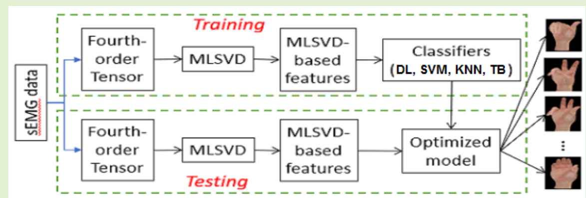
A Tensor-Based Approach Using Multilinear SVD for Hand Gesture Recognition From sEMG Signals

Sibasankar Padhy, *Member, IEEE*

***Abstract*—With the increasing number of sensors/channels, hand gesture recognition from multiple channels has attracted the attention of the researchers in recent years. The analysis of a variety of gestures is still a challenging task in real-time applications such as in wearable devices because of the large number of channels used. This paper proposes a tensor-based approach using multilinear singular value decomposition (MLSVD) for hand gesture recognition where all available channels were used during training whereas only a single channel was used for recognition of new gestures. Tensor decompositions have very limited use in gesture recognition using surface Electromyography (sEMG) despite these signals naturally have multi-way structures. The sEMG data of different subjects were first segmented to reshape it as a fourth-order tensor of the form *channels* × *time* × *gestures* × *subjects*. The MLSVD was applied to model the tensor and extract features, which were then fed into various classifiers such as support vector machine (SVM), K-nearest neighbors (KNN), TreeBagger (TB), and dictionary learning (DL) classifiers in order to compare their performance. The proposed method was evaluated on three publicly available databases (NinaPro, CapgMyo (DB-a, DB-b and DB-c), CSL-HDEG) by performing intra-session, inter-session and inter-subject evaluations on each database. The experimental results indicated that the proposed method achieved the best accuracy (Acc) using DL, achieving Acc of 77.6% and 89.6% with CapgMyo DB-b and CSL-HDEMG databases, respectively, during inter-session evaluation, and 75.2%, 75.4%, 68.3%, and 67.7% with NinaPro, CapgMyo DB-b, CapgMyo DB-c, and CSL-HDEMG databases, respectively, during inter-subject evaluation. The proposed method performed better during both inter-session and inter-subject evaluations than the state-of-the-art methods.**



This work intends to recognize hand gestures from sEMG data, reshaped as fourth-order tensor (*channels × time × gestures × subjects*), using MLSVD. MLSVD-based features were obtained by coupling two modes, followed by solving Kronecker product equation.

***Index Terms*—EMG, gesture recognition, Kronecker product equation, tensor decomposition, multilinear singular value decomposition, support vector machine.**

I. Introduction

ELECTROMYOGRAPHY (EMG) is a well-known diagnostic tool used for measuring and evaluating the electrical activity produced by skeletal muscles, where electrodes are placed on the surface of the skin, called as surface EMG (sEMG) or by inserting wire or needles inside muscles, called as intramuscular EMG (iEMG). Study on sEMG-based muscle activities involves a wide range of applications including hand prosthetic control where hand gesture recognition is carried out in, broadly called, muscle-computer interface (MCI) field [1]–[5].

Manuscript received September 27, 2020; revised November 29, 2020; accepted November 29, 2020. Date of publication December 4, 2020; date of current version February 5, 2021. The associate editor coordinating the review of this article and approving it for publication was Prof. Chang-Hee Won.

The author is with the School of Electronics, Vellore Institute of Technology, Vellore 632014, India (e-mail: sibasankar.padhy@vit.ac.in).

Digital Object Identifier 10.1109/JSEN.2020.3042540

For accurate gesture recognition, a multiple number of electrodes are used that help analyze the activity of groups of muscles involved in certain movements and for analysis of spatio-temporal dynamics of muscle activities as well [2]. The existing gesture recognition methods are classified as either sparse multichannel sEMG (s-sEMG) or high-density sEMG (HD-sEMG) based methods depending on the number of electrodes used [6]. Though multichannel EMG data are represented in matrix form with indices *channels* × *time*, it naturally turns to a higher-order tensor data with the addition of extra modes due to repetitions of subjects or gestures or data collected in different sessions. It has been shown in many works from different fields including in EMG related works (for example in [7]–[12]) that the traditional vector- or matrix based methods do not take advantage of this higher-order data structure as they cannot capture multi-way couplings [13], [14]. This means that some information about the interaction between modes may be lost in those approaches. Another limitation associated with these multiple electrode recording systems is ease-of-use or precise positioning of electrodes [15]. So, the reduction of redundant electrodes could be a potential solution for this. The hypothesis of this work is that higher-order tensor decomposition will be more beneficial for hand gesture recognition, by reducing number of channels and coupling information from different modes.

Tensors, which are higher-order generalizations of scalars, vectors, or matrices, are the effective ways to store multimodal or multivariate data while preserving the inherent structure of data. Not only this representation retains important characteristics but the tensor decomposition methods also help discover hidden components present within multimodal data. Though firstly proposed almost hundred years back, tensor analysis is gaining increasing interest in the last two decades in the analysis of multimodal data [13], [16] including on different biomedical signals and images like electroencephalogram, electrocardiogram, magnetic resonance imaging [8], [10], [17]–[22]. However, tensor decompositions have very limited use in EMG applications that include elbow movement classification using non-negative Tucker decomposition [23], muscle synergy analysis using parallel factor analysis and Tucker decomposition [11], [12], and task-dependent modulation using canonical polyadic decomposition [24]. However, the application of tensor decomposition for hand gesture recognition is lacking.

In this work, the effectiveness of tensor and tensor decomposition in analyzing spatio-temporal changes of muscle activities is demonstrated. Here, multilinear singular value decomposition (MLSVD), a prominent tensor decomposition technique, is applied on the higher-order sEMG tensor to extract features from the decomposed matrices that help recognize gestures accurately. Another advantage of the proposed work is the number of channels used. Here, a model is created by training all channels whereas a reduced number of channels or a single channel is used during testing. This may increase the possibility of deploying the method in real-time applications. This paper, however, does not present a readily wearable interface rather presents the basis for follow-up work on EMG signals using tensor-based techniques. The mail goal of this paper is to access whether the extracted features from the decomposed matrices from reduced number of channels with different machine learning methods succeed in hand gesture recognition. The comprehensive approach conducted in this study has the following contributions:

- Reshaping matrix form sEMG data to a fourth-order tensor,
- Modeling the tensor using MLSVD,
- Representing gestures of a channel as a Kronecker product equation (KPE) (See Subsection II-A for Kronecker Product),
- Extracting features from decomposed matrices that characterize gestures,
- Extensive analysis on intra-session, inter-session, and inter-subject evaluations,
- Comparison of different machine learning methods for gesture recognition.

