Construction and Building Materials 48 (2013) 853–857

Contents lists available at ScienceDirect

Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Pavement crack classification based on tensor factorization

Offei Amanor Adarkwa, Nii Attoh-Okine *

Department of Civil and Environmental Engineering, University of Delaware, United States

HIGHLIGHTS

- Tensor factorization method of classifying cracks easy to understand.
- Higher level of classification accuracy achieved with larger training image sets.
- To ensure accuracy there must be appreciable variation within and across training sets.

ARTICLE INFO

Article history:
Received 3 May 2013
Accepted 24 July 2013
Available online 24 August 2013

Keywords:
Roads and highways
Maintenance and inspection
Management

ABSTRACT

This paper presents tensor factorization as a means of classifying cracks in pavements. Several crack classification algorithms exist and they are mostly based on other machine learning methods. These may come with their own problems which may be classification errors and long processing times. Tensors are multidimensional arrays. The nature of tensors enables the analysis of the images to be carried out in a 3D space which ensures a more robust and accurate analysis tool. The levels of accuracy obtained after using the algorithm implies that crack classification based on tensor factorization is one that can be successfully employed by state agencies nationwide and around the world which use digital image processing systems as part of their pavement management programs.

Published by Elsevier Ltd.

1. Introduction

Infrastructure systems are necessary for supporting society's functioning as well as economic growth. They are capital intensive assets whose proper functioning is critical to the development of modern societies worldwide. The assets deteriorate over time and as such there is the need for maintenance to ensure they function efficiently during their design life. The most common sign of deterioration in road pavements is cracking. There are various types of cracks namely longitudinal, transverse, alligator and block cracks. Usually, manual surveys are carried out by experienced inspectors who walk along the roadway and note the various types of cracks at different sections of the road. This is a very subjective way of monitoring the condition of the pavement. It is also time consuming and may also pose serious safety risks for the inspectors. In recent times, automated methods which process pavement images for condition monitoring have been developed. Images are taken by cameras attached to specialized vehicles after which processing algorithms are used to determine the types of cracks on the pavement. These algorithms are usually based on neural networks. They are usually associated with problems such as classification errors and long processing times.

This paper presents the basic concepts of tensor factorization and also outlines the use of tensor decomposition in pavement crack classification. The two main types of cracks; longitudinal and transverse cracks are considered in this work.

The paper has six main parts. The first part is the introduction. This is followed by the second part which describes previous work done in the area of pavement crack classification. The third part focuses on tensors and the basic underlying concepts of tensor factorization. The fourth, fifth and sixth parts are on crack classification based on tensors, results and conclusion respectively.

2. Previous works in pavement crack classification

The phenomenon of crack formation is very important in transportation since it is the commonest form of pavement deterioration. These damages cost about $10 billion annually in the US [9]. This explains why the Federal Government and State Departments of Transportation (DOTs) nationwide place much emphasis on monitoring and maintenance of pavements. Since the 1970s, automated methods for studying cracks have been developed [9–11] and upgraded because the manual means are subjective, labor-intensive and time-consuming [14].

In Rababaah's work [9], the accuracy of three different classification algorithms were compared. They consisted of two supervised learning algorithms; Genetic Algorithm (GA) and

* Corresponding author. Tel.: +1 3027382337.
E-mail addresses: adarkwa@udel.edu (O.A. Adarkwa), okine@udel.edu (N. Attoh-Okine).

