ASPHALT PAVEMENT CRACK CLASSIFICATION: A COMPARISON OF GA, MLP, AND SOM

Haroun Rababaah
Dana Vrajitoru
James Wolfer

Computer and Information Sciences
Indiana University South Bend
1700 Mishawak Avenue
P.O. Box 7111

ABSTRACT
Asphalt pavement distress, the various defects such as holes and cracks, represent a significant engineering and economic concern. It is estimated that pavement defects cause damage costing $10 billion each year in the United States alone [10]. One important step in managing this problem is accurately assessing the pavement condition and its change over time. In this paper we compare three methods for automatically classifying pavement cracks, genetic algorithms, multilayer perceptrons, and self-organizing maps. We also discuss the impact of feature representation on the resulting classification. Our best classifiers demonstrated accuracies between 86 and 98%.

Keywords
Classification, pavement cracks, multilayer perceptron, genetic algorithms, self-organizing maps, computer vision.

1. INTRODUCTION

Large structures are usually constructed with materials that exhibit distress over time due to loading, environmental conditions, and normal wear. These large structures include pavement, chimneys for nuclear power plants, skyscrapers, and pipelines among others. Often the distress are present in the form of surface cracking [1]. For this investigation, we focus on asphalt pavement. Since pavement forms a large part of our transportation infrastructure, a number of distresses have been identified and their characteristics cataloged [5]. Four common pavement crack types are illustrated in Figure 1.1, and form the target of our classification system.

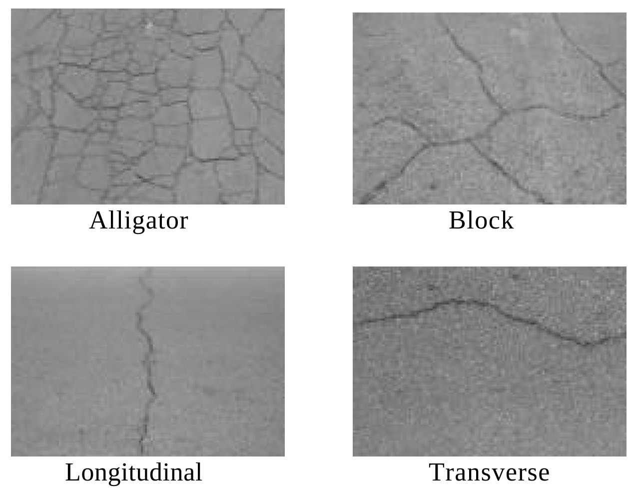

Figure 1.1 Types of pavement cracks

Motivation

Observing methods employed by experts in the pavement assessment industry [2] we note the following limitations in current practice:

- They involve computationally expensive features of crack objects.
- A hard coded criterion is not likely to perform well in the harsh crack-pavement environment, which includes a significant amount of noise and stochastic distribution and geometry of cracks.
- The processing speed is important, since the project of pavement assessment involves surveying 100% of target pavement. Processing time per image plays an important factor in motivating us to design more efficient techniques, especially if the analysis application required real-time processing.

Given these limitations, we believe that pavement crack analysis provides a viable test case for a real-world classification comparison. In this case, we compare the results of two supervised approaches, a genetic algorithm

(GA) and the multilayer perceptron (MLP) with the unsupervised self-organizing map (SOM).

We introduced three new methods to improve the overall system efficiency. Linear regression technique was used for crack objects detection. This method reduced the original image space into two-dimensional boolean matrix. The projection method was used for image representation. This method constructed the feature vectors with only two components to characterize the different crack types. The class map, and the GA encoding method were used to evolve an efficient GA classifier by evolving a crack-type map of two-dimensional matrix. The classification model used the two components of feature vectors as a lookup coordinates on the space of the evolved map.

2. PREVIOUS WORK

Chou et al. [6] approached the problem of pavement crack classification using moment invariant and neural networks. After preprocessing and thresholding into binary images, they calculated Hu, Bamieh, and Zemike moments. Eighteen moments were supplied as an input vector to a multilayer perceptron with seven nodes indicating the output. After training and testing, they report a one-hundred percent classification accuracy.

