Paper No. 01-3089

Novel Approach to Pavement Cracking Detection Based on Neural Network

H. D. Cheng, Jingli Wang, Y. G. Hu, C. Glazier, X. J. Shi, and X. W. Chen

The collection of pavement surface condition data is usually done by conventional visual and manual approaches, which are very costly, time-consuming, dangerous, labor-intensive, and subjective. These approaches have high degrees of variability, are unable to provide meaningful quantitative information, and almost always lead to inconsistencies in cracking details over space and across evaluations. A novel pavement crack detection approach based on neural network and computer vision, pattern recognition, and image-processing techniques is proposed. The thresholding approach is used to separate crack pixels from the background. The selection of the thresholds is critical to the performance of automated crack detection systems. Statistical values (mean and standard deviation) are used as the features, and they are used to train the neural network for selection of the thresholds. Because of the noise, the resulting images have some isolated spots that can be eliminated by a curve detector. Finally, Hough transformation is used to detect or classify all cracks in parallel. The experimental results have demonstrated that the cracks are correctly and effectively detected by the proposed method, which will be useful for pavement management.

Statistics published by FHWA indicate that maintenance and rehabilitation of highway pavements in the United States require more than $17 billion a year. Currently, maintenance and repairs account for nearly one-third of all federal, state, and local government road expenditures. According to the U.S. Department of Transportation, more than $315 billion will be needed through 2000 to maintain current road conditions (*1*).

Crack detection is very important to pavement management. Conventional visual and manual pavement cracking analysis approaches are very costly, time-consuming, dangerous, labor-intensive, and subjective. They possess the drawbacks of having high degrees of variability, they are unable to provide meaningful quantitative information, and they almost always lead to inconsistencies in cracking details over space and across evaluations. Automated pavement crack detection and classification systems have attracted the attention of many researchers. An automated crack detection system should ideally detect all types of cracking and other surface distresses of all sizes and at any collection speed. It should be affordable, easy to operate, and capable of operation during daylight hours. Pavement management programs would be adequately served by devices that give meaningful, repeatable distress ratings to sections of pavement, providing critical information for maintenance-related decision making (*2–6*).

An approach to recognition of segmented pavement distress images has been studied (*7*). It uses directional filters to classify the cracks. The crack is longitudinal if there is a high concentration of object pixels in a narrow interval of x (transverse) coordinates; it is transverse if there is a high count of object pixels in a narrow interval of y (longitudinal) coordinates. However, it is not clear how other crack types can be identified by analyzing these counts. Another statistical approach recognized the imperfections of segmentation that cause difficulty in distinguishing pavement crack types, especially between block and alligator cracks (*8*). By this method, the original image is enhanced by subtracting an average of a few plain images that have no cracks (nondistressed areas) from the same series to compensate for lighting variations. Segmentation is done by assigning one of four values to each pixel, based on its probability of being an object pixel. Classification is based on the identification of cracks in the segmented image:y block, joint, transverse, longitudinal, and alligator cracks and images with no cracks. A crack is a block crack if it contains a few joints with several horizontal and vertical cracks and is an alligator crack if it contains several joints and diagonal cracks.

Sophisticated techniques need to be used to improve the accuracy of classification, additional descriptors need to be used, and the feature vectors need to be evaluated. On the basis of the assumption that the histogram of a pavement containing cracks is bimodal (although this may not always be true), Li et al. proposed an algorithm for pavement crack detection (*9*). A standard model was proposed to represent pavement surface images so that a unified and automated means of acquisition of key characteristics can be used for improvement of data quality (*10*). However, how such a model should be used in crack detection and classification systems was not discussed.

Laser ranging is executed within a subset of the source image that covers the area of interest. The laser-ranging data would corroborate or reject the image data so that dark areas not caused by pavement distress, such as tire marks, oil spills, shadows, and recent fillings, may be ruled out. The development of a crack identification system by the laser-ranging technique was reported; however, it could not provide massive amounts of information across the lane by the use of only two or three narrow beams of laser light (*11*).

Fuzzy image enhancement and classification through a neural network trained with feature vectors was studied (*12*). The components of feature vectors are the central moments. A nonlinear equation calibrated with sample images is used to compensate for the nonuniform illumination. Some of the sample images presented thresholding imperfections, probably because of inadequate illumination correction, and the method could not determine the parameters of the membership function according to the nature of the images automatically.

Shadow Moire interferometry was used to measure coarse pavement distress, such as abnormal elevations and big potholes. This method allows detection of areas of the pavement that deviate from specified flatness criteria. The shadow Moire interferograms provide

H. D. Cheng, J. Wang, Y. G. Hu, X. J. Shi, and X. W. Chen, Department of Computer Science, Utah State University, Logan, UT 84322-4205. C. Glazier, Utah Department of Transportation, 4501 South 2700 West, Salt Lake City, UT 84119.

surface elevation variation measurements that cannot be obtained through ordinary videotaping. They can detect severe road elevation deformations caused by heavy loads and potholes with undefined borders, which optical methods cannot detect. The computational time is quite long (about 15 s/frame) (*13*).

A new pavement crack enhancement algorithm and an analysis-classification algorithm based on a new data structure were discussed (*14*). The algorithms could detect and classify cracks with high degrees of accuracy, but the computational times were high (>10 s/frame). Automated real-time crack analysis is even more challenging. A novel approach to crack detection based on fuzzy logic and fuzzy set theory has been proposed (*15*). Cheng et al. (*6*) proposed a new approach to the detection of cracks. This system is divided into two stages. At the first stage, the system detects the distress with a lower degree of accuracy and needs to answer only "yes" or "no" and indicate the orientation of the potential cracks. If the answer is "yes," the system will send the image to the frame buffer for further processing and will process the next image. At the second stage, the system processes the images in the frame buffer one by one and analyzes the types and severities of the cracks. Thus, the algorithms for the second stage have higher degrees of time complexity and accuracy.

