

**PAVEMENT CRACK CLASSIFICATION BASED ON  
TENSOR FACTORIZATION**

by

Offei Amanor Adarkwa

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master in Civil Engineering

Fall 2012

© 2012 Offei Amanor Adarkwa  
All Rights Reserved

**PAVEMENT CRACK CLASSIFICATION BASED ON  
TENSOR FACTORIZATION**

by

Offei Amanor Adarkwa

Approved: \_\_\_\_\_  
Nii Attoh-Okine, Ph.D.  
Professor in charge of thesis on behalf of the Advisory Committee

Approved: \_\_\_\_\_  
Harry W. Shenton III, Ph.D.  
Chair of the Department of Civil and Environmental Engineering

Approved: \_\_\_\_\_  
Babatunde A. Ogunnaike, Ph.D.  
Interim Dean of the College of Engineering

Approved: \_\_\_\_\_  
Charles G. Riordan, Ph.D.  
Vice Provost for Graduate and Professional Education

## **ACKNOWLEDGMENTS**

I thank God for how far He has brought me. I thank Him for His guidance throughout the entire period of my work.

My sincerest gratitude goes to my graduate advisor, Prof. Attoh-Okine who has supported and encouraged me from the very first day. I also wish to thank Yaw Adu-Gyamfi and Leslie Mills, PhD candidates at the University of Delaware, for their help and contributions to my work. I also thank Chris Reoli for her invaluable assistance.

Finally, I thank my family and all loved ones who have been there for me in order for this dream to become a reality.

## TABLE OF CONTENTS

LIST OF FIGURES .....	vi
ABSTRACT.....	vii
Chapter	
1 INTRODUCTION .....	1
1.1 General.....	1
1.2 Objectives of Research .....	3
1.3 Statement of Problem.....	4
1.4 Thesis Organization .....	5
2 BACKGROUND & LITERATURE REVIEW .....	7
2.1 General.....	7
2.2 Crack Classification.....	7
2.3 Tensors.....	10
2.4 Selected Applications of Tensors.....	13
3 TENSOR ANALYSIS .....	22
3.1 Definition .....	22
3.2 General Notation.....	22
3.3 Indices .....	23
3.4 Subarrays of Tensors.....	23
3.5 Matricization & Vectorization of Tensors .....	25
3.6 Vectorization of Tensors.....	26
3.7 Special Tensors .....	27
3.8 Basic Operations in Tensors .....	27
3.8.1 Addition of Tensors .....	27
3.8.2 Multiplication of Tensors.....	28
3.9 Tensor Decomposition.....	30
3.9.1 Higher-Order Singular Value Decomposition (HOSVD).....	31
3.9.2 Tucker Decomposition.....	31
3.9.3 The CP Model.....	33

3.10 Rank & Multilinear Rank of a Tensor .....	34
3.11 Determining the Number of Components in a Model .....	34
3.12 Software .....	35
3.13 Applications .....	36
4 DATA ANALYSIS & RESULTS .....	38
4.1 Source of Data.....	38
4.2 Description.....	39
4.3 Analysis.....	40
4.4 Results.....	44
5 CONCLUSION & RECOMMENDATIONS .....	48
5.1 General.....	48
5.2 Conclusion .....	48
5.3 Recommendations.....	49
REFERENCES .....	50
Appendix	
A CLASSIFICATION RESULTS.....	52
B REPRINT PERMISSION LETTERS .....	58

## LIST OF FIGURES

Figure 1: Laser Road Imaging System, (Mandli Comm. Inc.) .....	2
Figure 2: A typical 3D tensor.....	11
Figure 3: Fibers of a 3D tensor .....	12
Figure 4: HOSVD of a 3D array (Savas et al 2006) .....	17
Figure 5: Training tensor for second algorithm (Savas et al 2006) .....	18
Figure 6: Application of Tensor in Infrastructure Management.....	21
Figure 7: 3rd-order tensor fibers .....	24
Figure 8: 3rd-order tensor slices .....	25
Figure 9: Illustration of matricization using a 3D tensor.....	26
Figure 10: The Tucker Model.....	32
Figure 11: The CP Model .....	33
Figure 12: Pavement Image Acquisition System, PIAS (Nejad et al 2011) .....	38
Figure 13: Original longitudinal and transverse cracks (Nejad et al 2011) .....	39
Figure 14: Steps in preprocessing of images for classification.....	41
Figure 15: Longitudinal and Transverse crack training sets .....	41
Figure 16: Frontal slices from HOSVD of longitudinal tensor of 3 images.....	43
Figure 17: First slices of the core arrays from HOSVD of longitudinal crack tensors....	44
Figure 18: First Slices of the core arrays from HOSVD of transverse crack tensors .....	45
Figure 19: Graph of number of images in training tensors against level of accuracy .....	45
Figure 20: Graph of processing times versus size of images.....	46
Figure 21: Comparison of processing times for various image sizes and training sets ...	47

## **ABSTRACT**

Cracks in pavements allow water to pervade the various layers reducing drastically its ability to perform the primary function of carrying traffic loadings. The various types of cracks that occur on road pavements raise concerns for engineers and infrastructure managers. Identifying the type of crack accurately and efficiently is essential in road maintenance as this will lead to the prescription of cost-effective maintenance and treatment procedures. In the past, various image processing techniques have been applied in the detection and classification of pavement cracks most of which employ machine learning methods. This thesis outlines the importance of tensor analysis and decomposition as an alternative means of pavement crack classification. Tensors are multidimensional arrays and are generalizations of scalars, vectors and matrices. Two main types of cracks; longitudinal and transverse cracking are considered in the study. Due to the nature of tensors, the training set of images used is analyzed in a 3-dimensional space which captures variation across all images and ensures a more robust tensor algorithm for accurate crack classification. The levels of accuracy obtained after using the algorithm implies that crack classification based on tensor decomposition is one that can be successfully employed by state agencies nationwide who use digital image processing systems as part of their pavement management programs.

# **Chapter 1**

## **INTRODUCTION**

### **1.1 General**

Infrastructure systems are necessary for supporting society's functioning as well as economic growth. Typical examples of civil infrastructure systems include roads, bridges, railways, canals and airports. The aforementioned examples 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. This thesis focuses on road pavements.

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 (See Figure 1). The algorithms for classification of crack types are usually based on Neural



Networks (NN). These may come with problems of their own including processing times and classification errors. Work is still being carried out in the area of image processing to produce robust and adaptive means for classifying cracks. This thesis is on a tensor-based approach for classifying pavement cracks.

Tensors are simply multidimensional arrays. The tensor concepts have been applied extensively across various fields including chemometrics and computer vision among others (Mørup 2011).



Figure 1: Laser Road Imaging System, (Mandli Comm. Inc.)

Tensors have also been applied in various domains which have similarities to pavement crack classification such as facial recognition (Vasilescu and Terzopoulos 2002) and handwritten digit recognition (Savas and Elden 2006). Tensor decomposition

and factorization can serve as a robust tool for the classification of crack types in road pavements.

## **1.2 Objectives of Research**

The main objective of the research is to introduce and use tensor decomposition as an advanced data analysis technique in civil infrastructure health monitoring. During pavement health monitoring and evaluation, large amounts of data are collected and analyzed and this makes condition monitoring a data-driven process. With increasing computing technology and capabilities, the amount of pavement data being collected and stored daily is vast and complex. This means that pavement condition monitoring and evaluation requires not only knowledge in civil engineering but also a strong foundation in advanced data analysis. The main objective of this thesis will be realized under the following sub-objectives.

Firstly, the concepts of tensor factorization and decomposition in data analysis will be introduced in order to spur interest in the area of pavement monitoring and evaluation. This will lead to the generation of ideas about how to apply the basic concepts of tensors in the analysis and interpretation of civil infrastructure datasets.

Secondly, tensor decomposition will be used in pavement crack classification. There are several pavement crack classification algorithms available. These are based on methods other than tensor analysis. The aim is to employ tensor decomposition as an alternative pavement crack classifier and assess its accuracy, speed and robustness.

### **1.3 Statement of Problem**

The quality of road pavements in a society is very important since it has a direct influence on development, trade and commerce. The condition of the pavements deteriorates over time and so there is the need for sound maintenance and management practices.

In maintaining the pavement infrastructure, the roads are monitored. This can be achieved through manual inspections by experienced engineers. They walk along the pavement and note all surface distresses and their respective locations and then from the information gathered, remedies are provided to slow down the rate of deterioration and improve the riding quality of the road. Manual inspections are very labor-intensive and time-consuming. It is also a subjective means of condition monitoring. Due to these reasons and the rapid advancement of computer technology, alternative means have been developed. These involve the use of algorithms to detect, classify and determine the geographical locations of cracks from captured images. These are stored and organized in a database which can then be analyzed with relative ease.

The automated method is a very time-efficient method of identifying cracks. The problem with this approach is usually the proper classification of the cracks into say transverse, longitudinal, alligator and block cracks. This is due to the fact that the computers may not have the recognition capabilities most humans have. Current automatic classification methods are based on neural networks and may be computationally expensive. For the methods, which also use thresholding, a major problem is to determine the appropriate threshold value to be used.

Tensors generally refer to arrays of three or more dimensions and they have been applied across various fields and have proven to be very efficient and useful (Kolda and Bader 2009). Tensor applications have been used in handwritten digit classification, chemometrics, facial recognition, web mining, bioinformatics among others. The approach employed in these uses, especially the hand-written digit recognition (Savas and Elden 2006) was used extensively in the crack classification. Due to its multidimensional properties, tensor analysis is able to capture variation of data in all modes simultaneously and so has the ability to produce realistic interpretations of datasets.

#### **1.4 Thesis Organization**

The thesis has five main chapters.

Chapter 1 is the introductory chapter and as such details the objectives of this research, statement of the problem as well as the structure of the thesis in addition to providing a brief introduction to the topic.

Chapter 2 has the background and literature review. Previous works done in the area of crack classification as well as selected work done with tensors in other fields are summarized. This chapter identifies current practices in the pavement health monitoring field. Chapter 3 explains the concept of tensors and tensor decomposition and factorization, software packages available and various application areas.

Chapter 4 describes the dataset that is used in the analysis. It includes the source of the data, general statistical description and the processing that is carried out. The summary of the analysis is presented here and there is a discussion of the results after the

summary. The conclusion and recommendations are outlined in chapter 5 of the document.

## **Chapter 2**

### **BACKGROUND & LITERATURE REVIEW**

#### **2.1 General**

Much work has been carried out in the area of crack detection and 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 (Rababaah et al 2005). This explains why the federal government and State Departments of Transportation (DOTs) nationwide place much emphasis on monitoring and maintenance of pavements. Most studies concentrate on three main types of cracks namely alligator cracking, transverse and longitudinal cracking. Since the 1970s, automated methods for studying cracks have been developed and upgraded because the manual means are subjective, labor-intensive and time-consuming (Xu et al 2003).

