Position-Invariant Neural Network for Digital Pavement Crack Analysis

Byoung Jik Lee

Department of Computer Science, Western Illinois University, Macomb, IL 61455, USA

&

Hosin "David" Lee*

Department of Civil & Environmental Engineering and Public Policy Center, University of Iowa, Iowa City, IA 52242, USA

Abstract: *This article presents an integrated neural network-based crack imaging system to classify crack types of digital pavement images. This system includes three neural networks: (1) image-based neural network, (2) histogram-based neural network, and (3) proximity-based neural network. These three neural networks were developed to classify various crack types based on the subimages (crack tiles) rather than crack pixels in digital pavement images. These spatial neural networks were trained using artificially generated data following the Federal Highway Administration (FHWA) guidelines. The optimal architecture of each neural network was determined based on the testing results from different sets of the number of hidden units, learning coefficients, and the number of training epochs. To validate the system, actual pavement pictures taken from pavements as well as the computer-generated data were used. The proximity value is determined by computing relative distribution of crack tiles within the image. The proximity-based neural network effectively searches the patterns of various crack types in both horizontal and vertical directions while maintaining its position invariance. The final result indicates that the proximity-based neural network produced the best result with the accuracy of 95.2% despite its simplest neural network structure with the least computing requirement.*

1 INTRODUCTION

Quantification of pavement crack data is one of the most important criteria in determining optimum pavement maintenance strategies (Hudson et al., 1992). Over the years, a significant amount of effort has been spent on developing methods to objectively evaluate the condition of pavements. The simplest method is to visually inspect the pavements and evaluate them by subjective human experts. This approach, however, involves high labor costs and produces unreliable and inconsistent results. Furthermore, it exposes the inspectors to dangerous working conditions on highways.

To overcome the limitations of the subjective visual evaluation process, several attempts have been made to develop an automatic procedure. Most current systems use computer vision and image processing technologies to automate the process. However, due to the irregularities of pavement surfaces, there has been a limited success in accurately detecting cracks and classifying crack types. In addition, most systems require complex algorithm with high levels of computing power. Though many attempts have been made to automatically collect pavement crack data, better approaches are needed to evaluate these automated crack measurement systems (Luhr, 1999; Guralnick et al., 1993). The goal of this research is to develop an integrated neural network system capable

*To whom correspondence should be addressed. E-mail: *hlee@engineering.uiowa.edu.*

© 2004 *Computer-Aided Civil and Infrastructure Engineering*. Published by Blackwell Publishing, 350 Main Street, Malden, MA 02148, USA,
and 9600 Garsington Road, Oxford OX4 2DQ, UK.

of automatically determining a crack type from digital pavement images.

2 BACKGROUND

The Distress Identifications Manual for the Long-Term Pavement Performance Project (SHRP-P-338) defines the crack types for asphalt concrete pavement, which include an alligator crack, a block crack, a longitudinal crack, and a transverse crack as follows:

- A longitudinal crack appears along the highway.
- A transverse crack is a crack perpendicular to the pavement centerline caused by temperature change.
- An alligator crack is a series of interconnected cracks, which has many-sided and sharp-angled pieces.
- A block crack is a pattern of rectangular pieces of asphalt surface developed from transverse cracks due to low temperature.

Due to the rough surface and nonuniform illumination of the pavement surface image, pavement surface images have a lot of background noise. Several approaches including median filtering and neighborhood averaging are commonly applied as methods of image enhancement (Gonzalez and Woods, 1987). Image segmentation is a process used to identify objects by dividing the original image into subgroups. Thresholding is the most commonly used technique in image segmentation (Haralick and Shapiro, 1985).

A neural network consists of a number of autonomous processing elements called *neurons* or *nodes*. These *nodes* receive input signals, evaluate the computation, and produce the output. These *nodes* are highly interconnected with connection weights. A neuron has many input paths and the weighted sum of all incoming paths is combined. The neural network learns to approximate the desired function by updating its connection weights on the basis of input and output data. The neural network is recommended, especially, when it is difficult to determine the class of proper and sufficient rules in advance. Additionally, the neural network is a promising approach when a traditional computing approach is not efficient to represent a solution.

Learning in a neural network requires many good example data for training and it usually takes a significant amount of time to learn. The major advantage of neural networks is their ability to learn how to respond appropriately to the input data, which might be incomplete or noisy. These benefits are difficult to achieve using traditional computing approaches (Hertz et al., 1991). The backpropagation learning algorithm is one of the most powerful supervised learning methods in the area of neural networks. Backpropagation learning has been shown to work well on a wide variety of problems (Rumelhart et al., 1986). Training a network by backpropagation in a neural network consists of two steps. The first step is the propagation of input signals forward through the network. The second step is the adjustment of weights based on error signals propagated backward through the network. As shown in Equation (1), the performance of the system is measured by a mean square difference between the desired target outputs and the actual outputs:

$$E = \frac{1}{2}\sum_{pk}(T_{pk} - O_{pk})^2 \tag{1}$$

where T_{pk} is the desired target output, O_{pk} is the actual output, k is a particular output unit, and p is the particular training data, respectively.

The goal of the neural network is to reduce this error. To minimize the error by the gradient descent method, as shown in Equation (2), the weights are updated by an amount negatively proportional to

$$\frac{\partial E}{\partial W_{ji}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial W_{ji}} \tag{2}$$

where E is the mean square difference between the desired output and the actual output as shown in Equation (1); W_{ji} is the connection weight from node i to node j; and O_j is the actual output of node j, respectively.

Recently, Owusu-Ababio (1998) looked at the use of an artificial neural network with backpropagation in predicting asphalt pavement cracks. Inputs include the surface thickness of the pavement, surface age of the pavement, and the traffic level. The system is composed of five hidden nodes and one output node, which is expressed in the cracking length per station. Forty-five samples were used for training and 15 samples were used for testing. The predicted crack types were limited only to longitudinal and transverse cracking and the results show a relatively large standard error.

