Kernelization of Tensor Discriminant Analysis with Application to Image Recognition

Cagri Ozdemir^{+}, Randy C. Hoover^{+}, Kyle Caudle*, Karen Braman*
$^{+}$Department of Electrical Engineering and Computer Science, *South Dakota Mines*
*Department of Mathematics, *South Dakota Mines*

***Abstract*—Multilinear discriminant analysis (MLDA), a novel approach based upon recent developments in tensor-tensor decomposition, has been proposed recently and showed better performance than traditional matrix linear discriminant analysis (LDA). The current paper presents a nonlinear generalization of MLDA (referred to as KMLDA) by extending the well known "kernel trick" to multilinear data. The approach proceeds by defining a new dot product based on new tensor operators for third-order tensors. Experimental results on the ORL, extended Yale B, and COIL-100 data sets demonstrate that performing MLDA in feature space provides more class separability. It is also shown that the proposed KMLDA approach performs better than the Tucker-based discriminant analysis methods in terms of image classification.**

***Index Terms*—tensor discriminant analysis, kernel method, image recognition, linear discriminant analysis.**

I. Introduction

Linear discriminant analysis (LDA) [1], [2] is one of the most popular shallow learning algorithms for feature extraction and widely used in many areas of image classification and pattern recognition for dimensionality reduction [3]–[5]. Traditionally, the reduced dimensional subspace of a large data set has been computed using a linear algebraic framework. In particular, each image is "row-scanned" into a column vector. However, converting two-dimensional (2D) image matrices into one-dimensional (1D) vectors is problematic in that: (a) it eliminates the spatial correlations within each image; and (b) it suffers from the curse of dimensionality and small sample size problem [6], [7]. In order to solve these problems, new methods have been proposed that rely on higher-order data structures that leave each image in its 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 higher-order discriminant analysis (HODA) [8]. In addition to HODA, discriminant analysis with tensor representation (DATER) [9] and general tensor discriminant analysis (GTDA) [10] provide iterative procedures to maximize the scatter ratio criterion. However, DATER does not converge over iterations. Although the iterative approximation method of GTDA converges, it converges to a local maximum. Thus, two effective algorithms, direct general discriminant analysis (DGTDA) and constrained multilinear discriminant analysis (CMDA) are proposed as outlined in [11]. DGTDA learns tensor subspaces by obtaining the global maximum scatter ratio without iteration, whereas CMDA guarantees convergence over iterations. All these supervised learning methods rely on Tucker decompositions and the n-mode product [12], [13].

Recently, a new approach to supervised learning has been introduced by extending traditional LDA to a multilinear framework (referred to as multilinear discriminant analysis (MLDA)) [14]. The approach is based on recent developments based upon Fourier theory and an algebra of circulants as outlined in [15]–[19]. It was shown that under the right tensor multiplication operator, a third order tensor can be written as a product of third order tensors in which the left tensor is a collection of eigenmatrices, the middle tensor is a front-face diagonal (denoted as f-diagonal) tensor of eigen-tuples, and the right tensor is the tensor inverse of the eigenmatrices resulting in a tensor-tensor eigenvalue decomposition (**t-eig**) that is similar to its matrix counterpart.

In an effort to improve and perfect the performance of MLDA, in this paper, we propose a kernel-based MLDA (referred to as KMLDA) that provides nonlinear generalization of MLDA. The kernel idea was originally applied in Support Vector Machines [20] and kernel Fisher discriminant analysis [21], [22] for second-order tensors. However, we introduce the polynomial kernels for third-order tensors that provide the nonlinear mapping explicitly. Moreover, we introduce the wavelet transform based third-order tensor definitions and mathematical operations using a special structured block matrix which are computationally more efficient compared to the fast Fourier transform (FFT) and the discrete cosine transform (DCT) based third-order operators [18], [23], [24].

The remainder of the paper is organized as follows. In Section II, we discuss the mathematical foundations of the tensor operators and the tensor-tensor eigendecomposition. In Section III, we propose our new framework for KMLDA. In Section IV, we compare the proposed method with the state-of-art Tucker structure based discriminant analysis and MLDA methods for classification applications, and Section V concludes the paper.

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.

II. Mathematical Background

In this section, we will define the wavelet transform-based third-order tensor definitions and mathematical operations using a special structured block matrix determined by the frontal slices of the third-order tensors. Fundamental to the results presented in this section is motivated by the tensor definitions based upon Fourier theory and algebra of circulants as outlined in [15]–[19], [23], [25]–[27].

A. Tensor operators

In the wavelet transform, a signal in the time domain is decomposed by passing it through high-pass filter (resulting in detail coefficients) and low-pass filter (resulting in approximation coefficients) to produce low-pass and high-pass wavelet coefficients referred to as a "*level-1 decomposition*". The low-pass version can be further decomposed by again passing it to a set of low-pass and high pass filters referred to as "*level-2 decomposition*". This process can be further continued to a pre-defined level as outlined in [28], [29]. While there are many different types of wavelets, arguably the most common are the Haar wavelet [30] and Daubechies wavelet [31]. In our work, we use the Haar wavelet due to its low computation cost and simplicity to apply as compared to other wavelets. The discrete Haar wavelet transform can be expressed in matrix form for each level decomposition [32]. Thus, the discrete Haar wavelet forward transformation matrix for level-1 decomposition can be written as:

$$H = \begin{bmatrix} h_0 & h_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & h_0 & h_1 \\ g_0 & g_1 & 0 & \cdots & \cdots & \cdots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & g_0 & g_1 \end{bmatrix}, \quad (1)$$

where h_0 and h_1 are scaling function coefficients, whereas g_0 and g_1 are wavelet function coefficients.