0950-0618/$ - see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.conbuildmat.2013.07.091

Nomenclature

$\mathcal{A} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$	3-dimensional tensor	$\mathcal{G}$	core array
T	normalized test image	U, V, and W	orthogonal components of decomposition
$\mathcal{A}_k^c$	slice of the outer product of core tensor and loading matrices in modes 1 and 2	α	scalar
		$g(k)$	scalar
C	training set class (transverse or longitudinal)		
K	number of images in training set		

Multi-layer Perceptron (MLP) and one unsupervised learning algorithm called the Self-Organizing Map (SOM). The work was motivated by the fact that existing classification algorithms were computationally expensive and that they were inefficient in terms of processing speeds. Alligator, block, transverse and longitudinal cracks were considered in the work. After preprocessing of images, two feature representation methods were used before they were subjected to the classifiers. The methods used were Hough Transform and the Projection-based approach. The best classification algorithm after the analysis was the MLP using the projection-based representation with a total accuracy of 98.6%. However, this approach is still computationally expensive and involves thresholding which may not provide accurate results occasionally.

Saar [10] also devised a means of classifying cracks. The method had an overall classification accuracy of 95%. It was based on a Neural Network (NN) Approach. Alligator cracks in addition to longitudinal and transverse cracks were considered in this work. The training set consisted of 61 images, with 41 images used each for validation and testing.

Oliveira and Correira [8] also proposed a very impressive system for identifying and classifying pavement crack images. 100% recall and precision values were obtained for the classification algorithm which made use of Bayesian classification techniques. The cracks considered were based on the description of cracks in the Portuguese Distress Catalog. The cracks were divided into three classes namely; longitudinal, transversal and miscellaneous. Normalization is done after the image is subdivided into pixels of size 65×65 in order to reduce the effect that different background illuminations may have on the results. The training set images are selected by choosing first, the images obtained after sorting database images in decreasing order of the longest component length. This is to ensure that the training images all contain cracks. 2D feature spaces are created for each image. For classification, the standard deviations of the row and column coordinates of the detected crack regions are used. There is a bisectrix which divides the feature space into two zones. Points which fall very close to the bisectrix indicate miscellaneous cracks. Points which are closer to the horizontal axis indicate transverse cracks and points closer to the vertical axis indicate longitudinal cracks.

3. Basic concepts in tensor analysis

A tensor is a multiway or multidimensional array [3,6]. This definition suggests that tensors are generalizations of scalars, vectors and matrices [6]. A vector is a one-dimensional tensor and a matrix is known as a two-dimensional tensor. A scalar is a zero-dimensional tensor. Higher-order tensors are those that generally have dimensions or modes greater than 2. The diagram in Fig. 1 shows a typical three-dimensional tensor.

The indices of a tensor with elements, x_{ijk} run from 1 to the capital letter form which means $i = 1, 2, \ldots I, j = 1, 2, \ldots, J$ and $k = 1, 2, \ldots, K$. Tensors have subarrays which are similar to columns and rows in matrices. Three-dimensional tensors have fibers and slices as examples of subarrays [3]. A slice is a two-dimensional fragment of a tensor obtained after allowing the tensor to vary in two modes whilst keeping one mode fixed. There are frontal, horizontal and lateral slices all shown in Fig. 2.

The horizontal slice is obtained by fixing mode i and allowing the other modes to vary. The lateral slice is obtained by fixing the tensor in mode j and allowing the others to vary. Frontal slices are obtained after fixing mode k and allowing modes i and j to vary. Fibers are also one dimensional fragments obtained when the tensor is allowed to vary in only mode. See Fig. 3. Row fibers are obtained when the tensor is fixed in modes i and k and is allowed to vary in mode j. Column fibers are obtained from allowing the tensor to vary in mode i whilst the two other modes are fixed. The tube fibers are obtained when the tensor is fixed in mode i and j and made to vary in mode k.

3.1. Matricization and vectorization of tensors

Matricization is the process of converting a tensor into a matrix for visualizations and computations. It is also known as flattening or unfolding in some fields. It is the rearrangement of entries in a tensor [1]. Matricization is carried out in a specific mode and so a n-mode mode matricization of a tensor will involve aligning all mode-n fibers as columns of a matrix $A_{(n)}$ in a forward cyclic manner. See Fig. 4 for an illustration.

