Wavelets for training a neural network to recognize objects

John F. Pedersen and Edward B. Saff

University of South Florida,
Institute for Constructive Mathematics,
Department of Mathematics,
Tampa, Florida 33620-5700

Kevin Schweiker

Hercules Defense Electronic Systems, Inc.,
P.O. Box 4648, Clearwater, Florida 34618

ABSTRACT

A technique is developed for classifying different objects in natural imagery by employing a wavelet transform and training a neural network on certain of the wavelet transform coefficients. The effectiveness of different choices of coefficients and neural network architectures is analysed.

1. INTRODUCTION

The wavelet transform has recently enjoyed much popularity in image processing applications, primarily for image compression, image enhancement and edge detection [5, 6, 9]. This paper focuses on its use for object recognition, specifically for different types of vehicles in landscape ladar imagery. Pattern recognition involves the assignment of observed data sets (samples) to one of two or more disjoint classes depending on some "features" of the data set. Once the features have been determined, the optimal method of classifying them into the desired classes becomes a statistical question, which has received much attention — a classic text is [8]. A more recent approach to pattern recognition is with neural networks [11], although some training algorithms used for them bear a strong relationship with iterative gradient descent methods of the more traditional statistical approaches.

The raw data were a number of 256 x 256 ladar range/intensity images, with intensity pixels having values 0-255 and range pixels having values $0\text{-}2^{16}$, but effectively $0\text{-}2^{12}$. This data was separated into range and intensity components and this initial study of feasibility of a combined wavelet transform/ neural network approach used only the intensity data. However, correlation with the range data for the purposes of object location and further validation of classification is a valuabe asset of this type of data, which should be explored in further studies.

2. WAVELETS

In Fourier analysis, functions are expressed as a linear combination of basis functions, the sines and cosines of various frequencies. Sine and cosine have non-compact support. Recently there has been great interest in using bases consisting of functions supported on a finite interval [3, 4, 9]. A single function with support, say $[0, 1)$, when dilated and translated by all integral powers of two, can provide such a basis. These functions are wavelets. Because of their local support, they have better approximation properties for certain classes of functions.

Wavelets are built from a function satisfying a dilation equation

$$\phi(x) = \sum c_k \phi(2x - k)$$

with ϕ normalized by $\int \phi dx = 1$ which has the consequence $\sum c_k = 2$. Examples are the box function, $\phi = \chi_{[0,1)}$ (where χ denotes the characteristic function), corresponding to $c_0 = 1, c_1 = 1$, $c_k = 0$ for $k \geq 2$, and the D_4 function of [3] with $c_0 = \frac{1}{4}(1+\sqrt{3})$, $c_1 = \frac{1}{4}(3+\sqrt{3})$, $c_2 = \frac{1}{4}(3-\sqrt{3})$, $c_3 = \frac{1}{4}(1-\sqrt{3})$, $c_k = 0$ for $k \geq 4$.

The generating ("mother") wavelet is then obtained from ϕ by taking differences:

$$W(x) = \sum (-1)^k c_{1-k}$$

0-8194-1196-5/93/$6.00

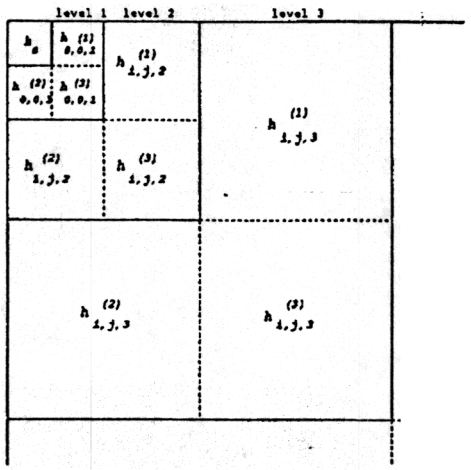


Figure 1: Arrangement of wavelet coefficients

For example, the (one-dimensional) Haar mother wavelet is $W_H(x) = \chi_{[0,\frac{1}{2})} - \chi_{[\frac{1}{2},1)}$.