It will be convenient to break a tensor $\mathcal{A}$ in $\mathbb{R}^{\ell \times m \times n}$ up into various slices 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 $\mathcal{A}^{(j)} \equiv \mathcal{A}(:, :, j)$. In order to discuss our new definitions we must first introduce the block matrix that can be diagonalized by the discrete Haar wavelet transform matrix H (illustrated in (1)). We call this block matrix as "*block dwt matrix*", denoted by **bdwt** for short. For example, consider the tensor $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ with $\ell \times m$ frontal slices then $\textbf{bdwt}(\mathcal{A})$ can be written as follows:

$$\textbf{bdwt}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(2)} & 0 & 0 & \cdots & 0 \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & 0 & 0 & \cdots & \vdots \\ 0 & 0 & \mathcal{A}^{(3)} & \mathcal{A}^{(4)} & \cdots & \vdots \\ 0 & 0 & \mathcal{A}^{(4)} & \mathcal{A}^{(3)} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathcal{A}^{(n-1)} & \mathcal{A}^{(n)} \\ 0 & 0 & \cdots & \cdots & \mathcal{A}^{(n)} & \mathcal{A}^{(n-1)} \end{bmatrix}. \quad (2)$$

A new block-diagonal form (3) can be constructed via left and right multiplication by a DWT matrix (1).

$$(H_n \otimes I_\ell) \cdot \textbf{bdwt}(\mathcal{A}) \cdot (H_n^T \otimes I_m) = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{bmatrix}, \quad (3)$$

where each of the D_i is a $\ell \times m$ matrix, I_ℓ is a $\ell \times \ell$ identity matrix, I_m is a $m \times m$ identity matrix, H_n is the $n \times n$ DWT matrix defined in (1), H_n^T is its transpose, and $\otimes$ is the Kronecker product.

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

Definition 2. If $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$, then $\textbf{unfold}(\mathcal{A})$ takes tensor $\mathcal{A}$ and returns a block $\ell n \times m$ matrix.

$$\textbf{unfold}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(n)} \end{bmatrix}.$$

The operation that takes $\textbf{unfold}(\mathcal{A})$ back to tensor form is the **fold** operator:

$$\mathcal{A} = \textbf{fold}\big(\textbf{unfold}(\mathcal{A})\big).$$

Definition 3. If $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$, then the **unbdwt** operator takes matrix $\textbf{bdwt}(\mathcal{A})$ and returns tensor $\mathcal{A}$.

$$\mathcal{A} = \textbf{unbdwt}\big(\textbf{bdwt}(\mathcal{A})\big).$$

Definition 4. Let $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ and $\mathcal{B} \in \mathbb{R}^{m \times \ell \times n}$. Then the wavelet product denoted by $\mathcal{A} *_w \mathcal{B} \in \mathbb{R}^{\ell \times \ell \times n}$ is defined as:

$$\mathcal{A} *_w \mathcal{B} = \textbf{fold}(\textbf{bdwt}(\mathcal{A}) \cdot \textbf{unfold}(\mathcal{B})),$$

where $\cdot$ is standard matrix multiplication.

Definition 5. Let $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$. Then the tensor transpose of the tensor $\mathcal{A}$ denoted by $\mathcal{A}^T \in \mathbb{R}^{m \times \ell \times n}$ is defined as:

$$\mathcal{A}^T = \textbf{unbdwt}\Big(\big(\textbf{bdwt}(\mathcal{A})\big)^T\Big).$$

Definition 6. Let $\mathcal{A} \in \mathbb{R}^{m \times m \times n}$. Then the tensor inverse of the tensor $\mathcal{A}$ denoted by $\mathcal{A}^{-1} \in \mathbb{R}^{m \times m \times n}$ is defined as:

$$\mathcal{A}^{-1} = \textbf{unbdwt}\Big(\big(\textbf{bdwt}(\mathcal{A})\big)^{-1}\Big).$$

Definition 7. 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 in the transform domain,

$$\mathcal{I} = \tilde{\mathcal{I}} \times_3 H_n^{-1},$$

where $\tilde{\mathcal{I}}(:,:,i) = \mathbf{I}$ for $i = 1, \ldots, n$ and $\mathbf{I}$ is the $m \times m$ identity matrix. H_n is the $n \times n$ Haar wavelet level-1 transformation matrix defined in (1). We note that $\times_3$ is a mode-3 product [13], [24]. This operation has the same effect as taking inverse wavelet transform along each tube in $\tilde{\mathcal{I}}$.

Definition 8. The tensor norm used through this paper is the Frobenious norm which for the tensor $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ is given by:

$$||\mathcal{A}||_F = \sqrt{\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n}\big(\mathcal{A}(i,j,k)\big)^2}.$$

Definition 9. An idempotent tensor $\mathcal{A} \in \mathbb{R}^{m\times m\times n}$ is the tensor which, when multiplied by itself, yields itself.

$$\mathcal{A} = \mathcal{A} *_w \mathcal{A}.$$

B. Wavelet Tensor Eigendecomposition

The final tool necessary for a multilinear LDA is to define a tensor-tensor eigenvalue decomposition. Motivated by the tensor eigendecomposition based on the Fourier transform-based tensor operators in [14], we will introduce the wavelet tensor eigendecomposition referred to as the **t-eig$_\mathbf{w}$** using the special block structure and tensor operators given in II-A.