Roberts and Attoh-Okine (1998) developed a neural network to predict the roughness of pavement with 105 samples. The system has ten inputs that include RT (rutting code), four FCs (fatigue cracking codes), three Ts (transverse cracking codes), BC (block cracking code), and EAL (equivalent axle loads). There is one output node, which is expressed in IRI (international roughness index). The correlation coefficient of the test data was reported as 0.74. The size of training samples and testing samples was 75 and 30, respectively. The model generally overpredicted the values of IRI below 125 and underpredicted the values of IRI above 125.

Senouci and Eldin (1995) trained a neural network with backpropagation learning to rate the condition of roadway sections. The system is composed of twenty input nodes containing rutting, bleeding, three severity levels of alligator crack, three severity levels of longitudinal crack, three severity levels of transverse crack, three severity levels of block crack, three severity levels of patching/pothole, and three severity levels of raveling. This system had one output node that represents the condition index (0.1 to 0.5 scale) of the pavement, with 0.1 indicating very good condition and 0.5 indicating very poor condition. The size of the training samples was 744. The neural network performed reasonably well compared to the experts' condition ratings.

Kaseko and Ritchie (1993) applied a neural network with backpropagation to classify the pavement crack type, crack severity, and crack extent of digitized video images. From 32 × 29 pixel pavement images, they computed 13 variables to use as input variables for the neural network. The 13 input variables included the number of uninterrupted sequences of object pixels in four directions, the variance of the number of object pixels per line in four directions, the projected crack length in four directions relative to width or length of the image, and the relative number of object pixels as a proportion of the total number of pixels. From these inputs, the neural network classified the subimage into five types: no crack, transverse crack, longitudinal crack, diagonal crack, and combination crack. The length of each crack type was then calculated for each row and column to produce an input matrix that was injected to the neural network to classify the entire pavement image. Relatively smooth asphalt-concrete pavement images have been used for this research resulting in high accuracy. Using quite complex computation procedures to extract some features from the pixels, the system correctly classified 97% of longitudinal cracks and 100% of transverse cracks, but the classification rate for alligator cracks and block cracks was degraded to 85.7%.

Cheng et al. (2001) utilized the neural network to determine the optimum threshold value for image segmentation. The inputs of the neural network are the mean and the standard deviation of the image. The chosen number of hidden nodes is 15, and the system was tested using 30 images. Test results indicate the close agreement between the desired threshold values and the threshold values computed by the neural network.

3 SYSTEM ARCHITECTURE

This system was developed to determine crack types using a spatial neural network concept. As shown in Figure 1, system architecture consists of four processes: (1) artificial training data generation, (2) training different neural network models, (3) crack tile generation, and (4) crack type classification.

The main limitation of pixel-based neural networks is its processing time because it deals with a pavement image that typically covers a 5 × 7 feet area with 288,000 (480 × 600) pixels. When we inject each pixel into an input unit of the neural network, we need 381,024 input units and a large number of hidden units. A typical neural network is fully connected between adjacent layers, and, therefore, a pixel-based neural network would require a very high level of computation in both training and testing. In addition, when there is a significant amount of noise in the image, a pixel-based approach could produce unreliable results.

The proposed neural network models in this article determine a crack type based on subimages of pavement rather than crack pixels. A pavement image is divided into 180 subimages called "tiles" and each tile is composed of 1,600 pixels (40 × 40). This tile-based computation significantly reduces computational complexity over pixel-based computation. As a result, it is possible to train the neural network in a reasonable period of time and quickly determine the crack type. It is less affected by background noises because a few noise pixels alone would not be sufficient for a tile to be classified as a crack tile.

3.1 Data generation

Pavement images are needed to train the artificial neural network and validate the system. However, it is difficult to obtain a set of real pavement images that would include a full range of crack types, extent, and severity. Therefore, a computer program was developed to generate an artificial set of training data following the FHWA guidelines (FHWA, 1989, 1991) on crack definition. Four hundred and fifty images, 90 images for each crack type, were generated.

Longitudinal cracks are allowed to be skewed from the centerline of the pavement by up to 30 degrees. Transverse cracks are angled from the centerline of the pavement from 60 to 90 degrees. An alligator crack forms small interconnected rectangles with a side length of from 3 to 4 tiles (1 tile = 0.4 feet). Block cracks were generated with a side length between 5 and 10 tiles.

3.2 Neural network training

The neural networks were trained with the computer-generated data using the backpropagation procedure (Rumelhart et al., 1986). All of the neural networks were trained with different sets of learning rates, and with various numbers of hidden nodes, and training cycles. To measure the performance of neural networks, we adopted two kinds of errors, MSE (mean square error) shown in Equation (3) and IRE(the number of incorrectly recognized images) shown in Equation (4).