For the work in this report we have used the two-dimensional Haar wavelets, for ease of computation and because we believe they preserve edges well. The generating functions in two dimensions are $\phi_0 = \phi(x)\phi(y)$, $\phi_1 = \phi(x)W_H(y)$, $\phi_2 = W_H(x)\phi(y)$, and $\phi_3 = W_H(x)W_H(y)$. The orthonormal basis is formed by taking ϕ_0 as the level $k = 0$ function and dilates and translates of ϕ_1, ϕ_2, ϕ_3 at level $k > 0$ given by

$$H^{(m)}_{i,j,k}(x,y) = 2^{k-1}\phi_m(2^{k-1}(x,y) - (i,j)), \quad 0 \le i,j < 2^{k-1},\ m = 1,2,3.$$

A given $N \times N$ image can be regarded as a function on $[0,1) \times [0,1)$ given by

$$F = \sum_{p=0,q=0}^{p=N,q=N} F_{pq}\chi_{[p/N,(p+1)/N)\times[q/N,(q+1)/N)}$$

where F_{pq} are the pixel values, $0 \le F_{pq} \le 255$. F can be rewritten as a linear combination of the wavelet basis functions:

$$F = h_0\phi_0 + \sum h^{(m)}_{ijk} H^{(m)}_{ijk}, \quad m = 1,2,3.$$

Convolution of the image with the basis functions $H^{(1)}$, $H^{(2)}$ and $H^{(3)}$ corresponds to horizontal, vertical and diagonal subband filtering. The wavelet coefficients $h_0, h^{(m)}_{i,j,k}$ can be arranged in a corresponding fashion in an $N \times N$ array, as shown in Figure 1. Thus, we will refer, for example, to the "top 8 x 8" block of wavelet coefficients, meaning the coefficients corresponding to levels $k = 0,1,2,3$.

Given the pixel values F_{pq}, there is a "fast wavelet transform" for recursively computing the coefficients at each level — see [5], for example. There is also a fast inverse transform for regaining the F_{pq} from the wavelet coefficients.

3. PATTERN AND OBJECT RECOGNITION

The theoretically best classifier, in the sense of minimizing the probability of classification error, assuming that the feature vectors are random and their distributions are given, is the Bayes classifier — see [8]. However, its computation is difficult in practice, and some of the assumptions may be unwarranted. In practice, a number of different parametric (linear, quadratic, etc.) and nonparametric (Parzen, k-nearest neighbor, etc.) classifiers are used. More recently, neural network methods have also gained popularity.

The question of how to extract an optimum set of features for a given classification problem is a difficult one. While many pattern recognition techniques are designed to minimize the error when there is a large number of samples

relative to the dimensionality of the feature or observation vectors, the situation we face is of a small number (tens to hundreds) of high-dimensional (65,536 pixels) observations. We may use the wavelet transform as a method of reducing this high dimensionality, by throwing away many of the wavelet coefficients. This approach has led to success with using wavelets for image compression, but it has to be emphasized, as is done forcefully by [8], and others, that the best features to extract for classification are not usually the same as the best ones for efficient signal representation. Indeed, our experiments reported below show that the choice of wavelet coefficients leading to the best classification results are not ones that will give good signal representation.

In their survey [13], the authors define object recognition as the task of finding and labeling parts of a two-dimensional image of a scene that correspond to objects in the scene. This involves not only pattern recognition but also some means of locating sets of pixels in the image which might be (part of) the objects of interest. In recognizing vehicles, for example, we may first locate part of an image which has some geometric feature believed indicative of a vehicle (object location), and then test the subimage to determine what type of vehicle, if any, it is (pattern recognition). There are established methods for locating geometrical shapes within images. Originally for lines, the Hough transform has been generalized to detect arbitrary shapes — see [2]. This method becomes computationally very intensive for complex shapes. Other object-specific methods based on simple geometrical properties present in the objects under consideration can be substituted. In our case, edge discontinuities in the range part of the image will be a fair detector of vehicle location. Wavelets have also been used for edge enhancement and detection — see [9] for example. In the experiments reported below we have focused mainly on the classification problem, although there are some tests comparing vehicles to scenery without vehicles.