Mohajeri and Manning [2], as part of a fully automated pavement management system, describe a rule-based system that incorporates knowledge about individual crack patterns and classifies by the process of elimination. In addition to classifying cracking by type, it is also capable of quantifying the crack severity with parameters such as length, width, and area.

Lee and Cheng [3] concentrate on image preprocessing and representation for input to a neural network. After tiling the image, they use local statistics to tag tiles which contain cracks, thus forming a binary crack matrix. This matrix is then summed along the X and Y axes, forming two histogram vectors. After additional processing these vectors are presented to an MLP for classification. They report ninety-five percent accuracy.

Wang et al. [4] describes an automated system capable of real-time assessment. Using analytical descriptions of pavement stress, they compare the images under consideration with a pre-defined database of typical crack characteristics such as location and geometry, ultimately producing surface crack indices.

Hsu et al. [12] described a moment invariant technique for feature extraction and a neural network for crack classification. The moment invariant technique reduces a two dimensional image pattern into feature vectors that characterize the image such as: translation, scale, rotation of an object in an image. After these features are extracted The overall results of this study were satisfactory and the classification accuracy of the introduced system was eighty-five percent.

3. METHODOLOGY

This paper describes an approach, which accepts binary images, processes them into an appropriate representation, and presents them to a variety of classifiers.

3.1. Preprocessing

First, the pavement images are manually thresholded to form a binary image containing the crack as illustrated[1] in Figure 3.1. The resulting binary image is then passed through a median filter to reduce noise, forming the image of Figure 3.2. The resulting filtered, binary, image is then subdivided into square tiles. A linear regression is applied to the pixels within each tile. The resulting correlation factor and number of pixels per tile are compared with pre-determined thresholds, ultimately forming a Boolean matrix, which indicates which tiles probably contain a distress fragment as illustrated in Figure 3.3. The projected vectors are shown in Figure 3.4.

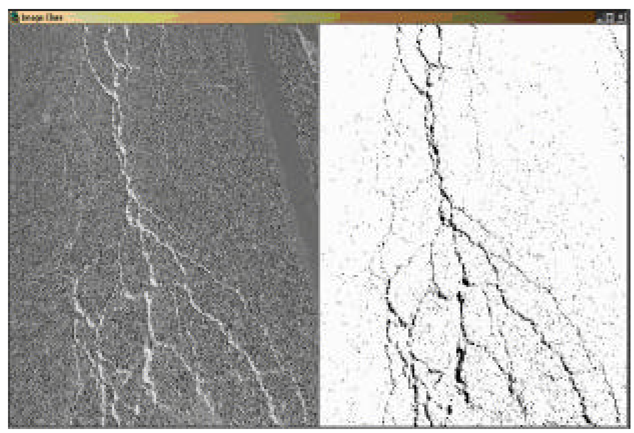

Figure 3.1. Thresholding

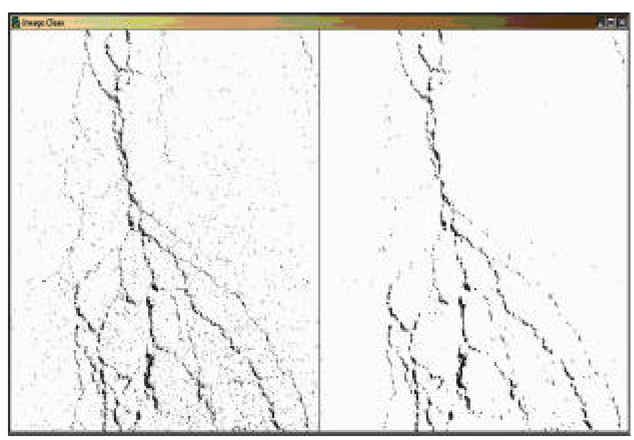

Figure 3.2. Median Filtering

[1] All images have inverted grayscale for printing.