Artificial neural networks (ANNs) attempt to model the information-processing capabilities of the nervous system. Some organizational principles (such as learning generalization, adaptability, fault tolerance, distributed representation, and computation) are used. In a neural network, the nodes are artificial neurons, and directed edges (with weights) are connections between the outputs and the inputs of the neurons. In recent years, because of their parallel processing capabilities, their learning capabilities, and their capabilities to solve nonlinear problems, ANNs have increasingly been applied to classification, prediction, optimization, control, image-processing, pattern recognition, and social science problems (*16–20*). Specifically, they are used with great success in many commercial products. Lately, many researchers have also paid a lot of attention to the use of ANNs in transportation fields (*21–25*).

PROPOSED METHOD

The proposed cracking detection algorithm consists of the following steps:

1. Collection of pavement image samples and calculation of the mean, standard deviation, and desired threshold values that will be used to train the neural network;
2. Design of a neural network;
3. Training and testing of the neural network and then use of the trained neural network to determine the threshold values for the newly acquired images;
4. Conversion of the gray-level images into binary images by using the thresholds obtained by the neural network;
5. Removal of isolated spots by the methods described elsewhere (*6, 14*);
6. Classification of the cracks into different types (longitudinal, transversal, diagonal, etc.) on the basis of the information from the Hough transformation algorithm; and
7. Calculation of crack severity by counting the number of pixels of the corresponding cracks. For alligator cracks, however, severity is determined by counting the number of pixels of the area surrounded by the crack.

Thresholding

Thresholding is a technique widely used for image segmentation. It converts a gray-level image into a bilevel image, which contains all of the essential information. The most common way is to select a single threshold value, and all the gray levels below this value will be classified as black (maximum intensity level = 0), whereas those above this value will be white (maximum intensity level = 255).

The threshold value must be determined according to the characteristics of the images. One simple approach is to use the image mean as the threshold value. This would result in about half of the pixels being white and about half of the pixels being black. Another easy way to find a threshold is by using the histogram of the image. In the easiest case, a procedure that determines the threshold needs to place the threshold only in the valley between the two modes (*26*). However, this condition cannot always be satisfied. Many more complicated and sophisticated methods have been proposed (*26*); however, they are not suitable for real-time processing. Finding of a fast, easy, and valid approach to determination of the threshold value is critical for many real-time applications.

In the present study statistical values (standard deviation and mean) were used as the features to train a neural network whose outputs will be the thresholds.

Let g_{pq} be the gray level of a pixel at location (p, q) in an $M \times N$ image, where M is the number of rows and N is the number of columns. The variance of image m (v_m) can be estimated as

$$v_m = \frac{1}{M \times N} \sum_{p=1}^{M} \sum_{q=1}^{N} (g_{pq} - \mu_m)^2 \quad (1)$$

where μ_m is the mean of the gray levels. The value of μ_m can be formulated as

$$\mu_m = \frac{1}{M \times N} \sum_{p=1}^{M} \sum_{q=1}^{N} g_{pq} \quad (2)$$

Solutions t_m of the following function need to be found:

$$t_m = f(v_m, \mu_m) \quad (3)$$

where t_m is the threshold of image m, in general, which is nonlinear.

Neural Network Calculation of Thresholds

Neural networks have been applied to many different disciplines. Their widespread acceptance as powerful and flexible forecasting and analysis tools, as well as the possibility of real-time processing, contribute to the fast growth of neural network studies.

Figure 1 shows the architecture of a feed-forward neural network. All processing elements (PEs) except those in the input layer perform two kinds of computations that determine the net input value to the PE and that set the output value. Each PE computes a net input value based on all its input weights. The net input (net_i) to the ith PE can be represented as (*17, 27, 28*)

$$\text{net}_i = \sum_j (X_j W_{ij} + \theta_j) \quad (4)$$

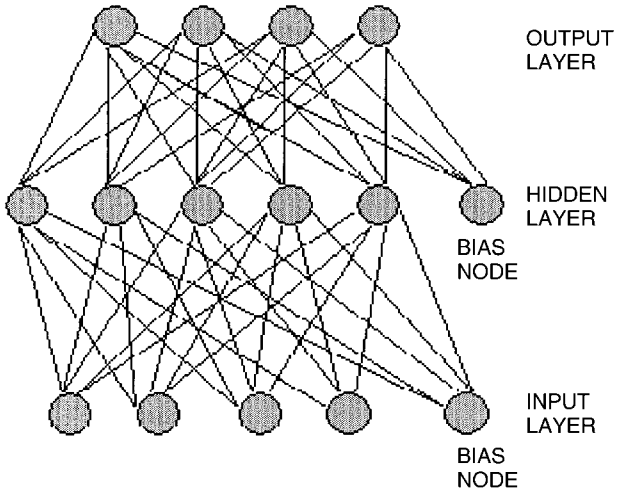


FIGURE 1 Structure of neural networks.

where

W_{ij} = weights,
X_j = inputs,
j = jth PE connected from ith PE, and
θ_j = bias.

Each PE will convert the net input to an activation value or simply activation:

$$a_i(t) = F_i[a_i(t-1); \text{net}_i(t)] \tag{5}$$

where

a_i = activation value,
$a_i(t)$ = a_i value at time t, and
F_i = function with two variables.