4. RESULTS

The first set of experiments focussed on discrimination among the different types of vehicles present in the images. Only the intensity data were used. From the original images, 20 subimages of each type of vehicle (car, van, jeep) were manually selected. A typical size for a subimage was 60 x 50 pixels, although this varied considerably. Each orientation of each vehicle was typically included in two subimages, at different scales and/or in different positions in the image. These subimages were rescaled to 256 x 256 (without smoothing). The images are given in Figures 2-7 at the end of the paper.

For each image, its Haar wavelet transform was computed, also represented as a 256 x 256 array, as illustrated in Section 2. Subsets of these coefficients were used as representations of the images for input to neural network training and testing programs. For each type of vehicle, 10 of the 20 images were used for training and the other 10 for testing. Two types of neural network training algorithms were used: a standard backpropagation algorithm and the cascade correlation algorithm of [7]. For both types of network, an example with a target value of 1 was considered correctly classified by the network if the output was no less than 0.7, and an example with a target value of 0 was considered correctly classified if the network output value was no more than 0.3.

The results of backpropagation training a fully connected three-layer (one hidden layer) feedforward network to discriminate between the jeep and the van are summarized in Table 1. The networks with 16 input nodes used the top 4 x 4 block of wavelet coefficients, those with 64 input nodes used the top 8 x 8 block of wavelet coefficients, while the ones with 256 nodes used the top 16 x 16 block. For each number of input nodes, different numbers of nodes in the hidden layer were tried, in an attempt to determine the architecture best suited to this problem. For a given architecture, 10 trials were made with random (small) initial weights and thresholds. In each trial the network was trained until the mean squared error over the 20 examples in the training set was less than 0.0001, or 10000 training cycles were completed. Each line of the table presents the results of one such set of 10 trials. The results from 10 trials of training with cascade-correlation for each input size are also given. These are marked by CAS, and the minimum and maximum numbers of hidden nodes generated by the algorithm over the 10 trials is given.

Similar experiments were performed on car against van and car against jeep. Results are summarized in Tables 2 and 3 below.

A second set of experiments attempted to distinguish a given vehicle from a mixture of other vehicles. For example, a network was trained to output 1 for a car and 0 for other vehicles (van or jeep). As before, 10 trials were run on each architecture. The results are summarized in Tables 4, 5 and 6.

Finally, an experiment was run with the vehicle subimages against intensity subimages of comparable size not containing vehicles. A set of 20 non-vehicle and 20 single vehicles (mixture of car, jeep, van) was used, with a target output of 1 for a vehicle, 0 for a non-vehicle. The results are given in Table 7.

Architecture	Average cycles to train	Success (20 cases)		
		Low	Ave	High
16/2/1	5494	0	11.4	18
16/4/1	2846	0	16.2	18
16/8/1	1362	17	17.9	18
16/16/1	668	18	18.0	18
CAS 16/0-1/1	37	18	18.0	18
64/4/1	2276	0	14.1	19
64/8/1	276	17	18.2	20
64/16/1	168	16	18.8	20
64/32/1	147	17	18.8	20
64/64/1	129	18	18.7	20
CAS 64/0/1	8	17	17.9	19
256/8/1	109	12	15.1	19
256/16/1	72	9	13.4	16
256/32/1	43	11	15.0	19
256/64/1	19	10	15.0	17
256/128/1	14	11	14.6	18
CAS 256/0/1	4	12	14.2	17

Table 1: Results on jeep vs. van

Architecture	Average cycles to train	Success (20 cases)		
		Low	Ave	High
16/2/1	10000	0	12.1	17
16/4/1	10000	0	11.0	17
16/8/1	9956	0	10.2	15
16/16/1	9344	12	14.6	17
CAS 16/1-7/1	245	13	14.9	16
64/4/1	495	9	10.5	13
64/8/1	355	7	10.9	14
64/16/1	319	9	10.8	13
64/32/1	287	10	11.6	14
64/64/1	259	9	10.2	12
CAS 64/0/1	12	10	11.3	12
256/8/1	232	10	11.5	15
256/16/1	98	9	12.0	14
256/32/1	91	8	11.3	15
256/64/1	48	7	11.4	15
256/128/1	43	9	11.4	14
CAS 256/0/1	5	9	12.5	15

