2DTPCA: A NEW FRAMEWORK FOR MULTILINEAR PRINCIPAL COMPONENT ANALYSIS

Cagri Ozdemir[†]*, Randy C. Hoover*[†]*, and Kyle Caudle*[‡]

South Dakota Mines
[†]Department of Computer Science & Engineering
[‡]Department of Mathematics
Rapid City, South Dakota

ABSTRACT

Two-directional two-dimensional principal component analysis ((2D)^{2}PCA) has shown promising results for it's ability to both represent and recognize facial images. The current paper extends these results into a multilinear framework (referred to as two-directional Tensor PCA or 2DTPCA for short) using a recently defined tensor operator for 3$^{\text{rd}}$-order tensors. The approach proceeds by first computing a low-dimensional projection tensor for the row-space of the image data (generally referred to as mode-1) and then subsequently computing a low-dimensional projection tensor for the column space of the image data (generally referred to as mode-3). Experimental results are presented on the ORL, extended Yale-B, COIL-100, and MNIST data sets that show the proposed approach outperforms traditional "tensor-based" PCA approaches with a much smaller subspace dimension in terms of recognition rates.

Index Terms— Two directional tensor PCA, tensor PCA, tensor singular value decomposition

1. INTRODUCTION

Principal component analysis (PCA) [1] is one of the most popular feature extraction methods and widely used in many areas of image classification and pattern recognition for dimensionality reduction. Traditionally such methods have been approached using a linear algebraic framework. In particular, given a collection of n images $\{I_1, I_2, \ldots, I_n\}$, each of size $h \times v$, each image is "vectorized" into a column vector $\mathbf{x}_i = \text{vec}(I_i) \in \mathbb{R}^{m\times 1}$ where $m = hv >> n$. While matrix PCA has shown much success over the years, there are two fundamental issues that arise when the data of interest is collections of images: a) vectorizing the images destroys the *natural representation* of the image thereby eliminating the spatial correlation within each image and b) because the number of pixels in an image m is generally significantly larger than the number of samples n, subspace methods can encounter the curse of dimensionality or small sample size problems.

To overcome these issues, new methods have been proposed that rely on higher-order data structures that leave each image in it's natural matrix form, and stack the collection of matrices into a tensor structure (commonly referred to as a n-way or n mode array, where n is not to be confused with the number of images but rather represents the different statistical modes of the data). Most notably is the so called High Order Singular Value Decomposition (HOSVD) that is based on Tucker decompositions [2–4] and Multilinear PCA (MPCA) [5] that performs dimensionality reduction in all statistical modes of the tensor simultaneously to produce a projection tensor that relies on basic multilinear algebra (commonly referred to as n-mode products) as outlined in the seminal papers [2, 3]. Additionally, two "tensor-like" decompositions have been proposed where the authors present a method to operate on the image collection as 2-mode tensors and reduce the dimensionality of the data through image covariance [6, 7]. Most notably was a reduced order feature space in the *row-pixels*, referred to as 2DPCA [6] that was later extended to both the *row-* and *column-pixels* referred to as (2D)^{2}PCA [7]. Finally, the author's show in [8–10] that an alternate approach to multilinear PCA (referred to as Tensor PCA) can be performed based on a recently defined multiplication operator for third order tensors [11–14], referred to as the ***t-product*** and resulting ***t-SVD*** (as outlined in Section 2). This operator is advantageous in that it allows for a tensor singular value decomposition to be defined that is analogous to its matrix counterpart, i.e., it decomposes a third-order tensor into three third-order tensors with similar orthogonality and diagonal structures.

The current paper builds on our prior work in [8–10] while capitalizing on the developments of both 2DPCA as well as

The current research was supported in part by the Department of the Navy, Naval Engineering Education Consortium under Grant No. (N00174-19-1-0014) and the National Science Foundation under Grant No. (2007367). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Naval Engineering Education Consortium or the National Science Foundation.