#### **2.2 Crack Classification**

In Rababaah's work (Rababaah et al 2005), the accuracy of three different classification algorithms were compared. They consisted of two supervised learning algorithms; Genetic Algorithm (GA) and 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. Images were preprocessed through binarization and median filtering in order to reduce noise. 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 et al also devised a means of classifying cracks (Saar et al, 2010). The method had an overall classification accuracy of 95%. Alligator cracks in addition to longitudinal and transverse cracks were considered in this work. It was noted that image classification comprised of three main steps; preprocessing, processing and classification. The training set consisted of 61 images, with 41 images used each for validation and testing. Initially, the images were of size  $3648 \times 2736$  but were resized to  $12 \times 12$  during the preprocessing phase. Thresholding was not used in the approach since it can affect the efficiency of image segmentation. However, the image was binarized and the pixels of probable crack areas were added up to give the first input for the Neural Network (NN). The elements of the binary matrix formed were summed up column-wise and row-wise. The maximum and median values in the two resulting vectors were also used as the second and third input values for the network. The maximum values gave an indication of

the length of the defects while the median value indicated the overall number of smaller defects. The binary matrix is then convolved by two convolution masks of size  $11 \times 11$ . The two resulting matrices are then summed to form the fourth and fifth input values for the network. The convolution process is done to suppress noise in the image.

Santhi et al proposed an algorithm for automated pavement crack detection (Santhi et al 2012). The method used in the detection of cracks consisted of 5 major steps namely; image acquisition, image preprocessing, image enhancement, image restoration and image segmentation. The restoration and segmentation phases involved using filters and edge-based methods for the detection of cracks. The first two steps after image acquisition were used in this thesis. They were the image pre-processing step which involved the resizing of the images and image enhancement which refers to gray scale image conversion.

Oliveira et al (2008) 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  $65 \times 65$  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.

Normalization is a key pre-processing step that ensures that the effects of non-uniform lighting are reduced. This produces accurate identification and classification results.

### **2.3 Tensors**

A tensor is a multiway or multidimensional array. This definition suggests that tensors are generalizations of scalars, vectors and matrices (Mørup 2011). Higher-order tensors are those that have  $N$  number of modes with  $N > 2$ . A typical three-dimensional tensor has elements  $x_{ijk}$  similar to a matrix with elements  $x_{ij}$  since matrices vary in only two directions. Early suggestions for multiway analysis started in the 1960s. Tucker came out with the Tucker decomposition model and Carroll and Chang and Harshman also proposed the second popular tensor decomposition model known as the CANDECOMP/PARAFAC (Canonical Decomposition/Parallel Factors). Tensors are now being applied across various fields namely psychology, chemometrics, signal processing and computer vision due to the fact that the multiway analysis takes into account the natural multidimensional structure of the data. Modern computers are also

capable of performing complex tensor computations which would have been very difficult a few years ago.

To eliminate the confusion in this area of tensors, standardized notations have been devised (Kiers 2000). Matrices are generally denoted by bold-face capital letters such as  $\mathbf{X}$  and tensors are denoted by bold-face underlined capital letters such as  $\underline{\mathbf{X}}$ . However, in most literature, tensors are also denoted by calligraphic letters, for example  $\mathcal{A}$ . The indices of a tensor with elements,  $x_{ijk}$  run from 1 to the capital format which means  $i = 1, 2, \dots, I, j = 1, 2, \dots, J$  and  $k = 1, 2, \dots, K$ . Alternatively, the indices can be  $i_1, i_2, \dots, i_N$  which means they will run from 1 to  $I_1, I_2, \dots, I_N$ . A typical three-way or three-dimensional array with the three orthogonal axes or modes is shown in figure 2. The three modes or ways are usually identified as modes  $i, j$  and  $k$  or modes 1, 2 and 3. For an  $n$ -dimensional tensor, there are subarrays which are similar to columns and rows in matrices. In tensors, there are slices, fibers and other subarrays which do not have intuitive names but can only be defined by the MATLAB notation for subarrays. Considering a three-way tensor, there are three fibers namely, vertical, horizontal and tube fibers. For higher-order tensors, a general definition of a fiber is a 1-dimensional fragment obtained by allowing the tensor to vary in only one mode.

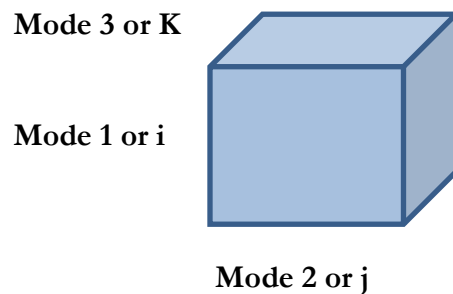


Figure 2: A typical 3D tensor

Row, column and tube fibers in the three-dimensional tensor can be obtained by fixing the other two modes and allowing the tensor to vary in the  $j$ ,  $i$  and  $k$  modes respectively.

This is shown in figure 3.

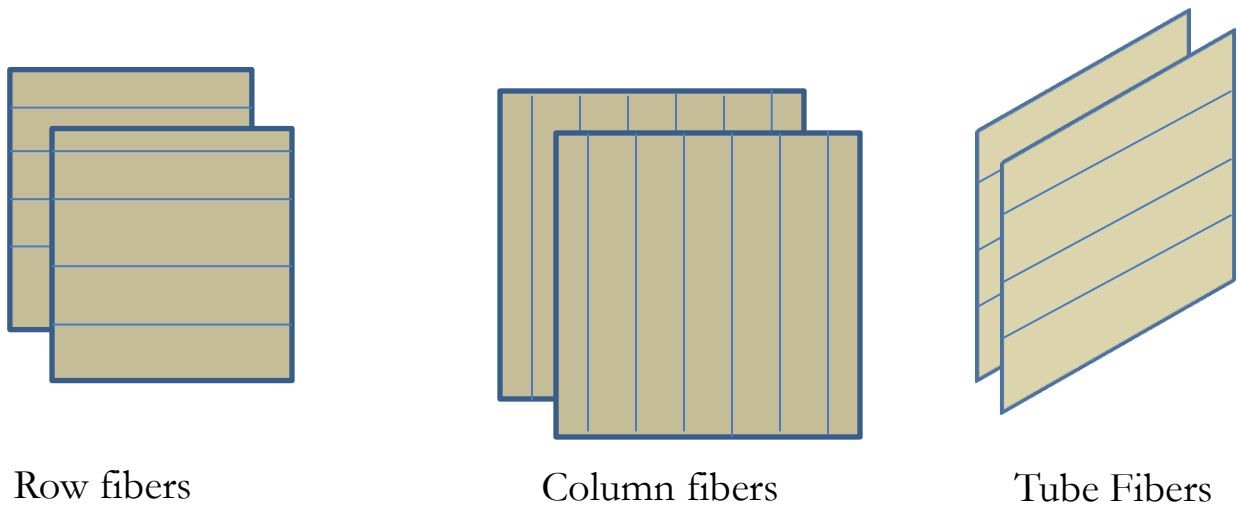


Figure 3: Fibers of a 3D tensor

Tensor slices are obtained by allowing the tensor to vary in two modes resulting in frontal, horizontal and longitudinal slices for a three-dimensional tensor.

Matricization is a process of transforming a tensor into a matrix. This process is the same as collecting all vectors in a particular mode. In chemometrics, the process of matricization is known as ‘unfolding’. The notations are  $\mathbf{X}_1$ ,  $\mathbf{X}_2$  and  $\mathbf{X}_3$  for matricization in modes 1, 2 and 3 respectively. N-mode matricizations are related to each other by cyclic permutations of the modes.

Vectorization involves transforming a tensor into a vector. In matrices, vectorization is done by stacking the columns of the matrix on each other to form one vector. In higher-order tensors, vectorization is achieved by vectorizing the matricized version of the tensor. Mathematically, they can be represented as:

$\mathbf{u} = \text{Vec}(\mathbf{U})$  –vectorization of matrix  $\mathbf{U}$

$\mathbf{x} = \text{Vec}(\mathbf{X}_i)$  –vectorization of matricized tensor in mode  $i$

A superdiagonal tensor is one with all elements zero except the leading diagonal. The unit superdiagonal tensor has the elements on the leading diagonal being ones but it must be noted that it does not perform the same function as the identity matrix in linear algebra.

A cubical tensor is one in which all the modes are of the same size such that  $\chi \in \mathbb{R}^{I \times I \times \dots \times I}$ . A cubical tensor is known as a symmetric tensor if it has its elements remaining the same when the indices are permuted (Kolda et al 2009).

## 2.4 Selected Applications of Tensors

The CP decomposition was first applied in chemometrics (Kolda et al 2009). Chemometrics refers to the study of chemical systems using data driven approaches. PARAFAC and Tucker 3 decompositions have also been used to study the behavior of soils irrigated with wastewater (Singh et al 2005). The 3D tensor analyzed had modes representing the soil sites, soil variables and depths at which the variables are measured in the soil. There were 24 variables some of which included pH, electrical conductivity, redox potential, salinity, organic carbon, and organic matter among others. Preprocessing

of the dataset which involved centering across the first mode and scaling with the second mode were done to ensure all modes had equal levels of influence on the model without compromising the variation in the individual modes. Using the loadings and score plots from the various models, it was concluded that heavy metals were mainly confined in the upper layers of the soils which meant that this can be taken up by crops planted in the soils. It is also worth noting that a 2D analysis, specifically Principal Component Analysis (PCA) was done and compared with the PARAFAC and Tucker 3 analysis. The 3D models were superior to the 2D models in explaining the behavior patterns of the soils.

Higher-order singular value decomposition (HOSVD) or n-mode decomposition has also been used as a mathematical framework for a more efficient facial recognition algorithm (Vasilescu et al 2002). Images are formed by a combination of several factors such as illumination and scene structure. Humans have much more robust recognition capabilities which enable them to identify faces with a lot of variance in the image contributing factors. However, this is not the case in computer vision whereby the system has difficulty in recognizing faces when several factors are varied. This is so because PCA and Independent Component Analysis (ICA) are only able to deal with variation in a single mode and as such are only accurate when person identity is the only factor being varied.

Consider a matrix  $D$ , with left and right matrices  $U_1$  and  $U_2$ . The SVD of  $D$  is

$$D = U_1 \Sigma U_2^T \tag{1}$$

where  $U_1 \in \mathbb{R}^{I_1 \times J_1}$  is the column space,  $\Sigma \in \mathbb{R}^{J_1 \times J_2}$  is a singular value matrix and  $U_2 \in \mathbb{R}^{I_2 \times J_2}$  is the row space.

The third-order extension is as follows:

$$\mathcal{D} = \Sigma \times_1 U_1 \times_2 U_2 \times_3 U_3 \quad (2)$$

where  $U_1$ ,  $U_2$  and  $U_3$  are the mode matrices and  $\Sigma$  is the core tensor.

The core tensor  $\Sigma$  in this case is not diagonal; rather it shows the interaction between the mode matrices  $U_n$ , for  $n=1, \dots, n$ .  $U_n$  contains orthonormal vectors that span the column space of the matrix  $D_{(n)}$  which comes from the matricizing process. As such, the algorithm for the n-mode SVD can be summarized in two main steps; first by computing the matrix  $U_n$  from the SVD of  $D_{(n)}$  which is the n-mode matricized form of  $\mathcal{D}$  and second by computing the core tensor,  $Z$  which can be obtained from the product the tensor  $\mathcal{D}$  and the transpose of the left matrices found in the first step.

$$Z = \mathcal{D} \times_1 U_1^T \times_2 U_2^T \dots \times_n U_n^T \quad (3)$$

Since they are orthogonal matrices, their transpose is equal to their inverse and so the product above yields the core tensor  $Z$ . A portion of the Weizmann face database was used for the analysis in this paper. The data consisted of 28 individuals in 5 different poses, 3 expressions and 3 illuminations. The multilinear analysis is able to represent each individual by a vector of dimension 28. In each column, the first eigenvector represents the average person and the remaining ones capture variability across the people. The eigenvector in any row has the same role in its respective column and so this is the reason why images of the individuals under various conditions are projected to the same coefficient vector.