Definition 10. Let $\mathcal{A} \in \mathbb{R}^{m\times m\times n}$, then the **t-eig$_\mathbf{w}$** of the tensor $\mathcal{A}$ is defined as:

$$\mathcal{A} = \mathcal{P} *_w \mathcal{D} *_w \mathcal{P}^{-1},$$

where $\mathcal{P} \in \mathbb{R}^{m\times m\times n}$ is a non-singular and $\mathcal{D} \in \mathbb{R}^{m\times m\times n}$ is a f-diagonal tensor such that the frontal slices are diagonal. A graphical illustration of the **t-eig$_\mathbf{w}$** is shown in Fig. 1.

Recall equation (3):

$$(H_n \otimes I_m)\cdot \mathbf{bdwt}(\mathcal{A})\cdot(H_n^T \otimes I_m) = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{bmatrix}.$$

To construct the **t-eig$_\mathbf{w}$**, the matrix eigenvalue decomposition is performed on each of the D_i as $D_i = P_i \Sigma_i P_i^{-1}$. Then we can write:

$$\begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_n \end{bmatrix} = \begin{bmatrix} P_1 & & \\ & \ddots & \\ & & P_n \end{bmatrix}\begin{bmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_n \end{bmatrix}\begin{bmatrix} P_1^{-1} & & \\ & \ddots & \\ & & P_n^{-1} \end{bmatrix}.$$

Applying $(H_n^T \otimes I_m)$ to the left and $(H_n \otimes I_m)$ to the right of each of the block diagonal matrices on the right hand side results in each being the **bdwt** structure. We can use the **unbdwt** operator given in **Definition 3** to take them back into tensor form as following:

$$\mathcal{P} = \mathbf{unbdwt}\left((H_n^T \otimes I_m)\begin{bmatrix} P_1 & & \\ & \ddots & \\ & & P_n \end{bmatrix}(H_n \otimes I_m)\right),$$

$$\mathcal{D} = \mathbf{unbdwt}\left((H_n^T \otimes I_m)\begin{bmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_n \end{bmatrix}(H_n \otimes I_m)\right),$$

$$\mathcal{P}^{-1} = \mathbf{unbdwt}\left((H_n^T \otimes I_m)\begin{bmatrix} P_1^{-1} & & \\ & \ddots & \\ & & P_n^{-1} \end{bmatrix}(H_n \otimes I_m)\right),$$

Therefore, that results in the decomposition:

$$\mathcal{A} = \mathcal{P} *_w \mathcal{D} *_w \mathcal{P}^{-1}.$$

For computational efficiency, we can compute the tensor eigendecomposition for any invertible transforms using spectral domain operations similar to the computation of the t-SVD using the FFT in place of spatial domain operations [15], [17], [18], [24], [33]. Previously, the FFT-based tensor eigendecomposition is defined in [14]. However, the DWT has time complexity of $O(N)$, whereas the FFT and DCT are both $O(NlogN)$, hence the reduction in computational complexity. TABLE I shows the time complexity of the tensor eigendecomposition of tensor $\mathcal{A} \in \mathbb{R}^{m\times m\times n}$ based on three most widely used invertable linear transforms, it is clearly seen that the DWT-based eigendecomposition (**t-eig$_\mathbf{w}$**) is the fastest method among others.

TABLE I
THE TIME COMPLEXITY OF THE FFT, DCT, AND DWT BASED TENSOR EIGENDECOMPOSITION.

Tensor $\mathcal{A} \in \mathbb{R}^{m\times m\times n}$	
FFT	$O\big(m^2 nlog(n)\big)$
t-eig after FFT	$O(2m^3)$
t-eig with FFT	$O\big(m^2 nlog(n)\big) + O(2m^3)$
DCT	$O\big(m^2 nlog(n)\big)$
t-eig after DCT	$O(m^3)$
t-eig with DCT	$O\big(m^2 nlog(n)\big) + O(m^3)$
DWT	$O(m^2 n)$
t-eig after DWT	$O(m^3)$
t-eig with DWT	$O(m^2 n) + O(m^3)$

III. PROPOSED KMLDA METHOD

In this section, we provide an overview of prior MLDA work proposed in [14] and illustrate some of the shortcomings of such an approach. We then turn our attention to nonlinear extension of MLDA work that improves the recognition performance of MLDA.

A. Overview of MLDA

To keep the current work self-contained, we present a brief overview of the work on MLDA outlined in [14]. However, unlike MLDA in [14] (which was based Fourier theory), we use the wavelet transform-based operators and definitions as outlined in Section II.

