Tensor Kernel Functions Based on Core Tensors Applied to the Recognition of Hand Movements

Flávio V. dos Santos, C. Alexandre R. Fernandes

***Abstract*—Kernel methods and Support Vector Machine (SVM) are widely used in machine learning. However, when multi-dimensional data are used, the classical vector-based kernel functions must vectorize the inputs, which breaks down the original tensor structure, leading to performance loss. To avoid this problem, tensor kernel functions can be used. In the present work, three novel tensor kernel functions are presented. The proposed methods are based on the core tensors of the Higher-Order Singular Value Decomposition (HOSVD) and Tensor-Train Decomposition (TTD). Two of the presented methods are fast kernel functions that ignore the factor matrices of these tensor decompositions, alleviating the time complexity burden. The presented techniques were evaluated in the classification of hand movements. A low-cost "smart glove" with accelerometers and gyroscopes was developed, generating tensor input samples with modes related to sensors, channels and features. The experiments showed a good performance of the proposed techniques when compared to state-of-the-art tensor kernel functions.**

***Index Terms*—machine learning, HOSVD, kernel function, SVM, tensor learning.**

I. Introduction

SUPPORT Vector Machines (SVMs) stand out as one of the most widely used tools in machine learning (ML) [1]. As many problems in ML are nonlinear, SVMs are generally used with kernel functions. Classical kernel functions include the radial basis function (RBF), polynomial and sigmoid [2].

In ML, vectors are usually used as inputs. When the input data are matrices or higher-order arrays (tensors), the inputs are generally transformed into vectors. However, this vectorization breaks down the multidimensional structure of the data, which often leads to performance loss [3]. The so-called tensor learning (TL) is used to avoid the vectorization of input and the consequent destruction of data structure. In TL, the multidimensional structure of the data is maintained, improving the performance in learning tasks [4], [5], [6].

In the particular case of the SVM, tensor inputs must be vectorized before being used by the classical vector-based kernel functions. To avoid this problem, tensor kernel functions can be used, allowing the use of tensor inputs without vectorization, preserving the data structure. Some works in the literature have proposed tensor kernel functions. In [7], the authors proposed the Dual Structure-preserving Kernel (DuSK), where the structure of the input tensors is preserved using the PARAllel FACtors (PARAFAC) decomposition [8], also known as Canonical decomposition (CANDECOMP) or Canonical Polyadic (CP). In [9], the authors presented two tensor-based kernels: Partial Least Squares (KTPLS) and Kernel Tensor Canonical Correlation Analysis (KTCCA).

In [10], the authors introduced a tensor kernel function based on the unitary factor matrices generated by the Higher-Order Singular Value Decomposition (HOSVD), also known as Multilinear SVD (MLSVD), which consists in a generalization of the singular value decomposition (SVD) for tensors [11]. The kernel function computes the angles between the subspaces generated by the matrices with the singular vectors.

The Fast Kernel Subspace Estimation based on Tensor Train (FAKSETT), proposed in [12], is a tensor-kernel based on the Tensor-Train Decomposition (TTD). The FAKSETT is similar to the method of [10], but it uses the matrix and tensor factors of the TTD to compute the angles between the subspaces. Other tensor kernel methods can be found in [13], [14].

In the present work, three novel tensor kernel functions are proposed. The first one, denoted by HOSVD Tensor-Core Kernel (H-TCK), is inspired by the tensor kernel function of [10], which uses the factor matrices of the HOSVD. However, the core tensor of the HOSVD is not used in [10]. The main idea of the H-TCK is to introduce an additional term in the kernel function that takes into account the core tensor of the HOSVD. This additional term introduces relevant information that is ignored by the kernel function of [10].

The second proposed tensor kernel function, denoted by Fast Tensor-Core Kernel (F-TCK), can be viewed as a simplification of the H-TCK that uses only the term corresponding to the core tensors of the HOSVD. The F-TCK does not use the unitary factor matrices of the HOSVD. Ignoring these factor matrices significantly simplifies the computation of the kernel function, alleviating the time complexity burden.