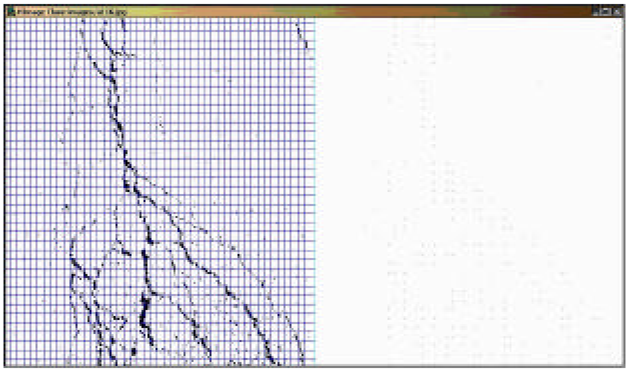

Figure 3.3. Tiling and Linear regression

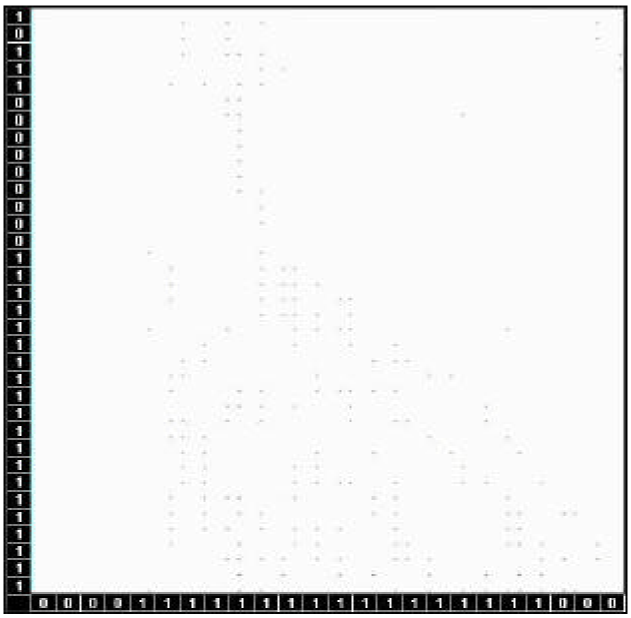

Figure 3.4. Vector projection

3.3. Feature Representation

We compared two different representations to reduce the data supplied to the classifiers, one based on the Hough transform, and one based on projecting the tile matrix along its major axes.

For the Hough method, a Hough transform was calculated for each image. Feature vectors were then formed from average angles in Hough Space, along with the total of the true-boolean values for each matrix.

Alternately, we applied a projection-based approach that creates a pair of binary vectors by projecting the Boolean matrix on the X and Y axes. This forms vectors with the following characteristics as illustrated in Figure 3.5.

- Alligator cracks: have projection points on both X and Y axis with higher frequency than block cracks.
- Block cracks: have projection points on both X and Y axis.
- Longitudinal cracks: have projection points mainly on the X-axis
- Transverse cracks: have projection points mainly on the Y-axis.

	X	Y
Alligator	1 1 1 1 1 1 1 1 0 1	1 1 1 1 1 0 1 1 1 1
Block	1 0 1 0 0 1 0 1 0 1	1 1 0 0 1 0 1 0 0 1
Longitudinal	1 1 1 1 1 0 0 0 0 0	0 0 0 0 0 1 1 0 0 0
Transverse	1 1 0 0 0 0 0 0 0 0	1 1 1 1 1 1 0 0 0 0

Figure 3.5. Feature vectors

The input vectors used in training and testing the three classifiers were formed as follows: in the projection method, the summation of the '1's (true) in each sub-vector then is used to form two dimensional vectors (x-sum, y-sum). In the Hough transform, the average of the angles in the Hough space, and the total number of crack-hits in the hit-patterns matrices were used to construct two-element vector for each image. In the GA classifier we used these input vectors with no further modification, where as in the MLP and SOM classifiers, the vectors were normalized.