Table 3: Results on car vs. jeep

Architecture	Average cycles to train	Success (20 cases)		
		Low	Ave	High
16/2/1	7148	0	6.0	15
16/4/1	3104	1	11.8	15
16/8/1	1000	14	14.4	15
16/16/1	915	14	14.6	15
CAS 16/0-1/1	54	16	16.9	18
64/4/1	1401	12	15.3	16
64/8/1	455	15	15.7	16
64/16/1	316	14	15.5	17
64/32/1	222	15	15.6	17
64/64/1	183	13	15.1	17
CAS 64/0/1	13	13	14.1	17
256/8/1	184	4	12.2	17
256/16/1	102	11	13.4	16
256/32/1	73	10	12.2	14
256/64/1	30	11	12.3	14
256/128/1	20	9	12.7	15
CAS 256/0/1	4	10	12.4	16

Table 2: Results on car vs. van

Architecture	Average cycles to train	Success (30 cases)		
		Low	Ave	High
16/2/1	10000	20	20.0	20
16/4/1	10000	3	17.9	20
16/8/1	10000	12	18.9	20
16/16/1	10000	17	19.1	20
CAS 16/0-3/1	174	21	22.1	25
64/4/1	1622	16	18.5	20
64/8/1	976	18	19.2	20
64/16/1	876	17	19.2	20
64/32/1	700	17	19.3	21
64/64/1	522	16	18.8	21
CAS 64/0/1	18	19	20.8	23
256/8/1	164	16	18.5	20
256/16/1	103	13	17.5	20
256/32/1	95	15	17.8	20
256/64/1	73	15	17.3	19
256/128/1	60	16	18.3	20
CAS 256/0/1	6	17	19.9	22

Table 4: Results on car vs. other vehicles

Architecture	Average cycles to train	Success (30 cases)		
		Low	Ave	High
16/2/1	10000	0	11.9	26
16/4/1	10000	0	19.4	26
16/8/1	10000	0	21.2	25
16/16/1	10000	18	23.8	27
CAS 16/0-9/1	430	24	24.7	25
64/4/1	538	21	22.3	25
64/8/1	405	20	21.4	23
64/16/1	398	19	21.0	23
64/32/1	329	20	21.4	23
64/64/1	294	20	21.4	24
CAS 64/0/1	9	18	22.2	26
256/8/1	101	18	20.3	23
256/16/1	105	17	21.2	25
256/32/1	64	18	20.1	23
256/64/1	35	17	20.7	23
256/128/1	35	18	20.7	22
CAS 256/0/1	5	14	19.6	23

Table 5: Results on jeep vs. other vehicles

Architecture	Average cycles to train	Success (30 cases)		
		Low	Ave	High
16/2/1	9635	0	12.5	25
16/4/1	6342	0	20.3	26
16/8/1	2646	25	25.8	26
16/16/1	2441	26	26.1	27
CAS 16/0/1	22	26	26.4	27
64/4/1	3301	0	18.4	27
64/8/1	348	25	26.0	27
64/16/1	245	25	25.8	27
64/32/1	254	24	26.1	28
64/64/1	224	24	25.6	27
CAS 64/0/1	10	23	24.5	27
256/8/1	129	20	22.9	25
256/16/1	82	22	24.0	28
256/32/1	48	20	22.5	25
256/64/1	28	21	23.1	25
256/128/1	21	21	23.5	25
CAS 256/0/1	5	18	21.6	25

Table 6: Results on van vs. other vehicles

Architecture	Average cycles to train	Success (20 cases)		
		Low	Ave	High
16/2/1	9443	0	11.8	16
16/4/1	8262	9	15.0	17
16/8/1	5438	14	15.5	17
16/16/1	2570	14	14.7	17
CAS 16/1-20/1	392	10	12.6	15
64/4/1	7544	12	12.9	16
64/8/1	474	12	13.9	16
64/16/1	342	12	14.3	15
64/32/1	294	12	14.5	16
64/64/1	192	12	13.7	15
CAS 64/0-1/1	47	9	9.5	11
256/8/1	190	8	10.6	13
256/16/1	113	7	10.1	12
256/32/1	58	7	9.6	11
256/64/1	61	8	10.2	14
256/128/1	36	8	10.3	13
CAS 256/0/1	6	9	12.0	15