Handwritten digit classification has also been done using HOSVD (Savas et al, 2006). Images can be considered as two-dimensional arrays but when they are placed together they form a three-dimensional array. The authors of the paper asserted the fact that it may be more beneficial to work with data in its natural multidimensional form. Data used was from the US Postal Service Database of handwritten digits. In this paper, two classification methods both based on HOSVD were compared.

In Singular Value Decomposition (SVD), a matrix  $F$  can be decomposed into constituent parts and written as:

$$F=U\Sigma V^T \quad (4)$$

where  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$  and  $\Sigma$  is an  $(m \times n)$  diagonal matrix with non-negative entries which are ordered from the largest to the smallest value. Columns of  $U$  are called left singular vectors and columns of  $V$  are called right singular vectors.  $U$  and  $V$  are orthogonal. The values in  $\Sigma$  are called singular values. Ideally, the SVD can be written as an n-mode product:

$$F = \Sigma \times_1 U \times_2 V \quad (5)$$

Thus, HOSVD for a 3<sup>rd</sup> order tensor  $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$  can be written as:

$$\mathcal{A} = \mathcal{G} \times_1 U \times_2 V \times_3 W \quad (6)$$

$U$ ,  $V$  and  $W$  are orthogonal matrices and  $\mathcal{G}$  has its elements ordered such that the energy of the core tensor is located around the  $(1, 1, 1)$  vicinity.

In matrices, compression is done by low rank approximation:

$$F = U\Sigma V^T \approx U_k \Sigma_k V_k^T \quad (7)$$

where  $\Sigma_k = \text{diag}(\sigma_1, \dots, \sigma_k)$  and  $U_k$  and  $V_k$  are the first  $k$  columns of  $U$  and  $V$  respectively.

When extended into 3-way arrays:

$$\mathbb{R}^{I \times J \times K} \ni \mathcal{A} = \mathcal{G} \times_1 U \times_2 V \times_3 W \approx \tilde{\mathcal{G}} \times_1 \tilde{U} \times_2 \tilde{V} \times_3 \tilde{W} \quad (8)$$

where  $\tilde{\mathcal{G}} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ ,  $\tilde{U} \in \mathbb{R}^{I \times k_1}$ ,  $\tilde{V} \in \mathbb{R}^{J \times k_2}$  and  $\tilde{W} \in \mathbb{R}^{K \times k_3}$ .

A small approximation error means that the part of the core tensor that is omitted is small and insignificant.

The third order tensor can be written as:

$$\mathcal{A} = \sum_{v=1}^K A_v \times_3 w_v \quad (9)$$

This is illustrated in figure 4 below.

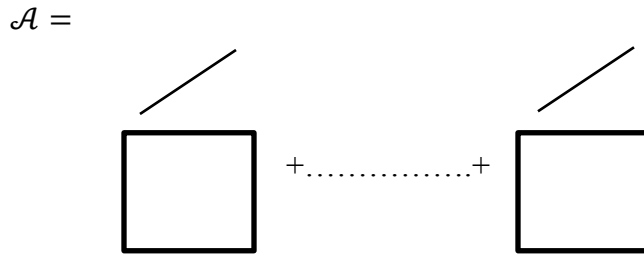


Figure 4: HOSVD of a 3D array (Savas et al 2006)

where  $A_v = \mathcal{G}(:, :, v) \times_1 U \times_2 V$

$\times_3$  is an outer product between  $A_v$  and  $w_v$  to form the tensor  $\mathcal{A}$ .



The images originally are  $16 \times 16$  and preprocessing is done using a blurring technique to form  $20 \times 20$  pixels.

In the first algorithm, basis matrices which span the dominant subspace of each class of digits are computed and are used in describing unknown digits. Tensors are formed for each class of digits. The tensor is decomposed into a finite sum which is truncated to obtain a dominant  $k$ -dimensional subspace for each digit cluster.

$$\mathcal{A}^\mu = \sum_{v=1}^k \mathcal{A}_v^\mu \times_3 w_v^\mu \quad (10)$$

Where  $\mu$  is the class of digits (0, 1... 9). After finding the basis matrices the classification comes down to a least squares problem.

$$\min_{\mu} = \left\| D - \sum_{v=1}^k \alpha_v^\mu A_v^\mu \right\| \quad (11)$$

where  $\alpha_v^\mu$  are scalars to be determined and  $D$  is the normalized original image.

The second algorithm involves compression of the training set before classification is done. It is much more efficient than the first algorithm because unknown digits are projected once. The training tensor is quite different from what exists in algorithm one. In this algorithm, the training tensor is shown in figure 5. Each slice represents digits for a particular class in the vectorized form.

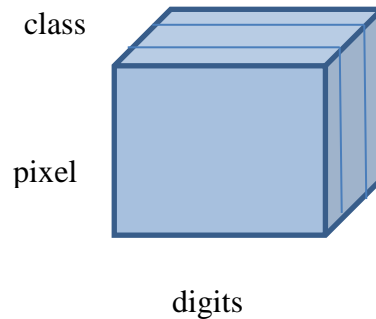


Figure 5: Training tensor for second algorithm (Savas et al 2006)

The decomposition of tensor  $D$  with algorithm two is as follows:

$$D = \mathcal{G} \times_1 U \times_2 V \times_3 W \approx \mathcal{F} \times_1 \tilde{U} \times_2 \tilde{V} \quad (12)$$

This implies that the core tensor,  $F$  can be calculated from the product of the original tensor,  $D$  and the transpose of the approximate loading matrices from  $U$  and  $V$ .

$$D_p = \mathcal{G} \times_1 U_p^T = \mathcal{F} \times_2 V_q \quad (13)$$

The slices of  $F$  contain the basis vectors for the different classes of digits. The  $F$  slices are decomposed using SVD and then the basis matrices  $B$  is used in the least squares equation.

$$\mathcal{F} = (B(B)^{\perp})\Sigma(Q)^T \quad (14)$$

The least squares problem is shown below:

$$\min \|d_p - Bx\| \quad (15)$$

where  $d_p = U_p^T d$ , low dimensional representation of the test image  $d$

$B$  is the basis matrix from the SVD of  $F$  and  $x = B d_p$ .

Analysis of the Enron Email corpus was also done based on tensor factorization (Bader et al 2007). Discussions over time were tracked using PARAFAC and nonnegative tensor factorization. The tensor that was decomposed was of the form  $m \times n \times p$ , where  $m$  represents the terms,  $n$  represents author and  $p$  represents time which was in months for one analysis and days for another analysis. The data used was made up of 53,733 messages sent over a period of 12 months in 2001.

The nonnegative tensor factorization method preserves natural nonnegativity and prevents subtractive basis vector interactions which sometimes occur in the Principal

Component Analysis (PCA). The algorithms used were written in MATLAB using the Tensor Toolbox. With the PARAFAC, a 25 component decomposition was carried out. One iteration took about 22.5s and in all 27 iterations were required to satisfy a tolerance of  $10^{-4}$  in change of fit for the term-author-month array. For the term-author-day array, the rank of the decomposition was 25 with 13 iterations satisfying a tolerance of  $10^{-4}$ . Under the NTF, most of the runs required less than 50 iterations in the monthly data whilst the daily data required about 17 iterations.

The two methods were able to identify the discussions and how each triad,  $\{A_j, B_j, C_j\}$  varied with time, either in days or months. However, the NTF was able to identify distinctly eight topics as opposed to the six topics identified by the PARAFAC for the term-author-month dataset. The determination of an optimal tensor rank may require more research in the future.

Several tools exist in MATLAB that can be used for manipulation and analysis of tensors. These are usually in the form of external toolboxes which have algorithms that can make multilinear analysis possible. They are the N-way Toolbox, Tensor Toolbox, PLS Toolbox and CuBatch. The N-way Toolbox can perform analysis with orthogonality and nonnegativity constraints. CuBatch is built on the N-way Toolbox and it provides an interface which can give results in the form of graphs making the analysis more intuitive (Gourvenec et al 2005).

Tensor analysis can be a very important tool in pavement distress analysis and other infrastructure systems. The examples of applications mentioned in the literature can be modified and used in infrastructure data analysis. Figure 6 shows a possible

application in infrastructure management which involves the collection and interpretation of large amounts of data.

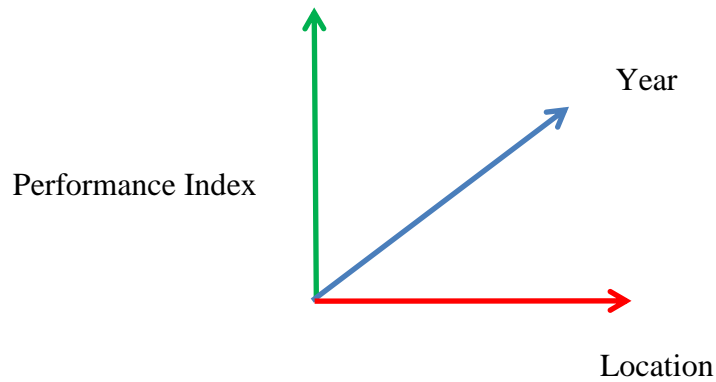


Figure 6: Application of Tensor in Infrastructure Management

The array in figure 6 can be used in studying the general behavior of an infrastructure system in which the performance indexes of crucial parts of the infrastructure are measured at various locations over years. This type of data is inherently multidimensional and so a tensor-based analysis will be able to capture and model the behavior of the system accurately.

## Chapter 3

### TENSOR ANALYSIS

#### 3.1 Definition

A tensor is a multidimensional array. In other words, tensors refer to N-th order arrays. This is not to be confused with tensors in physics and engineering; typical examples of such being stress tensors (Kolda et al 2009). Tensors can also be considered as a generalization of vectors, scalars and matrices. A zero-order tensor refers to a scalar, a first-order tensor is a vector and a second-order tensor is a matrix. Tensor analysis methods are increasingly being used in various fields since data that may arise from observations may be inherently multidimensional. As such, a multidimensional analysis will be the most appropriate means of interpreting the data. The techniques used in tensor analysis are able to capture the variation of the complex structure of the data that would have been lost in a two-way analysis (Mørup 2011).

#### 3.2 General Notation

The order of a tensor refers to its number of dimensions or modes. This definition forms the basis for the idea that matrices, vectors and scalars are all forms of tensors. A scalar is represented by a lowercase letter. Vectors are represented by boldface lowercase letters. Matrices are represented by boldface capital letters whilst higher-order tensors,

referring to tensors of dimensions greater than 2, are represented by boldface Euler script letters (Kolda et al 2009). Higher-order tensors may also be denoted by boldface underlined capital letters (Kiers 2000). Mathematically,

$y \in \mathbb{R}$  represents a Scalar;

$Y \in \mathbb{R}^n$  represents a Vector;

$\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2}$  represents a Matrix; and

$\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$  represents a n-dimensional tensor.