Consider the situation where we have a collection of ℓ image samples, each of size $m \times n$ pixels. We can construct our data tensor $\mathcal{A} \in \mathbb{R}^{m\times \ell\times n}$ where each lateral slice of $\mathcal{A}$ (i.e., $\mathcal{A}_i$,

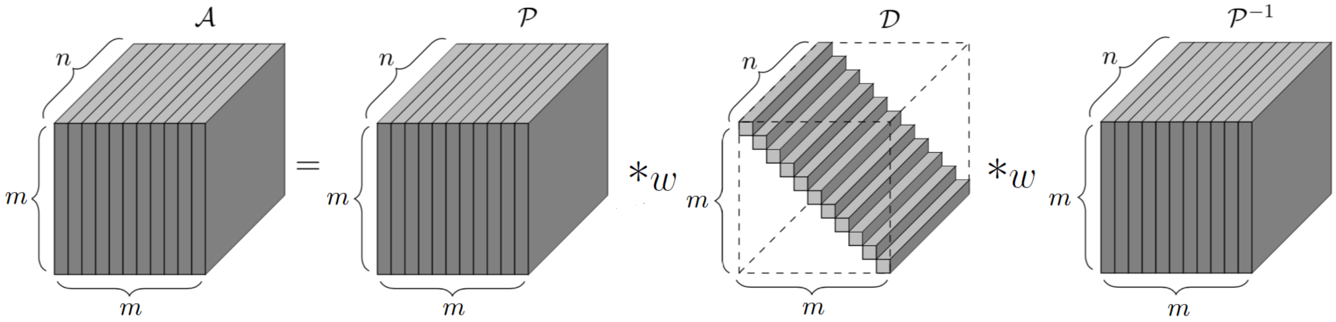

Fig. 1. Graphical illustration of the **t-eig$_\mathbf{w}$**.

for $i = 1, 2, \ldots, \ell$) is an $m \times n$ sample image. From this construction, the within-class scatter tensor can be written as:

$$\mathcal{S}_w = \sum_{i=1}^{C} \sum_{\mathcal{A}_j \in c_i} (\mathcal{A}_j - \mathcal{M}_i) *_w (\mathcal{A}_j - \mathcal{M}_i)^T, \tag{4}$$

where C is the total number of classes, $\mathcal{A}_j \in \mathbb{R}^{m\times 1\times n}$ is the j^{th} lateral slice of class i denoted by c_i, and $\mathcal{M}_i \in \mathbb{R}^{m\times 1\times n}$ is the mean of class c_i. The transpose operator and the multiplication operator are outlined in **Definition 5** and **Definition 4** respectively. We define the between-class scatter tensor as:

$$\mathcal{S}_b = \sum_{i=1}^{C} N_i (\mathcal{M}_i - \mathcal{M}) *_w (\mathcal{M}_i - \mathcal{M})^T, \tag{5}$$

where $\mathcal{M}$ is the mean of all data samples and N_i is the number of samples in the class i.

The goal is to find a projection tensor $\mathcal{U}$ so as to maximize the between-class scatter while minimizing the within-class scatter (generally written as a scatter ratio). It can be shown that the projection tensor in question can be computed by solving the generalized tensor eigenvalue problem as:

$$(\mathcal{S}_w{}^{-1} *_w \mathcal{S}_b) *_w \mathcal{U} = \mathcal{D} *_w \mathcal{U}, \tag{6}$$

where $\mathcal{U} = [\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_k] \in \mathbb{R}^{m\times k\times n}$ are the eigenmatrices corresponding to the k largest eigen-tuples of the diagonal tensor $\mathcal{D}$ using the tensor norm defined in **Definition 8** and the tensor inverse operation is outlined in **Definition 6**. Note that similar to its matrix counterpart, there are at most $C-1$ nonzero eigentuples of (6), therefore the projection space has at most dimension $C-1$. The projection tensor $\mathcal{U}$ can be obtained via the **t-eig$_\mathbf{w}$** defined in Section II-B. Finally, the tensor data $\mathcal{A} \in \mathbb{R}^{m\times \ell\times n}$ can be projected onto the new multilinear subspace $\mathcal{U} \in \mathbb{R}^{m\times k\times n}$ resulting in the new reduced feature tensor.

$$\mathcal{B} = (\mathcal{U}^T *_w \mathcal{A}) \in \mathbb{R}^{k\times \ell\times n}. \tag{7}$$

B. Proposed KMLDA Extension

Suppose that $\mathcal{A} \in \mathbb{R}^{m\times \ell\times n}$ is a data tensor consists of $m \times n$ sample images as lateral slices, i.e., $\mathcal{A}_i \in \mathbb{R}^{m\times 1\times n}$, for $i = 1, 2, \ldots, \ell$. Define Φ as a function that maps the tensor input to feature space, i.e., $\Phi(\mathcal{A}_i) \in \mathbb{R}^{t\times 1\times n}$. We note that we generally have $t \geq m$. Motivated by the definition of the kernel of two vectors outlined in [34], [35], we define the kernel of two tensors $\mathcal{A}_1 \in \mathbb{R}^{m\times 1\times n}$ and $\mathcal{A}_2 \in \mathbb{R}^{m\times 1\times n}$ by:

$$\mathbf{k}(\mathcal{A}_1, \mathcal{A}_2) = \Phi(\mathcal{A}_1)^T *_w \Phi(\mathcal{A}_2) \in \mathbb{R}^{1\times 1\times n}, \tag{8}$$

which results in a tubal scalar and measures similarity between the two tensors [24]. Because the function Φ is usually not available, computation of dot products in feature spaces can be done efficiently by using the kernel functions. There exists different types of kernels functions defined for vectors [36], [37]. In this paper, we define polynomial kernels for third-order tensors based on the inner product definition between two third order tensors.

$$\textbf{Polynomial: } \mathbf{k}(\mathcal{A}_1, \mathcal{A}_2) = (\mathcal{A}_1^T *_w \mathcal{A}_2 + \mathbf{c})^d \tag{9}$$

where $\mathbf{c} \in \mathbb{R}^{1\times 1\times n}$ is a tubal-scalar constant, and d is a scalar constant. In our experiments, we use $\mathbf{c}_j = 1$ for $j = 1 \cdots n$, where $\mathbf{c}_j$ denotes the j^{th} entry of that tubal scalar, and $d = 0.8$.