$$\mathrm{MSE} = \frac{1}{2}\sum_{p,k}(T_{p,k} - O_{p,k})^2 \tag{3}$$

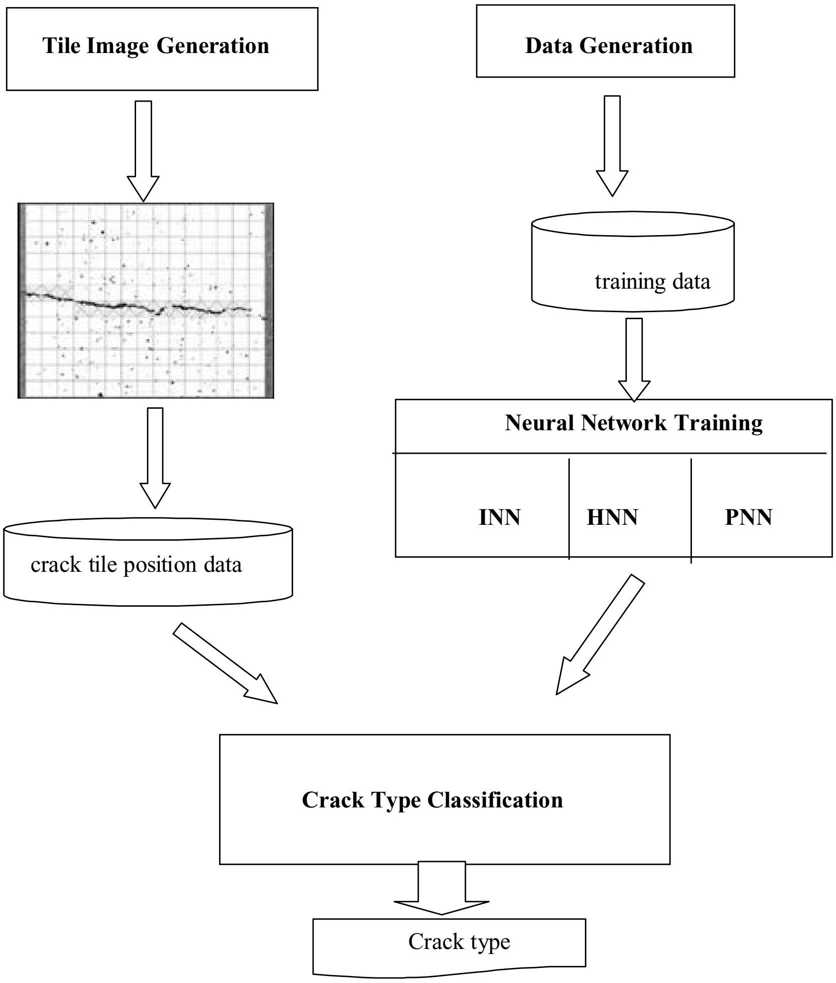


Fig. 1. System architecture of the position invariant neural network.

where p is pth sample data; k is kth output units; $T_{p,k}$ is the target output in pth data and kth output unit, and $O_{p,k}$ is the actual output in pth data and kth output unit, respectively.

$$\mathrm{IRE} = \sum_t \mathrm{diff}(T_t, A_t) \tag{4}$$

where t is the tth data; T_t is the target crack type for the tth data; and A_t is the actual crack type produced by the system for the tth data, respectively.

Figure 2 illustrates the flow chart of the training procedure. Artificial pavement images are read from the external file. The actual outputs of the neural network are computed through the feedforward procedure, and connection weights are adjusted by the backpropagation procedure. After the training is done, the performance on real pavement images and other artificial images is tested using the crack-type classification process.

3.3 Tile image generation

An original image is filtered and divided into 12 × 15 partitions called "tiles." Each partition has 1,600 pixels (40 rows and 40 columns). For the image enhancement, a median filtering was chosen to reduce the noise in the pavement surface image. A median filtering computes the median value of the determined filtering size by sorting the pixels in ascending or descending order. A different thresholding value is then determined using a regression equation, which is developed as a function of the average brightness level of each tile. By applying this threshold value to each tile, each pixel is converted into a binary value. Whether a tile is a crack tile or not is determined based on the percentage of crack pixels in a tile. If the percentage of crack pixels in a tile is greater than the predefined threshold, the tile is considered as a crack tile (Jitprasithsiri et al., 1996).

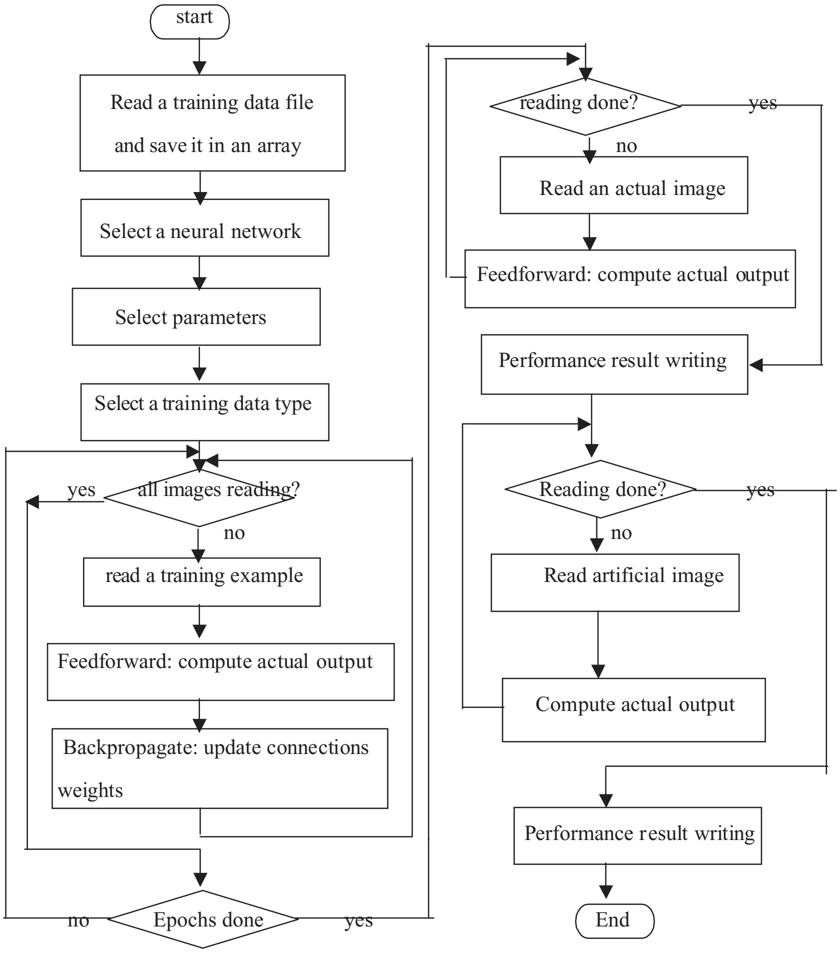

start
Read a training data file and save it in an array
Select a neural network
Select parameters
Select a training data type
all images reading?
yes
no
read a training example
Feedforward: compute actual output
Backpropagate: update connections weights
Epochs done
no
yes
reading done?
yes
no
Read an actual image
Feedforward: compute actual output
Performance result writing
Reading done?
yes
no
Read artificial image
Compute actual output
Performance result writing
End

Fig. 2. Flowchart of spatial neural network training.

3.4 Crack type classification

The crack type of actual pavement images is determined using a specific neural network trained using artificial images. There are two groups of inputs in this process: the saved connection weights of the neural network and the actual pavement tile images. The connection weights of the neural network were obtained from the training process. These weights are loaded into the system to instantiate the neural network. The tile generation process reads an actual pavement image and produces a crack tile (subimage) image. This crack tile image is then converted into a two-dimensional binary image with crack tiles and no crack tiles. The two-dimensional binary image is injected into the process to classify its crack type using the chosen neural network. Figure 3 shows the flowchart of the crack-type classification process.