### 3.3 Indices

In a typical third-order tensor of modes  $i$  or  $1$ ,  $j$  or  $2$  and mode  $3$  or  $k$ , an element within the tensor is denoted as  $a_{ijk}$  which defines the exact position of the element in the tensor and is analogous to matrix and vector representations of elements;  $a_{ij}$  for matrix of modes  $i$  and  $j$ , and  $a_i$  for vector of mode  $i$ .

### 3.4 Subarrays of Tensors

Slices and fibers are subarrays of tensors which require understanding in tensor analysis. The fundamental concepts of slices and fibers are used in computations involving tensors. The subarrays are obtained when one or more modes are fixed in a tensor.

Fibers refer to 1-dimensional fragments of tensors obtained by fixing every index in the tensor except one. In a third-order tensor, this implies fixing two modes and allowing

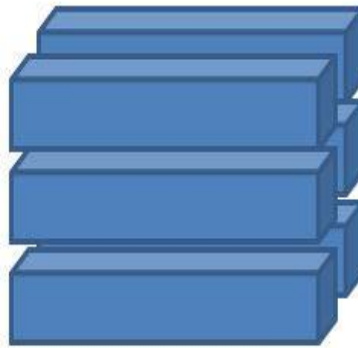
only one mode to vary. They are the higher-order analogues columns and rows in matrices. A column fiber is obtained by fixing modes  $j$  and  $k$ . A row fiber is obtained by fixing modes  $i$  and  $k$ , and a tube is obtained by fixing modes  $i$  and  $j$ .

Slices are 2-dimensional fragments of tensors obtained by fixing all indices except two. In a 3D tensor, a frontal slice is obtained when the  $k$ -th mode is fixed and the  $i$ -th and  $j$ -th mode are allowed to vary. A lateral slice is obtained by fixing the  $j$ -th mode and a horizontal slice is obtained by fixing mode  $i$ .

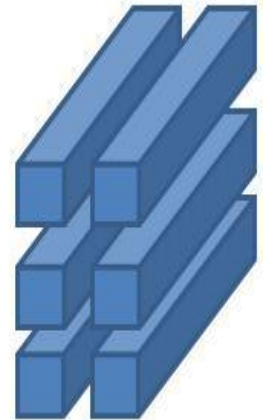
The tensor fibers are illustrated in figure 7.



Column fibers:  $x_{:,jk}$



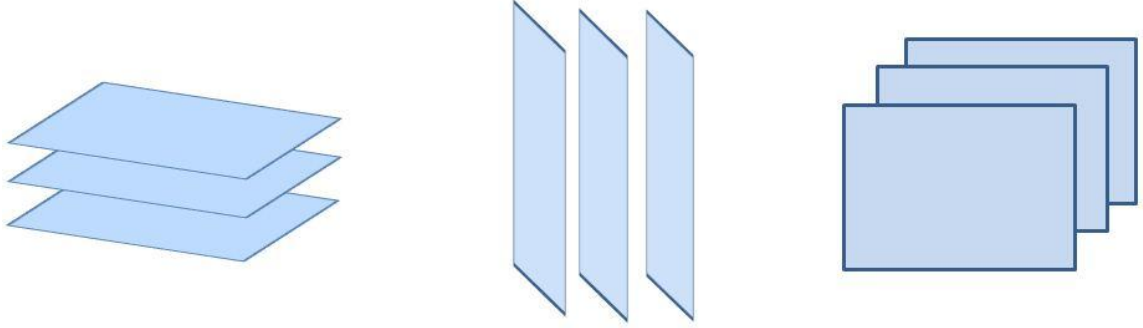
Row fibers:  $x_{i,:k}$



Tubes:  $x_{ij,:}$

Figure 7: 3rd-order tensor fibers

Tensor Slices are also shown in figure 8.



Horizontal Slices:  $x_{i::}$

Lateral Slices:  $x_{:j}$

Frontal Slices:  $x_{::k}$

Figure 8: 3rd-order tensor slices

### 3.5 Matricization & Vectorization of Tensors

Matricization of a tensor is the process by which a tensor is converted into a matrix for visualization and computations. Also known as unfolding or flattening, it is a mode-specific operation in which the  $n$ -mode matricization of a tensor  $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$  means that all  $n$ -mode fibers of  $\mathcal{A}$  are aligned as columns of the matrix  $\mathcal{A}_{(n)}$  in a forward cyclic manner (Savas et al 2006).

Mathematically,

$$\mathbb{R}^{I \times JK} \ni \mathcal{A}_{(1)}: a_{ijk} = a_{iv}^{(1)}, v = j + (k - 1)K \quad (16)$$

$$\mathbb{R}^{J \times IK} \ni \mathcal{A}_{(2)}: a_{ijk} = a_{jv}^{(2)}, v = k + (i - 1)I \quad (17)$$

$$\mathbb{R}^{K \times IJ} \ni \mathcal{A}_{(3)}: a_{ijk} = a_{kv}^{(3)}, v = i + (j - 1)J \quad (18)$$



This is further illustrated in figure 9.

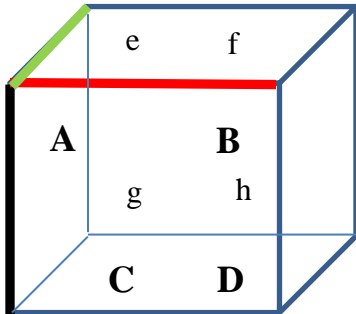


Figure 9: Illustration of matricization using a 3D tensor

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.

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

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

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

### 3.6 Vectorization of Tensors

A tensor can also be converted into a vector. In matrices, vectorization implies arranging successive columns of a matrix below each other in a single vector. A similar process exists in tensors whereby the matricized version of the tensor is vectorized. For

the 3D tensor in figure 9, the vectorized version,  $\mathbf{a} = \begin{bmatrix} A \\ B \\ \vdots \\ h \end{bmatrix}$ .

### 3.7 Special Tensors

Cubical tensors refer to tensors with modes of identical size. Mathematically, a tensor  $\mathcal{A}$  is cubical if  $\mathcal{A} \in \mathbb{R}^{I \times I \times \dots \times I}$ . Symmetric tensors are tensors that have the same elements regardless of the permutation of its indices which means that  $\mathcal{A}_{ijk} = \mathcal{A}_{jki} = \mathcal{A}_{kij} = \mathcal{A}_{kji} = \mathcal{A}_{jik} = \mathcal{A}_{ikj}$ .

A partially symmetric tensor is symmetric in two or more modes. Considering, a typical 3D tensor which is partially symmetric in its frontal slices,  $\mathcal{A}_k = \mathcal{A}_k^T$ .

A diagonal tensor is one with nonzero elements along its diagonal which means that all other elements are zero. Mathematically,  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is diagonal if  $a_{i_1 \times \dots \times i_N} \neq 0$ , if  $i_1 = i_2 = \dots = i_N$

A unit superdiagonal tensor is a tensor with ones as the only nonzero elements in its leading diagonal. It should not be treated as an identity since it does not perform the same function as an identity matrix.

### 3.8 Basic Operations in Tensors

#### 3.8.1 Addition of Tensors

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

$$a_{ijk} + b_{ijk} = c_{ijk} \quad (19)$$

It involves addition of corresponding elements in the tensors.

### 3.8.2 Multiplication of Tensors

Multiplication in tensors can be done in several ways. A tensor can be multiplied by another tensor, matrix, scalar or a vector.

#### *Scalar Multiplication*

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

$$\alpha \mathcal{A} = \alpha a_{ijk} \text{ for } \mathcal{A} \in \mathbb{R}^{I \times J \times K} \quad (20)$$

#### *N-mode multiplication*

The process of multiplying a tensor by a matrix or vector is carried out in a specific mode. Mode-n multiplication of a tensor  $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  by a matrix  $U \in \mathbb{R}^{J \times I_n}$  denoted by  $\mathcal{A} \times_n U$  is of size  $I_1 \times I_2 \times \dots \times I_{(n-1)} \times J \times I_{(n+1)} \times \dots \times I_N$ . 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  $\mathcal{A} \times_n U = U \times \mathcal{A}_n$ .

#### *Inner product*

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

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i,j,k} a_{i,j,k} b_{i,j,k}. \quad (21)$$

It involves the multiplication of corresponding elements and the subsequent summing of the products to produce a scalar.

This implies that the Frobenius norm of a tensor will be:

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} \quad (22)$$

### *Kronecker Product*

The Kronecker Product for  $A \in \mathbb{R}^{I \times J}$  and  $B \in \mathbb{R}^{K \times L}$  denoted by

$$A \otimes B = C^{IK \times JL};$$

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1J}B \\ \dots & \dots & \dots \\ a_{I1}B & \dots & a_{IJ}B \end{bmatrix}$$

### *Khatri-Rao Product*

This is also known as the columnwise Kronecker product. The two matrices must have the same number of columns before this multiplication is possible. For

$$A \in \mathbb{R}^{I \times J} \text{ and } B \in \mathbb{R}^{K \times J},$$

$$A \odot B = C^{IK \times J} = (a_1 \odot b_1 | \dots | a_J \odot b_J) \quad (23)$$

### *Hadamard Product*

This is also known as the elementwise product and it can be performed with matrices of the same order. For matrices  $A \in \mathbb{R}^{I \times J}$  and  $B \in \mathbb{R}^{I \times J}$ , the Hadamard product is

$$A * B = \begin{bmatrix} a_{11}b_{11} & \dots & a_{1J}b_{1J} \\ \dots & \dots & \dots \\ a_{I1}b_{I1} & \dots & a_{IJ}b_{IJ} \end{bmatrix}.$$

The following are some properties based on the various products discussed (Kolda et al 2009):

$$(A \otimes B)(C \otimes D) = AC \otimes BD \quad (24)$$

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger \quad (25)$$

$$(A \otimes B)^\dagger = ((A^T A) * (B^T B))^\dagger (A \odot B)^T \quad (26)$$

$$A \odot B \odot C = (A \odot B) \odot C = A \odot (B \odot C) \quad (27)$$

where A, B, C and D are matrices and  $A^\dagger$  represents the Moore-Penrose Pseudoinverse.

### 3.9 Tensor Decomposition

Tensors are decomposed for analysis similar to matrix decomposition. 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 SVD of a matrix is given by  $A = U\Sigma V^T$ , written as:

$$\mathbf{A} = \sum_{i=1}^R \sigma_i (\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}) \quad (28)$$

where  $\mathbf{u}^{(i)}$  and  $\mathbf{v}^{(i)}$  represent the  $i$ th columns of U, left singular matrix and V, right singular matrix respectively,  $\sigma_i$  represents the numbers on the diagonal also known as the singular values. R represents the rank of the matrix.

### 3.9.1 Higher-Order Singular Value Decomposition (HOSVD)

In tensors, application of the SVD principles leads to a method known as the HOSVD. 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  $\mathbf{A}$ , with  $U$  and  $V$  orthogonal components, it can be written as

$$\mathbf{A} = \Sigma \times_1 U \times_2 V \quad (29)$$

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

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

$$\mathcal{A} = \mathcal{G} \times_1 U \times_2 V \times_3 W \quad (30)$$

where  $U$ ,  $V$  and  $W$  are the orthogonal singular vectors and  $\mathcal{G}$  is the array with the singular values. Variations of the above representation results in the CP and Tucker decompositions. The elements of  $\mathcal{G}$  are ordered such that the most of the energy is concentrated in the vicinity of (1, 1, 1) (Savas et al 2006).