The rest of the paper is organized as follows: Section II discusses the useful concepts of tensor such as notations, definitions, and the tensor decomposition method, MLSVD, used in this work. Section III discusses the databases used in this work. The proposed tensor-based approach is discussed in Section IV. Results and discussions of the proposed method with comparison evaluation performance using different classifiers are presented in Section V. Finally, conclusions are given in Section VI.

II. Tensor Preliminaries

A. Notations and Definitions

An Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multiway array that varies over N modes (e.g. sensors, time samples, etc.) with dimensions $I_1, I_2, \cdots, I_N$, respectively. We denote scalars, vectors, and matrices of respective tensor orders $N = 0, 1,$ and 2 by italic lower-case letters x, boldface lower-case letters $\mathbf{x}$, boldface upper-case letters $\mathbf{X}$, respectively, whereas, higher-order tensors ($N > 2$) by calligraphic letters $\mathcal{X}$. For example, a third-order tensor is denoted by $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and its elements as x_{ijk} with $i = 1, \ldots, I$, $j = 1, \ldots, J$ and $k = 1, \ldots, K$.

A mode-n tensor *fiber* is the vector that is obtained by fixing all except the nth index. For example, the mode-1 fiber of a third-order tensor is analogue to a column vector and is denoted by $\mathbf{x}_{:jk}$. The mode-2 and the mode-3 fibers are the row and the tube vectors and are denoted by $\mathbf{x}_{i:k}$ and $\mathbf{x}_{ij:}$, respectively.

Slices of a tensor are obtained by fixing all but two indices, and its structure is similar to a matrix. A third-order tensor comprises of three types of slices, namely 'horizontal', 'vertical' and 'frontal' slices. The 'horizontal' slice where index i is fixed is denoted as $\mathbf{X}_{i::}$. Other two slices, keeping j and k fixed, are represented as $\mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$, respectively.

A tensor $\mathcal{X}$ can be matricized by stacking all fibers of a certain mode-n in a matrix. This is also known as the mode-n unfolding of $\mathcal{X}$ and is written as $\mathbf{X}^{(n)}$.

The mode-n product between a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \cdot_n \mathbf{A}$ and is defined element-wise as: $(\mathcal{X} \cdot_n \mathbf{A})_{i_1 i_2 \cdots j \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_n \cdots i_N} a_{j i_n}$. Hence, each mode-$n$ vector of the tensor $\mathcal{X}$ is multiplied with the matrix $\mathbf{A}$, i.e., $(\mathcal{X} \cdot_n \mathbf{A})_{(n)} = \mathbf{A}\mathbf{X}^{(n)}$.

Another useful product in tensor algebra is Kronecker product, denoted as $\otimes$. If $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ are two matrices, the Kronecker product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IK \times JL}$ is represented as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \tag{1}$$

B. Tensor Decomposition

Two well-known tensor decomposition methods are the canonical polyadic decomposition (CPD) [25] and the Tucker decomposition (TD) [26]. Later on, these two are adopted or presented differently with different names with little variations by different authors. The CPD is also known as the parallel factor analysis (PARAFAC) or canonical decomposition

(CANDECOMP). Similarly, TD is also known as Multilinear Singular Value Decomposition (MLSVD) or Higher-Order SVD (HOSVD). In this work, MLSVD is used to decompose the tensor introduced later in the next section.

Multilinear Singular Value Decomposition

The MLSVD is the higher-order generalization of the SVD (a matrix decomposition technique) for tensors [16], [27]. It has been used in various signal processing and machine learning applications such as for dimensionality reduction or compression, feature extraction, classification, etc. (for specific applications the readers are encouraged to refer the citations in [13], [14], [16], [27]. The MLSVD of an Nth-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is given by

$$\mathcal{Y} = \mathcal{Z} \cdot_1 \mathbf{U}^{(1)} \cdot_2 \mathbf{U}^{(2)} \cdot_3 \cdots \cdot_n \mathbf{U}^{(n)} \tag{2}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ are orthogonal factor matrices and can be found by applying SVD on unfolded versions of the tensor in respective mode, and $\mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is all-orthogonal and ordered core tensor and governs interaction between these factor matrices.

The low-rank approximation of $\mathcal{X}$ is given by

$$\mathcal{Y} = \hat{\mathcal{Z}} \cdot_1 \hat{\mathbf{U}}^{(1)} \cdot_2 \hat{\mathbf{U}}^{(2)}) \cdot_3 \cdots \cdot_n \hat{\mathbf{U}}^{(n)} \tag{3}$$

where $\hat{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times r_n}$ are the truncated matrices obtained by keeping the first r_n columns of $\mathbf{U}^{(n)}$ [27]. The multilinear-rank of $\hat{\mathcal{Z}}$ is obtained by using an energy criterion rule to determine the minimum number of singular values to be kept along each mode as:

$$r_n = \arg\min_i \sum_{l=1}^{i} \sigma_l^{(n)} \quad \text{s.t.} \quad \frac{\sum_{l=1}^{i} \sigma_l^{(n)}}{\sum_{l=1}^{I_n} \sigma_l^{(n)}} > 0.95 \tag{4}$$

where $\sigma_l^{(n)}$ is the lth singular value of the matrix obtained from the SVD of the unfolding $\mathbf{Y}^{(n)}$. From now on, for a simpler representation purpose, the Hat symbols $(\hat{\cdot})$ in (3) have been omitted.

III. Databases

Several databases containing multi-channel sEMG recordings of hand gesture recognition are publicly available, and a summary of these is presented in [3]. In this work, we used three freely available databases, NinaPro ((Non-Invasive Adaptive Hand Prosthetics) [28]–[30], CapgMyo [2], [6], and CSL-HDEMG [31]. The former one is a s-sEMG database whereas the latter two are HD-sEMG databases. A detailed discussion on the databases is beyond the scope of this work, we present briefly on the same here and encourage the readers to refer the above cited original sources.

A. NinaPro Database

It is the most widely accepted benchmark database for the hand gesture recognition problem and consists of ten sub-databases (DB1-DB10) to this date, recorded by using 10-16 sparsely located electrodes. For a quick summary of these, we encourage the readers to refer [3]. In this work, the first sub-database DB1 was considered for comparison purposes. It consists of 52 gestures performed by 27 subjects recorded by using 10 sparsely located electrodes.

B. CSL-HDEMG Database

The sEMG signals were recorded at a sampling rate of 2,048 Hz, by using 192 electrodes, covering the upper forearm muscles of 5 subjects performing 27 gestures, where each subject was recorded over 5 sessions and performed 10 trials for each gesture in each session. Every eighth channel contains redundant information and, hence, were ignored from data analysis leading to a total of 168 channels of usable data [31].