The output value can be computed by the output that is a function of a_i (O_i):

$$O_i = f_i(a_i) \tag{6}$$

where f_i is also a function. Usually, a_i is equal to net_i and the output value is

$$O_i = f_i(\text{net}_i) \tag{7}$$

To apply the most commonly used weight update algorithm, a generalized delta rule, the output function f_i must be differentiable. A commonly used function is the sigmoid function:

$$O_i = f_i(\text{net}_i) = \left(1 + e^{-\text{net}_i}\right)^{-1} \tag{8}$$

Each time the computed output value is compared with the desired response. A predefined learning algorithm is used to adjust the weights of the interconnections according to the error obtained from the comparison.

Network weights are adjusted to minimize the error measured by the difference between the desired and the actual network outputs.

$$E_p = 1/2 \sum_k \left(t_{pk} - O_{pk}\right)^2 \tag{9}$$

$$E = \sum_p E_p \tag{10}$$

where

E_p = error function
E = total error,
p = index for the training samples, and
k = index for the output PEs.

The following weight update rule can be used for the PEs in the output layer:

$$\Delta W_{kj}(t+1) = \Delta W_{kj}(t) + \eta \delta_{pk} i_{pj} \tag{11}$$

where η is the learning rate,

$$i_{pj} = \frac{\partial(\text{net}_{pk})}{\partial W_{kj}} \tag{12}$$

where net_{pk} is the net input to the pth PE, and

$$\delta_{pk} = \left(t_{pk} - O_{pk}\right) f_k'\left(\text{net}_{pk}\right) \tag{13}$$

where f_k' is the derivative. The weight update rule for the PEs in the hidden layers is

$$\Delta W_{kj}(t+1) = \Delta W_{kj}(t) + \eta \delta_{pk} i_{pj} \tag{14}$$

where $\Delta W_{kj}(t+1)$ is the weight at time $t+1$ and η is the learning rate, and

$$\delta_{pk} = f_k'\left(\text{net}_{pk}\right) \sum_j \delta_{pj} W_{jk} \tag{15}$$

Data Set

Pavement images were acquired with a video camera under different daylight conditions (morning, noon, and afternoon) and with the vehicle traveling at 56 to 105 km/h (35 to 65 mph). Images of the following types were selected: alligator cracks, transverse cracks, longitudinal cracks, 45-degree diagonal or 135-degree diagonal cracks, sealed cracks, no cracks, and shadows.

A total of 150 images were used. Of those images, 120 were randomly chosen for use in the training set and the remaining 30 were used for the test set. The training data cannot be used during the testing stage.

Variable Selection

The following variables for the neural networks were used in the experiments:

1. Mean of image m, η_m;
2. Standard deviation of image m, σ_m; and
3. Threshold value for image m, t_m.

Each value is normalized to achieve computational consistency. After normalization, each value will be between 0 and 1. The normalization value can be calculated as

$$U_m = \frac{\mu_m}{\mu_{\max}}$$

$$\sum_m = \frac{\sigma_m}{\sigma_{\max}}$$

$$T_m = \frac{t_m}{255}$$

where $\mu_{\max}$ is equal to max $\{\mu_m\}$ and $\sigma_{\max}$ is equal to max$\{\sigma_m\}$. The value 255 is the maximum gray level. (U, Σ, T) represents a set of normalized values.

Configuration of Neural Network

In the experiment described here, the number of nodes in the input layer is 3. One of them is used for the bias. Only a single node is required in the output layer, and only one hidden layer is needed. The configuration can be varied by changing the number of nodes in the hidden layer. Determination of the number of nodes in the hidden layer is quite important. The number of nodes in the hidden layer is adjusted from 2 to 20 to obtain a neural network with better performance.

Figure 2 shows that the output root mean square (RMS) error is minimum when the number of the nodes in the hidden layer is 15. The final configuration of the neural network is as follows:

- Architecture: three-layer backpropagation neural network;
- Input of the neural network: mean and standard deviation;
- Output of the neural network: threshold value;
- Number of hidden neurons: 15;
- Momentum: 0.5;
- Learning rate: 0.5;
- Learning rule: delta rule; and
- Activation function: sigmoid.

Training the Neural Network

Network weights were randomly initialized, and the training set was presented to the network in a random order to minimize bias and prevent the neural network from memorizing the data in the training set.

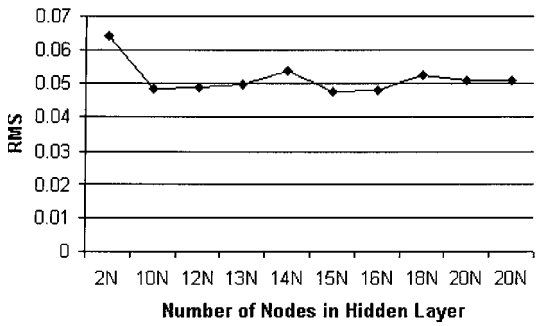


FIGURE 2 RMS value versus number of nodes.

The training process was terminated when the number of epochs reached 3,000.

Removal of Isolated Pixels and Shadow Detection

Because the pavement image has some tars (bleeding), it is difficult to analyze the crack type on the image. Meanwhile, under some circumstances, the image may have some noise after thresholding. The connectivity-checking algorithm (*6, 29*) should be applied to remove small isolated object pixel clusters that are considered noise. The thinning algorithm reduces the object width to just one pixel so that it is quite easy to determine the length of the object. The number of object pixels within each connected cluster is counted, and all clusters having a size below a given threshold are assigned the background attribute. If the threshold is too large, it eliminates small clusters accidentally disconnected from the distress during the segmentation process. In the experiments, the threshold value of 3 produced acceptable results. The shadows can be distinguished from cracks by using the projection and thinning algorithms described elsewhere (*6*).