The 3D tensor above has elements A, B, C, D in the first frontal slice and elements e, f, g, h are in the second frontal slice.

$$\text{Mode} - 1 \text{ matricization} = \mathcal{A}_{(1)} = \begin{bmatrix} A & B & e & f \\ C & D & g & h \end{bmatrix} \tag{1}$$

$$\text{Mode} - 2 \text{ matricization} = \mathcal{A}_{(2)} = \begin{bmatrix} A & C & e & g \\ B & D & f & h \end{bmatrix} \tag{2}$$

$$\text{Mode} - 3 \text{ matricization} = \mathcal{A}_{(3)} = \begin{bmatrix} A & B & C & D \\ e & f & g & h \end{bmatrix} \tag{3}$$

Vectorization is also done by arranging successive columns of a matricized tensor below each other to form a vector. For the 3D tensor in Fig. 4, the vectorized version, $\boldsymbol{a} = \begin{bmatrix} A \\ B \\ \vdots \\ h \end{bmatrix}$

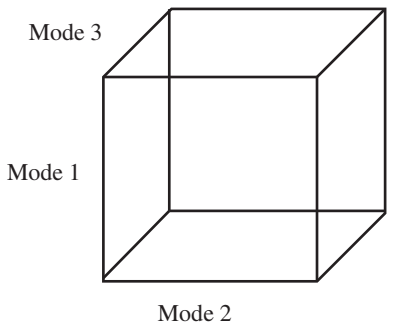

Fig. 1. Typical three-dimensional tensor.

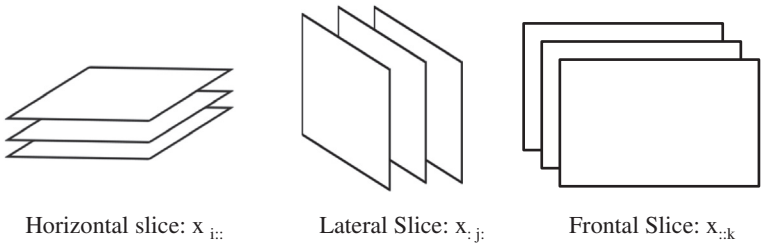

Fig. 2. 3D tensor slices.

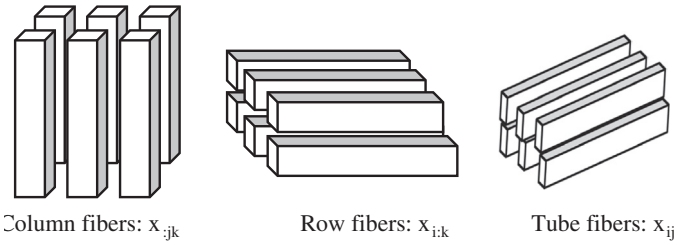

Fig. 3. 3D tensor fibers.

3.2. Basic operations in tensors

3.2.1. Addition of tensors

Similar to matrices, tensors of identical dimensions can be added. Addition of two tensors $\mathcal{A} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$ and $\mathbb{B} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$ can be expressed in element wise form as:

$$a_{ijk} + b_{ijk} = c_{ijk} \tag{4}$$

It involves addition of corresponding elements in the tensors.

3.2.2. Multiplication of tensors

Scalar multiplication of a tensor is such that every individual element of the tensor is multiplied by the scalar shown below as:

$$\alpha\mathcal{A} = \alpha a_{ijk} \text{ for } \mathcal{A} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}} \tag{5}$$

3.2.3. N-mode multiplication

The process of multiplying a tensor by a matrix is carried out in a specific mode. Mode-n multiplication of a tensor $\mathcal{A} \in \mathbb{R}^{\mathbb{I}_1\times\mathbb{I}_2\ldots\mathbb{I}_N}$ by a matrix $U \in \mathbb{R}^{\mathbb{J}\times\mathbb{I}_n}$ denoted by $\mathcal{A}\times_n U$ is of size $I_1 \times I_2 \ldots I_{(n-1)} \times J \times I_{(n+1)} \ldots I_N$. In order to make understanding of the n-mode multiplication easier, it is usually seen as a multiplication of the matricized form of the tensor in the specified mode being premultiplied by the matrix. The mathematical representation is $\mathcal{A}\times_n U = U \times \mathcal{A}_n$.