C. CapgMyo Database

This database consists of 3 sub-databases (DB-a, DB-b, and DB-c), recorded using 128 channels (8 × 16 electrode array) with sampling frequency of 1000 Hz from 23 subjects; 8 hand gestures were obtained from 18 and 10 subjects in DB-a and DB-b, respectively, and 12 basic movements of the fingers were obtained from 10 of the 23 subjects in DB-c. The total 20 gestures are subsets of gestures in NinaPro database. The gestures in DB-a and DB-b correspond to Nos. 13-20 whereas gestures in DB-c to Nos. 1-12 in the NinaPro database.

The records in this database are available in two forms: raw data and preprocessed data. In the latter, power-line interference was removed by using a second-order Butterworth band-stop filter (45-55 Hz). As the resulting gestures performed by the subjects may not perfectly match the label as a result of human reaction times, only the static part of the movement was used to evaluate the recognition algorithms. For each gesture and trial, the middle one-second window of data (e.g. 1000 samples in case of CapgMyo database) was used. The same preprocessing approach, i.e., power-line interference removal and extracting one-second data for further processing purpose, was also followed for NinaPro and CSL-HDEMG databases.

IV. Methods

A. Tensor Construction

The first step to create a tensor model is to prepare the sEMG matrix data in higher-order tensor form using a suitable tensorization (process of transforming the lower-order data to higher-order) technique. 'Segmentation' is one of the tensorization techniques where lower-order data is segmented and smaller segments are thereafter juxtaposed to form higher-order tensor. The sEMG data intuitively with two modes *channels* × *time* has many epochs where each corresponds to a specific hand gesture or task. Gestures of all channels after segmentation were juxtaposed to form a fourth-order tensor of the form *channels* × *time* × *gestures* × *subjects*. Each frontal slice has temporal information of all channels corresponding to a specific gesture whereas each horizontal slice in total has a channel information and each vertical slice gives information about timestamp of each channel and gesture.

B. Tensor Model Construction Using MLSVD

The fourth-order sEMG tensor $\mathcal{Y} \in \mathbb{R}^{C \times T \times G \times S}$ with modes *channels*, *time*, *gestures*, and *subjects* for hand gesture recognition using truncated MLSVD can be modeled using (3) as follows:

$$\mathcal{Y} = \mathcal{Z} \cdot_1 \mathbf{U} \cdot_2 \mathbf{V} \cdot_3 \mathbf{W} \cdot_4 \mathbf{X} \tag{5}$$

where M, N, P and Q are reduced dimensions along the different modes, $\mathbf{U} \in \mathbb{R}^{C\times M}$, $\mathbf{V} \in \mathbb{R}^{T\times N}$, $\mathbf{W} \in \mathbb{R}^{G\times P}$ and $\mathbf{X} \in \mathbb{R}^{S\times Q}$ are orthonormal factor matrices, and $\mathcal{Z} \in \mathbb{R}^{M\times N\times P\times Q}$ is the core tensor that governs the interaction between these factor matrices. The column vectors of the above four orthonormal matrices span the *channel*, *time*, *gestures* and *subjects* space, respectively. Observe in (5) Hat symbols ($\hat{.}$) have been omitted from the truncated matrices and core tensor just for simpler representation.

C. MLSVD-Based Gesture Recognition: Testing Phase

Given the model and factor matrices in (5), the recognition of a new gesture q using a channel p can be computed by solving Kronecker Product Equations (KPE). Use of KPEs for ECG applications and image processing were done in [9], [32], [33] where KPEs were solved for a third-order tensor. In this work, this has been extended for a fourth-order tensor as per the requirement. Setting up KPEs that help recognize hand gestures later for this problem is discussed here:

The time series information of all subjects $\mathcal{Y} \in \mathbb{R}^{1\times t\times 1\times s}$ at a particular channel (say, pth channel) by a particular gesture (say, qth gesture) can be represented as

$$\mathcal{Y}_{p,q} = \mathcal{Z} \cdot_1 \mathbf{u}_p^T \cdot_2 \mathbf{V} \cdot_3 \mathbf{w}_q^T \cdot_4 \mathbf{X} \tag{6}$$

where $\mathbf{u}_p^T$ and $\mathbf{w}_q^T$ are pth and qth row of $\mathbf{U}$ and $\mathbf{W}$, respectively. Equivalently, each $\mathcal{Y}_{p,q}$ of $\mathcal{Y}$ is a slice ($\mathbf{Y}_{p:q:}$) (Ref: Subsection II-A) that can be modeled by KPEs. Unfolding over the second and the fourth mode can equivalently be expressed as

$$\begin{aligned}\mathbf{Y}_{p,q}^{(2,4)} &= (\mathbf{X} \otimes \mathbf{V})\mathbf{Z}^{(2,4)}(\mathbf{w}_q \otimes \mathbf{u}_p)\\ &= \mathbf{B}(\mathbf{w}_q \otimes \mathbf{u}_p)\end{aligned} \tag{7}$$

where the rows and columns of multilinear subspace $\mathbf{B} = (\mathbf{X} \otimes \mathbf{V})\mathbf{Z}^{(2,4)}$ form basis for subjects and time series, respectively. The coefficients $\mathbf{u}_p$ and $\mathbf{w}_q$ describe the data $\mathbf{Y}_{p,q}^{(2,4)}$ in the subspace $\mathbf{B}$, i.e., they correspond to the gesture of a channel.

Next, we explain how to recognize new gestures from sEMG multiple channels. First, a fourth-order tensor $\bar{\mathcal{Y}}$ of new data was created as in Subsection IV-A followed by computation of the MLSVD to obtain the different mode factor matrices and the core tensor. As every gesture of a channel can be expressed as a KPE, a new gesture (*q'*) from a channel (*p'*) can be recognized by solving following KPE, i.e., by finding coefficient vectors in the multilinear basis $\mathbf{B}$:

$$\bar{\mathbf{Y}}_{p',q'}^{(2,4)} = \mathbf{B}(\bar{\mathbf{w}}_q \otimes \bar{\mathbf{u}}_p) \tag{8}$$

The solution of (8) gives the estimates $\tilde{\mathbf{u}}_p$ and $\tilde{\mathbf{w}}_q$ that express the new data in the *channel* and *gesture* subspace. The coefficients $\tilde{\mathbf{u}}_p$ and $\tilde{\mathbf{w}}_q$ were compared with the rows of $\mathbf{U}$ and $\mathbf{W}$ (5), respectively, using the Frobenius norm of the difference (after fixing scaling and sign invariance) as in [32], [33].