3.2 Crack Classification

We compared three classifiers for this study, the GA, MLP, and SOM. To ensure comparable results we formed training and testing data as follows:

1. Five-hundred images representative of the four crack types were collected and independently classified by an industry expert [2].
2. The classified images were preprocessed. This stage produced two master files of vectors: one for projection technique, and the other one for the Hough transform.
3. For each master file, five different files for training (400 vectors each) and five different files (100 vectors each) for testing were generated, using the cross validation technique [11].

3.2.1. MLP Classification

The MLP consisted of two input units, one output unit for each of the four possible crack types, along with a variable number of hidden units. The number of hidden units was assessed empirically with both the Hough and projection representations, resulting in three hidden units being chosen for this study.

3.2.2. GA Classification

We approached the genetic classification by evolving a classification matrix that would map the target classes into distinct regions on a matrix that could then be indexed by feature coordinates from the testing set. A typical evolved GA matrix is illustrated in Figure 3.6.

3.2.3. SOM Classification

In addition to providing a similar output representation as the genetic approach described above, the SOM is an example of an unsupervised neural network. A typical trained map is shown in Figure 3.7. The same method for classification and resolution used in GA, was used in SOM.

The methods used for classification are publicly well-known, and further information about these methods can be found in [7], [8], and [9].

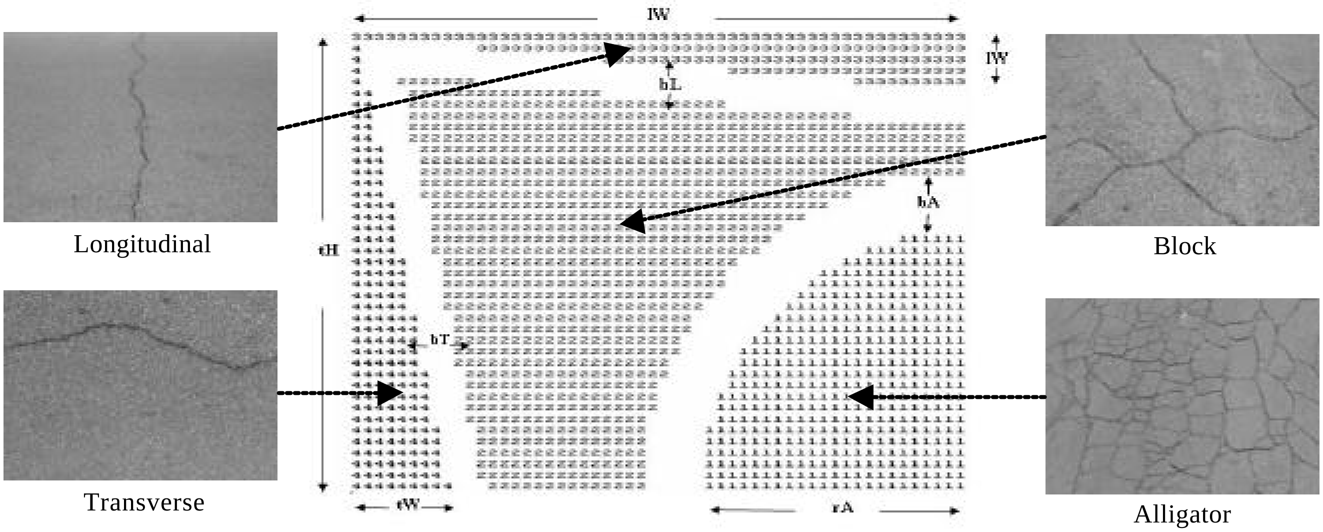

Figure 3.6. Typical GA classifier matrix

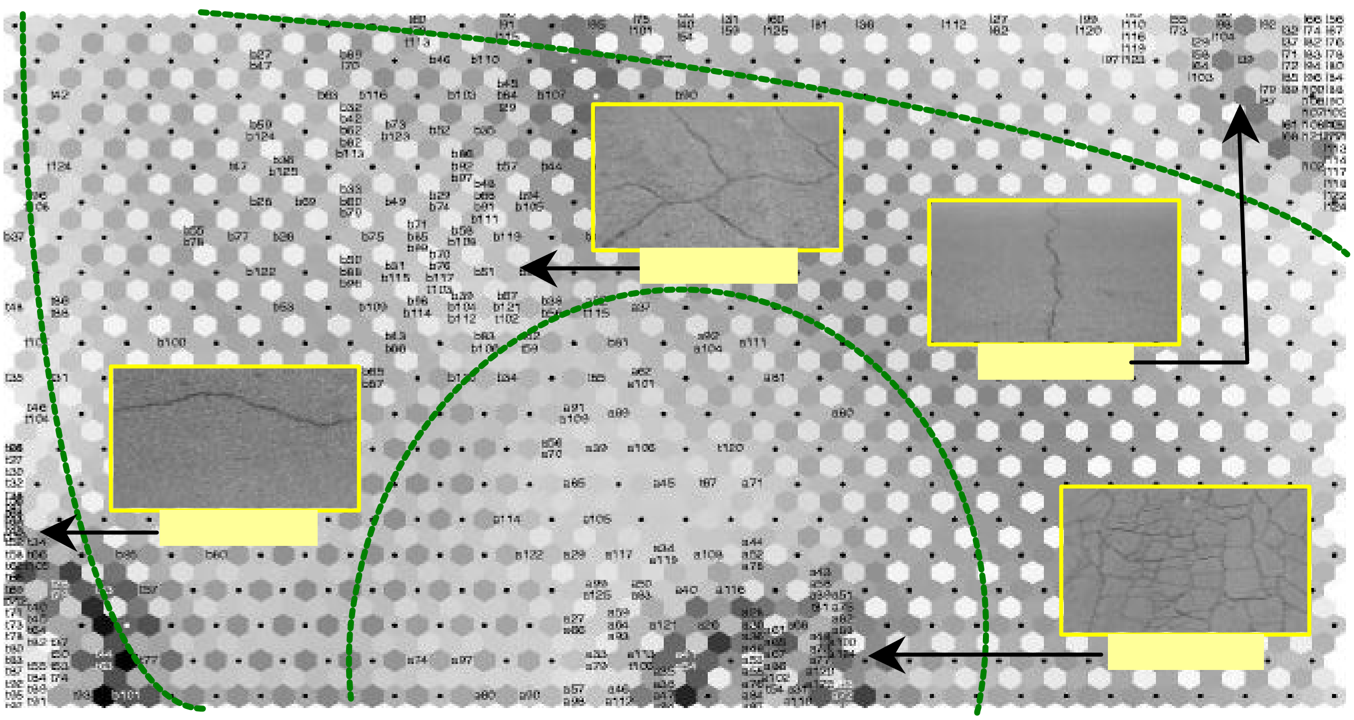