Table 7: Results on vehicles vs. scenery

Problem	Highest average		Highest correct	
	Architecture	Success	Architecture	Success
jeep/van	64/16/1	94.0%	64/8/1	100.0%
car/van	CAS 16/0-1/1	84.5%	CAS 16/1	90.0%
car/jeep	CAS 16/1-7/1	74.5%	16/2/1	85.0%
jeep/other	CAS 16/0-9/1	82.3%	16/16/1	90.0%
van/other	CAS 16/0/1	88.0%	64/32/1	93.3%
car/other	CAS 16/0-3/1	76.3%	CAS 16/1/1	83.3%
vehicle/non	16/8/1	77.5%	16/4/1	85.0%

Table 8: Summary of results

A summary of the various problems and the best networks found for them appears in Table 8. When more than one architecture gave the same best result, the one with fewest nodes is listed.

DISCUSSION AND CONCLUSIONS

The results demonstrate that neural networks can be trained using wavelet coefficients as inputs to achieve good generalization in discriminating vehicle types. Perhaps counterintuitively, Table 8 shows that for all of the problems considered, the best results were achieved using only the first two or three levels (4 x 4 or 8 x 8) of wavelet coefficients, while using another level of coefficients (16 x 16) gave inferior results. In general, the best networks give good generalization (83% - 100% correct). The car proved to be the hardest vehicle to distinguish, especially from the jeep, as might be expected. The results for distinguishing vehicles from scenery are not as good, on average, as for distinguishing vehicles from each other. It is difficult to judge the significance of such results in light of the small number of training and test images. As for the two training methods, cascade-correlation provided comparable generalization results to backpropagation, but with much lower training times and fewer nodes in its successful networks. The production by the cascade-correlation algorithm of networks with no hidden layer for all training sets with 64 or 256 inputs, and some of those with 16 inputs, demonstrates that these sets are linearly separable, that is, there is a separating hyperplane. This is clear in the 64 and 256 cases just from dimensionality considerations.

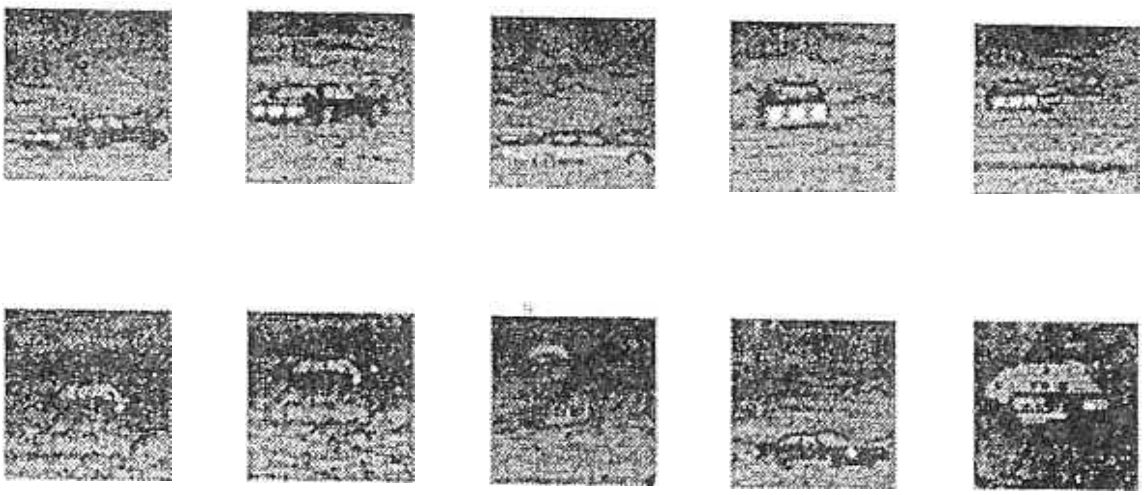
Although it takes some time to find and train the right neural network, once a suitable network has been found, it is very fast to use it, especially when the number of inputs is relatively small. All that has to be done is to present the inputs and do one feedforward pass, computing the output value.