Hough Transformation Algorithm

After use of the thresholding and neural network algorithm, the image becomes a bilevel image and has a clear structure for determination of the type of crack. Hough transformation is used to detect lines in the image. Hough transformation has successfully been applied to the detection of parameter curves (e.g., straight lines and circles) (*30–32*). By use of the Hough transformation, one can obtain information about the locations, magnitudes, and orientations of these lines in the form of their start and end coordinates and their orientations. Hough transformation specifies a straight line by an angle θ with the x-axis and has a norm ρ. The equation of a line is given by

$$\rho = x\cos(\theta) + y \times \sin(\theta) \tag{16}$$

where (x, y) is any point on the line. A set of collinear points with a given angle in image space would yield the same value of ρ. A straight line in image space corresponds to a parametric point in the ρ-θ plane. After the transformation, the peaks in the parameter space hold information about the lines in the image space. Thus, scanning of those peaks above a threshold value would yield information about the number of lines and one can determine the crack types that the pavement images have. This algorithm can efficiently classify the types of crack: transverse, longitudinal, diagonal, or alligator cracks or a combination. In the experiment described here, the Hough transformation is implemented at 5-degree intervals from 0 to 180 degrees. The selection of ρ is set at steps of 5 pixels. If there is only a single line with $\theta = 0$, 45 (or 135), and 90 degrees, the crack is classified as transverse, diagonal, and longitudinal, respectively. If there are more lines, the crack is an alligator crack or a combination crack. Alligator and combination cracks can be discriminated by the number of lines above the threshold in the Hough transformation. That is, if the crack is a combination crack, the number of lines is small, and if it is an alligator crack, the number of lines is quite large.

Because the image size is fixed (as is the area of the pavement covered), one can count the number of pixels of the corresponding cracks to represent the severity. However, the severity of an alligator crack

TABLE 1 Testing Data and Neural Network Outputs

Mean	Var.	Thre.	NMean	NVar.	NThre.	ONN.	Diff.	Thre.	TNN.
120.47	25.62	65.00	0.9163	0.6868	0.2549	0.2492	0.0057	65.00	63.55
116.65	20.36	64.00	0.8873	0.5456	0.2510	0.2573	-0.0063	64.00	65.61
117.78	24.88	63.00	0.8959	0.6668	0.2471	0.2533	-0.0063	63.00	64.60
115.20	24.09	61.00	0.8762	0.6456	0.2392	0.2460	-0.0068	61.00	62.73
115.72	24.18	61.00	0.8802	0.6480	0.2392	0.2417	-0.0025	61.00	61.64
118.04	26.44	58.00	0.8978	0.7087	0.2275	0.2350	-0.0076	58.00	59.93
119.90	25.46	65.00	0.9120	0.6825	0.2549	0.2540	0.0009	65.00	64.78
116.97	22.54	62.00	0.8897	0.6042	0.2431	0.2441	-0.0010	62.00	62.24
118.87	29.44	60.00	0.9042	0.7891	0.2353	0.2386	-0.0033	60.00	60.83
116.01	37.31	55.00	0.8824	1.0000	0.2157	0.2133	0.0024	55.00	54.39
114.46	30.10	60.00	0.8706	0.8068	0.2353	0.2336	0.0017	60.00	59.56
116.15	24.40	64.00	0.8835	0.6541	0.2510	0.2424	0.0086	64.00	61.82
131.47	30.40	62.00	1.0000	0.8148	0.2431	0.2408	0.0023	62.00	61.40
121.33	25.61	65.00	0.9229	0.6865	0.2549	0.2494	0.0055	65.00	63.60
114.84	34.49	62.00	0.8735	0.9244	0.2431	0.2387	0.0044	62.00	60.87
117.37	22.05	64.00	0.8928	0.5909	0.2510	0.2398	0.0112	64.00	61.15
114.08	22.67	64.00	0.8677	0.6076	0.2510	0.2425	0.0085	64.00	61.84
118.40	25.53	60.00	0.9006	0.6843	0.2353	0.2463	-0.0110	60.00	62.81
114.75	24.54	60.00	0.8728	0.6578	0.2353	0.2375	-0.0022	60.00	60.56
118.09	18.95	65.00	0.8982	0.5078	0.2549	0.2552	-0.0003	65.00	65.07
111.87	29.96	60.00	0.8509	0.8031	0.2353	0.2374	-0.0021	60.00	60.54
116.69	23.55	65.00	0.8876	0.6312	0.2549	0.2519	0.0030	65.00	64.23
117.08	21.64	62.00	0.8905	0.5800	0.2431	0.2447	-0.0015	62.00	62.39
110.65	29.72	58.00	0.8416	0.7966	0.2275	0.2332	-0.0057	58.00	59.46
121.96	30.50	60.00	0.9277	0.8175	0.2353	0.2314	0.0039	60.00	59.01

Mean: mean value of the image; **Var.:** standard deviation of the image; **Thre.:** desired threshold value; **NMean:** normalized mean value; **NVar.:** normalized standard deviation; **NThre.:** normalized desired threshold; **ONN.:** neural network output; **Diff.:** difference between normalized desired threshold and neural network output; **TNN:** scaled neural network output (ONN x 255).

is computed by counting the number of pixels of the area surrounded by the crack. The widths of the cracks can be computed by using the projection and morphology algorithms described elsewhere (*6*) after the orientations of the cracks are determined as described above.