D. Classification

In the final stage, a (set of) new gesture(s) of a channel/channels were assigned with the label having the closest match. Three classifiers such as K-Nearest Neighbors (KNN) with $K = 7$, support vector machine (SVM) with linear (SVM-lin), polynomial (SVM-pol), and radial basis function (SVM-rbf) kernels, and treebagger (TB) were used for the comparison purpose. The optimization of hyperparameters was done with F-fold and leave-one-out cross-validation techniques. Although the traditional classifiers mentioned earlier do the job, it is suggested in the literature that dictionary learning approaches such as in [34]–[37] perform better for the experiments that involve intra- and inter- class variations. In this work, the centralized class-specific dictionary pair learning (DL) method proposed in [36] was also used for classification purpose. The rows of $\mathbf{U}$ and $\mathbf{W}$ along with the labels were used as the training database and the coefficients $\tilde{\mathbf{u}}_p$ and $\tilde{\mathbf{w}}_q$ as the testing input features.

The recognition or classification performance was carried out in two ways: intra-session – where a classification model was created by training a part of the data of a subject from one session and testing of the model was performed on another part from that same session, and inter-session or inter-subject – where a classification model was created by training data from a set of subjects and testing was performed on another set of subjects. In all the experiments, it was ensured that the train and the test sets were independent of each other, i.e., there was no overlapping of data on the train and the test sets.

E. Performance Evaluation Metrics

To evaluate the performance of the developed method, a statistical analyses using one-way Analysis of Variance (ANOVA) followed by a post-hoc test analysis were performed. These tests were carried out to verify whether means of different gesture across sessions or subjects are significantly different or not. ANOVA assesses the relative size of variance among category means compared to the average variance within categories. The post-hoc test using Tukey's honest significant difference (hsd) test then was executed to perform specific comparisons for the purpose of discovering the origin(s) of the difference.

The results of gesture recognition were further quantified by calculating accuracy of the system.

F. Computational Analysis

In the following, the complexity of the algorithm is analyzed.

In the proposed method, the fourth-order tensor $\mathcal{Y} \in \mathbb{R}^{C\times T\times G\times S}$ is decomposed into four orthonormal matrices ($\mathbf{U} \in \mathbb{R}^{C\times M}$, $\mathbf{V} \in \mathbb{R}^{T\times N}$, $\mathbf{W} \in \mathbb{R}^{G\times P}$ and $\mathbf{X} \in \mathbb{R}^{S\times Q}$) and a core tensor $\mathcal{Z} \in \mathbb{R}^{M\times N\times P\times Q}$ (See (5)). The computation of MLSVD involves the finding of the left orthonormal factor matrices using SVD along each mode. That means to find, for example, the matrix $\mathbf{U}$, SVD is applied on the mode-1 unfolding of tensor $\mathcal{Y}$, i.e., $\mathbf{Y}^{(1)}$ (Ref. Section II-A). The computational complexity of SVD of a matrix $\mathbf{A}$ of size $m \times n$ is $\mathcal{O}(4mnr)$ where r is the rank of the matrix [38], [39].

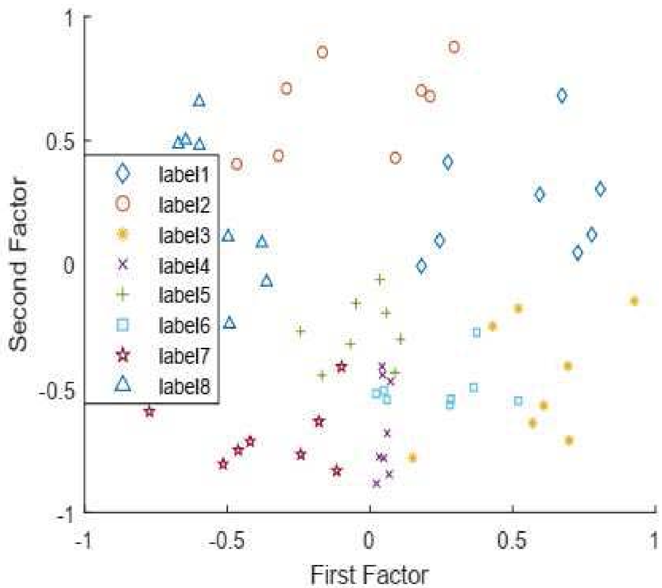


Fig. 1. Scatter plot of the first and the second factors of factor matrix **W** (corresponding to gestures mode) obtained from the trained data of CapgMyo DB-b during intra-session evaluation.

So, the computational complexities to find **U** (SVD of $\mathbf{Y}^{(1)}$), **V** (SVD of $\mathbf{Y}^{(2)}$), **W** (SVD of $\mathbf{Y}^{(3)}$), and **X** (SVD of $\mathbf{Y}^{(4)}$) are $\mathcal{O}(4MCTGS)$, $\mathcal{O}(4NCTGS)$, $\mathcal{O}(4PCTGS)$, and $\mathcal{O}(4QCTGS)$, respectively. Next, the n-mode product of a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ has a computational complexity of $\mathcal{O}(\prod_{i=1, i\neq n}^{N} I_i J_n)$. So, the computational complexity of (6) is $\mathcal{O}(NPQ + TPQ + TQ + TS)$. In (7), $(\mathbf{X} \otimes \mathbf{V})$ is a $ST \times NQ$ matrix, $\mathbf{Z}^{(2,4)}$ is a $NQ \times MP$ matrix, and $(\mathbf{w}_q \otimes \mathbf{u}_p)$ is a row vector $MP \times 1$, whose matrix-matrix multiplication has computational complexity of $\mathcal{O}(MNPQST + MPST) = \mathcal{O}(MNPQST)$. In the whole procedure, the orders of magnitude C, T, G, and S are usually $\mathcal{O}(10)$ or $\mathcal{O}(10^2)$, $\mathcal{O}(10^3)$, $\mathcal{O}(10)$, and $\mathcal{O}(10)$, respectively, and computation of factor matrices using MLSVD (5) is an upper bound. So MLSVD is guaranteed to converge and runs with the amount of computation in $\mathcal{O}(10^7)$ – $\mathcal{O}(10^8)$ at each repetition, which is acceptable for using multiple records of a database.

V. Results and Discussion

We illustrate the proposed MLSVD-based method on the databases mentioned in Section III for hand gesture recognition. All experiments in this work were performed using MATLAB R2019b in a personal computer with CPU @2.70 GHz and 4 GB RAM, and computations using Tensorlab [40]. Tensorlab provides various tools for tensor computations, tensor decompositions, tensorization techniques, etc.