The third proposed kernel function, denoted by TTD Fast Tensor-Core Kernel (TF-TCK), is similar to the F-TCK, but it uses the core tensor of the TTD instead of the HOSVD. Similarly to the F-TCK, the TF-TCK does not use the factor matrices of the TTD, which significantly simplifies the computation of the kernel function. Moreover, the TTD is more generic than the HOSVD and it can be estimated in a shorter processing time.

The proposed tensor kernel functions were evaluated in the classification of hand movements. For this purpose, a low-cost "smart glove" equipped with two MPU-6050 modules was developed. The glove captures different angular and acceleration variations measured by accelerometers and gyroscopes, adding up to 12 channels.

This work was supported in part by the FUNCAP/Brazil under Grant BP5-0197-00183.01.06/23 and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq/Brazil). The authors are with the Federal University of Ceara, Rua Coronel Estanislau Frota, 563, 62010-560, Sobral, Brazil (e-mails: eng.flaviovasconcelos@gmail.com, alexandrefernandes@ufc.br). ORCID: 0009-0001-8339-6068, 0000-0002-9933-9930

Digital Object Identifier: 10.14209/jcis.2025.08.

Submission: 2025-04-03, First decision: 2025-05-21, Acceptance: 2025-10-13, Publication: 2025-10-22.

A database was built using the smart glove, with five different types of movements being considered. The database was used to train and test an SVM with several tensor kernel functions. The multidimensional nature of the data generated by the smart glove is exploited to form tensor inputs. Each sample movement was organized in a 3^{th} order tensor with dimensions corresponding to sensors, channels and features. The use of tensor algebra in this kind of database, constructed from wearable devices, is a new application of TL. The experiments showed that the proposed kernel functions provided good accuracy and processing time, when compared with state-of-the-art methods, especially the TF-TCK.

Notation: The following notation is used throughout the work. Scalars are denoted by lowercase letters (x), vectors by lowercase boldface letters (**x**), matrices by uppercase boldface letters (**X**) and tensors (arrays with order higher than 2) by calligraphic letters ($\mathcal{X}$).

II. SVM FOR INPUT TENSOR DATA

The SVM classifier generally uses kernel functions when dealing with non-linearly separable cases. However, for tensor inputs, the use of vector-based kernel functions requires the input vectorization, which breaks the data structure and usually leads to a loss of performance [3]. To avoid this problem, tensor kernel functions can be used, preserving the multidimensional structure of the data. The dual form of the SVM classifier for tensor patterns is given by [1]:

$$\max_{\alpha} \sum_{i=1}^{P} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{P} \alpha_i \alpha_j y_i y_j k(\mathcal{X}_i, \mathcal{X}_j), \tag{1}$$

subject to the following restrictions: $0 \leq \alpha_i \leq C, \quad \forall_i = 1, \ldots, P$ and $\sum_{i=1}^{P} \alpha_i y_i = 0$, where P is the number of samples, C is the box constant parameter, α_i and α_j represent the Lagrange coefficients, $k(.,.)$ is the tensor kernel function, $\mathcal{X}_i \in R^{I_1 \times \cdots \times I_N}$ and $\mathcal{X}_j \in R^{I_1 \times \cdots \times I_N}$ are the N^{th}-order input tensors, and y_i are the sample labels.

III. PROPOSED TENSOR KERNELS

A. HOSVD Tensor-Core Kernel (H-TCK)

Let the HOSVDs of the input data be given by:

$$\mathcal{X}_i = \mathcal{W} \times_1 \mathbf{U}^{(1)} \times_2 \ldots \times_N \mathbf{U}^{(N)}, \tag{2}$$

$$\mathcal{X}_j = \mathcal{Y} \times_1 \mathbf{V}^{(1)} \times_2 \ldots \times_N \mathbf{V}^{(N)}, \tag{3}$$

where $\times_n$ denotes the mode-n product [11], $\mathbf{U}^{(n)} \in R^{I_n \times R_n}$ and $\mathbf{V}^{(n)} \in R^{I_n \times R_n}$, for $1 \leq n \leq N$, are unitary matrices with the singular vectors, and $\mathcal{W} \in R^{R_1 \times \ldots \times R_N}$ $\mathcal{W} \in Y^{R_1 \times \ldots \times R_N}$ are the core tensors of $\mathcal{X}_i$ and $\mathcal{X}_j$, respectively. The block-diagram of the HOSVD of a 3^{th}-order tensor is shown in Fig. 1. The HOSVD decomposition can be interpreted as a core tensor undergoing successive orthogonal transformations along each mode independently, using the mode-n product.