### 3.9.2 Tucker Decomposition

In the Tucker Decomposition, the tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  is decomposed in a similar manner as in the SVD. However, there are no orthogonality constraints on the resulting matrices and so the solution may have matrices which are not orthogonal. The core matrix  $\mathcal{G}$ , in this case, accounts for all the interactions between components of a particular mode. The mathematical representation of the Tucker model is below:

$$\mathcal{X}^{I \times J \times K} \approx \sum_{lmn} G_{l,m,n} A_l^I \circ B_m^J \circ C_n^K \quad (31)$$

where  $G$  is the core array and  $A$ ,  $B$  and  $C$  are loading matrices.

The model decomposes the tensor into mode specific loading matrices in addition to a core array. See figure 10 below.

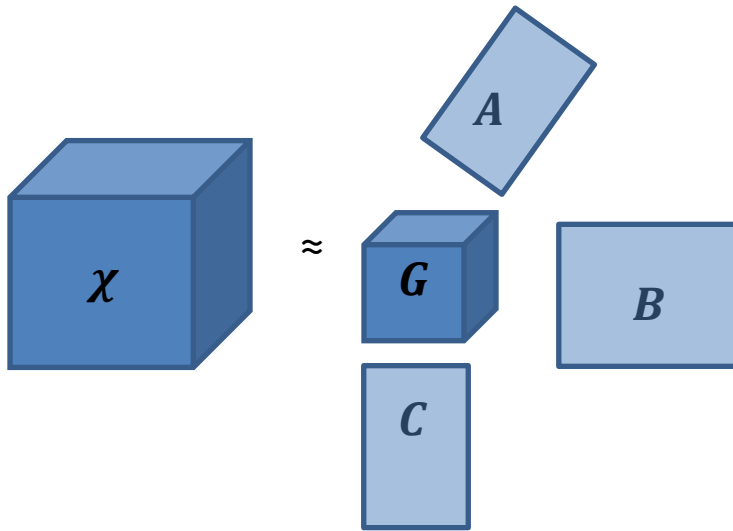


Figure 10: The Tucker Model

### *Estimating the Tucker Model*

This can be done by updating elements of each mode using an Alternating Least Squares (ALS) approach. The loading matrices for the model above can be calculated as:

$$A \leftarrow X_{(1)}(G_{(1)}((C \otimes B)^T)^\dagger) \quad (32)$$

$$B \leftarrow X_{(2)}(G_{(2)}(C \otimes A)^{T\dagger}) \quad (33)$$

$$C \leftarrow X_{(3)}(G_{(3)}(B \otimes C)^{T\dagger}) \quad (34)$$

The core array is determined by the convergence of this function

$$\mathcal{G} \leftarrow \mathcal{X} \times_1 A^\dagger \times_2 B^\dagger \times_3 C^\dagger \quad (35)$$

### 3.9.3 The CP Model

This is another variation of the HOSVD. In this decomposition, the core array is a cubical array and has nonzero elements in its leading diagonal. The CP model can be written as:

$$\mathcal{X}^{I \times J \times K} \approx \mathcal{D}^{D \times D \times D} \times_1 A^{I \times D} \times_2 B^{J \times D} \times_3 C^{K \times D} \quad (36)$$

where  $\mathcal{D}$  is the core tensor and A, B and C are the loading matrices. In some representations, the core tensor is absorbed into the loading matrices.

It can also be expressed as a finite sum of rank-1 components

$$\mathcal{X}^{I \times J \times K} \approx \sum_d a_d^I \circ b_d^J \circ c_d^K \quad (37)$$

The illustration for the CP model is in figure 11.

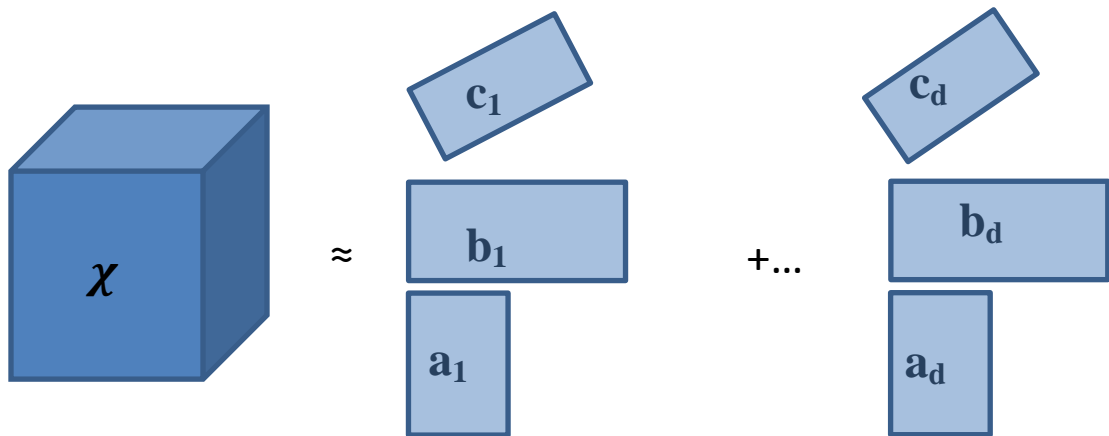


Figure 11: The CP Model



### 3.10 Rank & Multilinear Rank of a Tensor

The rank of a matrix refers to the number of linearly independent rows and columns. There is a well-defined approach to determine the rank of a matrix but in tensors the rank is determined by fitting various CP models. The model which gives the best fit will then provide the rank of the tensor.

The rank of a tensor is the minimal sum of rank one components  $R$  such that:

$$\mathcal{X} = \sum_{r=1}^R a_r \circ b_r \circ c_r \quad (38)$$

where  $R$  is the rank of the tensor.

In other words, the rank of a tensor refers to the minimum number of CP components that sum up to the tensor. The definition of a matrix rank implies that the rank cannot be greater than the number of its rows and columns. However, the rank of a tensor can be greater than  $\min(I, J, K)$ .

The multilinear rank of a tensor arises when the Tucker model is used. For the Tucker model shown in figure 10, the multilinear rank is given in mode-1, mode-2 and mode-3 as  $L$ ,  $M$  and  $N$  respectively.

### 3.11 Determining the Number of Components in a Model

The number of components in both the CP and Tucker models is very important since the features depend on these components. As a result, the right interpretation of a phenomenon is only possible if the right number of components is used in the model. Approaches used in estimating the order of models are the Core Consistency Diagnostic (CORCONDIAG) and the Automatic Relevance Determination (ARD) methods.

The CORCONDIAG is used for estimating number of components in the CP model. A core consistency diagnostic value above 90% indicates a very trilinear model whilst a value near zero indicates an invalid model. It is done by calculating the Tucker core array using the components obtained from the CP model by using both the CP and Tucker core arrays in the expressions below:

$$\mathcal{G} \leftarrow \mathcal{X} \times_1 A_{CP}^\dagger \times_2 B_{CP}^\dagger \times_3 C_{CP}^\dagger \quad (39)$$

$$\text{CORCONDIAG} = 100 \times \left( 1 - \frac{\|I - \mathcal{G}\|_F^2}{\|I\|_F^2} \right) \quad (40)$$

where  $A_{CP}, B_{CP}, C_{CP}$  are components of the CP model,  $I$ =diagonal CP core array, and  $\mathcal{G}$ = corresponding Tucker core array obtained from the CP model loadings.

In the ARD, there is no need to evaluate potential model orders. The priors are given hyperparameters that are able to give the variation of each component. Through optimization, components with values falling below the threshold are removed.

### 3.12 Software

There are several software packages available for tensor decompositions. There is the TensorToolbox that can be downloaded as an add-on for use in Matlab. Other external toolboxes that can be used in Matlab are the N-way Toolbox and the PLS\_Toolbox which is a commercial package. The Multilinear Engine is a FORTRAN code used for computing models. In C++, the HUJI Tensor Library (HTL) can be used.

### 3.13 Applications

Originally applied in Psychometrics in the 1970s by Carroll, Chang and Harshman, tensor decompositions are now being used across a wide variety of fields. An application area in behavioral science addresses the question of ‘which group of subjects behaves differently on which variables under which conditions?’ This is a multidimensional problem and as such tensor decompositions are used since they are able to extract loadings that span the dynamics of each mode.

In chemometrics, tensors were first used by Appleof and Davidson in 1981. The CP model is used in fluorescence spectroscopy which is based on the Beer-Lambert’s law that states that there is a linear relationship between the absorbance of light and the concentration of a compound. As such tensor decompositions make it possible to analyze the phenomenon as shown:

$$\chi^{Excitation \times Emission \times Sample} \approx \sum_{d=1}^{D \# compounds} a_d^{Excitation} \circ b_d^{Emission} \circ c_d^{Samples} \quad (41)$$

In bioinformatics, HOSVD is used in interpreting cellular states and biological processes by explaining different combinations of patterns. Also the Tucker model was used in analyzing the metabolic response of rats to toxins. The NMR spectra of the rat urine samples were used in the analysis represented in tensor form as:

$$\chi^{Sample \times Spectra \times Time} \approx \sum_{lmn} g_{l,m,n} a_l^{sample} \circ b_m^{spectra} \circ c_n^{time} \quad (42)$$

In computer vision, tensor factorizations have been used extensively in the facial recognition software known as TensorFaces. Facial images taken from different subjects under different illumination conditions and expressions are used as the dataset and it has

been found out that, TensorFaces are very accurate when compared with other pattern recognition techniques.

Tensor decompositions have also been applied in web mining. The Enron email dataset when analyzed revealed the patterns of communication between various officers and the periods in which communication took place in a more interpretable manner using tensors. Tensor decompositions were also used in the Netflix Collaborative Filtering competition for predicting users' ratings of movies.

Tensor factorization is a powerful tool for data analysis. This is evident in its use in several application areas. However, no work has been done on tensor application in road pavement health monitoring.

## Chapter 4

### DATA ANALYSIS & RESULTS

#### 4.1 Source of Data

The pavement images used for the analysis were part of the dataset from Fereidoon Moghadas Nejad and Hamzeh Zakeri (A comparison of multi-resolution methods for detection and isolation of pavement distress, Amirkabir University of Technology, Tehran, Iran, 2010). The images were acquired using a Pavement Image Acquisition System (PIAS) shown in the figure 12. It is made up of three lighting sources, a Global Positioning System (GPS), camera, frame grabber, monitoring system, computer and a power generator.



Figure 12: Pavement Image Acquisition System, PIAS (Nejad et al 2011)

The GPS mounted on the PIAS setup is advantageous because it makes it possible to indicate the exact location of the cracks.

## 4.2 Description

26 images from the larger dataset were used in the analysis. They were  $3501 \times 2550$  color images. The entire dataset of 1830 images was divided into two main groups. The first group of images had distresses and the second group of images was without distresses. Under the first group, there were longitudinal, transverse, diagonal, block, alligator and pothole distresses. However, only longitudinal and transverse cracks were used for this study. Figure 13 shows samples of the images used.