A. Intra-Session Evaluation

In this experiment, a classification model was created using some part of the data of all subjects from one session as a training set and the remaining of the same session as a test set. Total number of gestures in a record were split in 80% training and 20% test set and then tensor is constructed by following this for all subjects. For example, the tensor constructed from the sub-database DB-a (of CapgMyo database) having 18 subjects, recorded with 8 hand gestures with 10 repetitions has the size of the training set $\mathcal{Y} \in \mathbb{R}^{128\times1000\times64\times18}$ and the test set $\mathcal{Y} \in \mathbb{R}^{128\times1000\times16\times18}$. Both sets had equal ratios of different gestures in all databases. Similar method is followed for all the databases.

As discussed earlier we deal with the coefficients of the factor matrices **U** and **W** essentially along two modes (*channels* and *gestures*) for the training and the testing purpose. It is important to see the spread of the features across the sessions in each database, but the results are shown for one database only. Figure 1 shows the scatter plot of the first and the second factors of trained **W** of 8 gestures of CapgMyo DB-b. These 8 gestures are 'Thumb up', 'Extension of index and middle, flexion of the others', 'Flexion of ring and little finger, extension of the others', 'Thumb opposing base of little finger', 'Abduction of all fingers', 'Fingers flexed together in fist', 'Pointing index', and 'Adduction of extended fingers' [2]. This figure suggests that the factor distributions of these gestures follow a clear separability. A statistical significant association was detected between the first and the second factors of both **U** and **W** ($p < 0.01$). This finding has an important implication as to the MLSVD-based factor matrices for recognizing different hand gestures.

To validate the effectiveness of the proposed method for this experiment, the coefficients of the factor matrices as explained in Subsection IV-C were used for gesture recognition using different classifiers. As mentioned earlier one of the advantages of the proposed method is the ability of recognizing gestures using single channel. Figure 2(a) shows the recognition accuracy of a randomly chosen segment (CapgMyo DB-b) on a per-channel basis. The original label of the shown segment is gesture 2 (Extension of index and middle, flexion of the others). We can observe that the most of the channels (49 out of 128 channels) recognize it as gesture 2 with the best accuracy of 97.3%. Although a set of channels outcomes this, it is not always true that the same set of channels will outcome similar results for the same gesture across the segments. The next most of channels (24) identify this segment as gesture 3 (Flexion of ring and little finger, extension of the others) which may be due to the similar type of muscle movements in both gestures.

Table I shows the recognition accuracy of MLSVD using features from **W** corresponding to *gestures* mode only in the first row and from both **U** and **W** corresponding to *channels* and *gestures* modes (See IV-C) in the second row, whereas the last row presents the p-value for each database. A statistical significant association was detected between the first and the second factors of both **U** and **W** with $p < 0.05$, $p < 0.01$, and $p < 0.05$ for NinaPro, CapgMyo, and CSL-HDEMG databases, respectively. Regarding the classifiers' performance, the dictionary learning classifier using combined features improved the recognition accuracy by 3.7–8.4%. When the number of gestures or labels are large (e.g. 52 in NinaPro DB-1), classifier using features from *gestures* mode only are not able to learn effectively for a large number of labels.

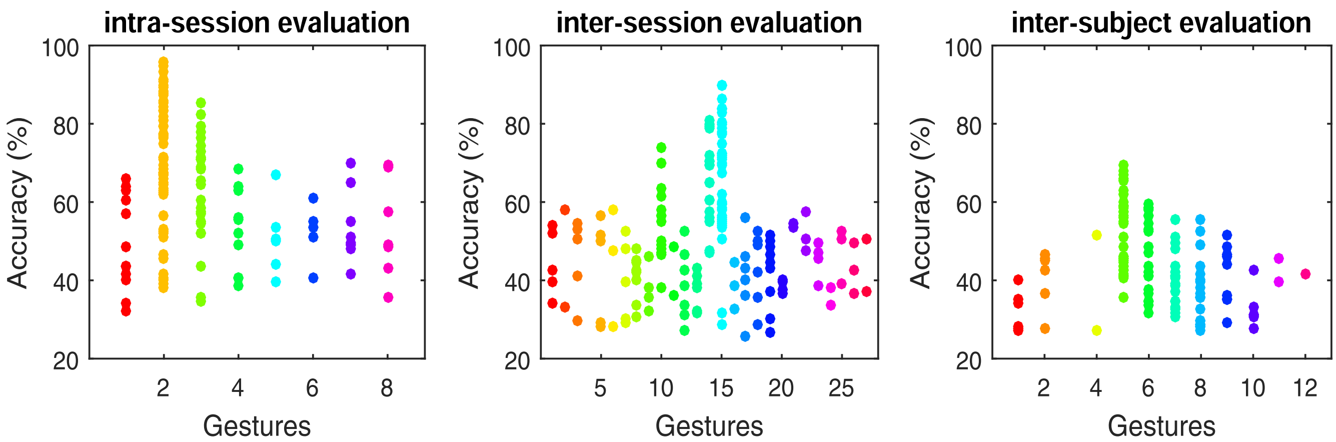


Fig. 2. An instance showing single-channel recognition accuracy of a randomly chosen segment during (a) intra-session of CapgMyo DB-b, Labeled gesture is 2 (Extension of index and middle, flexion of the others), (b) inter-session, (c) inter-subject evaluations. Accuracy results are shown using dictionary learning classifier [36].

TABLE I

COMPARISON OF THE RECOGNITION ACCURACY (IN %) OF THE INTRA-SESSION EVALUATION RESULTS OF THE PROPOSED METHODS WITH OTHER METHODS FROM LITERATURE. FOR EACH SESSION, RESULTS ARE AVERAGED OVER THE SUBJECTS. NUMBERS SHOWN FOR MLSVD METHOD IN THE FIRST TWO ROWS CORRESPOND TO SVM-RBF CLASSIFIER AND IN THE THIRD ROW TO DICTIONARY LEARNING APPROACH. 'X' REPRESENTS NOT APPLICABLE

Methods	DB	S1	S2	S3	S4	S5	Avg
Atzori [29]	1	76.1	x	x	x	x	76.1
Amma [31]	3	87.4	91.6	93.0	91.0	88.9	90.3
ConvNet [2]	1	96.7	x	x	x	x	96.7
	2		98.6	99.2	x	x	
	3	95.9	97.7	98.0	95.5	96.2	96.8
MLSVD	1	86.4	x	x	x	x	86.4
(F = 5)		94.8	x	x	x	x	94.8
		97.4	x	x	x	x	97.4
		(0.021)					
	2	90.2	93.4	94.2	89.7	x	
		93.5	97.8	98.1	93.1	x	
		97.1	98.7	98.5	97.9	x	
		(0.004)	(0.007)	(0.006)	(0.009)		
	3	94.3	94.1	93.6	92.9	95.0	94.0
		97.6	98.3	98.7	97.5	98.3	98.1
		98.4	98.1	99.2	98.6	98.9	98.6
		(0.025)	(0.038)	(0.014)	(0.023)	(0.018)	

*For DB: 1 – NinaPro, 2 – CapgMyo (S1: DB-a, S2: DB-b, S3: DB-b, S4: DB-c), 3 – CSL-HDEMG.
**Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$, $p < 0.01$, and $p < 0.05$ for DB-1, DB-2, and DB-3, respectively.)