In order to perform MLDA in feature space, we first map the training and testing tensors into a feature space. Suppose $\mathcal{A} \in \mathbb{R}^{m\times \ell\times n}$ and $\mathcal{B} \in \mathbb{R}^{m\times p\times n}$ are the training and testing tensors respectively where each lateral slice is a $m \times n$ sample image. The training tensor in the feature space $\mathcal{A}^\Phi \in \mathbb{R}^{\ell\times \ell\times n}$ is the kernel tensor of the training data and the testing tensor in the feature space $\mathcal{B}^\Phi \in \mathbb{R}^{\ell\times p\times n}$ is the kernel tensor of the training data and the testing data. They can be defined using the polynomial tensor kernel in (8) as:

$$\mathcal{A}^\Phi = \begin{bmatrix} \mathbf{k}(\mathcal{A}_1, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_1, \mathcal{A}_2) & \cdots \mathbf{k}(\mathcal{A}_1, \mathcal{A}_\ell) \\ \mathbf{k}(\mathcal{A}_2, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_2, \mathcal{A}_2) & \cdots \mathbf{k}(\mathcal{A}_2, \mathcal{A}_\ell) \\ \vdots & \vdots & \vdots \\ \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_2) & \cdots \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_\ell) \end{bmatrix}, \tag{10}$$

$$\mathcal{B}^\Phi = \begin{bmatrix} \mathbf{k}(\mathcal{A}_1, \mathcal{B}_1) & \mathbf{k}(\mathcal{A}_1, \mathcal{B}_2) & \cdots \mathbf{k}(\mathcal{A}_1, \mathcal{B}_p) \\ \mathbf{k}(\mathcal{A}_2, \mathcal{B}_1) & \mathbf{k}(\mathcal{A}_2, \mathcal{B}_2) & \cdots \mathbf{k}(\mathcal{A}_2, \mathcal{B}_p) \\ \vdots & \vdots & \vdots \\ \mathbf{k}(\mathcal{A}_\ell, \mathcal{B}_1) & \mathbf{k}(\mathcal{A}_\ell, \mathcal{B}_2) & \cdots \mathbf{k}(\mathcal{A}_\ell, \mathcal{B}_\ell) \end{bmatrix}. \tag{11}$$

The within-class scatter tensor can now be reformulated as:

$$\mathcal{S}_w^\Phi = \sum_{i=1}^{C} \mathcal{W}^{\Phi_i} *_w \mathcal{C}^{\Phi_i} *_w \mathcal{C}^{\Phi_i} *_w (\mathcal{W}^{\Phi_i})^T,$$

where $\mathcal{C}^{\Phi_i} \in \mathbb{R}^{k \times k \times n}$ is the centering tensor of the i^{th} class which is an idempotent tensor defined in **Definition 9**.

$$\mathcal{C}^{\Phi_i} = \mathcal{C}^{\Phi_i} *_w \mathcal{C}^{\Phi_i}.$$

Therefore,

$$\mathcal{S}_w^{\Phi} = \sum_{i=1}^{C} \mathcal{W}^{\Phi_i} *_w \mathcal{C}^{\Phi_i} *_w (\mathcal{W}^{\Phi_i})^T, \tag{12}$$

where $\mathcal{W}^{\Phi_i} \in \mathbb{R}^{\ell \times k \times n}$ is the kernel tensor of the entire training data and the training data of the i^{th} class. If, for example, the first three lateral slices of the tensor $\mathcal{A}$ belong to the i^{th} class, we can write $\mathcal{W}^{\Phi_i} \in \mathbb{R}^{\ell \times 3 \times n}$ as:

$$\mathcal{W}^{\Phi_i} = \begin{bmatrix} \mathbf{k}(\mathcal{A}_1, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_1, \mathcal{A}_2) & \mathbf{k}(\mathcal{A}_1, \mathcal{A}_3) \\ \mathbf{k}(\mathcal{A}_2, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_2, \mathcal{A}_2) & \mathbf{k}(\mathcal{A}_2, \mathcal{A}_3) \\ \vdots & \vdots & \vdots \\ \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_1) & \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_2) & \mathbf{k}(\mathcal{A}_\ell, \mathcal{A}_3) \end{bmatrix}.$$

Definition 11. The tensor $\mathcal{C}^{\Phi} \in \mathbb{R}^{m \times m \times n}$ is called a centering tensor when right multiplied with a tensor $\mathcal{A} \in \mathbb{R}^{m \times m \times n}$ has the same effect as subtracting the mean of all lateral slices of the tensor and left multiplied with a tensor has the same effect as subtracting the mean of all horizontal slices of the tensor.

