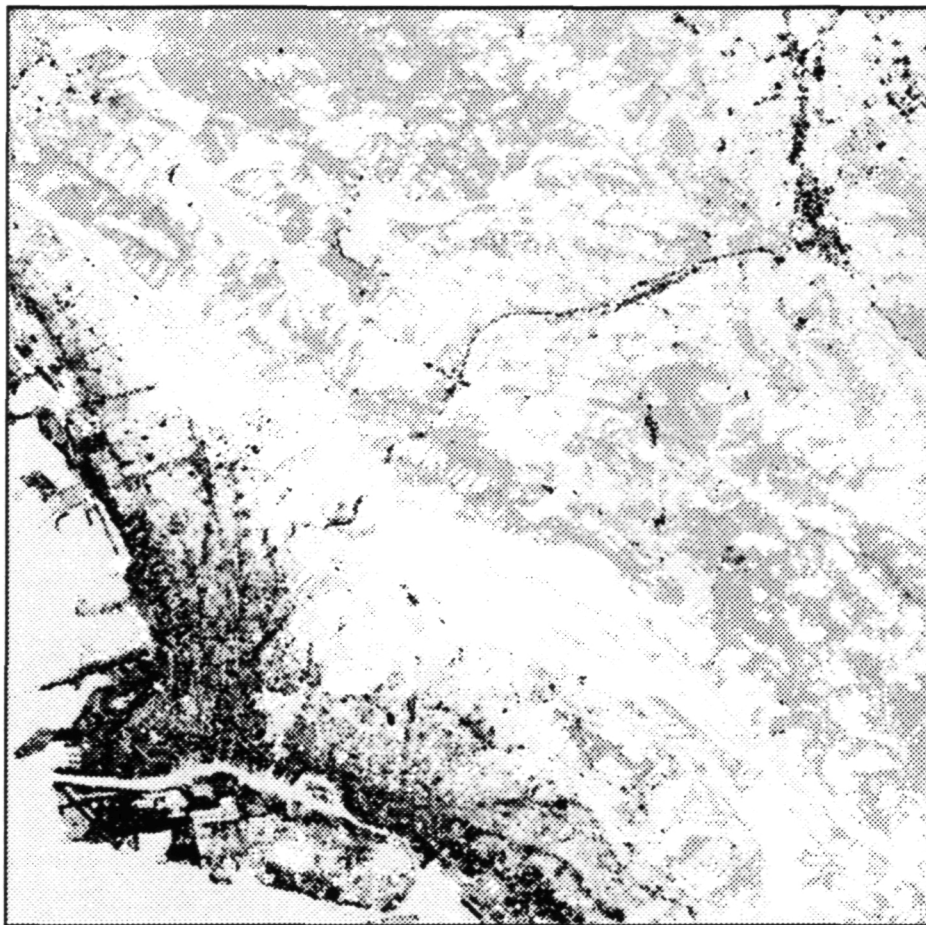
a) a TM image of Oakland, California with no relative calibration.



b

Fig. 2 (continued): The network as trained in Fig. 1b, applied to:

- b) a TM image of Washington, D.C., which was calibrated relative to the Tucson image using a simple mean and standard deviation match in each band.



C

Fig. 2 (continued): The network as trained in Fig. 1b, applied to:

C) a 29:1 JPEG compressed version of the Oakland TM image.



Mail Stop T041-5
NASA Ames Research Center
Moffett Field, CA 94035