3.2.4. Inner product

The inner product of two tensors $\mathcal{A} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$ and $\mathcal{B} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$ is denoted by

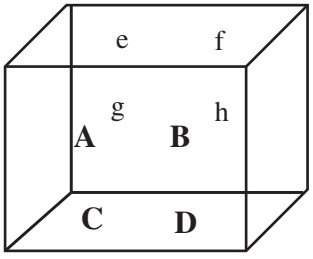

Fig. 4. Illustration of matricization.

$$\mathcal{A}, \mathcal{B} = \sum_{i,j,k} a_{i,j,k} b_{i,j,k}. \tag{6}$$

It involves the multiplication of corresponding elements and the subsequent summing of the products to produce a scalar. There are other tensor products namely Khatri-Rao product, Hadamard product and so on.

3.3. Tensor decomposition

Tensors are decomposed for analysis similar to matrix decomposition [13]. The Singular Value Decomposition (SVD) used in second-order tensors (matrices) is extended to *n*-th order arrays. *N*-th order variations of the matrix SVD results in the Higher-order Singular Value Decomposition (HOSVD) as well as the two main tensor decomposition methods; Canonical Decomposition Parallel Factorization (CANDECOMP/PARAFAC/CP) and Tucker Decomposition. The HOSVD which was used in the analysis is discussed further in this paper.

3.3.1. Higher order singular value decomposition (HOSVD)

Extension of the SVD principles in tensors leads to a method known as the HOSVD [1,4]. It is interesting to note that the SVD explained above can be written in a tensor-like mode-1 and mode-2 multiplication. For a matrix $\boldsymbol{A}$, with U and V orthogonal components, it can be written as

$$\boldsymbol{A} = \Sigma\times_1 U\times_2 V \tag{7}$$

where $\times_1$ and $\times_2$ represent mode-1 multiplication and mode-2 multiplication respectively.

For a 3rd-order tensor $\mathcal{A} \in \mathbb{R}^{\mathbb{I}\times\mathbb{J}\times\mathbb{K}}$ its higher-order decomposition can be written as:

$$\mathcal{A} = \mathcal{G}\times_1 U\times_2 V\times_3 W \tag{8}$$

where U, V and W are the orthogonal singular vectors and $\mathcal{G}$ is the array with the singular values. Variation of the above representation results in the CP and Tucker decompositions. The elements of

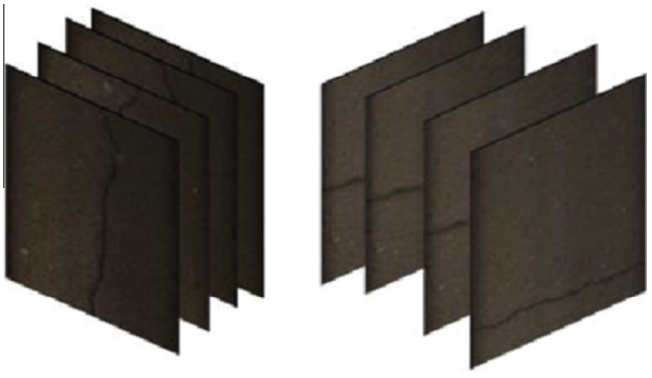

Fig. 5. Longitudinal and transverse training tensors.