The mean of the lateral slices of the tensor $\mathcal{A}$ can be computed as:

$$\mathcal{M} = \frac{1}{k} \sum_{i=1}^{k} \mathcal{A}^{(i)}.$$

To remove the mean $\mathcal{M} \in \mathbb{R}^{k \times 1 \times n}$ from the tensor $\mathcal{A}$, we subtract the mean from the lateral slices of the tensor $\mathcal{A}$.

$$\bar{\mathcal{A}} = \mathcal{A} - \mathcal{M}.$$

$\mathcal{C}^{\Phi} \in \mathbb{R}^{k \times k \times n}$ is a centering tensor can be written as:

$$\mathcal{C}^{\Phi} = \mathcal{I} - \frac{1}{k}\mathcal{J},$$
$$\mathcal{J} = \tilde{\mathcal{J}} \times_3 H_n^{-1},$$

where $\tilde{\mathcal{J}} \in \mathbb{R}^{k \times k \times n}$ is a tensor whose each entry is one, $\mathcal{I} \in \mathbb{R}^{k \times k \times n}$ is the identity tensor defined in (7) and H_n is the $n \times n$ Haar wavelet level-1 transformation matrix defined in (1). We remind the reader that $\times_3$ is a mode-3 product and details can be found in [13], [24]. Multiplication by the centering tensor is a convenient analytical tool of of removing the mean from a tensor.

$$\bar{\mathcal{A}} = \mathcal{A} *_w \mathcal{C}^{\Phi}.$$

The fact that left multiplication removes the mean of the horizontal slices from the tensor can be similarly shown.

The between-class scatter tensor can be written as:

$$\mathcal{S}_b^{\Phi} = \sum_{i=1}^{C} (\mathcal{M}^{\Phi_i} - \mathcal{M}^{\Phi}) *_w (\mathcal{M}^{\Phi_i} - \mathcal{M}^{\Phi})^T, \tag{13}$$

where $\mathcal{M}^{\Phi_i} \in \mathbb{R}^{\ell \times 1 \times n}$ is the mean of the lateral slices of the kernel tensor $\mathcal{W}^{\Phi_i} \in \mathbb{R}^{\ell \times k \times n}$, whereas $\mathcal{M}^{\Phi} \in \mathbb{R}^{\ell \times 1 \times n}$ is the mean of the lateral slices of the kernel tensor $\mathcal{A}^{\Phi} \in \mathbb{R}^{\ell \times \ell \times n}$.

The projection tensor $\mathcal{U}_w^{\Phi}$ can then be computed by solving the generalized tensor eigenvalue problem as:

$$(\mathcal{S}_w^{\Phi^{-1}} *_w \mathcal{S}_b^{\Phi}) *_w \mathcal{U}^{\Phi} = \mathcal{D}^{\Phi} *_w \mathcal{U}^{\Phi}, \tag{14}$$

where $\mathcal{U}^{\Phi} \in \mathbb{R}^{\ell \times C-1 \times n}$ consists of eigenmatrices. Note that there are at most $C-1$ nonzero eigen-tuples of (14), therefore the projection space has at most dimension $C-1$. Finally, the training tensor data $\mathcal{A}^{\Phi} \in \mathbb{R}^{\ell \times \ell \times n}$ and the testing training tensor data $\mathcal{B}^{\Phi} \in \mathbb{R}^{\ell \times k \times n}$ can be projected onto the new multilinear subspace $\mathcal{U}^{\Phi} \in \mathbb{R}^{\ell \times C-1 \times n}$ resulting in the new reduced feature tensor.

$$\bar{\mathcal{A}}^{\Phi} = \mathcal{U}^{\Phi^T} *_w \mathcal{A}^{\Phi} \in \mathbb{R}^{C-1 \times \ell \times n}, \tag{15}$$

$$\bar{\mathcal{B}}^{\Phi} = \mathcal{U}^{\Phi^T} *_w \mathcal{B}^{\Phi} \in \mathbb{R}^{C-1 \times k \times n}. \tag{16}$$

Once the projections $\bar{\mathcal{A}}^{\Phi}$ and $\bar{\mathcal{B}}^{\Phi}$ have been computed, classification is performed via nearest neighbor search for the closest match using the tensor norm defined in **Definition 8**.

IV. Experimental Results

In this section, we compare our proposed KMLDA method with other multilinear subspace learning methods in the literature, namely, DATER [9], HODA [8], CMDA [11] , DGTDA [11], and MLDA [14].

For our experimental validation, we choose three common data sets used in the literature, namely: (a) the ORL data set consists of 10 different images of 40 distinct subjects under varying lighting conditions, facial expressions, and facial details [38]; (b) the extended Yale-B data set that contains images of human faces under varying facial expressions and illumination directions [39]. There are 38 subjects and 59 images per subject; (c) the COIL-100 data set that contains images of 100 different objects being rotated about a single degree of freedom to obtain 72 poses per object [40]. For computational efficiency, we use the first 20 objects with 72 poses of each object to produce total data set of 1440 images. A subset of example images for each of the three data sets are illustrated in Fig. 3. We note that, for computational efficiency, each image in the all databases were resized to 32 × 32. For each of the four data sets, 20 different classification runs were evaluated where a random 70/30 (training/testing) split was performed to ensure complete coverage of the training/testing space. Information regarding image size, training set, testing set, and class size is outlined in Table II.

TABLE II
Specifications on the data sets used for experimental validation.