The proposed H-TCK is inspired by the tensor kernel of [10], which uses the angle between the subspaces spanned by the factor matrices $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ of the HOSVD, for $n = 1, ..., N$. However, the core tensors $\mathcal{W} \in R^{I_1 \times \ldots \times R_N}$ and $\mathcal{Y} \in R^{R_1 \times \ldots\ldots \times R_N}$ are not used in [10]. The main idea

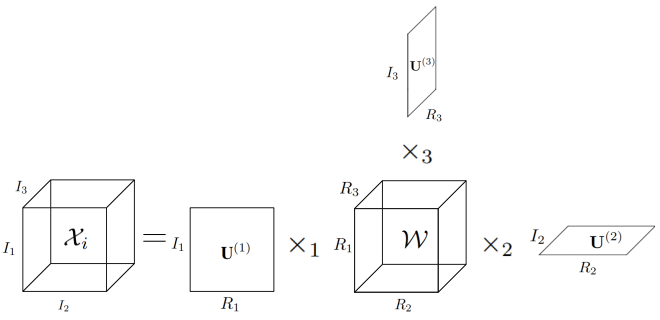

Fig. 1. Block-diagram of the HOSVD of a 3^{th}-order tensor.

of the H-TCK function is to introduce an additional term to the kernel function that takes the core tensors of the HOSVDs into account. The proposed H-TCK is defined as:

$$k(\mathcal{X}_i, \mathcal{X}_j) = \exp(-2\gamma ||\mathcal{W} - \mathcal{Y}||_F^2) \prod_{n=1}^{N} k_n(\mathbf{U}^{(n)}, \mathbf{V}^{(n)}), \tag{4}$$

where $||\cdot||_F^2$ denotes the Frobenius norm and $k_n(\mathbf{U}^{(n)}, \mathbf{V}^{(n)})$ is the kernel function of [10], given by:

$$k(\mathcal{X}_i, \mathcal{X}_j) = \prod_{n=1}^{N} k_n(\mathbf{U}^{(n)}, \mathbf{V}^{(n)}). \tag{5}$$

with $k_n(\mathbf{U}^{(n)}, \mathbf{V}^{(n)}) = \exp(-\gamma \sin^2(\theta_n))$, for $1 \leq n \leq N$, where γ is a scalar constant and θ_n is the angle between the subspaces spanned by $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$, obtained from: $\sin^2(\theta_n) = 2||\mathbf{U}^{(n)}\mathbf{U}^{(n)^T} - \mathbf{V}^{(n)}\mathbf{V}^{(n)^T}||_F^2$. The tensor kernel of [10] will be denoted, from now on, HOSVD kernel.

The difference between the proposed H-TCK and the HOSVD kernel function is the term $\exp(-2\gamma||\mathcal{W}-\mathcal{Y}||_F^2)$ that uses the core tensors of the HOSVD. This term introduces relevant information of the input tensors that is ignored by the kernel function of [10]. Indeed, the core tensor of the HOSVD is a crucial component that captures the interactions between the different modes of a tensor. While it is not superdiagonal and its matrix slices are not diagonal, a key property is that it is all-orthogonal and ordered [11]. The all-orthogonality property means that its subtensors along each mode are mutually orthogonal, and its norms relate directly to the singular values of the unfolded tensor along each mode. This core tensor provides a compact, interpretable representation of the original tensor's structure. Moreover, it ensures that the information captured by the core tensor is distinct and non-redundant across its modes, making it a powerful representation for preserving the multidimensional structural information of the data. The H-TCK leverages this property by incorporating the HOSVD core tensor to introduce information that might otherwise be overlooked by methods relying solely on factor matrices.