The addition of another feature set from *channels* mode improved the accuracy by 8.4%, while this improvement is around 2% for other databases where the number of gestures is between 7 and 12. The adopted dictionary pair learning classifier in [36] performs better than the SVM-rbf classifier achieving accuracy 97.4%, 98.0%, and 98.6% for NinaPro, CapgMyo, and CSL-HDEMG databases, respectively.

Table I also shows the average five-fold cross-validation recognition accuracy of the MLSVD-based method along with the comparison among other methods from the literature. The results shown are for one session each from NinaPro DB-1, CapgMyo DB-a and DB-c, two sessions from CapgMyo DB-b, and five sessions from CSL-HDEMG. Though the performance of the proposed method is almost similar to ConvNet [6], the recognition accuracy using the MLSVD-based technique has been increased significantly; improvement of 18.7% and 21.3% using SVM-rbf and DL classifiers, respectively, over Atzori's work in NinaPro database [2], and 3.7–8.3% over the method by Amma *et al.* in CSL-HDEMG database [31]. The results with CSL-HDEMG database using MLSVD-based method are slightly better than the ConvNet method.

B. Inter-Session Evaluation

In this experiment as stated earlier, the inter-session evaluation was conducted by training on a set of sessions data of a subject and testing on another set of sessions from that subject, and hence the size of the tensor $\mathcal{Y}$ along the fourth mode during training is one ($S = 1$). That is the training and the testing data belonged to the same subject but were recorded in different sessions or days. Inter-session evaluation is possible on the database that has more than one session counts (e.g., CSL-HDEMG and CapgMyo DB-b). Depending on the number of sessions data available, the ratio between the training and the testing data vary; 50:50 for CapgMyo DB-b and 80:20 for CSL-HDEMG as the former has two sessions data whereas the latter has five sessions data for a subject. Unlike the previous experiment, recognition accuracy measures were evaluated using the leave-one-out cross-validation technique, (i.e., one session, in turn, was used as the testing set). The same evaluation procedure was followed as in the previous studies [2], [31].

Figure 2(b) shows the recognition accuracy of a randomly chosen segment (CSL-HDEMG DB) on a per-channel basis. The original label of the shown segment is gesture 15 (Pinch

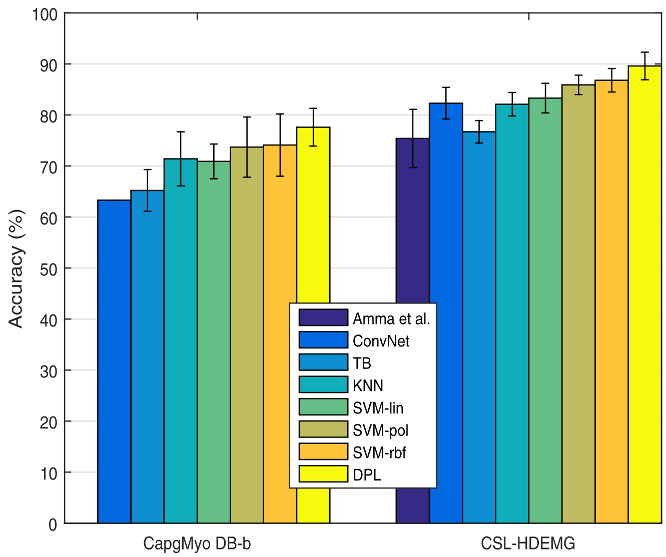


Fig. 3. Comparison of the recognition accuracy (in %) of the inter-session evaluation results of the proposed method with other methods from literature. Best results of [31] and [2] are shown here. Null hypothesis is rejected when $H_0 = 0 (p < 0.05$ for both CapgMyo DB-b and CSL-HDEMG databases).

TABLE II

COMPARISON OF THE RECOGNITION ACCURACY (IN %) OF THE INTER-SUBJECT EVALUATION RESULTS OF THE PROPOSED METHOD WITH OTHER METHODS FROM LITERATURE. NUMBERS SHOWN FOR MLSVD METHOD IN THE FIRST TWO ROWS CORRESPOND TO SVM-RBF CLASSIFIER AND IN THE THIRD ROW TO DICTIONARY LEARNING APPROACH. SVM RESULTS ARE ONLY SHOWN AS IT GIVES GOOD RESULTS AMONG OTHER CLASSIFIERS. 'X' REPRESENTS NOT APPLICABLE

DB	1	2-b	2-c	3
AL [41]	40.0	x	x	x
ConvNet [2]	67.4	55.3	35.1	x
MLSVD	71.8 (4.16)	61.4 (6.42)	60.6 (2.85)	59.4 (3.14)
	74.9 (4.30)	71.8 (5.78)	63.2 (2.43)	62.8 (2.81)
	75.2 (3.12)	75.4 (3.78)	68.3 (2.04)	67.7 (2.93)

*For DB: 1 – NinaPro, 2-b – CapgMyo DB-b, 2-c – CapgMyo DB-c, 3 – CSL-HDEMG.
**Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$, $p < 0.01$, and $p < 0.1$ for NinaPro, CapgMyo, and CSL-HDEMG databases, respectively.)

grips using ring finger with the thumb). The most of the channels (38 out of 168 channels) recognize it as gesture 15 with the best accuracy of 90.2%. The next most of channels (22) identify this segment as gesture 14 (Pinch grips using middle finger with the thumb).