4 THREE SPATIAL NEURAL NETWORK MODELS

Three proposed neural networks are named image-based neural network, histogram-based neural network, and proximity-based neural network. Each of these neural networks is a three-layered feedforward neural network. Figure 4 illustrates the basic framework of the image-based neural network model. Each neural network model has a different number of input nodes and different input values. The output layer includes five nodes that represent (1) alligator crack, (2) block crack, (3) longitudinal crack, (4) transverse crack, and (5) no crack.

Several different neural network architectures were explored with different sets of hidden nodes (30, 60, 90, 120, and 150), learning coefficients (0.1, 0.05, 0.01, and 0.001), and a number of training epochs (500, 1,000,

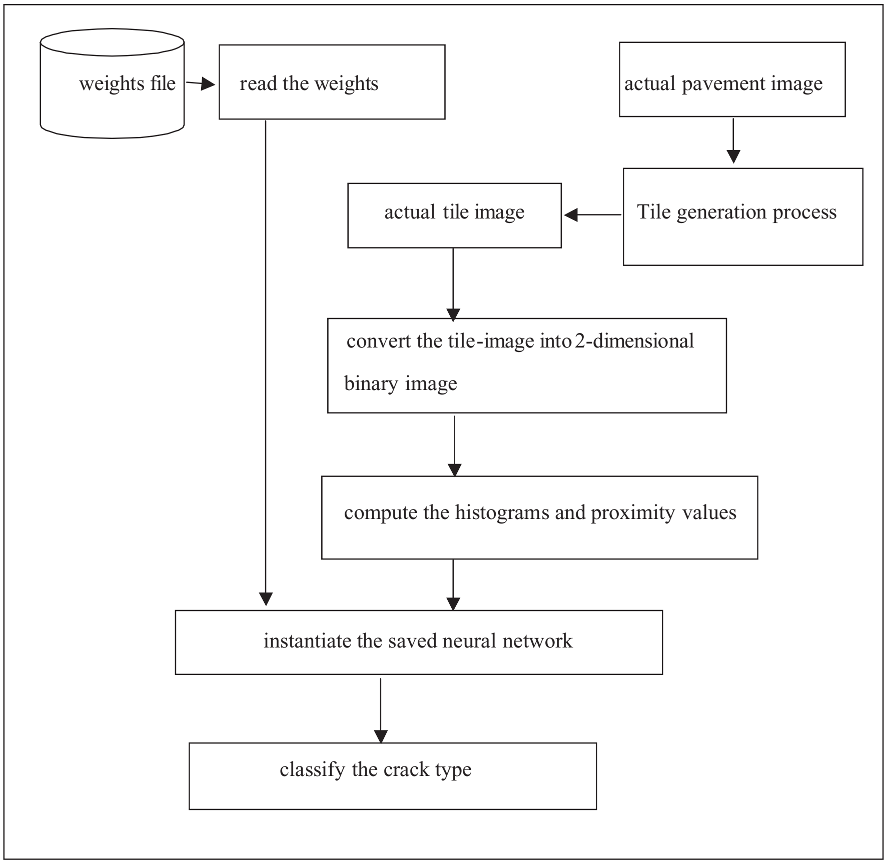

weights file
read the weights
actual pavement image
Tile generation process
actual tile image
convert the tile-image into 2-dimensional binary image
compute the histograms and proximity values
instantiate the saved neural network
classify the crack type

Fig. 3. A flowchart of crack-type classification.

1,500, and 2,500) to find an optimal architecture. An artificially generated data set was used to find an optimal structure for each neural network model. One hundred and twenty-four actual pavement images and 150 artificial images were used to test these neural network models.

4.1 Image-based neural network (INN)

This image-based neural network is similar to the pixel-based neural network except that it utilizes crack tiles. If the tile in the *i*th row and the *j*th column is determined to be a crack tile, then *crack_image[i, j]* is designated as 1. Otherwise, *crack_image[i, j]* is designated as 0. For INN, the matrix of crack tiles, *crack_image[i, j]*, is injected into the image input layer of the neural network. The *crack_image[i, j]* array has 150 binary values, which are represented by the number of rows (12) and the number of columns (15) as shown in Figure 4. A 12 × 15 matrix of a transverse crack image would be injected into the input layer of the INN as shown in Figure 5.

Inputs are read sequentially from the upper left corner tile (1st) to the bottom right corner tile (180th). However, the order of nodes in a layer is not significant in fully connected neural network. If the tile in the image is a crack, then the value for the corresponding node is 1. Otherwise, the value for that node is 0. Table 1 shows the comparison among different architectures for MSE and IRE for both training and testing data. It also lists accuracy rates with the learning coefficient of 0.01 and the number of epochs being 1,500 and 2,500. In all cases, INN performed well for training data; however, it performed poorly for test images. Table 2 shows the set of parameters that performed the best for each learning rate. As shown in Table 2, 60 hidden units performed the best with every learning coefficient. The 1,500 epochs with a learning coefficient of 0.1 or 0.01 were selected because they performed the best (70.1%) in the test.

4.2 Histogram-based neural network (HNN)

From the image of crack tiles, two kinds of histograms are computed: a vertical histogram and a horizontal histogram. The histograms in HNN are defined as the distributions of the number of cracked tiles in