Figure 13: Original longitudinal and transverse cracks (Nejad et al 2011)

### 4.3 Analysis

The analysis was carried out in MATLAB using the Tensor Toolbox (Brett W. Bader, Tamara G. Kolda and others, MATLAB Tensor Toolbox Version 2.5, Available online, January 2012. URL: <http://www.sandia.gov/~tgkolda/TensorToolbox/>). This is a free add-on downloaded from the URL above. The four major steps involved in the analysis are listed and explained further below:

1. Preprocessing and formation of training tensors;
2. Preprocessing of test data;
3. Higher Order Singular Value Decomposition of Training Tensors; and
4. Classification

#### *1. Preprocessing and Formation of training tensors*

This is the first step which involves preprocessing of the training dataset after which tensors are formed. First, the images are converted from color (RGB) into grayscale format. The next step involves resizing the images into  $100 \times 100$  pixel size images. The images were originally of size  $3501 \times 2550$ . Finally, the images are separated into two groups; transverse and longitudinal cracks after which the images in each class or group are stacked one after the other to form the training tensor sets. Note that in order for the Tensor Toolbox algorithms to function, the 3D arrays obtained after this step must be converted into tensor format in the MATLAB environment. See figure 15 for a diagram of the training tensors.

Figure 14 shows the flow of the steps involved in the preprocessing of the images before classification.

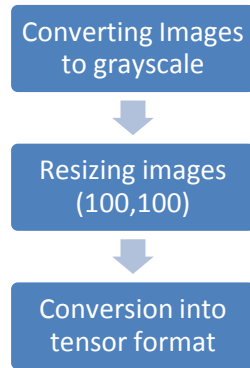


Figure 14: Steps in preprocessing of images for classification

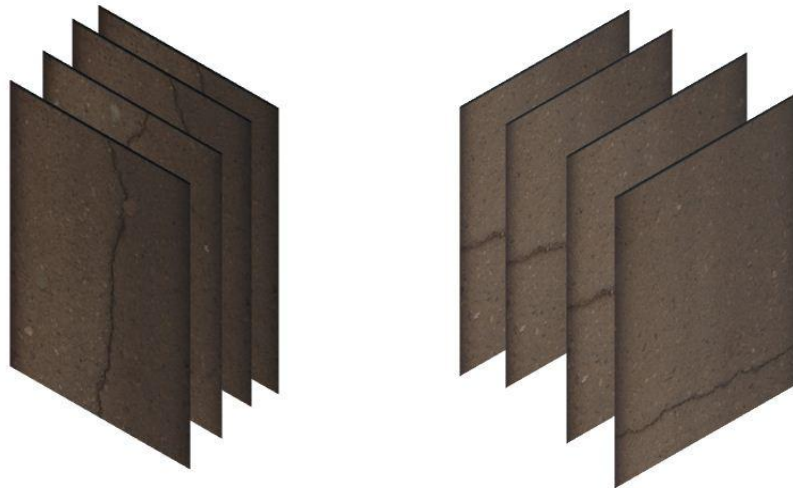


Figure 15: Longitudinal and Transverse crack training sets



## 2. Preprocessing of Test Data

The images to be tested also go through the previous preprocessing steps. However, training tensor sets are not formed for the test images. The preprocessing of test data involves conversion to grayscale and resizing of the test images to the same size as the images in the training sets, in this case  $100 \times 100$ .

## 3. Higher Order Singular Value Decomposition (HOSVD) of training tensor

Each training tensor is decomposed using HOSVD. After the decomposition, the outer product of the core array and the loading matrices in modes 1 and 2 are computed. The resulting tensor is used as the basis of classification. Mathematically, the HOSVD is expressed as:

$$\mathcal{A} = \mathcal{G} \times_1 B \times_2 C \times_3 D \quad (43)$$

where  $\mathcal{A} \in \mathbb{R}^{100 \times 100 \times K}$ , the training tensor

$\mathcal{G}$  is the core array

$K$ =number of images in training set

$B, C$  and  $D$  are the loading matrices in modes 1, 2 and 3 respectively.

Each matrix in  $A_k$  forms the basis for classification and they can be expressed as:

$$A_k^c = \mathcal{G}(:, :, k) \times_1 B \times_2 C \quad (44)$$

where  $k$  represents number of images in training set and  $c$  is the image class.

For example, the frontal slices of  $\mathcal{A}_3^{longitudinal}$  are shown in figure 16.

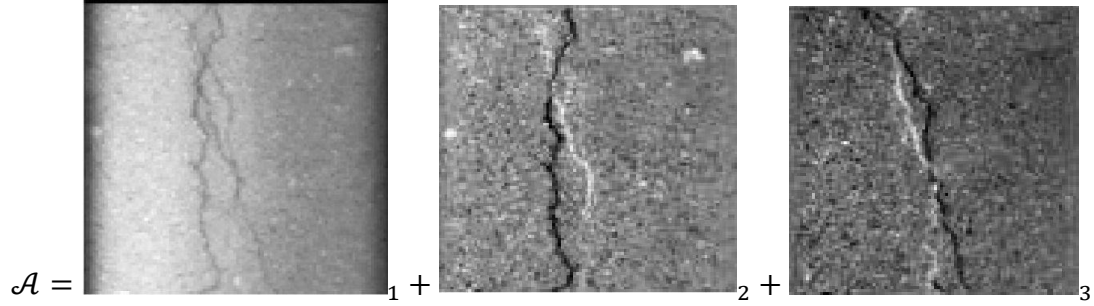


Figure 16: Frontal slices from HOSVD of longitudinal tensor of 3 images

The individual images that form the slices of  $\mathcal{A}_k^{longitudinal}$  and  $\mathcal{A}_k^{transverse}$  form the basis for the classification. It can be seen from the illustration above that each slice contains several longitudinal cracks which belonged to the original longitudinal crack training set.

#### 4. Classification

The classification of the pavement cracks is based on the function below:

$$\sum_{k=1}^K \|V - g(k)\mathcal{A}_k^c\| \quad (45)$$

where V 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

$$g(k) = V \cdot \mathcal{A}_k^c$$

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  $c$ .

#### 4.4 Results

Using longitudinal and transverse training sets of 10 images each, all the images were classified correctly. 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. In fact, the accuracy of this classification method depends more on the variations exhibited by images within the same classes than on the number of images. See Appendix A.

A closer look at the first slices of the core arrays of the training sets with 2, 3, 5, 7, 9 and 10 images shows the variation captured by the core tensors and hence its ability to correctly classify test images. See figures 17 and 18.

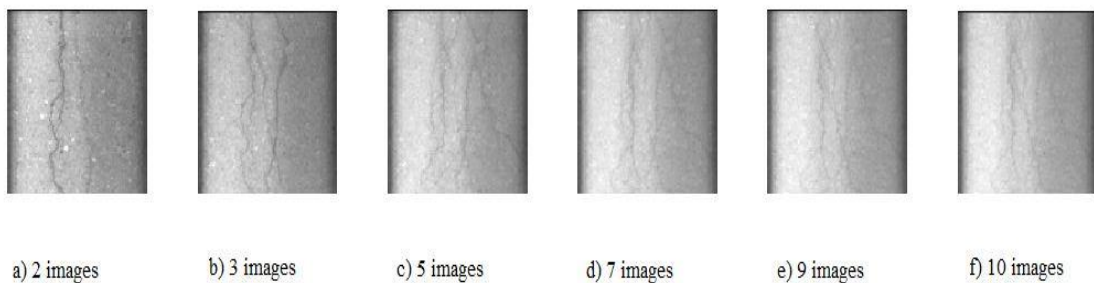


Figure 17: First slices of the core arrays from HOSVD of longitudinal crack tensors

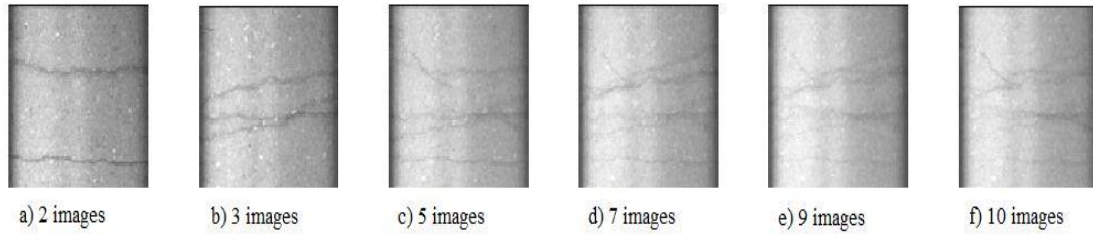


Figure 18: First Slices of the core arrays from HOSVD of transverse crack tensors

It can be observed from the figure above that the slice from the core arrays consisting of only two images show the two cracks that form the tensor and therefore its limited classification capability. The core tensors obtained from the training sets with 10 images contain and show various types of cracks within the same class and as such are more likely to classify test images correctly. See graph in figure 19.

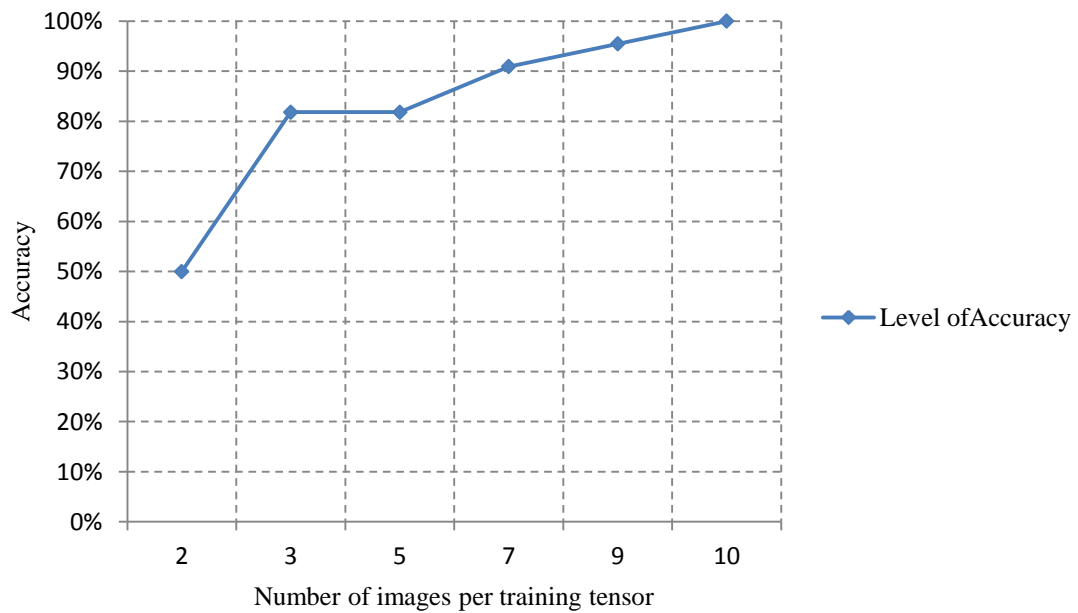


Figure 19: Graph of number of images in training tensors against level of accuracy

The general trend was that the level of accuracy increased with the number of images in the training set. The training sets consisting of 3 and 5 images both had the same level of accuracy which means the accuracy of the method is not solely dependent on the number of training images but also the variation of the images within the same class. A level of accuracy of 50% which was the lowest value was obtained when only two images were used in the training set. Conversely, a level of accuracy of 100% was obtained with 10 images in both longitudinal and transverse training sets.

The processing times for the algorithm was also affected by the size of the images used. For images of size  $100 \times 100$  which were used for the most part of the analysis, the processing time was less than a second. Ten images were used in all the training sets for the analysis of processing times and image sizes. Generally, the processing times increase with increase in the size of the image. This is illustrated in the figure 20 below.