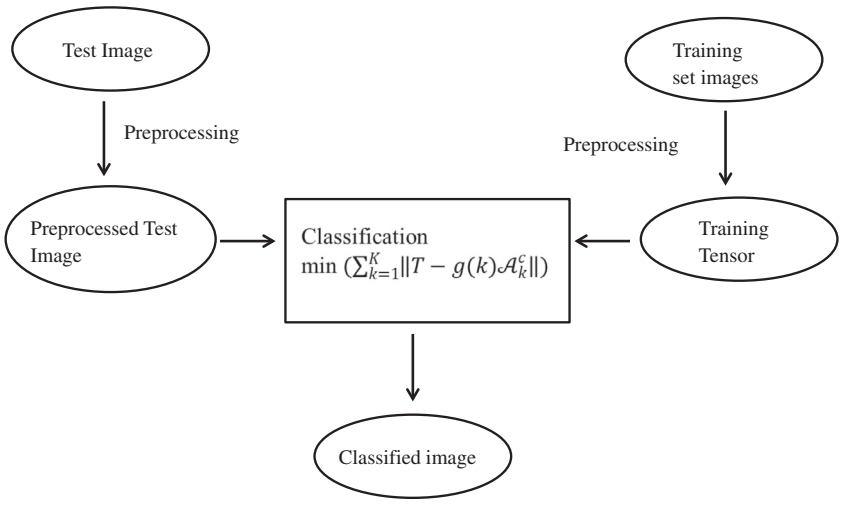

Fig. 6. A flow-chart of the classification process.

Fig. 7. Longitudinal and transverse images from dataset [12].

Table 1
Classification using training tensors of 10 images in each class.

Image	Longitudinal set	Transverse set
1*l*	9.043	9.0601
2*l*	9.0516	9.0608
3*l*	9.0572	9.0701
4*l*	9.0625	9.0746
5*l*	9.0515	9.0594
6*l*	9.0648	9.0721
7*l*	9.0641	9.0725
8*l*	9.0623	9.0678
9*l*	9.043	9.0601
10*l*	9.0525	9.0627
13*l*	9.0572	9.0701
1*t*	9.0601	9.0503
2*t*	9.0663	9.0596
3*t*	9.0718	9.0517
4*t*	9.0582	9.0522
5*t*	9.0677	9.0572
6*t*	9.0734	9.0623
7*t*	9.0742	9.0692
8*t*	9.0718	9.0517
9*t*	9.0601	9.0503
10*t*	9.0663	9.0596
11*t*	9.0658	9.0611

$\mathcal{G}$, are ordered such that the most of the energy is concentrated in the vicinity of $(1,1,1)$ [12].

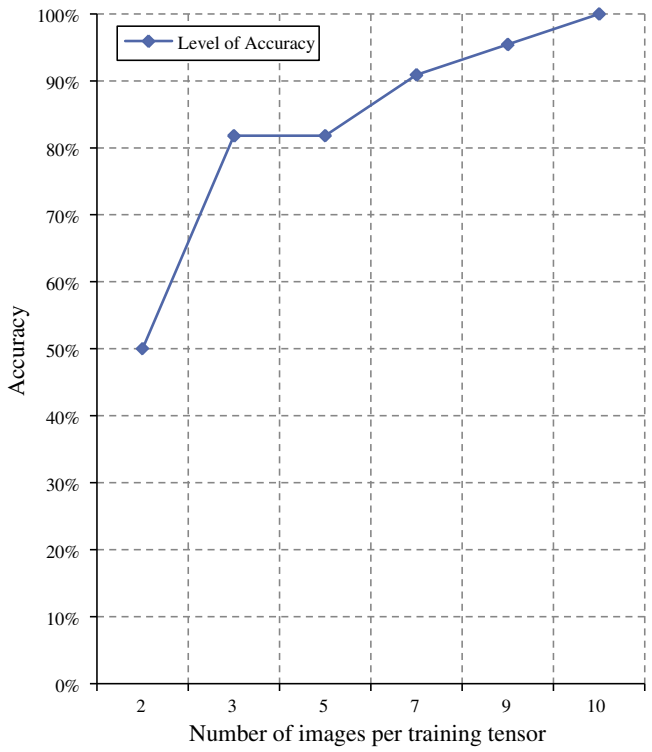

Fig. 8. Graph of number of images in training tensor versus accuracy.