EXPERIMENTAL RESULTS

The algorithms discussed above have been applied to process pavement images. Three evaluation criteria were chosen: (*a*) performances of different pavement types, (*b*) performances under different light conditions and circumstances, and (*c*) performances when there are non-crack-related scenes on the image. One can apply the proposed algorithms to process various videotape images, even those captured by other companies. Light condition is an important factor when crack information is being determined. Because of the unbalanced enhancement, the image usually does not have a uniform gray level. Therefore, images were captured under different light conditions. Table 1 summarizes some of the results obtained with 30 images.

Figure 3 shows the differences between the outputs of the neural network and the desired threshold values (normalized). Figures 4 to 6 show the original images (panels *a*), the images obtained after application of the desire threshold values (panels *b*), and the images obtained after application of the threshold values computed by the neural network (panels *c*). It can be found that the images in panels *b* and *c* are quite close. Figure 7 has an alligator crack, and the images in Figures 7*b* and 7*c* are quite close as well. In the experiments, 30 images with no cracks were also used to demonstrate that the proposed algorithm is sensitive without producing false-positive results. Figure 8*c* demonstrates the effectiveness of the isolated pixel removal algorithm.

In summary, the experimental results prove that the proposed system can effectively and accurately detect and classify cracks. The computational time is quick enough to meet the real-time requirement [the vehicle speed could be 56 to 105 km/h (35 to 65 mph)], and it will be very useful for pavement management systems.

CONCLUSIONS

In this paper, a novel approach uses the parallel and nonlinear nature of neural networks and computer vision, pattern recognition, and

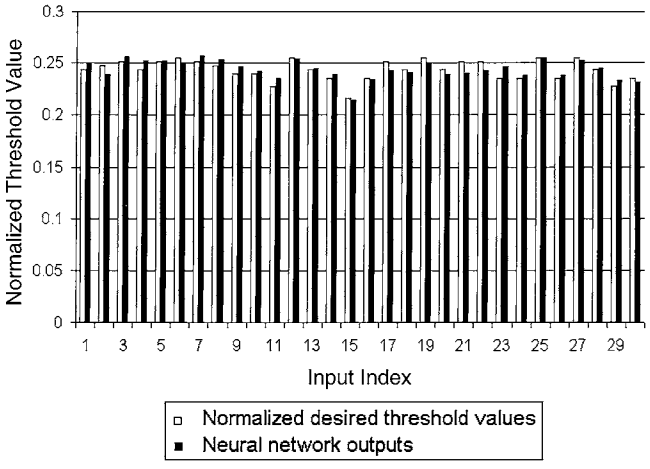
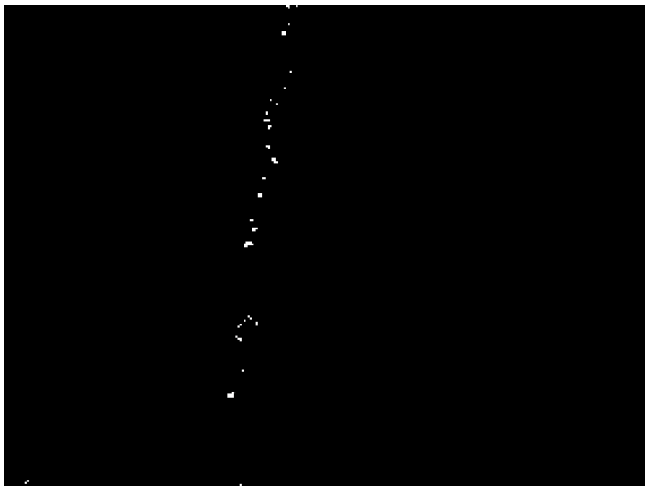


FIGURE 3 Comparison of neural network outputs with desired threshold values.

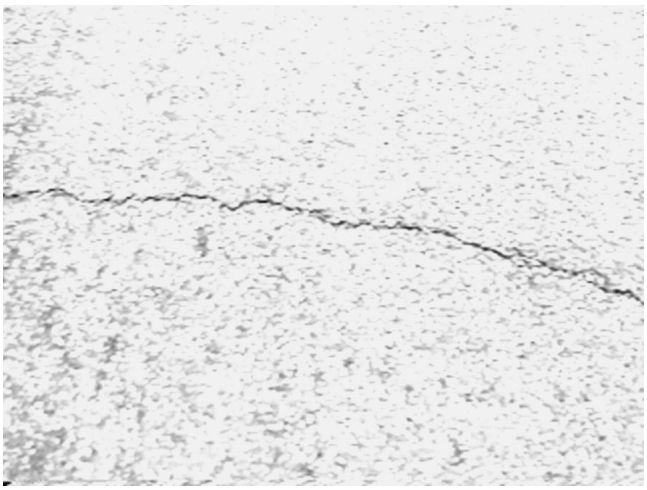
(a)

(b)

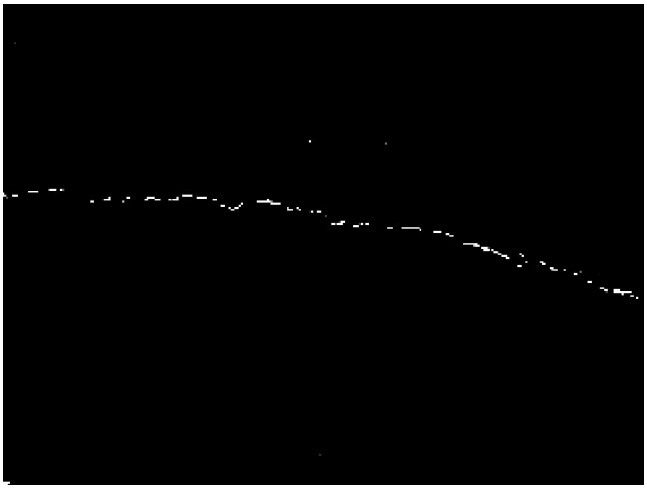


(c)