B. Fast Tensor-Core Kernel (F-TCK)

The second proposed tensor kernel function is a simplification of the H-TCK that uses only the term corresponding to the core tensors $\mathcal{W}$ and $\mathcal{Y}$ of the HOSVDs to compute the kernel function. The F-TCK function does not use the factor

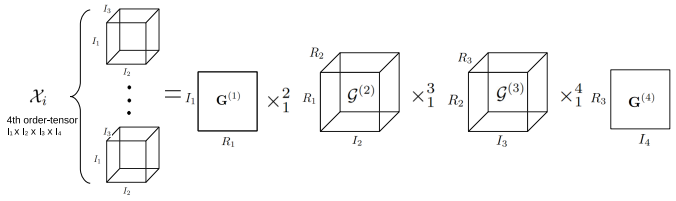

Fig. 2. Block-diagram of the TTD of a 4^{th}-order tensor.

matrices $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$, for $n = 1, ..., N$, of the HOSVD, due to complexity reasons. Ignoring these factors significantly simplifies the computation of the kernel function, alleviating the complexity burden. The F-TCK is given by:

$$K(\mathcal{X}_i, \mathcal{X}_j) = \exp(-2\gamma||\mathcal{W} - \mathcal{Y}||_F^2). \quad (6)$$

Using the core tensors of the input tensors assures the preservation of the tensor structure of the input data, as these core tensors $\mathcal{W}$ and $\mathcal{Y}$ depend on the structure of the input tensors $\mathcal{X}_i$ and $\mathcal{X}_j$. Using only the core tensor of the HOSVD in the expression of the kernel function preserves the multidimensional structural information of the input data while reducing computational complexity.

C. TTD Fast Tensor-Core Kernel (TF-TCK)

The third proposed kernel function is applied to the case of third-order input tensors. In this case, the TTD of $\mathcal{X}_i \in R^{I_1 \times I_2 \times I_3}$ and $\mathcal{X}_j \in R^{I_1 \times I_2 \times I_3}$ can be written as:

$$\mathcal{X}_i = \mathbf{G}^{(1)} \times_1^2 \mathcal{G}^{(2)} \times_1^3 \mathbf{G}^{(3)}, \mathcal{X}_j = \mathbf{H}^{(1)} \times_1^2 \mathcal{H}^{(2)} \times_1^3 \mathbf{H}^{(3)}, \quad (7)$$

where $\times_j^i$ denotes the contraction operation in modes i and j [15], $\mathcal{G}^{(2)} \in R^{R_1 \times I_2 \times R_2}$ and $\mathcal{H}^{(2)} \in R^{R_1 \times I_2 \times R_2}$ are the tensor-train cores, and $\mathbf{G}^{(1)} \in R^{I_1 \times R_1}$, $\mathbf{G}^{(3)} \in R^{R_2 \times I_3}$, $\mathbf{H}^{(1)} \in R^{I_1 \times R_1}$ and $\mathbf{H}^{(3)} \in R^{R_2 \times I_3}$ are the factor matrices.

The proposed TF-TCK function is similar to the F-TCK; however, it uses the core tensors $\mathcal{G}^{(2)}$ and $\mathcal{H}^{(2)}$ of the TTD instead of the core tensors of the HOSVD, as follows:

$$K(\mathcal{X}_i, \mathcal{X}_j) = \exp(-2\gamma||\mathcal{G}^{(2)} - \mathcal{H}^{(2)}||_F^2). \quad (8)$$

Similarly to the TCK, due to complexity reasons, the TF-TCK does not use the factor matrices $\mathbf{G}^{(1)}$, $\mathbf{G}^{(3)}$, $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(3)}$, which significantly simplify the kernel function computation.

The block-diagram of the TTD of a 4^{th}-order tensor is shown in Fig. 2 While the HOSVD has a core tensor that undergoes orthogonal transformations along each mode, the TTD does not treat all modes simultaneously. Instead, it breaks the tensor into a sequence of smaller tensors (except the first and last, which are matrices), each interacting only with its immediate neighbors. The TTD is a sequence of tensors and matrices connected in a chain-like structure, where each core interacts with adjacent modes and has no orthogonality restriction. This approach reduces the number of parameters for high-order tensors.