4. Crack classification using tensors

The analysis was carried out in MATLAB using the MATLAB Tensor Toolbox 2.5. There are four major steps involved in the analysis. They are:

- Preprocessing and formation of Training Tensors;
- Preprocessing of test data;
- Higher Order Singular Value Decomposition of Training Tensor; and
- Classification.

Preprocessing involves conversion of the images from RGB into grayscale format. The images are then converted from 3501×2550

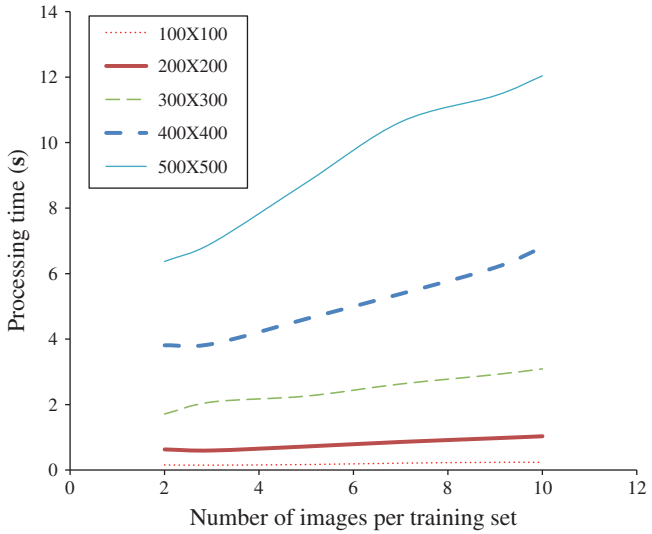

Fig. 9. Comparison of processing times for various image sizes and training sets.

into 100 × 100 size images. The training set of images are then divided into two main groups; transverse and longitudinal cracks. Images in each class or group are stacked one after the other to form the training tensor sets. For the Tensor Toolbox algorithms to function, the 3D arrays obtained after this step must be converted into tensor format in the MATLAB environment. The test image is also preprocessed. The training tensor is then decomposed and the test image is classified using the minimization function below.

$$\sum_{k=1}^{K} \|T - g(k)\mathcal{A}_k^c\| \tag{9}$$

where T is the normalized test image; $\mathcal{A}_k^c$ is the slice of the outer product of core tensor and loading matrices in modes 1 and 2; c is the training set class (transverse and longitudinal); k is the number of images in training set; and

$$g(k) = \text{a scalar} \tag{10}$$

As a result, V will belong to the class c, for which the above function gives a smaller value since that will suggest a higher degree of similarity between V and the images in the training class c. Fig. 5 illustrates the concept of the formation of training tensors. See also Fig. 6 for an illustration of the classification process.

5. Data and results

The pavement images used for the analysis were part of the dataset from Fereidoon Moghadas Nejad and Hamzeh Zakeri [7]. The images were acquired using a pavement image acquisition system (PIAS). 26 images from the larger dataset were used in the analysis. The larger dataset had 1830 images divided into two main groups; with defects and without defects. Fig. 7 shows images from the dataset. They were originally of size 3501 × 2550 .

Using longitudinal and transverse training sets of 10 images each, all the images were classified correctly. Table 1 shows the results obtained with a training set of 10 images.

The images were identified with numbers. The letters l and t after the numbers indicate whether the image contained longitudinal or transverse cracks respectively.

However, the level of accuracy of the algorithm changed with changes in the number of images in the training sets. It is evident that the accuracy does not depend on only the number of images in the training set but also the variability within and across classes. See Fig. 8.

From the graph, the level of accuracy for 3 and 5 images in the training set is similar which implies that the classification accuracy does not only depend on the number of training set images but also the variation within the class. The size of the images also affected the processing time for the algorithm as shown in Fig. 9.