Figure 3 shows the accuracy results, averaged over the subjects, of the MLSVD-based method using different classifiers (dictionary learning, SVMs, KNN, and TB) along with the comparison among two other methods. The differences between the SVMs with two kernels are negligible (Acc = 73.7% (pol), 74.1% (rbf) in CapgMyo DB-b and Acc = 85.9% (pol), 86.8% (rbf) in CSL-HDEMG). The KNN classifier (Acc = 71.4% in CapgMyo DB-b and Acc = 82.1% in CSLHDEMG) performs slightly worse than the SVMs while the performance of the TB (Acc = 65.2% in CapgMyo DB-b and Acc = 76.7% in CSL-HDEMG) classifier is significantly worse. The dictionary learning method performs better among all these classifiers achieving accuracy of 77.5% and 89.6% in CapgMyo DB-b and CSL-HDEMG databases, respectively. Also, it can be observed from the figure that the accuracy results obtained with the CSLHDEMG dataset over the CapgMyo DB-b dataset by all methods are comparatively better. The reason behind this is the amount of data used for training and testing (80% training data in the former database compared to 50% in the latter). The proposed method performs better than the state-of-the-art techniques. The MLSVD-based method enhanced the inter-session recognition accuracy with an improvement of 4-14%, reaching accuracy to 77.5% and 89.6% in CapgMyo DB-b and CSL-HDEMD databases, respectively.

C. Inter-Subject Evaluation

The next experiment is based on inter-subject analysis where sEMG data of a set of subjects was used as the training set and another independent set of subjects as the testing set. The inter-subject evaluation was carried out on all three databases considered in this work. For a fair comparison, the training and the testing datasets were separated as was in the original articles and being carried out in the literature [2], [6], [41]. A leave-one-out cross-validation technique was followed for both CapgMyo (separately for DB-b and DB-c) and NinaPro databases.

Figure 2(c) shows the recognition accuracy of a randomly chosen segment (CapgMyo DB-c) on a per-channel basis. The original label of the shown segment is gesture 5 (Ring flexion). The most of the channels (37 out of 128 channels) recognize it as gesture 5 with a best accuracy of 70.1%.

Table II shows the mean and standard deviation of gesture recognition accuracy of the proposed method) along with the comparison between two other methods. Patricia *et al.* performed experiments using different adaptive learning methods with the NinaPro DB-1 and obtained an average accuracy of approximately 40% [41]. ConvNet method by Du *et al.* achieved an accuracy of 67.4% (improvement of approximately 27%), whereas the method achieved an accuracy of 55.3% and 35.1% using CapgMyo's two sub-databases DB-b and DB-c, respectively. The results shown in the case of CSL-HDEMG database are averaged over sessions, i.e., for each session inter-subject evaluation was performed and then averaged. Like the other two experiments, dictionary learning method performs better than the other three classifiers attaining improvement of average accuracy up to 7%. Among the other classifiers, SVM-rbf and KNN perform similarly but results using SVM-rbf only (for MLSVD-based method) are shown in this table. Finally, the MLSVD-based method, achieving the accuracy in the range 58–79.5%, improved the average results over the ConvNet by 8%, 20%, and 33% in NinaPro DB-1, CapgMyo DB-b, and CapgMyo DB-c, respectively. The results of inter-subject evaluation using the tensor-based

TABLE III
COMPUTATIONAL TIME REQUIRED FOR TRAINING AND TESTING DURING THREE EVALUATIONS

Evaluation	DB	Dictionary learning		SVM-rbf	
		Training	Testing	Training	Testing
intra-session	1	39m	1.95s	2h 34m	8.625s
	2-a	1h 12m	6.08s	4h 27m	22.351s
	2-b	47m	6.32s	3h 06m	24.246s
	2-c	59m	6.95s	3h 51m	28.437s
	3	2h 01m	7.54s	7h 41m	32.239s
inter-session	2-b	34m	6.24s	2h 11m	25.187s
	3	1h 32m	6.91s	5h 34m	29.214s
inter-subject	1	48m	1.99s	3h 02m	9.009s
	2-b	59m	6.37s	4h 08m	26.240s
	2-c	1h 22m	7.56s	4h 10m	33.129s
	3	2h 54m	8.89s	7h 39m	36.831s

For DB: 1 – NinaPro, 2-a – CapgMyo DB-a, 2-b – CapgMyo DB-b, 2-c – CapgMyo DB-c, 3 – CSL-HDEMG

method performs better than the state-of-the-art techniques but are not too promising and hence demands improvement. The reason is the underlying assumption in (8) where the *subject* subspace does not change for new gestures because **X** is fixed.

Computational Time

In the following, the computation time required for the training and the testing of the proposed tensor-based algorithm during three evaluations are presented. Table III shows the time spent during computation in a personal computer (CPU @2.70 GHz, 4 GB RAM and MATLAB 2019b software) using the SVM-rbf and a dictionary learning classifier. The training time corresponds to the time needed starting from the tensor construction to train the classifier. Values shown for testing are average time per gesture, i.e., total channel-wise time taken during testing divided by total number of gestures. So, for example, the average elapsed time per channel for dictionary learning classifier is approximately 47.5ms (= 6.08s/128) in case of CapgMyo DB-a database during intra-session evaluation and 237ms (= 22.35s/128) for SVM-rbf classifier for the same evaluation. For any evaluation and database, we can observe that the dictionary learning classifier is computationally more effective than the multiclass SVM-rbf classifier.

VI. CONCLUSION AND FUTURE WORK

This work presented a tensor decomposition method using MLSVD that analyze spatio-temporal dynamics of muscle activities for hand gesture recognition. The results demonstrate that the extracted features from different factor matrices, specifically from gestures subspace, are potentially effective for hand gesture recognition. As the way of conducting evaluations gets complex, i.e., from intra- session to inter-subject evaluations the recognition accuracy obtained by a single channel decreases. But, comparison with existing techniques shows the proposed MLSVD-based method outperforms the state-of-the-art methods while considering inter-session and inter-subject evaluations.

It is also important to discuss the limitations and possible future directions of the current work:

- Currently, although a large number of channels are recorded for gesture recognition, it is recommended to use less number of channels for ease of use and in wearable sensors [15]. In the current work, we give an attempt to recognize gestures from a single lead though training was conducted using all channels. The results shown in all the experiments are the best results obtained from any one of the channels. So, it remains an open problem if and how a particular set of channels can be effectively used for gestures recognition. Additionally, it will be interesting to investigate the localization concept of sensors for gesture recognition, i.e., which channel(s) effectively is/are responsible for which specific gesture(s).
- There are many factors that limit the use of available algorithms over different databases. Future work will focus on adapting and testing the tensor-based method(s) for inter-database analysis making a generic method for gesture recognition.
- As future work, we would like to also explore LIME [42] or Grad Cam [43] explainability algorithms to understand the misclassification results provided by this model.