Figure 3.7. Typical trained SOM

4. RESULTS AND DISCUSSION

The tables Figures 4.1-4.3 of the aggregate results have the following uniform format:

- MLP_hough: is the label of the table, which combines the name of the classifier (MLP), and the image representation method used in this particular experiment.

- For each crack type, the table lists the classification distribution of the 125 test images. Image representation method, and gives the aggregate C/T for each crack type. The column chart plots the crack type vs. the C/T for that crack type.

The final stage in the system were classification and testing. Of the classification models, GA and MLP representing supervised learning and the SOM an example of an unsupervised approach. The testing phase produced a set of thirty classification results, ten for each approach. These are summarized and presented in Figures 4.1-4.3.

From Figures 4.1-4.3 we observe that the projection representation for our images produced consistently better results than the Hough representation. This suggests that simple representations should be considered among alternatives when exploring classification.

The best classifier was the MLP with the projection representation, with an overall accuracy of 98.6%. The other classifiers demonstrated similar accuracy, with the GA at 98.2% and 98.4% for the SOM. Both the supervised and unsupervised models produced similar classification performance.

While not perfect, since we used different image sets, it is still useful to compare our approach and results to other approaches presented in previous work. In [6], although their results were perfect, we noticed that their feature extraction method was computational expensive. It involves producing a vector of eighteen features computed by variety of moments. These moments range from 1st to 3rd order equations.