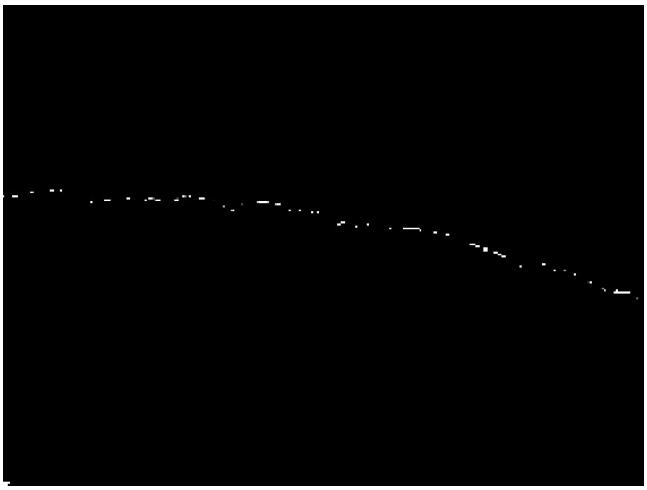
FIGURE 4 Longitudinal crack: (*a*) original image; (*b*) result obtained by using desired threshold; (*c*) result obtained by using neural network to compute threshold.



(a)



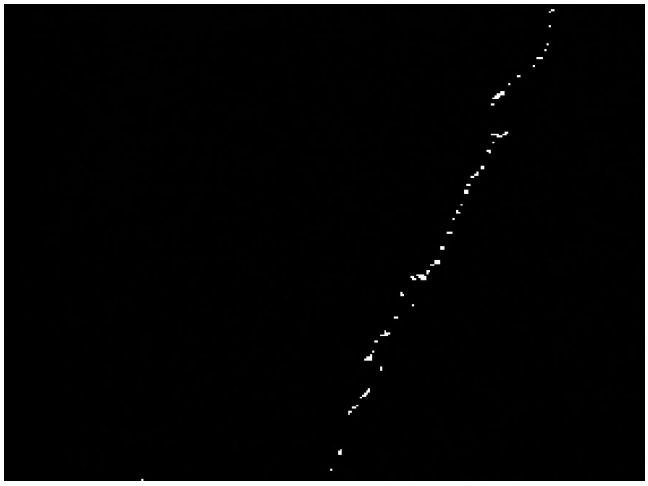
(b)



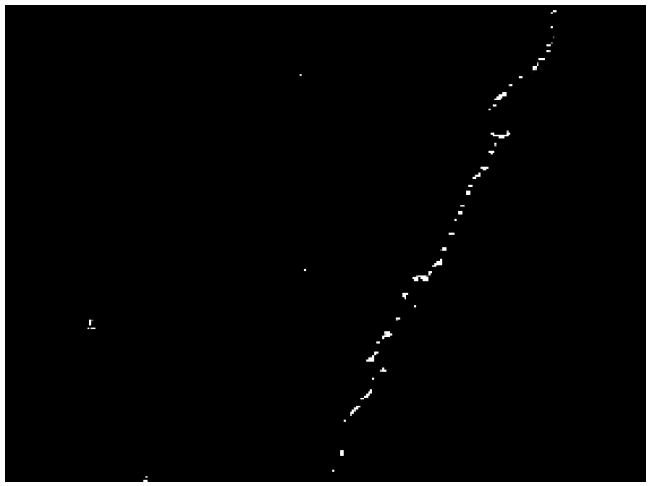
(c)

FIGURE 5 Transverse crack: (*a*) original image; (*b*) result obtained by using desired threshold; (*c*) result obtained by using neural network to compute threshold.

(a)



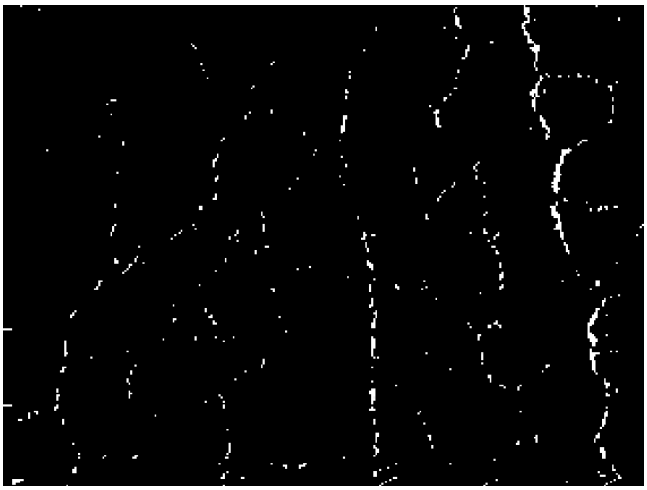
(b)