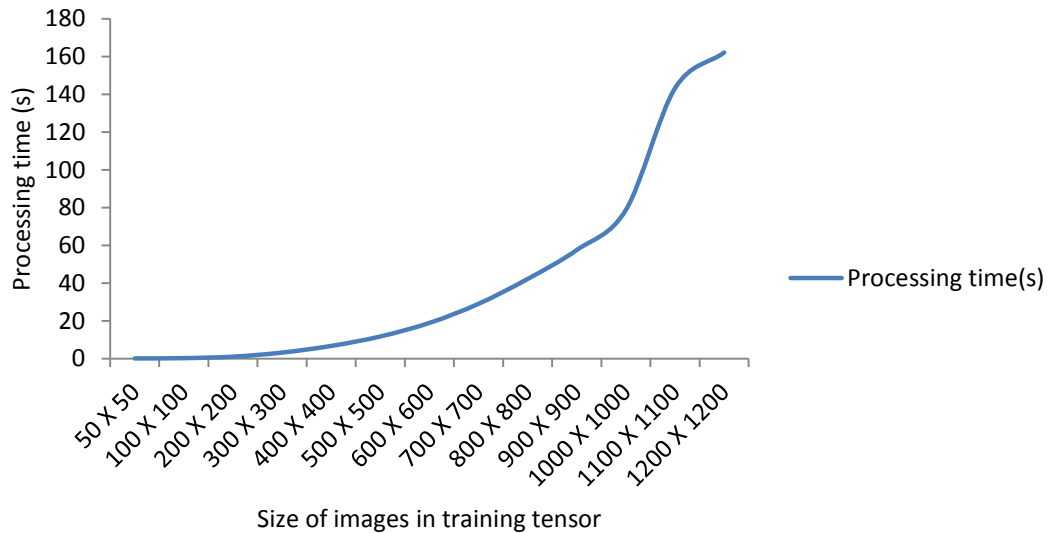


Figure 20: Graph of processing times versus size of images

For the same image size, the processing times do not differ significantly with increase in the number of images per training size. However, this is not the case when considering larger image sizes. See figure 21 for an illustration. Hence, for larger image sizes, the processing times increase significantly with an increase in the size of the training set

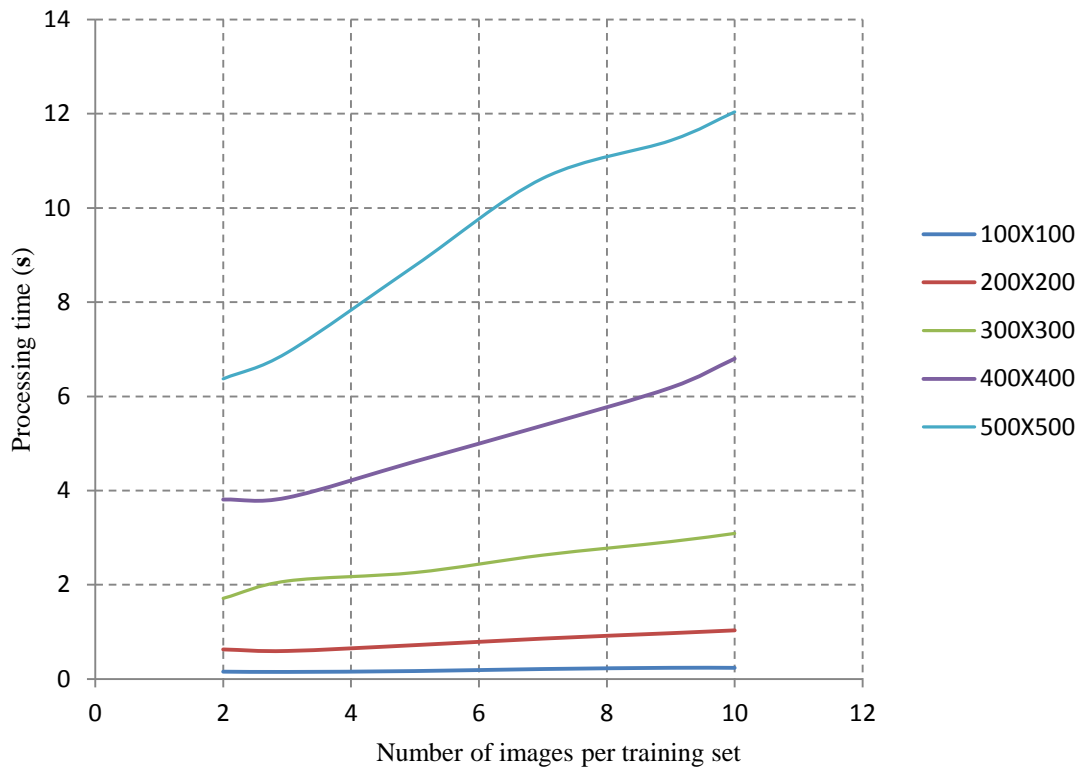


Figure 21: Comparison of processing times for various image sizes and training sets

## **Chapter 5**

### **CONCLUSION & RECOMMENDATIONS**

#### **5.1 General**

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.

#### **5.2 Conclusion**

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.

Generally, the algorithm was very accurate in classifying the cracks. With a training set of only three images, an 81.81% level of accuracy was obtained which is relatively

high considering the number of images in the training tensor. 100% accuracy was recorded with ten images which showed variations of longitudinal and transverse cracks. Furthermore, the classification was done at appreciable speeds with image sizes of about 1200 X 1200 which is very promising since the algorithm will be implemented on higher speed computers with images of higher resolution. The approach can be used for automated pavement crack classification.

### **5.3 Recommendations**

- The algorithm must be tested on a larger dataset of crack images. This will mean having larger training tensors for analysis.
- It will be very beneficial to extend the crack classification analysis to consider other forms of pavement defects such as alligator cracking and box cracking. The nature of the tensor analysis is such that it may be very useful for classifying these types of cracks whose classification has been very problematic in the past.
- The algorithm must be tested on quad-core computers and other processors with higher capabilities in order to reduce processing times making the classification much more efficient.
- It is also recommended that parallel processing be employed in the future for classification based on tensor analysis in order to speed up the entire process.
- Further research focuses on utilizing tensor analysis in the health monitoring of other civil infrastructure systems since these involve large datasets which are usually multidimensional.



## REFERENCES

- (1) Mørup M. (2011). "Applications of tensor factorizations and decompositions in data mining." *John Wiley & Sons, Inc.*, Volume 1.
- (2) Saar T<sup>1</sup>., Talvik O<sup>2</sup>. (2010). "Automatic Asphalt Pavement Crack Detection and Classification using Neural Networks." <sup>1</sup>Department of Electronics, TUT, Ehitajate tee 5, 19086 Tallinn, Estonia, <sup>2</sup>Department of Transportation, TUT, Ehitajate tee 5, 19086 Tallinn, Estonia, *12<sup>th</sup> Biennial Electronics Conference*.
- (3) Gourvenec S., Tomasi G., Durville C., Di Crescenzo E., Saby C.A., Massart D.L., Bro R., Oppenheim G. (2005). "CuBatch, a MATLAB interface for n-mode data analysis." Elsevier Ltd, *Chemometrics and Intelligent Laboratory Systems* 77 122-130.
- (4) Bader B. W., Berry M. W., Browne M. (2007). "*Survey of Text Mining II, Clustering, Classification and Retrieval, Chapter 8, Discussion Tracking in Enron Email Using PARAFAC*".
- (5) Kolda T. G., Bader B. W. (2009). "Tensor Decompositions and Applications." *SIAM Review*, Vol. 51, No. 3, pp. 455-500.
- (6) Rababaah H., Vrajitoru D., James W. (2005). "Asphalt Pavement Crack Classification: A Comparison of GA, MLP and SOM." *GECCO, Genetic and Evolutionary Computation Conference Late-Breaking Paper*.
- (7) Savas B., Elden L. (2006). "Handwritten digit classification using higher order singular value decomposition." *The Journal of Pattern Recognition Society, Elsevier Ltd, Pattern Recognition* 40 (2007) 993-1003.
- (8) Vasilescu M.A.O., Terzopoulos D. (2002). "Multilinear Analysis of Image Ensembles: TensorFaces." *ECCV '02*, Copenhagen, Denmark.
- (9) Cichoki, A., Zdunek R., Phan, A. H., Amari, S.I. (2009). "Nonnegative Matrix and Tensor Factorizations, Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation." *John Wiley and Sons Inc.*, p 1-11, p 40-47.
- (10) Kiers, A.L.H. (2000). "Towards a standardized notation and terminology in multiway analysis." Heyman Institute (PA), University of Groningen, Grote

Kruisstraat 2/1, NL-9712 TS Groningen, The Netherlands, *Journal of Chemometrics*, 14:105-122.

- (11) Elden, L. (2008). "Tensor Computations and Applications in Data Mining." *Department of Mathematics, Linköping University, Sweden, SIAM AM*.
- (12) Kilmer M. E. and Martin M. C. D. (2004), "Decomposing a Tensor." *SIAM News*, Volume 37, Number 9.
- (13) Anderson A.C. and Bro R. (2000). "The N-way Toolbox for MATLAB." *Chemometrics Intelligent Laboratory Systems*, 52(1): 1-4.
- (14) Singh K.P., Malik A., Singh V.K., Sinha S. (2005). "Multiway Analysis of soils irrigated with wastewater- A case study." *Chemometrics and Intelligent Laboratory Systems* 83 1-12.
- (15) Mørup M., Hansen L.K. (2009). "Automatic relevance determination for multiway models." *Wiley Interscience* ([www.interscience.wiley.com](http://www.interscience.wiley.com)) DOI:10.1002/cem.1223.
- (16) Nejad F.M., Zakeri H. (2008). "A comparison of multi-resolution methods for detection and isolation of pavement distress." Amirkabir University of Technology, Tehran, Iran, *Expert Systems with Applications* 38 (2011) 2857-2872.
- (17) Faloutsos C., Kolda T. G., Sun J. (2007). "Mining Large Time-evolving Data Using Matrix and Tensor Tools." <http://www.cs.cmu.edu/~jimeng/papers/ICMLtutorial.pdf>. Time accessed: 11/01/12, 11.45 am
- (18) Xu B., Yaxiong H. (2003). "Automated Pavement Cracking Rating System: A Summary." *Center for Transportation Research The University of Texas at Austin*, Project Summary Report 7-4975-S.
- (19) Santhi B., Krishnamurthy G., Siddharth S., Ramakrishnan P. K. (2012). "Automatic Detection of Cracks in Pavements using Edge Detection Operator." *Journal of Theoretical and Applied Information Technology*, Vol. 36 No. 2.
- (20) Oliveira H., Correia P.L. (2008). "Identifying and Retrieving Distress Images from Road Pavement Surveys." *15<sup>th</sup> IEEE International Conference*, P57-60.