(2D)^{2}PCA as outlined in [6,7]. In particular, we illustrate that through the ***t-product*** and resulting ***t-SVD***, (2D)^{2}PCA can be re-formulated into a multilinear framework that decomposes both the *row-* and *column-pixels* into reduced dimensional tensor models of order three (i.e., reduced dimensional 3-mode tensors) referred to as 2D Tensor PCA (2DTPCA). We present experimental results from some classic image data sets (namely ORL, extended Yale-B, COIL-100, and MNIST) and show that this new formulation performs better than the aforementioned methods in terms of image classification.

2. MATHEMATICAL PRELIMINARIES

In this section, we provide a brief overview of the tensor definitions and mathematical operations required to keep this work self contained, extended details may be found in [8–14].

2.1. Mathematical Preliminaries

Multidimensional data can be defined as multidimensional arrays of numbers, commonly referred to as *tensors* [2, 3]. The dimensions of a tensor are called *ways* or *modes*. The number of modes determines the *order* of a tensor. Fundamental to the results presented in this paper is a recently defined multiplication operation on third-order tensors which itself produces a third-order tensor [11–13].

First, we introduce the basic notations and definitions outlined in [11–13]. It will be convenient to break a tensor $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ up into various slices and tubal elements, and to have an indexing on those. The i^{th} lateral slice will be denoted $\mathcal{A}_i$ whereas the j^{th} frontal slice will be denoted $\mathcal{A}^{(j)}$. In terms of MATLAB indexing notation, this means $\mathcal{A}_i \equiv \mathcal{A}(:, i, :)$ while $\mathcal{A}^{(j)} \equiv \mathcal{A}(:, :, j)$.

We use the notation $\mathbf{a}_{ik}$ to denote the i, k^{th} tube in $\mathcal{A}$; that is $\mathbf{a}_{ik} = \mathcal{A}(i, k, :)$. The j^{th} entry in that tube is $\mathbf{a}_{ik}^{(j)}$. Indeed, these tubes have special meaning for us in the present work, as they will play a role similar to scalars in $\mathbb{R}$.

Definition 1. *An element* $\mathbf{c} \in \mathbb{R}^{1 \times 1 \times n}$ *is called a* **tubal-scalar** *of length* n.

In order to discuss multiplication between two tensors we must first introduce the concept of converting $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ into a block circulant matrix. If $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ with $\ell \times m$ frontal slices then $\texttt{circ}(\mathcal{A})$ is a block circulant matrix of size $\ell n \times mn$.

We anchor the `MatVec` operator to the frontal slices of the tensor. $\texttt{MatVec}(\mathcal{A})$ takes an $\ell \times m \times n$ tensor and returns a block $\ell n \times m$ matrix

$$\texttt{MatVec}(\mathcal{A}) = \left[(A^{(1)})^T, (A^{(2)})^T, \ldots, (A^{(n)})^T\right]^T$$

The operation that takes $\texttt{MatVec}(\mathcal{A})$ back to tensor form is the `fold` operator: $\texttt{fold}(\texttt{MatVec}(\mathcal{A})) = \mathcal{A}$.

Definition 2. *Let* $\mathcal{A} \in \mathbb{R}^{\ell \times p \times n}$ *and* $\mathcal{B} \in \mathbb{R}^{p \times m \times n}$ *be two third order tensors. Then the* ***t-product*** $\mathcal{A} * \mathcal{B} \in \mathbb{R}^{\ell \times m \times n}$ *is defined as* $\mathcal{A} * \mathcal{B} = \texttt{fold}\left(\texttt{circ}(\mathcal{A}) \cdot \texttt{MatVec}(\mathcal{B})\right)$.

Definition 3. *The* ***identity tensor*** $\mathcal{I} \in \mathbb{R}^{m \times m \times n}$ *is the tensor whose frontal slice is the* $m \times m$ *identity matrix, and whose other frontal slices are all zeros.*

Definition 4. *If* $\mathcal{A}$ *is* $\ell \times m \times n$*, then the* ***tensor transpose*** $\mathcal{A}^T$ *is the* $m \times \ell \times n$ *tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices through* n.