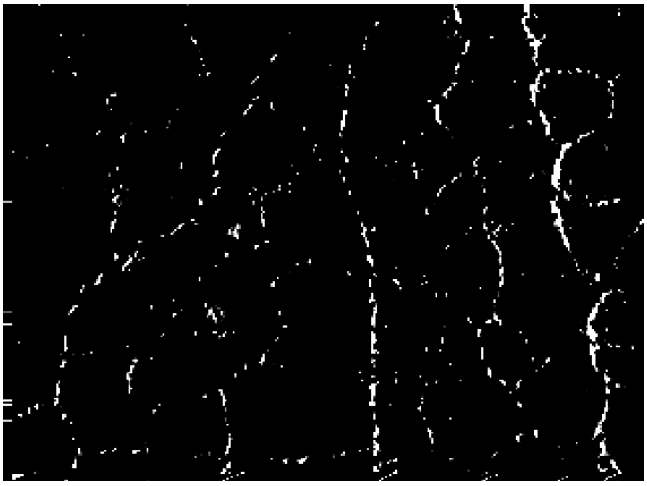
(c)

FIGURE 6 Diagonal crack: (*a*) original image; (*b*) result obtained by using desired threshold; (*c*) result obtained by using neural network to compute threshold.

(a)

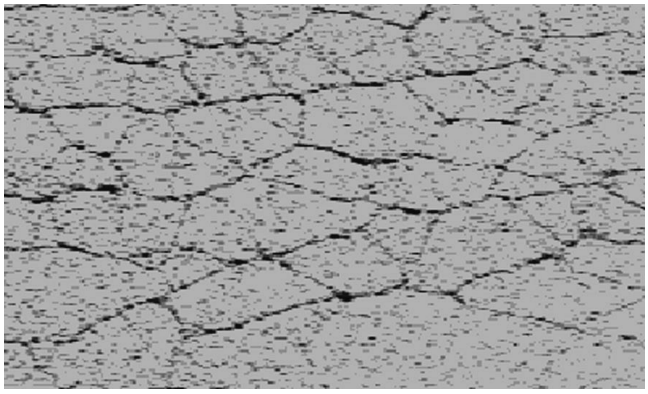


(b)

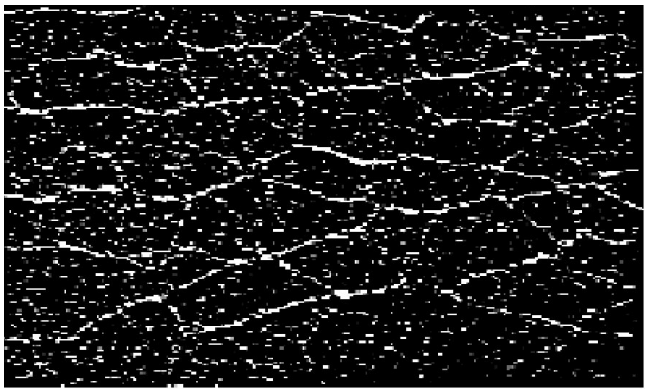


(c)

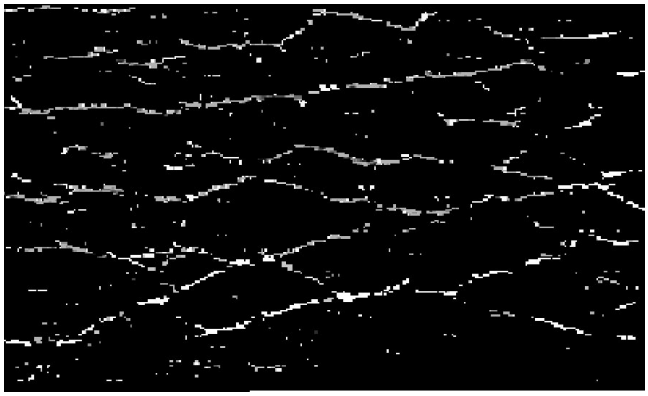
FIGURE 7 Alligator crack: (*a*) original image; (*b*) result obtained by using desired threshold; (*c*) result obtained by using neural network to compute threshold.



(a)



(b)



(c)

FIGURE 8 (*a*) Original image; (*b*) result after thresholding; (*c*) result after isolated pixel removal.

image-processing techniques to detect and classify cracks in real time. The preliminary results have shown the efficiency and effectiveness of the proposed algorithm. This system can effectively and accurately identify the types of cracks, improve the safety and efficiency of data collection, offer an objective standard of analysis and classification of cracks, and help identify cost-effective maintenance and repair plans. The proposed algorithm will be useful for pavement management.

ACKNOWLEDGMENT

This work was sponsored in part by a grant from the U.S. Department of Transportation.