Moreover, the TTD is more generic than the HOSVD and it can be estimated from Tensor-Train Singular Value Decompositions (TT-SVD), using sequential SVDs [15]. This property enables the TTD to exhibit a shorter processing time than the HOSVD, which implies that the TF-TCK demonstrates a reduced processing time compared to F-TCK. Another significant difference between the two decompositions is that the HOSVD uses the mode-n product, whereas the TTD employs the contraction operation.

The core tensor of the TTD plays a similar role as the core tensor of the HOSVD in preserving the multidimensional structure of the data. The TTD cores allow for highly efficient storage and computation, especially for high-dimensional tensors, by decomposing the tensor into low-rank components.

IV. Hand Movements System

A hand movement device was used in the experiments to evaluate the tensor kernel functions. The developed prototype is a smart glove with two MPU-6050 modules, each offering 6 degrees of freedom (from a 3-axis accelerometer and a 3-axis gyroscope), resulting in a total of 12 degrees of freedom, with a sampling rate of 2,150 Hz.

The MPU-6050 modules are fixed at the thumb and middle finger. The thumb has greater mobility and rotation than the other fingers, allowing a greater variety of movements, while the middle finger is at the center of the hand and represents an average of the movements of four fingers. The MPU-6050 sensors are low cost and have a fast connection with the microcontroller. They are used together with the Wemos D1 Mini microcontroller, as it has an I^2C connection, connectivity to the Arduino platform, and compatibility with Python 3.

A database of hand movements was built using the smart glove. Five hand movements were used: 1) open hand, 2) hand down, 3) hand up, 4) thumb up, and 5) inner side. The hand movements in the dataset were entirely performed by the first author of this article. The multidimensional nature of the data generated by the glove is exploited to form tensor input samples. Each sample was organized in a 3^{rd} order tensor with dimensions $2 \times 6 \times 500$, where the first mode corresponds to the MPU modules, the second one to the channels, and the third one to the 500 samples of each signal.

The feature extraction is performed by the Multilinear Principal Component Analysis (MPCA) [16]. The MPCA is a multidimensional dimensionality reduction technique that can be viewed as an extension of the PCA to tensor data.

The final database has five classes, with 300 sample movements for each class, adding up to 1,500 tensor samples. The database used in this article is available in the repository [17]. The classification system is composed of the following steps. Firstly, the MPCA method is applied to the tensor inputs. In the sequel, the tensor kernel function is chosen and the SVM classifier is used. Finally, the processing time and accuracy are used as parameters to evaluate the performance of the methods.

V. Results and Discussion

In the experiments, the SVM for input tensor data described in Subsection II is used along with the proposed tensor kernel functions and the following state-of-the-art tensor kernel functions: the tensor kernel of [10], denoted here by HOSVD kernel, the FAKSETT kernel [12], and the DuSK [7]. The dataset was split into two subsets: 80 % of the samples

TABLE I
HYPERPARAMETERS THAT PROVIDED THE BEST ACCURACIES.

Method	γ	C	I_1	I_2	I_3
DuSK	0.53	5.27	2	3	11
HOSVD kernel	0.001	100	2	5	91
FAKSETT	1.32	10.53	2	4	15
H-TCK	0.26	5.27	2	5	4
F-TCK	1.32	5.27	2	3	19
TF-TCK	0.26	31.58	2	5	4

TABLE II
ACCURACY AND PROCESSING TIME OBTAINED BY THE METHODS

Tensor Kernel Function	**Accuracy (%)**	**Processing Time (s)**
DuSK	78.7	227.8
HOSVD kernel	86.7	132.5
FAKSETT	79.0	31.4
H-TCK	92.0	40.0
F-TCK	87.3	16.6
TF-TCK	94.0	15.7

were used for hyperparameter tuning through 10-fold cross-validation, while the remaining 20 % were reserved for testing. The test set was not used at any stage of the hyperparameter tuning process.