Definition 5. *A tensor* $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ *is called orthogonal if* $\mathcal{A} * \mathcal{A}^T = \mathcal{A}^T * \mathcal{A} = \mathcal{I}$

Definition 6. *If* $\mathcal{A}$ *is* $m \times \ell \times n$*, then the* ***tensor rotation*** $\texttt{rot}(\mathcal{A})$ *is the* $n \times \ell \times m$ *tensor obtained by transposing each of the lateral slices.*

2.2. Computation of the t-SVD

The final tool necessary for tensor PCA is a tensor singular value decomposition (referred to as the ***t-SVD***. In [11–13] the authors show that, for $\mathcal{A} \in \mathbb{R}^{n \times n \times n}$, there exists orthogonal tensors $\mathcal{U}$ and $\mathcal{V}$, a *front-face diagonal* (f-diagonal) tensor $\mathcal{S}$ such that

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T = \sum_{i=1}^{min(\ell, m)} \mathcal{U}_i * \mathbf{s}_i * \mathcal{V}_i^T \qquad (1)$$

where $\mathcal{U} \in \mathbb{R}^{\ell \times \ell \times n}$, such that $\mathcal{U} * \mathcal{U}^T = \mathcal{I}_{n \times n}$ is the tensor of left-singular matrices, $\mathcal{V} \in \mathbb{R}^{m \times m \times n}$, such that $\mathcal{V} * \mathcal{V}^T = \mathcal{I}_{m \times m}$ is the tensor of right-singular matrices, $\mathcal{S} \in \mathbb{R}^{\ell \times m \times n}$ is an *f-diagonal* tensor where $\mathbf{s}_i = \mathcal{S}(i, i, :)$ are the singular tuples. In the context of the current work, the lateral slices of $\mathcal{U}$ (the left-singular matrices) are analogous to the left-singular vectors of the matrix SVD and serve as the *tensor* principal components. A graphical depiction of the ***t-SVD*** is shown in Fig. 1 where the tensor on the left-hand-side is the image data tensor $\mathcal{A}$ and each lateral slice $\mathcal{A}^{(i)}$ is an $l \times n$ image in our data set. We note that, l is the row-space of our image set and referred to as mode-1, n is the column-space of our image set and referred to as mode-3, whereas m is the "temporal-space[1]" of our image set and referred to as mode-2.

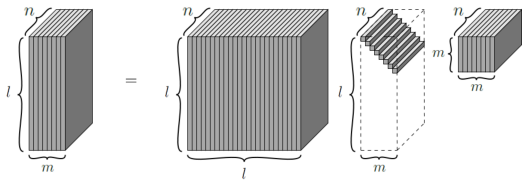

Fig. 1: Graphical depiction of the ***t-SVD***.

[1]Temporal in this context need not refer to time but rather samples in our training/testing set.

3. PROPOSED 2DTPCA APPROACH

In this section we provide an overview of our prior work on multilinear Tensor PCA as outlined in [8–10] and illustrate some of the shortcomings of such an approach. We then turn our attention to extensions of our prior work combined with the work of [6,7] to capitalize on correlations in both the row- and column-dimensions of the image set. Such extensions result in a multilinear variant of two-directional PCA referred to as 2DTPCA.

3.1. Overview of Tensor PCA

To keep the current work self-contained, we present a brief overview of our work on Tensor PCA outlined in [8–10] that capitalizes on the definitions outlined in Section 2. As shown in Fig. 1, we construct the image data tensor $\mathcal{A} \in \mathbb{R}^{l\times m\times n}$ where each image is of size $l \times n$ rotated laterally (i.e., a 2-mode tensor of dimension $l\times 1\times n$) and m is the total number of images in the set. Multilinear Tensor PCA can then be performed by computing the ***t-SVD*** of the image data tensor as $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ where $\mathcal{U} \in \mathbb{R}^{l\times l\times n}$ is the tensor containing the left-singular matrices (that are analogous to the principal components) ordered from left to right according to their importance. Dimensionality reduction (in mode-1) is performed by projecting the data tensor $\mathcal{A}$ onto the first k left singular matrices via

$$\mathcal{Y} = \mathcal{U}_k^T * \mathcal{A} \in \mathbb{R}^{k\times m\times n}, \tag{2}$$

is the reduced dimensional feature tensor used for online classification.