REFERENCES

1. Tsao, S., N. Kehtarnavaz, P. Chan, and R. Lytton. Image-Based Expert-System Approach to Distress Detection on CRC Pavement. *Journal of Transportation Engineering*, Vol. 120, No. 1, Jan.–Feb. 1994, pp. 52–64.
2. Guralnick, S. A., E. S. Sun, and C. Smith. Automating Inspection of Highway Pavement Surfaces. *Journal of Transportation Engineering*, Vol. 119, No. 1, 1993, pp. 1–12.
3. Haas, C., and S. McNeil. Criteria for Evaluating Pavement Imaging Systems. In *Transportation Research Record 1260*, TRB, National Research Council, Washington, D.C., 1990, pp. 64–73.
4. Koutsopoulos, H. N., and I. El Sanhouri. Methods and Algorithms for Automated Analysis of Pavement Images. In *Transportation Research Record 1311*, TRB, National Research Council, Washington, D.C., 1991, pp. 103–111.
5. Koutsopoulos, H. N., R. G. Mishalani, and A. B. Downey. Automated Analysis of Pavement Distress Data. *Proc., 2nd International Conference on Applications of Advanced Technologies in Transportation Engineering*, 1991, pp. 233–237.
6. Cheng, H. D., X. Jiang, J. Li, and C. Glazier. Automated Real-Time Pavement Distress Analysis. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1655*, TRB, National Research Council, Washington, D.C., 1999, pp. 55–64.
7. Mohajeri, M. H., and P. J. Manning. ARIA: An Operating System of Pavement Distress Diagnosis by Image Processing. In *Transportation Research Record 1311*, TRB, National Research Council, Washington, D.C., 1991, pp. 120–130.
8. Koutsopoulos, H. N., and A. B. Downey. Primitive-Based Classification of Pavement Cracking Images. *Journal of Transportation Engineering*, Vol. 19, No. 3, 1993, pp. 402–418.
9. Li, L., P. Chan, A. Rao, and R. L. Lytton. Flexible Pavement Distress Evaluation Using Image Analysis. *Proc., 2nd International Conference on Applications of Advanced Technologies in Transportation Engineering*, 1991, pp. 473–477.
10. Haas, C., and C. Hendrickson. Computer-Based Model of Pavement Surfaces. In *Transportation Research Record 1260*, TRB, National Research Council, Washington, D.C., 1990, pp. 91–98.
11. Walker, R. S., and R. L. Harris. Noncontact Pavement Crack Detection System. In *Transportation Research Record 1311*, TRB, National Research Council, Washington, D.C., 1991, pp. 149–157.
12. Chou, J., W. A. O'Neill, and H. Cheng. Pavement Distress Evaluation Using Fuzzy Logic and Moment Invariants. In *Transportation Research Record 1505*, TRB, National Research Council, Washington, D.C., 1995, pp. 39–46.
13. Guralnick, S. A., E. S. Suen, and C. Smith. Automating Inspection of Highway Pavement Surfaces. *Journal of Transportation Engineering*, Vol. 119, No. 1, 1993, pp. 1–12.
14. Cheng, H. D., and M. Miyojim. Novel System for Automatic Pavement Distress Detection. *Journal of Computing in Civil Engineering*, Vol. 12, No. 3, 1998, pp. 145–152.
15. Cheng, H. D., J. R. Chen, C. Glazier, and Y. G. Hu. Novel Approach to Pavement Distress Detection Based on Fuzzy Set Theory. *Journal of Computing in Civil Engineering*, Vol. 13, No. 4, Oct. 1999, pp. 270–280.
16. Sun, C., C. M. U. Neale, J. J. McDonnell, and H. D. Cheng. Monitoring Landsurface Snow from SSM/I Data Using an Artificial Neural Network Classifier. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 35, No. 4, July 1997, pp. 801–809.
17. Mueller, R. J., H. D. Cheng, and A. R. Barakat. Incoherent Radar Spectrum Processing Using Neural Network. *Signal Processing*, Vol. 69, 1998, pp. 117–129.

18. Haykin, S. *Neural Networks: A Comprehensive Foundation,* 2nd ed., Prentice-Hall, Englewood Cliffs, N.J., 1999.
19. Looney, C. G. *Pattern Recognition Using Neural Networks.* Oxford University Press, New York, 1997.
20. Nelson, M. M., and W. T. Illingworth. *A Practical Guide to Neural Nets.* Addison-Wesley, Reading, Mass., 1991.
21. Bullock, D., J. Garrett, and C. Hendrickson. A Neural Network for Image-Based Vehicle Detection. *Transportation Research C,* Vol. 1, No. 3, 1993, pp. 235–247.
22. Faghri, A., and J. Hua. Evaluation of Artificial Neural Network Applications in Transportation Engineering. In *Transportation Research Record 1358,* TRB, National Research Council, Washington, D.C., 1992, pp. 71–80.
23. Hua, J., and A. Faghri. Applications of Artificial Neural Networks to Intelligent Vehicle-Highway Systems. In *Transportation Research Record 1453,* TRB, National Research Council, Washington, D.C., 1994, pp. 83–90.
24. Wang, S. S. *Combining Image Processing and Neural Networks to Vehicle Classification.* M.S. thesis. Institute of Transportation and Communication Management, National Cheng Kung University, Taiwan, Republic of China, 1995.
25. Wei, C. H., and S. S. Wang. Automatic Toll Collection with Artificial Neural Networks. *Proc., 8th Conference of Road Engineering Association of Asia and Australasia,* Vol. 1., April 1995, pp. 423–427.
26. Fu, K. S., and J. K. Mui. A Survey on Image Segmentation. *Pattern Recognition,* Vol. 13, 1981, pp. 3–16.
27. Zurada, J. M. *Introduction to Artificial Neural Systems.* West Publishing, St. Paul, Minn., 1992.
28. Hertz, J., A. Krough, and R. G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood City, Calif., 1991.
29. Cheng, H. D., C. Tong, and Y. J. Lu. VLSI Curve Detector. *Pattern Recognition,* Vol. 23, No. 1/2, 1990, pp. 35–50.
30. Hough, P. V. C. *Method and Means for Recognizing Complex Patterns.* U.S. Patent 3069,654, Dec. 18, 1962.
31. Duda, R. O., and P. E. Hart. Use of the Hough Transformation to Detect Line and Curves in Pictures. *Journal of Graphics and Image Processing,* Vol. 15, No. 1, Jan. 1972.
32. Ballard, D. H. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition,* Vol. 13, No. 2, 1981, pp. 111–122.

Publication of this paper sponsored by Committee on Pavement Monitoring, Evaluation, and Data Storage.