Data Set	Image Size	Train Set	Test Set	# of Classes (C)
ORL	32 × 32	280	120	40
Ext. Yale-B	32 × 32	1569	673	38
COIL-100	32 × 32	1008	432	20

Fig. 2 shows the classification performance of MLDA and KMLDA methods on the three data sets. It is clearly seen that

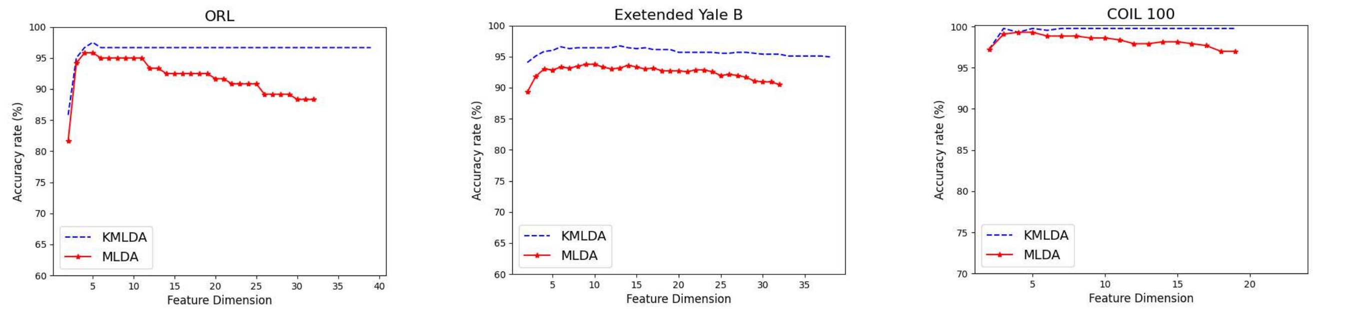

Fig. 2. Classification accuracy comparison on the ORL, extended Yale B, and COIL 100 databases.

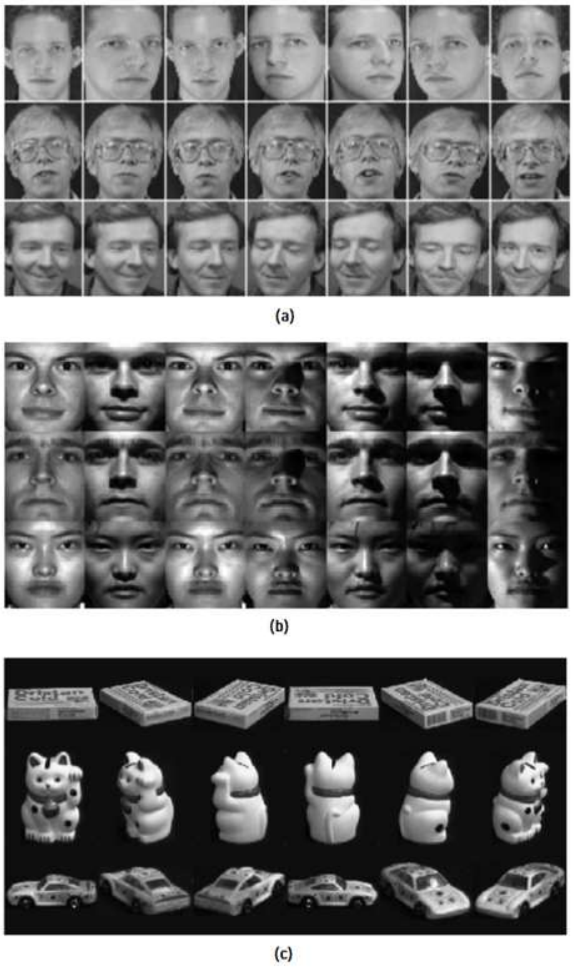

Fig. 3. A subset of images from each data set used in this research. (a) The ORL data set, (b) the extended Yale-B face database, (c) the COIL-100 data set.

the defined tensor kernel space yields more class separability for each feature dimension on the three data sets. We note that since the projection space has at most dimension $C-1$ (where C is the total number of classes), KMLDA may reach the maximum projection dimension, whereas the image size may not allow MLDA to reach the maximum as seen on the ORL and extended Yale B data sets.

Table III shows the classification accuracy for all six methods applied to the data sets outlined in Table II. Because the classification results are computed for 20 different cycles (each with a random $70/30$ split), we show the mean classification accuracy $\pm$ the standard deviation in classification accuracy across all 20 runs. We note that for classification, we selected a $5-$dimensional subspace for all six methods and performed nearest-neighbor search with tensor Frobenious norm defined in Section II **Definition 8**. As can be seen from Table III, the proposed KMLDA approach outperforms all other methods across all three data sets evaluated.

TABLE III
CLASSIFICATION ACCURACY FOR EACH OF THE SIX DIFFERENT METHODS AND ALL THREE DATA SETS.

Method	**ORL**	**Extended Yale B**	**COIL 100**
DATER [9]	56.75 ± 2.75	86.73 ± 2.13	94.02 ± 1.45
HODA [8]	75.04 ± 3.44	62.35 ± 4.66	97.55 ± 0.67
CMDA [11]	71.42 ± 4.01	74.71 ± 4.08	95.56 ± 1.01
DGTDA [11]	72.29 ± 2.85	61.45 ± 4.04	97.76 ± 0.69
MLDA[1] [14]	94.83 ± 2.56	92.31 ± 1.23	98.22 ± 0.63
KMLDA	$\mathbf{96.21 \pm 1.72}$	$\mathbf{95.81 \pm 0.76}$	$\mathbf{99.16 \pm 0.59}$

V. CONCLUSIONS AND FUTURE WORK

This paper presented a new approach to tensor discriminant analysis computing the discriminant features in some feature space which is nonlinear related to the input space. Furthermore, the wavelet transform-based MLDA was introduced using the wavelet transform based mathematical operations which are more computationally efficient compared to the FFT and the DCT-based operators. An analysis was presented in the context of classification of the ORL, the extended Yale B, and the COIL-100 data sets. It was illustrated that for all three data sets, our current approach outperformed the Tucker-based discriminant analysis and MLDA methods in terms of classification accuracy. Future work will be dedicated to reproducing kernel space applied to the Tucker-based discriminant analysis as well as evaluating radial basis kernel functions for third-order tensors.