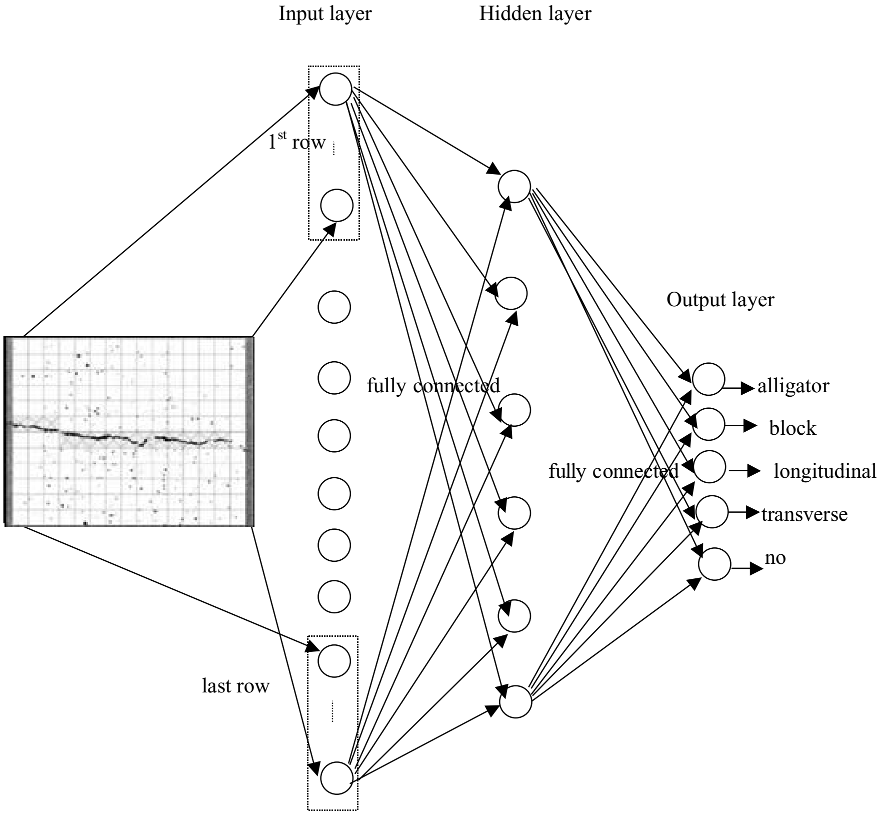
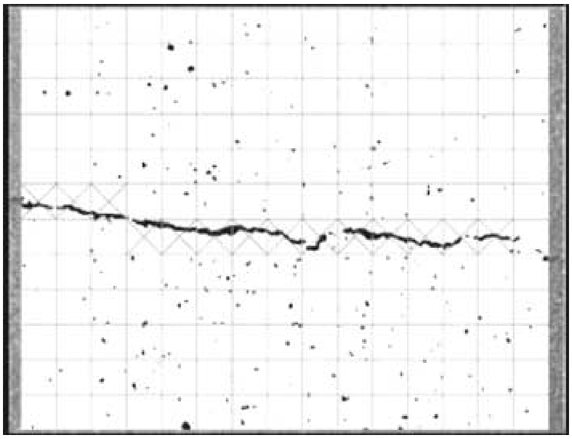


Fig. 4. An architecture of image-based neural network.

each column and row. Histograms show a clear pattern of a crack. If a crack is developed in a longitudinal direction, there is a clear peak in the vertical histogram. If a crack is developed in a transverse direction, then there is a clear peak in the horizontal histogram. If a crack is an alligator crack, the peaks can be found in both vertical and horizontal directions. For a block crack, peaks can also be found in both histograms but with a smaller magnitude than those of an alligator crack.

Table 1
A comparison of mean square error, incorrect recognition error, and accuracy rate in INN for data set with a learning coefficient (η) 0.01 and epochs of 1,500 and 2,500

Learning coefficient	*Epochs*	*Input-hidden-output layer*	*Training 300 artificial images*			*Testing Actual pavement 124 images*		
			MSE^{train}	*IRE^{train}*	*Accuracy*	*MSE^{test}*	*IRE^{test}*	*Accuracy (%)*
0.01	1,500	180-30-5	6.52	10	96.7%	31.29	39	68.5
		180-60-5	5.73	9	97	30.44	37	70.2
		180-90-5	6.94	12	96	32.20	45	63.7
		180-120-5	6.37	10	96.7	34.14	43	65.3
		180-150-3	6.42	11	96.3	35.12	43	65.3
	2,500	180-30-5	5.61	10	96.7	31.06	39	68.5
		180-60-5	6.41	12	96	32.18	45	63.7
		180-90-5	4.98	9	97	30.06	37	70.2
		180-120-5	4.92	9	97	34.10	44	64.5
		180-150-3	5.89	11	96	35.37	44	64.5



(a)

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

Fig. 5. A transverse crack (a) cracked tiles; (b) an input matrix to INN.

To determine the vertical histogram, the number of cracked tiles is recorded for each row using Equation (5). For example, the vertical histogram of Figure 6 is (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0):

$$\left(Hv[i] = \sum_{j=1}^{Nr} crack\_image[j, i]\right), \quad \forall i, 1 \ldots Nc \tag{5}$$

where Hv is vertical histogram; Nr is number of rows; and Nc is number of columns, respectively.

For a horizontal histogram, the number of cracked tiles is recorded for each column using Equation (6). For example, the horizontal histogram of Figure 6 is (0, 0, 0, 0, 0, 3, 11, 0, 0, 0, 0, 0):

$$\left(Hh[i] = \sum_{j=1}^{Nc} crack\_image[i, j]\right), \quad \forall i, 1 \ldots Nr \tag{6}$$

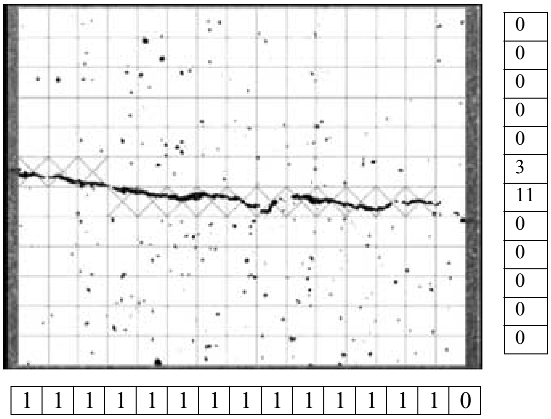
Table 2
Summary of the best INN performance in each learning coefficient

Learning coefficient	*Hidden units*	*Epochs*	*Test data*	
			*IRE^test^*	*Accuracy (%)*
0.1	60	1,500, 2,500	37	70.2
0.05	60	1,000, 1,500, 2,500	39	68.5
	90	2,500	39	68.5
0.01	60	1,500	37	70.2
	90	2,500	37	70.2
0.001	60	2,500	47	62.1

where Hh is the horizontal histogram; Nr is the number of rows; and Nc is the number of columns, respectively.