6. Conclusion

In summary, pavement crack classification based on tensor analysis is a useful tool that can be employed in the digital image processing systems of DOTs. Due to its relatively easy and intuitive manner of training, DOTs can train the algorithm to classify the road defects based on their standards and definitions. Several runs of the algorithm proved that a higher level of accuracy was achieved when the training tensors were built with larger datasets, which means more images within the same class. However, it is required that the larger datasets exhibit variation within and across the various classes in order to ensure accurate results. A few of the existing crack classification methods were reviewed and compared with the tensor based method. The tensor decomposition methods of classifying cracks were much easier to understand and implement as compared to other methods which required much more complex input and were difficult to understand the underlying concepts of the approach.

It will be very useful to employ tensor classification to other forms of pavement defects. Parallel processing can also be incorporated to speed up the processing time of algorithm.

Acknowledgments

The authors of this paper would like to thank Yaw Adu- Gyamfi, PhD candidate at the University of Delaware, John Cava, Mandli Communications, and Fereidoon Moghadas Nejad and Hamzeh Zakeri from Amikabir University of Technology, Iran.

References

[1] Elden L. Tensor computations and applications in data mining. Sweden, SIAM AM: Department of Mathematics, Linkoping University; 2008.
[2] Faloutsos C, Kolda TG, Sun J. Mining large time-evolving data using matrix and tensor tools; 2007. <http://www.cs.cmu.edu/~jimeng/papers/ICMLtutorial.pdf> [Time accessed 11.01.12, 11.45 am].
[3] Kiers ALH. Towards a standardized notation and terminology in multiway analysis. Heyman Institute (PA), University of Groningen, Grote Kruisstraat 2/1, NL-9712 TS Groningen. Netherlands, J Chemometr 2000;14:105–22.
[4] Kilmer ME, Martin MCD. Decomposing a tensor. SIAM News 2004;37(9).
[5] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM Rev 2009;51(3):455–500.
[6] Mørup M. Applications of tensor factorizations and decompositions in data mining. John Wiley & Sons, Inc.; 2011. vol. 1.
[7] Nejad FM, Zakeri H. A comparison of multi-resolution methods for detection and isolation of pavement distress. Amirkabir university of technology. Tehran, Iran, Expert Syst Appl 2008;38(2011):2857–72.
[8] Oliveira H, Correira PL. Identifying and retrieving distress images from road pavement surveys. In: 15th IEEE international conference; 2008. p. 57–60.
[9] Rababaah H, Vrajitoru D, James W. Asphalt pavement crack classification: a comparison of GA, MLP and SOM. GECCO. Genetic and evolutionary computation conference late-breaking paper; 2005.
[10] Saar T. Automatic asphalt pavement crack detection and classification using neural networks. Department of electronics, TUT, Ehitajate tee 5, 19086 Tallinn, Estonia. 12th Biennial electronics conference; 2010.Talvik O. Automatic asphalt pavement crack detection and classification using neural networks. Department of transportation, TUT, Ehitajate tee 5, 19086 Tallinn, Estonia. 12th Biennial electronics conference; 2010.
[11] Santhi B, Krishnamurthy G, Siddharth S, Ramakrishnan PK. Automatic detection of cracks in pavements using edge detection operator. J Theor Appl Inf Technol 2012;36(2).
[12] Savas B, Elden L. Handwritten digit classification using higher order singular value decomposition. J Pattern Recogn Soc, Els Ltd., Pattern Recogn 2006;40(2007):993–1003.
[13] Vasilescu MAO, Terzopoulos D. Multilinear analysis of image ensembles: tensorfaces. ECCV'02, Copenhagen, Denmark; 2002.
[14] Xu B, Yaxiong H. Automated pavement cracking rating system: a summary. Center for transportation research the university of Texas at Austin. Project Summary Report 7-4975-S; 2003.