[1] MLDA is computed using the wavelet based tensor operators and $\mathbf{t\text{-}eig_w}$ in Section II-A. Classification performance of MLDA based on the FFT [14] might be slightly different than the DWT-based MLDA.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, July 1997.

[2] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, Sept. 1936.

[3] M. Mahdianpari, B. Salehi, F. Mohammadimanesh, B. Brisco, S. Mahdavi, M. Amani, and J. E. Granger, "Fisher linear discriminant analysis of coherency matrix for wetland classification using PolSAR imagery," *Remote Sensing of Environment*, vol. 206, pp. 300–317, 2018.

[4] R. Fu, Y. Tian, P. Shi, and T. Bao, "Automatic detection of epileptic seizures in EEG using sparse CSP and fisher linear discrimination analysis algorithm," *Journal of medical systems*, vol. 44, no. 2, pp. 1–13, 2020.

[5] A. Kalsoom, M. Maqsood, M. A. Ghazanfar, F. Aadil, and S. Rho, "A dimensionality reduction-based efficient software fault prediction using fisher linear discriminant analysis (FLDA)," *The Journal of Supercomputing*, vol. 74, no. 9, pp. 4568–4602, Sept. 2018.

[6] K. Fukunaga, *Introduction to statistical pattern recognition*. Elsevier, 2013.

[7] A. Sharma and K. K. Paliwal, "Linear discriminant analysis for the small sample size problem: an overview," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 3, pp. 443–454, 2015.

[8] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear theory and its applications, IEICE*, vol. 1, no. 1, pp. 37–68, 2010.

[9] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 526–532.

[10] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 10, pp. 1700–1715, 2007.

[11] Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2524–2537, 2014.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

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

[14] R. C. Hoover, K. Caudle, and K. Braman, "Multilinear discriminant analysis through tensor-tensor eigendecomposition," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, Dec.2018, pp. 578–584.

[15] M. E. Kilmer, C. D. Martin, and L. Perrone, "A third-order generalization of the matrix SVD as a product of third-order tensors," *Tufts University, Department of Computer Science, Tech. Rep. TR-2008-4*, Oct. 2008.

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

[17] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[18] M. E. Kilmer, K. Braman, N. Hao, and R. 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*, vol. 34, no. 1, pp. 148–172, Feb. 2013.

[19] D. F. Gleich, C. Greif, and J. M. Varah, "The power and arnoldi methods in an algebra of circulants," *Numerical Linear Algebra with Applications*, vol. 20, no. 5, pp. 809–831, Oct. 2013.

[20] R. Soentpiet *et al.*, *Advances in kernel methods: support vector learning*. MIT press, 1999.

[21] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*. Ieee, 1999, pp. 41–48.

[22] B. Ghojogh, F. Karray, and M. Crowley, "Fisher and kernel Fisher discriminant analysis: Tutorial," *arXiv preprint arXiv:1906.09436*, 2019.

[23] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, "Facial recognition using tensor-tensor decompositions," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 437–463, Feb. 2013.

[24] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra and its Applications*, vol. 485, pp. 545–570, 2015.

[25] R. C. Hoover, K. S. Braman, and N. Hao, "Pose estimation from a single image using tensor decomposition and an algebra of circulants," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2011, pp. 2928–2934.

[26] C. Ozdemir, R. C. Hoover, and K. Caudle, "2DTPCA: A new framework for multilinear principal component analysis," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 344–348.

[27] C. Ozdemir, R. C. Hoover and K. Caudle, "Fast tensor singular value decomposition using the low-resolution features of tensors," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2021, pp. 527–533.

[28] G. Strang and T. Nguyen, *Wavelets and filter banks*. SIAM, 1996.

[29] A. Jensen and A. la Cour-Harbo, *Ripples in mathematics: the discrete wavelet transform*. Springer Science & Business Media, June 2001.

[30] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.

[31] I. Daubechies, "Orthonormal bases of compactly supported wavelets ii. variations on a theme," *SIAM Journal on Mathematical Analysis*, vol. 24, no. 2, pp. 499–519, Mar. 1993.

[32] P. Porwik and A. Lisowska, "The haar-wavelet transform in digital image processing: its status and achievements," *Machine graphics and vision*, vol. 13, no. 1/2, pp. 79–98, Nov. 2004.

[33] C. Ozdemir, R. C. Hoover, K. Caudle, and K. Braman, "High-order multilinear discriminant analysis via order-*n* tensor eigendecomposition," *arXiv preprint arXiv:2205.09191*, 2022.

[34] R. Herbrich, *Learning kernel classifiers: theory and algorithms*. MIT press, 2001.

[35] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[36] V. Vapnik and V. Vapnik, "Statistical learning theory wiley," *New York*, vol. 1, no. 624, p. 2, 1998.

[37] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," *Advances in neural information processing systems*, vol. 12, 1999.

[38] (2018) AT&T database of of faces. [Online]. Available: http://cam-orl.co.uk/facedatabase.html

[39] D. J. Kriegman, P. N. Belhumeur, and A. S. Georghiades, "Representations for recognition under variable illumination," in *Shape, Contour and Grouping in Computer Vision*. Springer, 1999, pp. 95–131.

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