These two histograms are injected into the HNN input layer, which has a total of 27 input nodes. Among the 30 input nodes, the first 12 nodes represent the horizontal histogram. The remaining 15 nodes represent the vertical histogram. Because the histograms in the pavement image are injected into the neural network instead of the pure image, the histogram-based neural network involves less computation than the image-input neural network.

Table 3 shows the comparison among different HNN architectures for MSE and IRE for both training and testing data. It also lists accuracy rates with the learning coefficient of 0.01 and the epoch numbers of 1,500 and 2,500. As shown in Table 4, a learning coefficient of 0.01 with 60 and 90 hidden units with an accuracy of 76.6% was selected as an optimal structure. Thirty hidden nodes were not enough to perform the nonlinear



(a) (b)

Fig. 6. A transverse crack (a) a vertical histogram; (b) a horizontal histogram.

Table 3
A comparison of mean square error, incorrect recognition error, and accuracy rate in HNN for data set with a learning coefficient (η) 0.01 and epochs of 1,500 and 2,500

Learning coefficient	*Epochs*	*Input-hidden-output layer*	*Training 300 artificial images*			*Testing Actual pavement 124 images*		
			MSE^{train}	IRE^{train}	*Accuracy (%)*	MSE^{test}	IRE^{test}	*Accuracy (%)*
0.01	1,500	27-30-5	2.31	3	99	23.46	32	74.2
		27-60-5	2.03	3	99	22.82	31	75
		27-90-5	2.36	4	98.7	22.92	29	76.6
		27-120-5	2.30	4	98.7	24.15	31	75
		27-150-3	1.27	2	99.3	23.93	31	75
	2,500	27-30-5	1.92	3	99	23.49	33	73.3
		27-60-5	1.29	2	99.3	22.33	29	76.6
		27-90-5	1.72	3	99	23.09	29	76.6
		27-120-5	2.16	4	98.7	24.27	31	75
		27-150-3	1.15	2	99.3	23.92	31	75

mapping of the input data while more than 90 hidden nodes did not improve the performance.

4.3 Proximity-based neural network (PNN)

For the proximity-based neural network, the proximity of the image was computed by accumulating the differences between adjacent histogram values. The proximity-based neural network distinguishes the crack type by finding the unique pattern of uniformity of an image by using these proximity values.

The vertical proximity is computed by accumulating the differences between the numbers of cracked tiles for adjacent image columns using Equation (7). In the example shown in Figure 7, the vertical proximity is 1 by adding 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, and 1. This low value indicates that there is little difference between any of the columns for this image:

$$Pv = \sum_{i=1}^{Nc-1} |Hv[i+1] - Hv[i]| \tag{7}$$

where Pv is vertical proximity; Hv is vertical histogram; and Nc is number of columns, respectively.

Horizontal proximity is computed by accumulating the differences between the numbers of cracked tiles for adjacent image rows using Equation (8). As shown in Figure 7, the horizontal proximity is 24 by adding 0, 0, 0, 0, 3, 8, 11, 0, 0, 0, and 0:

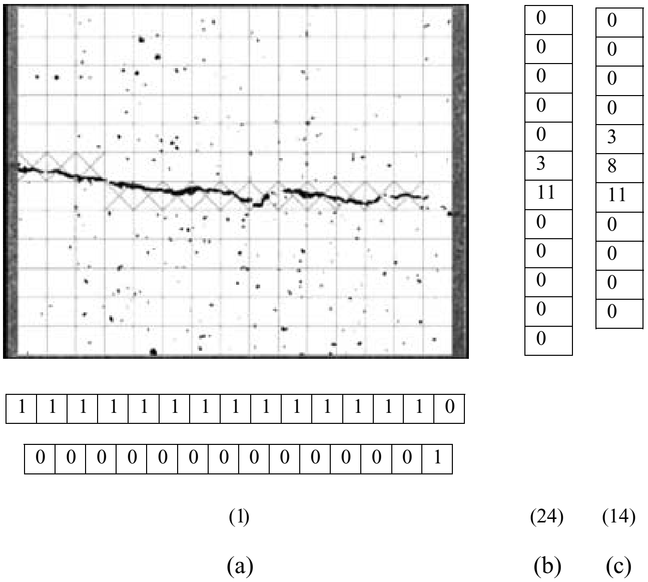
$$Ph = \sum_{i=1}^{Nr-1} |Hh[i+1] - Hh[i]| \tag{8}$$

where Ph is horizontal proximity, Hh is horizontal histogram, and Nr is number of rows.

These two values, *horizontal proximity* and *vertical proximity*, are injected into the proximity input layer. Like the previous two networks, PNN is a three-layered feedforward neural network. The same sets of hidden nodes, learning coefficients, and number of epochs such as INN and HNN were tested. Table 5 shows the comparisons of different PNN architectures for MSE and IRE for both training and testing data. It also lists accuracy rates with the learning coefficient of 0.01 and epoch

Table 4
Summary of the best HNN performance in each learning coefficient

Learning coefficient	*Hidden units*	*Epochs*	*Test data*	
			IRE^{test}	*Accuracy (%)*
0.1	120	500, 1000, 1500, 2500	30	75.8
0.05	60, 90, 120, 150	500, 1000, 1500, 2500	30	75.8
0.01	60	2500	29	76.6
	90	500, 1000, 1500, 2500	29	76.6
0.001	60	2500	30	75.8



1 1 1 1 1 1 1 1 1 1 1 1 1 1 0

0 0 0 0 0 0 0 0 0 0 0 0 0 1

0 0 0 0 0 3 11 0 0 0 0 0

0 0 0 0 3 8 11 0 0 0 0

Fig. 7. A transverse crack (a) a vertical proximity 1; (b) a horizontal proximity value 24; and (c) 14 crack tiles.

numbers of 1,500 and 2,500. As shown in Table 6, PNN has 3 input nodes, 60 hidden nodes, and 5 output nodes. Two nodes in the input layer represent horizontal proximity and vertical proximity. The third node in the input layer represents the number of cracked tiles in the image.