It's important to note here that while the above formulation reduces the dimensionality of the image data tensor along mode-1 (the row-pixels), the dimensionality along mode-3 (the column pixels) remains unchanged. The implications of this are that Tensor PCA capitalizes on mode-1 correlations but is unable to capitalize on the correlations along mode-3 as one would hope. This drawback is a direct consequence of the computation of the ***t-SVD*** as outlined in [11–13] which will be restated here for completeness. It is well known in matrix theory that a circulant matrix can be diagonalized via left and right multiplication by a discrete Fourier transform (DFT) matrix. Similarly, a block circulant matrix can be block diagonalized via left and right multiplication by a block diagonal DFT matrix. For example, consider the tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$, then

$$(F_n \otimes I_n)\texttt{circ}(\mathcal{A})(F_n^* \otimes I_n) = \texttt{diag}(D_1, D_2, \ldots, D_n) \tag{3}$$

where each of the D_i are $n \times n$, I_n is an $n \times n$ identity matrix of dimension, F_n is the $n \times n$ DFT matrix, F_n^* is its conjugate transpose, and $\otimes$ is the Kronecker product. To construct the ***t-SVD*** defined in (1), the matrix SVD is performed on each of the D_i, i.e., $D_i = U_i S_i V_i^T$. Taking the first block column of each block circulant matrix and applying the `fold` operator results in the decomposition $\mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Note that for simplicity, as well as computational efficiency, this entire process can by performed using the fast Fourier transform in place of the DFT matrix as illustrated in [11–13].

3.2. Proposed 2DTPCA method

One of the fundamental drawbacks associated with computing the ***t-SVD*** in this fashion is the choice of flattening the data tensor through the `circ(·)` operator. By construction, `circ(·)` operates of the frontal slices of $\mathcal{A}$ (mode-1), however this "choice" is somewhat arbitrary (e.g. we could just as easily reformulate the problem to operate on mode-2 or mode-3). As such, while we capture correlations in the image data along the row-pixels, we neglect the correlations in the image data along the column-pixels.

To overcome these drawbacks, we propose a two step procedure to capitalize on both the mode-1 and mode-3 correlations in the dataset (referred to as 2DTPCA). In particular, we illustrate the successive application of the ***t-SVD*** along the two different modes aims to capitalize on the correlations in both the row- and column-dimensions. Unfortunately, this approach also produces two different sets of subspaces (one for mode-1 and one for mode-3) that require fusion prior to data projection. Toward this end, the we define the **tensor rotation** operator (definition 6 in Section 2) that produces a rotated version of a tensor, i.e., given some $\mathcal{A} \in \mathbb{R}^{k\times m\times n}$, $\texttt{rol}(\mathcal{A}) = \bar{\mathcal{A}} \in \mathbb{R}^{n\times m\times k}$.

We proceed by computing Tensor PCA as illustrated in Section 3.1 (and detailed in [8–10]) resulting in the feature tensor $\mathcal{Y} \in \mathbb{R}^{k\times m\times n}$ that captures the correlation in the image data along the mode-1 dimension (row-pixels). To capture the correlations along the mode-3 dimension (column-pixels), we apply the tensor rotation operator

$$\texttt{rol}(\mathcal{Y}) = \bar{\mathcal{Y}} \in \mathbb{R}^{n\times m\times k}.$$

We then compute

$$\textbf{\textit{t-SVD}}(\bar{\mathcal{Y}}) = \bar{\mathcal{U}} * \bar{\mathcal{S}} * \bar{\mathcal{V}}^T$$

where $\bar{\mathcal{U}}$ contains the left singular matrices associated with the correlations in the mode-3 dimension of the data set. Similar to our prior work on Tensor PCA, we keep the first p lateral slices of $\bar{\mathcal{U}} = \bar{\mathcal{U}}_p$ to perform mode-3 dimensionality reduction. It should be noted that in general, the size of the reduced dimensional space for mode-1 (k) will not be the same as the size of the reduced dimensional space for mode-3 (p). We then compute the overall feature tensor via

$$\mathcal{W} = \bar{\mathcal{U}}_p^T * \bar{\mathcal{Y}} \in \mathbb{R}^{p\times m\times k}, \tag{4}$$

that capitalizes on correlations in both mode-1 and mode-3. Online classification is then performed in the reduced dimensional feature space $\mathcal{W}$ via single sample projection and nearest-neighbor search.