MLP_Hough

	C/T	a	b	l	t	total
a	0.984	**123**	2	0	0	125
b	0.928	1	**116**	5	3	125
l	0.984	0	2	**123**	0	125
t	0.848	10	9	0	**106**	125
Over all	**0.936**					**500**

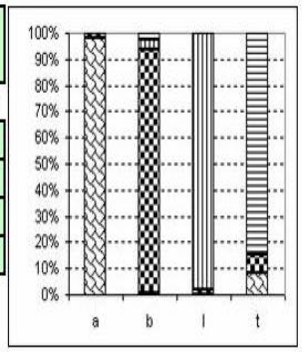

Figure 4.1-a Aggregate results of MLP/Hough

MLP_Projection

	C/T	a	b	l	t	total
a	1	**125**	0	0	0	125
B	0.976	1	**122**	0	2	125
L	0.984	0	1	**123**	1	125
T	0.984	0	1	1	**123**	125
Over all	**0.986**					**500**

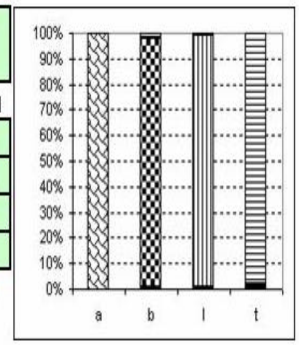

Figure 4.1-b Aggregate results of MLP/Projection

GA_Hough

	C/T	a	b	l	t	total
A	0.96	**120**	5	0	0	125
B	0.928	0	**116**	0	9	125
L	0.864	0	17	**108**	0	125
T	0.816	7	16	0	**102**	125
Over all	**0.892**					**500**

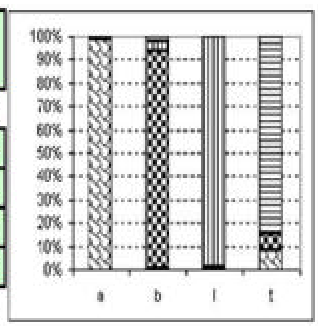

Figure 4.2-a Aggregate results of GA/Hough

GA_Projection

	C/T	a	b	l	t	total
a	0.968	**121**	4	0	0	125
b	0.976	3	**122**	0	0	125
l	1	0	0	**125**	0	125
t	0.984	0	1	1	**123**	125
Over all	**0.982**					**500**

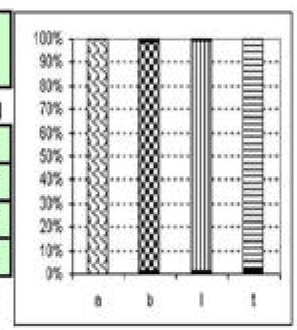

Figure 4.2-b Aggregate results of GA/Projection

SOM_Hough

	C/T	a	b	l	t	total
a	0.792	**99**	2	0	24	125
b	0.832	1	**104**	3	17	125
l	0.976	0	3	**122**	0	125
t	0.84	9	11	0	**105**	125
Over all	**0.86**					**500**

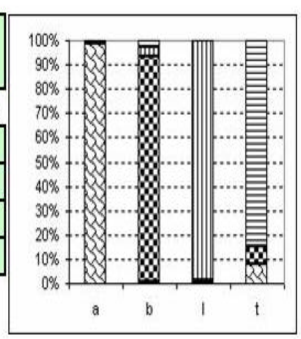

Figure 4.3-a Aggregate results of SOM/Hough

SOM_Projection

	C/T	a	b	l	t	total
a	1	**125**	0	0	0	125
b	0.968	1	**121**	2	1	125
l	0.992	0	1	**124**	0	125
t	0.976	0	2	1	**122**	125
Over all	**0.984**					**500**

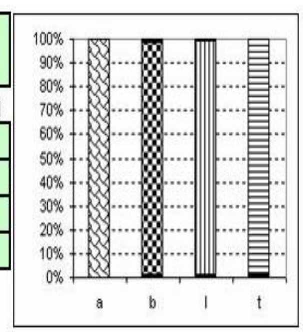

Figure 4.3-b Aggregate results of SOM/Projection

In [12] the reported accuracy for a similar system was 85%, and [3] reported a 93.7% overall accuracy. Since we used a similar approach to tiling the images as [3], it is worth comparing them on a variety of crack types, as in Table 4.1. In addition to obtaining higher overall accuracy, we note that our MLP classifier used far fewer hidden layer nodes which should enhance real-time performance.