5 ANALYSIS RESULT

Two kinds of test data were used. One set of data is the 124 actual picture images taken from the road. The other set of data is 150 computer-generated data following the FHWA guideline. The main criteria of system performance are computation time and the accuracy of the system. Two kinds of criteria were considered in assessing computation time. One is the time for training the neural network and the other is for processing real pavement images.

Table 6
Summary of the best PNN performance in each learning coefficient

Learning coefficient	*Hidden units*	*Epochs*	*Test data*	
			IRE^{test}	*Accuracy*
0.1	120, 150	1000	15	87.9%
0.05	60	2500	14	88.7
	150	500	14	88.7
0.01	60	500	6	95.2
	90	2500	6	95.2
0.001	60	500	7	94.3

5.1 Computation time

In backpropagation algorithm, there are two computational phases; (1) propagation of input signal from input layer to output layer through the hidden layer and (2) backpropagation of error signal from output layer to input layer through the hidden layer. There is no standard rule to decide the optimal number of hidden units. However, there is a rule of thumb used by some researchers that the number of hidden units should be (*number of input units* + *number of output units*) × 2/3. The number of hidden units is usually dependent on the

Table 5
A comparison of mean square error, incorrect recognition error, and accuracy rate in PNN for data set with a learning coefficient (η) 0.01 and epochs of 1,500 and 2,500

Learning coefficient	*Epochs*	*Input-hidden-output layer*	*Training 300 artificial images*			*Testing Actual pavement 124 images*		
			MSE^{train}	*IRE^{train}*	*Accuracy (%)*	*MSE^{test}*	*IRE^{test}*	*Accuracy (%)*
0.01	1,500	3-30-5	18.70	35	88.3	8.54	13	89.5
		3-60-5	10.92	13	95.7	7.55	7	94.3
		3-90-5	11.93	11	96.3	7.67	8	93.5
		3-120-5	9.61	13	95.7	7.15	10	91.9
		3-150-3	7.21	9	97.0	7.28	8	93.5
	2,500	3-30-5	18.60	35	88.3	8.73	12	90.3
		3-60-5	7.49	9	97.0	6.60	8	93.5
		3-90-5	8.63	10	96.7	7.04	6	95.2
		3-120-5	9.72	12	96.0	7.02	8	93.5
		3-150-3	6.81	8	97.3	8.28	7	94.4

Table 7
A training time of three tile-based neural networks for data set

Neural networks	*Number of input units*	*Number of hidden units*	*Training time with artificial data (sec)*
INN	180	180	3644
HNN	27	30	80
PNN	3	3	34

number of input units and the number of training data. When the number of input units is n, we began by assuming that the number of hidden units is n. Because there is complete connection between the input layer and hidden layer, the number of multiplications between them is n^2. Because the number of output units is 5, the number of multiplication between the hidden layer and output layer is $5n$. Because two phases are necessary for the backpropagation procedure, the number of multiplication becomes $2(n^2 + 5n)$. As a result, the backpropagation algorithm is $O(n^2)$.

Table 7 shows the training time of each neural network for 300 artificial images of a data set on a computer equipped with a Pentium III processor with 750 MHz. The number of epochs is 1,500. The input sizes are 180, 30, and 3, respectively. As expected and as is shown in Table 7, when the number of hidden units is equal to the number of input units, training time with PNN is significantly reduced to 1% and 42% compared to INN and HNN, respectively. Training time with proximity input layer took the least time. When a testing image is fed into the NeuralCrack system, all three neural network models took about 1.5 seconds on a Pentium III-750 MHz machine to determine a crack type for a 504 × 756 pixel pavement image.

5.2 Training and testing accuracy

Three hundred artificial images were used for training. One hundred and twenty-four actual pavement images and 150 artificial images were then used for testing. As shown in Table 8, all neural networks achieved a high training accuracy of 95% or higher. In the testing, INN had an accuracy of 76% for artificial images and 70.2% for actual pavement images. HNN achieved an accuracy of 91.3% for artificial images; however, the performance for the actual images was 75%. PNN achieved the highest accuracy of 93.3% for artificial images and 95.2% for actual pavement images.

Tables 9 and 10 show the detailed test results of INN for 150 artificial data and 124 actual test images, respectively. For all crack types, the accuracy of INN significantly deteriorated when it was tested against actual pavement images except for longitudinal cracks and no cracks.

As shown in Table 11, based on the artificial data set, HNN performed well for all crack types except block cracks. For actual test images, the accuracy of HNN significantly deteriorated for transverse cracks, alligator cracks, and block cracks (Table 12).

As shown in Table 13, based on the artificial data set, PNN produced 100% accuracy for all crack types except block cracks. As shown in Table 14, based on actual pavement images, PNN also performed well achieving accuracies of 88%, 90%, 98%, and 100% for alligator and block cracks, transverse cracks, longitudinal cracks, and no cracks, respectively.

6 SUMMARY AND CONCLUSION

This article presents three spatial neural networks to classify crack types of digital pavement images: (1) image-based neural network (INN), (2) histogram-based neural network (HNN), and (3) proximity-based neural network (PNN). These neural network models utilize crack tiles instead of crack pixels. For training, 300 artificial images were generated following FHWA guidelines. One hundred and twenty-four actual pavement images and 150 artificial images were used as testing data. To find the optimal architecture of each neural network, different sets of the number of hidden units (30, 60, 90, 120, and 150), learning coefficients (0.1, 0.05, 0.01, 0.001), and training epochs (500, 1,000, 1,500, and 2,500) were tested.