## Appendix A

### CLASSIFICATION RESULTS

Image	Longitudinal set	Transverse set	
1l	1.0671	1.0657	*
2l	1.0677	1.0687	
3l	1.0725	1.0809	
4l	1.0825	1.0756	*
5l	1.0344	1.0678	
6l	1.0803	1.0859	
7l	1.0549	1.0848	
8l	1.0771	1.0687	*
9l	1.0671	1.0657	*
10l	1.0651	1.0645	*
13l	1.0725	1.0809	
1t	1.0691	1.0726	*
2t	1.0762	1.0689	
3t	1.0761	1.0799	*
4t	1.0694	1.0445	
5t	1.075	1.0803	*
6t	1.0809	1.0751	
7t	1.0842	1.0845	*
8t	1.0761	1.0799	*
9t	1.0691	1.0726	*
10t	1.0762	1.0689	
11t	1.0779	1.04	

2 images in each training set

longitudinal set: 5l,7l

transverse set: 4t, 11t

\*-wrong classification

Accuracy=50%

Image	Longitudinal set	Transverse set	
1l	2.0424	2.0665	
2l	2.0646	2.067	
3l	2.0731	2.0788	
4l	2.0496	2.0777	
5l	2.0473	2.0632	
6l	2.0821	2.0816	*
7l	2.0792	2.0812	
8l	2.0725	2.0668	*
9l	2.0424	2.0665	
10l	2.0612	2.064	
13l	2.0731	2.0788	
1t	2.0671	2.0544	
2t	2.0708	2.0468	
3t	2.0824	2.072	
4t	2.0637	2.0661	*
5t	2.0743	2.0749	*
6t	2.0783	2.0501	
7t	2.0782	2.0857	
8t	2.0824	2.072	
9t	2.0671	2.0544	
10t	2.0708	2.0468	
11t	2.0694	2.0666	

3 images in each training set

longitudinal set: 5l,4l,1l

transverse set: 1t, 6t, 2t

\*-wrong classification

Accuracy=81.81%

Image	Longitudinal set	Transverse set	
1l	4.0459	4.065	
2l	4.0611	4.0635	
3l	4.06833	4.0777	
4l	4.0585	4.0769	
5l	4.049	4.062	
6l	4.0608	4.08	
7l	4.0734	4.0791	
8l	4.0712	4.0639	*
9l	4.0459	4.065	
10l	4.0515	4.0621	
13l	4.0683	4.0777	
1t	4.0624	4.0692	*
2t	4.0684	4.0509	
3t	4.0758	4.056	
4t	4.0605	4.0547	*
5t	4.0691	4.0728	
6t	4.0754	4.0532	
7t	4.0752	4.052	
8t	4.0758	4.056	
9t	4.0624	4.0692	*
10t	4.0684	4.0509	
11t	4.0676	4.0519	

5 images in each training set  
longitudinal set: 1l, 4l, 5l, 6l, 10l  
transverse set: 11t, 10t, 8t, 6t, 4t  
\*-wrong classification

Accuracy=81.81%

Image	Longitudinal set	Transverse set	
1l	6.0476	6.0591	
2l	6.06	6.0596	*
3l	6.0564	6.0705	
4l	6.0609	6.073	
5l	6.0496	6.0585	
6l	6.0634	6.0727	
7l	6.0713	6.0727	
8l	6.0625	6.0657	
9l	6.0476	6.0591	
10l	6.0517	6.0606	
13l	6.0564	6.0705	
1t	6.0609	6.0478	
2t	6.0664	6.0553	
3t	6.0735	6.0538	
4t	6.0583	6.0583	
5t	6.0682	6.0669	*
6t	6.0738	6.0596	
7t	6.0745	6.0671	
8t	6.0735	6.0538	
9t	6.0609	6.0478	
10t	6.0664	6.0553	
11t	6.0665	6.0558	

7 images in each training set  
longitudinal set: 1l, 3l, 4l, 5l, 6l, 10l, 8l  
transverse set: 1t, 2t, 6t, 8t, 9t, 11t, 7t  
\*-wrong classification

Accuracy=90.91%

Image	Longitudinal set	Transverse set	
1l	8.0417	8.0585	
2l	8.051	8.0608	
3l	8.0565	8.0697	
4l	8.062	8.0728	
5l	8.0589	8.0587	*
6l	8.064	8.0724	
7l	8.0635	8.0729	
8l	8.0654	8.067	
9l	8.0417	8.0585	
10l	8.0531	8.0612	
13l	8.0565	8.0697	
1t	8.0607	8.0488	
2t	8.0676	8.0584	
3t	8.0727	8.0585	
4t	8.0589	8.0512	
5t	8.0684	8.0579	
6t	8.0749	8.0623	
7t	8.0742	8.0667	
8t	8.0727	8.0585	
9t	8.0607	8.0488	
10t	8.0676	8.0584	
11t	8.0666	8.0598	

9 images in each training set  
longitudinal set: 13l, 10l, 9l, 8l, 7l, 6l, 4l, 11l, 12l  
transverse set: 11t, 10t, 9t, 8t, 7t, 6t, 4t, 1t, 5t  
\*-wrong classification  
Accuracy=95.45%

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

10 images in each training set

longitudinal set:

transverse set:

\*-wrong classification

Accuracy=100%



## Appendix B

### REPRINT PERMISSION LETTERS

Offei,

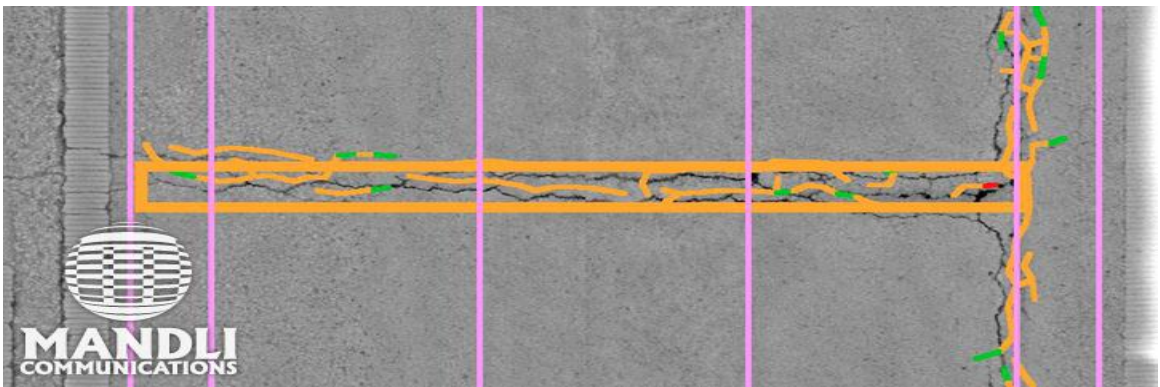
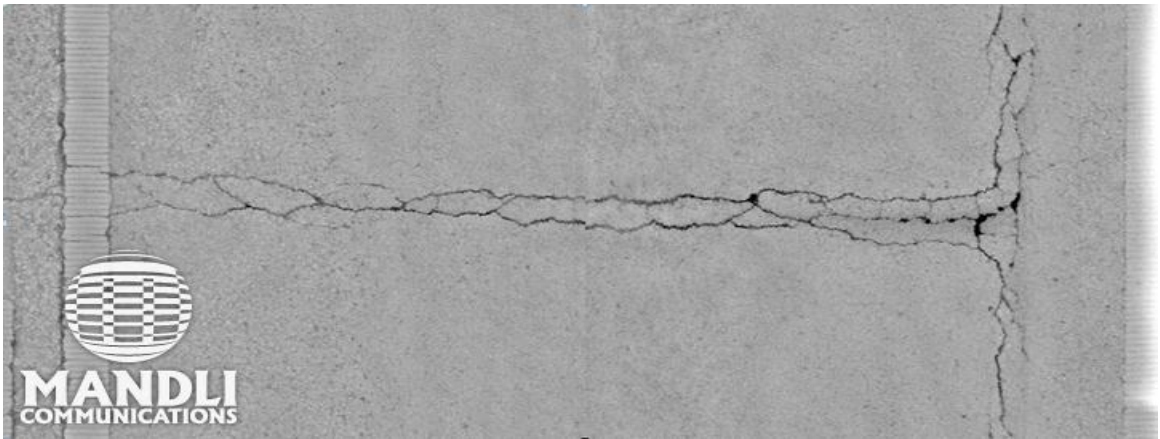
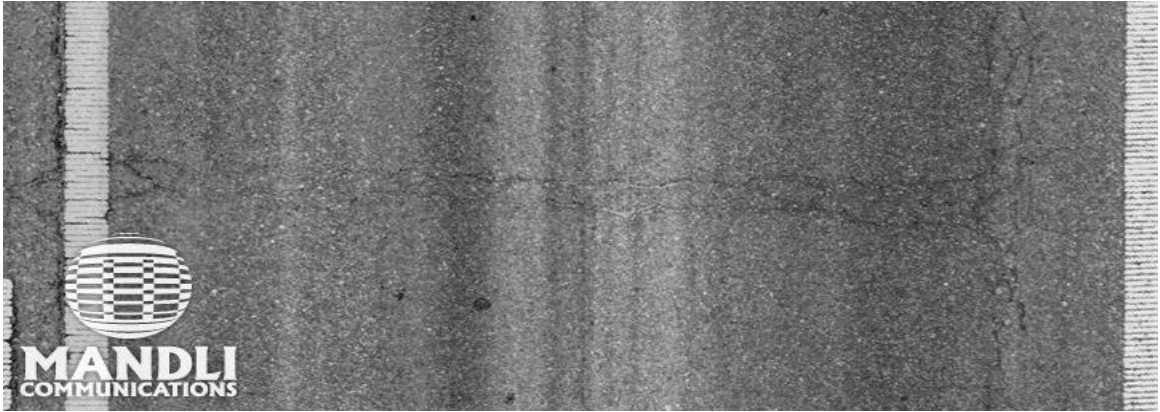
Attached are a few images you can use for your presentation. We would be very interested in seeing the final results.

Thank you,

John

John M. Caya  
Director of Business Development  
Mandli Communications, Inc.  
4801 Tradewinds Parkway  
Madison, WI 53718  
w [608.835.3500 Ext. 1001](tel:608.835.3500)  
c [608.213.5200](tel:608.213.5200)  
[www.mandli.com](http://www.mandli.com)





## PERMISSION LETTER FROM F.M. NEJAD AND H. ZAKERI

Sure, we will be glad . Additionally, we encouraging him to see our new publications below.

Can you send us a copy of your thesis in future?

1- M.H. Fazel Zarandi, Fereidoon Moghadas Nejad and H. Zakeri ,(2012). *A Type-2 Fuzzy Model Based on Three Dimensional Membership Functions for Smart Thresholding in Control Systems*, Fuzzy Controllers-Recent Advances in Theory and Applications, Sohail Iqbal, Nora Boumella and Juan Carlos Figueroa Garcia (Ed.), ISBN: 978-953-51-0759-0

2- Fereidoon Moghadas Nejad, Hamzeh Zakeri ,(2013). [\*The Hybrid Method and its Application to Smart Pavement Management\*](#), Metaheuristics in Water, Geotechnical and Transport Engineering, Xin-She Yang, Amir Hossein Alavi,(Ed.) 2013, Pages 439-484, ISBN: 978-0-12-398296-4.

F. Moghadas Nejad, *PhD*  
Associate Professor  
Dept. of Civil & Environmental Engineering  
Amirkabir University of Technology  
Hafez Avenue  
Tehran 15875, Iran  
Tel: [+98\(21\)64543004](tel:+982164543004), Fax: [+98 \(21\) 66414231](tel:+982166414231)  
Email: [moghadas@aut.ac.ir](mailto:moghadas@aut.ac.ir)(Other Email: [f.moghadas@yahoo.com](mailto:f.moghadas@yahoo.com))

---