REFERENCES

[1] U. Cote-Allard *et al.*, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.
[2] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation," *Sensors*, vol. 17, no. 3, p. 458, Feb. 2017.
[3] P. Kaczmarek, T. Mańkowski, and J. Tomczyński, "putEMG—A surface electromyography hand gesture recognition dataset," *Sensors*, vol. 19, no. 16, p. 3548, Aug. 2019.
[4] S. Lobov, N. Krilova, I. Kastalskiy, V. Kazantsev, and V. Makarov, "Latent factors limiting the performance of sEMG-interfaces," *Sensors*, vol. 18, no. 4, p. 1122, Apr. 2018.
[5] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 4832–4840, Sep. 2013.
[6] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Sci. Rep.*, vol. 6, no. 1, Dec. 2016, Art. no. 36571.
[7] S. Padhy and S. Dandapat, "Exploiting multi-lead electrocardiogram correlations using robust third-order tensor decomposition," *Healthcare Technol. Lett.*, vol. 2, no. 5, pp. 112–117, Oct. 2015.
[8] S. Padhy and S. Dandapat, "Third-order tensor based analysis of multilead ECG for classification of myocardial infarction," *Biomed. Signal Process. Control*, vol. 31, pp. 71–78, Jan. 2017.
[9] S. Geirnaert, G. Goovaerts, S. Padhy, M. Boussé, L. D. Lathauwer, and S. V. Huffel, "Tensor-based ECG signal processing applied to atrial fibrillation detection," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 799–805.
[10] G. Goovaerts, B. Vandenberk, R. Willems, and S. Van Huffel, "Automatic detection of T wave alternans using tensor decompositions in multilead ECG signals," *Physiolog. Meas.*, vol. 38, no. 8, pp. 1513–1528, Jul. 2017.
[11] A. Ebied, L. Spyrou, E. Kinney-Lang, and J. Escudero, "On the use of higher-order tensors to model muscle synergies," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 1792–1795.
[12] A. Ebied, E. Kinney-Lang, L. Spyrou, and J. Escudero, "Muscle activity analysis using higher-order tensor decomposition: Application to muscle synergy extraction," *IEEE Access*, vol. 7, pp. 27257–27271, 2019.
[13] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[14] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[15] S. Tam, M. Boukadoum, A. Campeau-Lecours, and B. Gosselin, "A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 2, pp. 232–243, Apr. 2020.

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

[17] J. Dauwels, K. Srinivasan, M. R. Reddy, and A. Cichocki, "Near-lossless multichannel EEG compression based on matrix and tensor decompositions," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 3, pp. 708–714, May 2013.

[18] H. Akbari, M. B. Shamsollahi, and R. Phlypo, "Fetal ECG extraction using pi tucker decomposition," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Sep. 2015, pp. 174–178.

[19] G. Goovaerts, C. Varon, B. Vandenberk, R. Willems, and S. Van Huffel, "Tensor-based detection of t wave alternans in multilead ECG signals," in *Proc. Comput. Cardiol.*, Sep. 2014, pp. 185–188.

[20] S. Padhy and S. Dandapat, "Validation of μ-volt T-wave alternans analysis using multiscale analysis-by-synthesis and higher-order SVD," *Biomed. Signal Process. Control*, vol. 40, pp. 171–179, Feb. 2018.

[21] K. Huang and L. Zhang, "Cardiology knowledge free ECG feature extraction using generalized tensor rank one discriminant analysis," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, p. 2, Dec. 2014.

[22] K. Srinivasan, J. Dauwels, and M. R. Reddy, "Multichannel EEG compression: wavelet-based image and volumetric coding approach," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 1, pp. 113–120, Jan. 2013.

[23] P. Xie and Y. Song, "Multi-domain feature extraction from surface EMG signals using nonnegative tensor factorization," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, Dec. 2013, pp. 322–325.

[24] K. Takiyama, H. Yokoyama, N. Kaneko, and K. Nakazawa, "Detecting task-dependent modulation of spatiotemporal module via tensor decomposition: Application to kinematics and EMG data for walking and running at various speed," *BioRxiv*, 2019.

[25] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Stud. Appl. Math.*, vol. 6, nos. 1–4, pp. 164–189, 1927.

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

[27] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[28] M. Atzori *et al.*, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Sci. Data*, vol. 1, no. 1, Dec. 2014, Art. no. 140053.

[29] M. Atzori *et al.*, "Characterization of a benchmark database for myoelectric movement classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 1, pp. 73–83, Jan. 2015.

[30] A. Gijsberts, M. Atzori, C. Castellini, H. Muller, and B. Caputo, "Movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 735–744, Jul. 2014.

[31] C. Amma, T. Krings, J. Böer, and T. Schultz, "Advancing muscle-computer interfaces with high-density electromyography," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst. - CHI*, 2015, pp. 929–938.

[32] M. Bousse, N. Vervliet, O. Debals, and L. De Lathauwer, "Face recognition as a kronecker product equation," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Dec. 2017, pp. 276–280.

[33] M. Boussé, G. Goovaerts, N. Vervliet, O. Debals, S. Van Huffel, and L. De Lathauwer, "Irregular heartbeat classification using kronecker product equations," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 438–441.

[34] S. M. Mathews, L. F. Polania, and K. E. Barner, "Leveraging a discriminative dictionary learning algorithm for single-lead ECG classification," in *Proc. 41st Annu. Northeast Biomed. Eng. Conf. (NEBEC)*, Apr. 2015, pp. 1–2.

[35] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "Maximum correntropy based dictionary learning framework for physical activity recognition using wearable sensors," in *Advances in Visual Computing*, vol. 10073. Cham, Switzerland: Springer, 2016, pp. 123–132.

[36] S. M. Mathews, C. Kambhamettu, and K. E. Barner, "Centralized class specific dictionary learning for wearable sensors based physical activity recognition," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[37] S. M. Mathews, "Dictionary and deep learning algorithms with applications to remote health monitoring systems," Ph.D. dissertation, Univ. Delaware, Newark, Delaware, 2017.

[38] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processig*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[39] R. Badeau and R. Boyer, "Multilinear singular value decomposition for structured tensors," *SIAM J. Matrix Anal. Appl., Soc. Ind. Appl. Math.*, vol. 30, no. 3, pp. 1–15, 2008.

[40] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer. (Mar. 2016). *Tensorlab 3.0*. [Online]. Available: https://www.tensorlab.net

[41] N. Patricia, T. Tommasit, and B. Caputo, "Multi-source adaptive learning for fast control of prosthetics hand," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 2769–2774.

[42] S. M. Mathews, "Explainable artificial intelligence applications in NLP, biomedical, and malware classification: A literature review," in *Intelligent Computing* (Advances in Intelligent Systems and Computing), vol. 998, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham, Switzerland: Springer, 2019, pp. 1269–1292.

[43] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.