4. EXPERIMENTAL RESULTS

In this section, we compare our proposed 2DTPCA method with other multilinear subspace learning methods in the literature, namely, our prior work on Tensor PCA [10], HOSVD [2, 3], and MPCA [5]. In addition, although not "strictly" multilinear (tensor) methods, we compare against both 2DPCA [6] and (2D)^{2}PCA [7] in terms of classification accuracy and overall dimensionality reduction.

For our experimental validation, we choose four common data sets used in the literature, namely: (a) the ORL and extended Yale-B data sets that contain images of human faces under varying facial expressions and illumination directions [15–17]; (b) The COIL-100 data sets that contain images of 100 different objects being rotated about a single degree of freedom [18]. For computational efficiency, each image was resized from size 128×128 to 32×32; and (c) the classic MNIST data set of handwritten digits [19] where we randomly select 200 samples of each digit to produce a total data set of size 2000 images. For completeness, we briefly describe each of the four different data sets along with the number of training vs. testing images available in each. For each of the four data sets, 20 different classification runs were evaluated where a random 80/20 (training/testing) split was performed to ensure complete coverage of the training/testing space. Information regarding image size, training set and testing sets is outlined in Table 1.

Table 1: Specifications on the data sets used for experimental validation.

Data Set	Img. Size	Train	Test
ORL	32×32	280	120
Ext. Yale-B	32×32	756	306
COIL-100	32×32	1020	420
MNIST	20×20	1400	600

Table 2 illustrates the size of the reduced dimensional feature space for each of the six methods across all four data sets. We note that for Tensor PCA, the third dimension is always the dimension of mode-3 as this approach does not have the ability to reduce the dimensionality in mode-3. We also note that both 2DPCA and (2D)^{2}PCA only have two dimensions shown for their respective feature spaces. This results from the not being strictly tensor methods, however, post projection will result in similar values for the second dimension (i.e., the total number of samples in the data set). For comparison, when possible, we choose the same subspace dimensions (for both k and p) across all six approaches for each of the four data sets.

Table 3 shows the classification accuracy for all six methods applied to the data sets outlined in Table 1. Because the classification results are computed for 20 different cycles (each with a random 80/20 split), the we show the mean classification accuracy $\pm$ the standard deviation in classification accuracy across all 20 runs. As can be seen from the table, the proposed 2DTPCA approach outperforms all other methods for each of the four data sets. Moreover, it can be seen that these results hold true even in the case where the reduced dimensional subspace is larger than that of 2DTPCA.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a new approach to multilinear two dimensional and two directional principal component analysis. The approach builds upon our prior Tensor PCA by capitalizing on the statistical correlations in both the row- and column-pixels within an image data set, all while recasting the problem in a multilinear framework. Experimental results are presented on the ORL, Extended Yale-B, COIL-100, and MNIST data sets comparing our proposed approach with the current state of the art in both tensor and two-dimensional PCA approaches. In all experiments performed, for a similar reduction in dimensionality, the proposed 2DTPCA approach outperforms the state of the art in terms of classification accuracy. Future work will focus on methods to determine both subspace dimensions k and p as well as evaluations of the computational cost associated with computing such subspace reduction.

Table 2: Reduced dimensional feature space for each of the six different methods and all four data sets.