Table 8
Three types of neural networks and their performance

Neural network	*Train data (artificial data: 300)*		*Test data (artificial data: 150)*		*Test data (actual images: 124)*	
Image-based NN	291	97%	114	76%	87	70.2%
Histogram-based NN	297	99%	137	91.3%	93	75%
Proximity-based NN	287	95.7%	140	93.3%	118	95.2%

Table 9
Performance of INN for 150 artificial data

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	30	0	0	0	0	0	100
Block crack	7	22	0	1	0	8	73
Longitudinal crack	0	0	25	4	1	5	83.3
Transverse crack	0	4	11	15	0	15	50
No crack	0	0	6	2	22	8	73.3
Overestimated	7	4	17	7	1	*Ed = 36*	*76*

Table 10
Performance of INN for 124 actual pavement images

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	6	4	3	3	0	10	38
Block crack	1	2	3	2	0	6	33
Longitudinal crack	3	0	38	7	1	11	78
Transverse crack	0	1	8	12	1	10	55
No crack	0	0	0	0	29	0	100
Overestimated	4	5	14	12	2	*Ed = 37*	*70*

Table 11
Performance of HNN for 150 artificial data group

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	30	0	0	0	0	0	100
Block crack	8	18	1	3	0	12	60
Longitudinal crack	0	0	30	0	0	0	100
Transverse crack	0	1	0	29	0	1	97
No crack	0	0	0	0	30	0	100
Overestimated	8	0	0	5	0	*Ed = 7*	*91*

Table 12
Performance of HNN for 124 actual pavement data

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	2	3	2	9	0	14	13
Block crack	1	0	2	5	0	8	0
Longitudinal crack	1	0	48	0	0	1	98
Transverse crack	0	2	6	14	0	8	64
No crack	0	0	0	0	29	0	100
Overestimated	2	5	10	14	0	*Ed = 31*	*75*

Table 13
Performance of PNN for 150 artificial images

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	30	0	0	0	0	0	100
Block crack	10	20	0	0	0	10	67
Longitudinal crack	0	0	30	0	0	0	100
Transverse crack	0	0	0	30	0	0	100
No crack	0	0	0	0	30	0	100
Overestimated	10	0	0	0	0	*Ed = 10*	*93.3*

Table 14
Performance of PNN for 124 actual pavement data

	System classification						
Target classification	*Alligator crack*	*Block crack*	*Longitudinal crack*	*Transverse crack*	*No crack*	*Under-estimated*	*Accuracy (%)*
Alligator crack	14	0	1	1	0	2	88
Block crack	1	7	0	0	0	1	88
Longitudinal crack	1	0	48	0	0	1	98
Transverse crack	2	0	0	20	0	2	90
No crack	0	0	0	0	29	0	100
Overestimated	4	0	1	1	0	*Ed = 6*	*95.2*

For all cases, 60 hidden units produced the best results along with a 0.01 learning coefficient. For the number of training epoch, 1,500 was selected as a reasonable number because all three neural networks produced the most accurate results within 1,500 epochs.

INN utilizes a whole tile image as an input resulting in 180 input units. When the identical crack types appear in different positions, INN could not successfully recognize these pavement images as the identical one. Furthermore, large input units required a heavy computational demand in the training of INN. HNN utilizes the vertical and horizontal histograms of the crack tile image. HNN requires 27 input units (15 for the vertical histograms and 12 for horizontal histograms). HNN was also inconsistent in identifying the identical crack types at different positions within the image.

PNN utilizes the uniformity patterns of each crack type. PNN computes the values of vertical and horizontal proximities by taking the differences between adjacent histograms. PNN is very robust in identifying cracks appearing in different parts of the image. This produced a high level of accuracy and also reduced an amount of training data and computation time. Furthermore, PNN works well on any digital pavement images irrespective of various crack locations, lengths, and thickness. The PNN produced the best result with the accuracy of 95.2% despite its simplest neural network structure with the least computing requirement. The PNN effectively searches the patterns of various crack types in both horizontal and vertical directions while maintaining its position-invariance.

REFERENCES

Cheng, H. D., Jiang, X. H. & Glazier, C. (2001), Novel approach to pavement cracking detection based on neural network, *Transportation Research Board*, No. 1764, 119–27.

FHWA (1989), Federal-Aid Highway Program Manual. Vol. 6, Federal Highway Administration, U.S. Department of Transportation, Washington DC.

FHWA (1991), An Advance Course in Pavement Management Systems. Federal Highway Administration, U.S. Department of Transportation, Washington DC.

Gonzalez, R. C. & Woods, R. (1987), *Digital Image Processing* 2nd edn., Addison-Wesley Publishing Co. Inc., Menlo Park, CA.

Guralnick, S. A., Sun, E. S. & Smith, C. (1993), Automating inspection of highway pavement surface, *Journal of Transportation Engineering*, **119**(1), 1–12.

Haralick, R. M. & Shapiro, L. G. (1985), Image segmentation techniques, *Computer Vision, Graphics, and Image Processing*, **29**, 100–32.

Hertz, J., Krogh, A. & Palmer, R. (1991), *Introduction to the Theory of Neural Computation*. vol. 1, Santa Fe Institute, Addison-Wesley, Boston, MA.

Huang, Y. H. (1993), *Pavement Analysis and Design*. Prentice Hall, Englewood Cliffs, NJ.

Hudson, S. W., Hudson, W. R. & Carmichael, R. F. (1992), Minimum requirements for standard pavement management systems, *Pavement Management Implementation*, STP 1121, ASTM, 19-31, Philadelphia.

Jitprasithsiri, S., Lee, H. & Sorcic, R. G. (1996), Development of digital image processing algorithm to compute a unified crack index for Salt Lake City, *Transportation Research Record*, No. 1526, 142–8.

Kaseko, M. S. & Ritchie, S. G. (1993), A neural network-based methodology for pavement crack detection and classification, *Transportation Research-C*, **1**(4), 75–291.

Luhr, D. R. (1999), A proposed methodology to quantify and verify automated crack survey measurements, *Presented at the 78th Annual TRB Meeting*, Washington, DC.

Owusu-Ababio, S. (1998), Effect of neural network topology on flexible pavement cracking prediction, *Computer-Aided Civil and Infrastructure Engineering*, **13**(5), 349–55.

Roberts, C. A. & Attoh-Okine, N. O. (1998), A comparative analysis of two artificial neural networks using pavement performance prediction, *Computer Aided Civil and Infrastructure Engineering*, **13**(5), 339–48.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), Learning internal representation by error propagation, *Parallel Distributed Processing*, **1**, 318–62.

Senouci, A. B. & Eldin, N. N. (1995), A pavement condition-rating model using back-propagation neural networks, *Microcomputers in Civil Engineering*, **10**, 433–441.

SHRP-P-338, Distress Identification Manual for the Long-Term Pavement Performance Project, Strategic Highway Research Program, National Research Council, Washington DC.