Table 4.1. Accuracy Comparison

	Lee and David	Rababaah
Alligator	88.00%	100.00%
Block	88.00%	97.60%
Longitudinal	98.00%	98.40%

5. FUTURE WORK

Given the performance of this system we believe that there is potential for future work. Specifically, we see the following as important extensions leading to deployment:

- To fully automate the system we need to incorporate adaptive thresholding into the process.
- The system should incorporate other subsets of pavement distress such as pot-holes, patching, polished aggregate, etc.
- Real-world assessment images often contain foreign objects such as oil residue, dirt, lane markings, vegetation debris, and other non-pavement artifacts. Algorithms need to be explored to distinguish them from legitimate cracks.

6. SUMMARY AND CONCLUSION

We have presented a comparative study of three approaches to automating pavement crack classification. The system consists of four phases: Image preprocessing, crack detection, crack representation, and crack classification. Two approaches to image representation were explored. The results indicate that the MLP proved the highest accuracy, 98.6%, and that the projection data representation was more effective in all classifiers studied. We believe that this system, in addition to producing competitive accuracy, has the positive attributes of design simplicity and computational efficiency.

7. REFERENCES

[1]. Timm, and McQueen. 2004. *A Study of Manual vs. Automated Pavement Condition Surveys*. Auburn Alabama.
http://www.eng.auburn.edu/users

[2]. Mohajeri and Manning. 1991. ARIA™, An Operating System of Pavement Distress Diagnosis by Image Processing. *Transportation Research Record 1311*, TRB, National Research Council, Washington, D.C., pp. 120-130.

[3]. Lee and "David" Lee. 2003. A Robust Position Invariant Artificial Neural Network for Digital Pavement Crack Analysis. *Technical Report TRB2003-000996*.
http://www.ltrc.lsu.edu

[4]. Wang, Watkins and Kuchikulla. 2002. Digital Distress Survey of Airport Pavement Surface. *Federal Aviation Administration Airport Technology Transfer Conference* 2002.
http://www.airporttech.tc.faa.gov

[5]. US department of Transportation, Federal Highway Adminstration. 2003. *Distress Identification Manual*. FHWA-03-031. Mclean, VA.
http://www.tfhrc.gov

[6]. Chou, O'Neill, and Cheng. 1994. Pavement Distress Classification Using Neural Networks. *IEEE-94, 0-7803-2129-4/94.*

[7]. Haykin. 1994. *Neural Networks A Comprehensive Foundation*.
New Jersey: Mcmillan Publications.

[8]. Greene. *Unsupervised Hierarchical Clustering via a Genetic Algorithm.* University of New Orleans, Computer Science Department.
http://www.uno.edu.

[9]. Kohonen. 1990. The Self-Organizing Map. *Proceedings of the IEEE, vol. 78, No. 9, September 1990.*

[10]. Robert Jay Dilger. 2003. *American Transportation Policy.* West Virginia University, Department of Political Science.
Westport, CT: Praeger Publishers, 2003.

[11]. Fredric M. Ham, Ivica Kostnic. 2001. *Principles of Neurocomputing for Science & Engineering*.
McGraw-Hill publications.

[12]. Chung-Jung Hsu, Chi-Farn Chen, Chau Lee, and Shu-Meng Huang. 2001. Airport Pavement Distress Image Classification Using Moment Invariant Neural Network. *22nd Asian Conference on Remote Sensing and processing (CRISP), Singapore*.