	2DTPCA	Tensor PCA	HOSVD	MPCA	2DPCA	(2D)^{2}PCA
ORL	$6 \times 280 \times 15$	$6 \times 280 \times 32$	$6 \times 280 \times 15$	$6 \times 280 \times 15$	6×32	6×15
Ext. Yale-B	$10 \times 756 \times 20$	$10 \times 756 \times 32$	$10 \times 756 \times 20$	$10 \times 756 \times 20$	10×32	10×20
COIL-100	$6 \times 1020 \times 10$	$6 \times 1020 \times 32$	$6 \times 1020 \times 10$	$6 \times 1020 \times 10$	6×32	6×10
MNIST	$6 \times 1400 \times 9$	$6 \times 1400 \times 20$	$6 \times 1400 \times 9$	$6 \times 1400 \times 9$	6×20	6×9

Table 3: Classification accuracy for each of the six different methods and all four data sets.

	2DTPCA	Tensor PCA	HOSVD	MPCA	2DPCA	(2D)^{2}PCA
ORL	$\mathbf{92.45 \pm 2.63}$	90.25 ± 2.47	89.77 ± 2.06	89.29 ± 1.72	88.79 ± 2.12	89.31 ± 2.11
Ext. Yale-B	$\mathbf{65.34 \pm 2.35}$	62.21 ± 2.37	61.41 ± 1.63	61.75 ± 2.05	59.26 ± 2.88	59.25 ± 2.40
COIL-100	$\mathbf{99.48 \pm 0.35}$	99.32 ± 0.34	99.32 ± 0.39	99.27 ± 0.49	99.40 ± 0.49	99.44 ± 0.40
MNIST	$\mathbf{93.68 \pm 1.19}$	93.35 ± 0.89	93.32 ± 1.00	93.53 ± 0.93	92.49 ± 0.94	92.80 ± 0.92

6. REFERENCES

[1] Ian T Jolliffe, "Principal components in regression analysis," in *Principal component analysis*, pp. 129–155. Springer, 1986.

[2] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, March 2000.

[3] Tamara G Kolda and Brett W Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[4] Ledyard R Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[5] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *IEEE transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[6] Jian Yang, David Zhang, Alejandro F Frangi, and Jingyu Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[7] Daoqiang Zhang and Zhi-Hua Zhou, "(2d) 2pca: Two-directional two-dimensional pca for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1-3, pp. 224–231, 2005.

[8] Ning Hao, Misha E. Kilmer, Karen S. Braman, and Randy C. Hoover, "New tensor decompositions with applications in facial recognition," *SIAM Journal on Imaging Science (SIIMS)*, vol. 6, no. 1, pp. 437–463, Feb. 2013.

[9] Misha E. Kilmer, Karen S. Braman, Ning Hao, and Randy C. Hoover, "Third order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, vol. 34, no. 1, pp. 148–172, Feb. 2013.

[10] Randy C. Hoover, Karen S. Braman, and Ning Hao, "Pose estimation from a single image using tensor decomposition and an algebra of circulants," in *Int. Conf. on Intel. Robots and Sys.*, 2011.

[11] Misha E. Kilmer, Carla D. Martin, and Lisa Perrone, "A third-order generalization of the matrix SVD as a product of third-order tensors," Tech. Rep. TR-2008-4, Tufts University, Department of Computer Science, October 2008.

[12] Misha E. Kilmer and Carla D. Moravitz Martin, "Factorization strategies for third-order tensors," *Linear Algebra and Its Applications*, , no. Special Issue in Honer of G.W.Stewart's 75^{th} birthday, 2009.

[13] Karen Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241 – 1253, 2010.

[14] Misha Kilmer, Karen Braman, and Ning Hao, "Third order tensors as operators on matrices: A theoretical and computational framework," Tech. Rep. TR-2011-01, Tufts University, Department of Computer Science, January 2011.

[15] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," vol. 19, no. 7, pp. 711–720, July 1997.

[16] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?," *Int. J. of Comp. Vis.*, vol. 28, no. 3, pp. 245–260, July 1998.

[17] David J. Kriegman, Peter N. Belhumeur, and Athinodoros S. Georghiades, "Representations for recognition under variable illumination," in *Shape, Contour and Grouping in Computer Vision*, 1999, pp. 95–131.

[18] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-100)," Tech. Rep. CUCS-006-96, Technical Report CUCS-006-96, Feb. 1996.

[19] "MNIST database of handwritten digits," 2018.