The feasibility of using some of the coefficients of the wavelet transform of the intensity part of ladar imagery to train a neural network to distinguish vehicle types has been demonstrated. For the data in this study, the results indicate that the best discrimination is obtained by using only the coarsest two or three levels of wavelet coefficients. Conclusions must be tentative because of the limited data set. A more extensive study is warranted, using larger training and test sets, more trials for each type of architecture, connection patterns which mirror the structure of the wavelet coefficient hierarchy and other neural network training methodologies, such as adaptive logic networks [1], and divide and conquer networks [12]. A comparison with traditional pattern classification techniques should also be made.

6. ACKNOWLEDGEMENTS

This work was supported by a grant from Hercules Defense Electronic Systems, Inc. The neural network simulators used were the public domain software PlaNet [10] (formerly SunNet) from the University of Boulder and a modified version of the Cascade-Correlation program of R. Scott Crowder, adapted from the LISP implementation of Scott Fahlman.

igur Trai ng examples of



Figu Examples of for testing

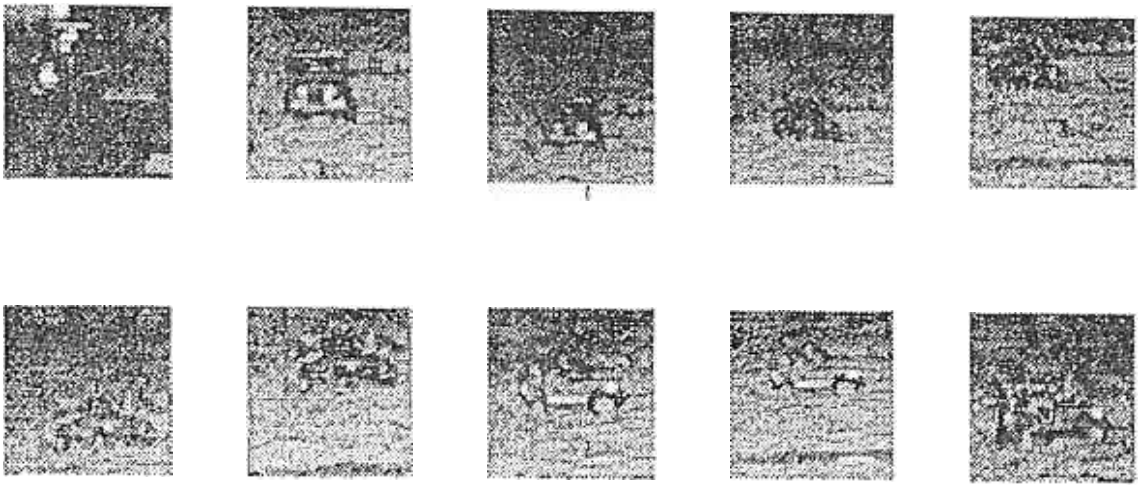


Figure 4: Training examples of jeeps

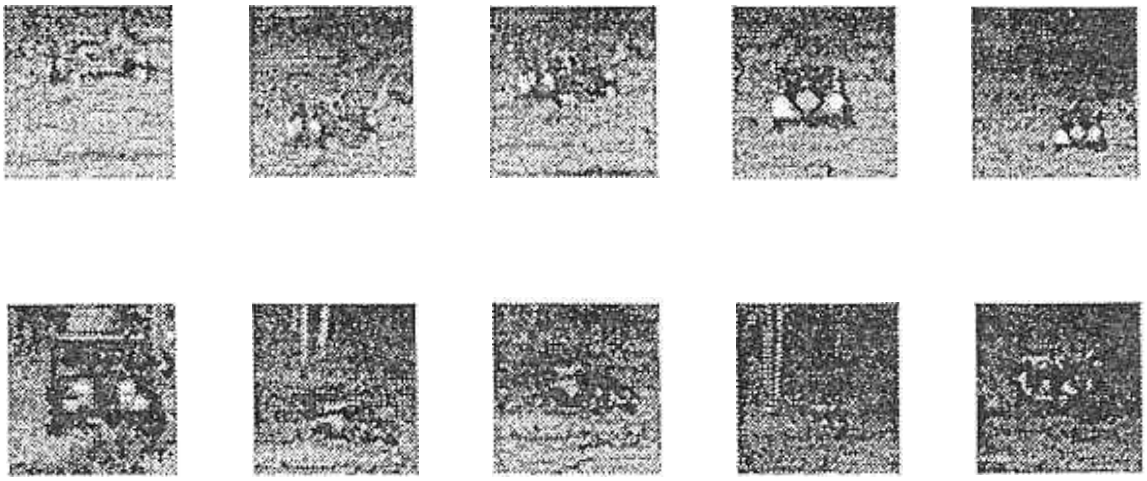


Figure 5: Examples of jeeps for testing

Figure 6: Training examples of vans

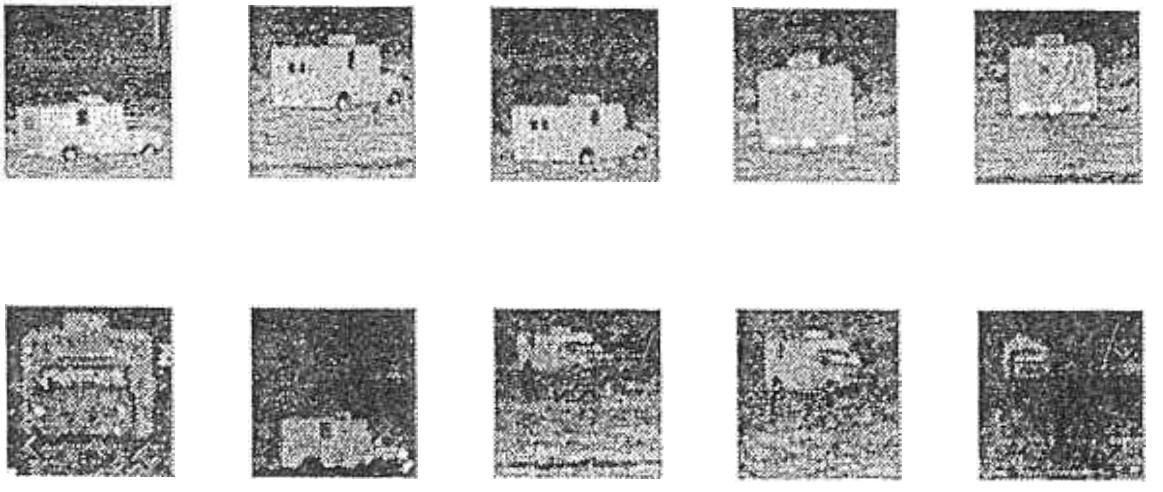


Figure 7: Examples of vans for testing

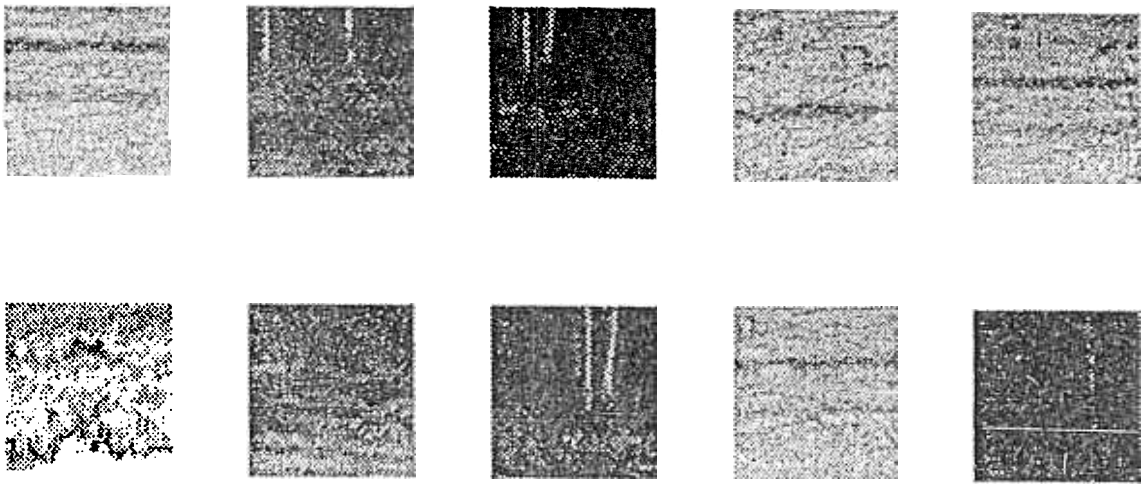


Figure 8: Training examples of scenery

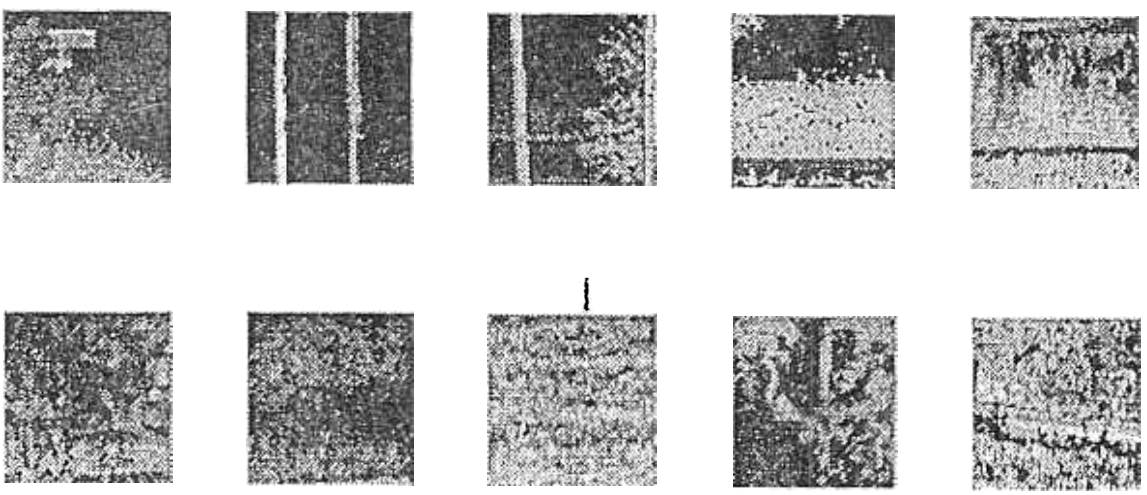


Figure 9: Examples of scenery for testing

References

[1] W. Armstrong, A. Dwelly, J. Liang, D. Lin and S. Reynolds, "Some results concerning adaptive logical networks," Technical Report, Dept. of Computing Science, U. Alberta, 23 pages, 1991.

[2] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, 13, pp. 111-122, 1981.

[3] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, 41, pp. 909-996, 1988.

[4] R. DeVore, B. Jawerth and B. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inf. Th.*, 38, pp. 719-746, 1992.

[5] R. DeVore and B. Lucier, "Wavelets," *Acta Numerica*, 1, pp. 1-61, 1992.

[6] R. DeVore and B. Lucier, "Fast wavelet techniques for near-optimal image processing," *MILCOM '92 Proceedings*, (to appear).

[7] S. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," *Carnegie Mellon Comp. Sci. Technical Report CMU-CS-90-100*, 1990.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Edition, Academic Press, New York, 1990.

[9] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAM-11, pp. 674-693, 1989.

[10] Y. Miyata, "A user's guide to PlaNet version 5.6," Technical Report, Computer Science Department, Univ. Colorado, 1991.

[11] Yoh-Han Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, Mass., Addison-Wesley, 1989.

[12] S. Romaniuk and L. Hall, "Dynamic neural networks and the use of divide and conquer," *Proc. Intl. Joint Conf. Neural Networks*, vol. 1, pp. 658-663, 1992.

[13] P. Suetens, P. Fua and A. Hanson, "Computational strategies for object recognition," *ACM Computing Surveys*, 24, pp. 5-61, 1992.