The hyperparameters I_2, I_3, γ, and C were adjusted using grid search within the ranges $2 \leq I_2 \leq 6$. $2 \leq I_3 \leq 200$, $0.01 \leq C \leq 100$ and $0.01 \leq \gamma \leq 5$, with 20 equally spaced values for γ and C. The performance of the techniques using different MPCA configurations is a challenging task due to the inherent trade-off between accuracy and processing time. Given the scope and space limitations, for each technique, the number of MPCA components that delivered the best performance in terms of accuracy was used.

The hyperparameters that provided the best accuracies are listed in Tab. 4. I. The MPCA led to a massive reduction in the number of inputs, for all the tested kernel functions, especially in the third dimension. For instance, for the H-TCK and TF-TCK, the dimensions of input tensors decreased from $2 \times 6 \times 500 = 6,000$ to $2 \times 5 \times 4 = 40$. Tab. II shows the accuracy and the processing time, in seconds, provided by the techniques. The processing time corresponds to the total time needed to perform training and testing. The tests were performed in Python 3.9 with the following computational setup: CPU Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz up to 4.5GHz, 32GB RAM dual channel, GPU NVIDIA GeForce RTX 2060 and a SSD M.2 with reading speed of 2400MB/S.

Two statistical significance tests were performed to verify whether the gains in accuracy provided by the proposed techniques are significant. The first is the Mann-Whitney U test (also known as the Wilcoxon rank-sum test), which evaluates the null hypothesis that two independent samples from continuous distributions share the same median. The second is the two-sample t-test (Welch's t-test), which assesses whether two samples originate from normal distributions with equal means, under the assumption of unequal variances. In what follows, the p-value from the Mann-Whitney U test and the binary outcome of Welch's t-test are reported in parentheses ($h = 0$: equal means; $h = 1$: different means).

The proposed H-TCK provided an accuracy higher than that of the HOSVD kernel, due to the additional term that takes the core tensors into account. The H-TCK also obtained an accuracy higher than that of the F-TCK, due to the fact that the F-TCK does not use the factor matrices. However, the gain in accuracy of the H-TCK with respect to the HOSVD and F-TCK kernels was not significant ($p = 0.08$, $h = 0$ and $p = 0.09$, $h = 0$, resp.). Moreover, the accuracy of the H-TCK was significantly higher than that of the DuSK and FAKSETT methods ($p = 0.001$, $h = 1$ and $p = 0.004$, $h = 1$, resp.).

The proposed TF-TCK provided the highest accuracy among the tested methods, with significant difference in accuracy with respect to the DUSK ($p = 0.001$, $h = 1$), FAKSETT ($p = 0.003$, $h = 1$), HOSVD Kernel ($p = 0.01$, $h = 1$), and F-TCK ($p = 0.02$, $h = 1$), and insignificant gain in accuracy with respect to the H-TCK ($p = 0.40$, $h = 0$). The two kernel functions based on the TTD provided the best results, showing that using the TTD is more efficient than using the HOSVD and the PARAFAC. As earlier mentioned, the TTD is more generic than the HOSVD and PARAFAC, which gave the TTD-based kernel functions a better ability to classify the data. Besides, using only the term relative to the core tensors of the TTD proved to be the most effective choice.

In addition, the proposed TF-TCK needed less processing time than the other tested tensor kernel functions, followed by the F-TCK. These two kernel functions are composed of only one term relative to a core tensor, showing that ignoring the factor matrices significantly simplifies the computation of the kernel function, alleviating the complexity burden.

The FAKSETT kernel is the third fastest method, with a processing time two times higher than that of the TF-TCK. The FAKSETT method provided a processing time smaller than the HOSVD kernel, due to the fact that the time complexity of the TTD is lower than that of the HOSVD. Besides, as expected, the processing time of the H-TCK is higher than that of the HOSVD kernel, as the H-TCK has an additional term when compared to the HOSVD kernel.

Overall, TF-TCK provided the most favorable results regarding accuracy and processing time. The accuracy obtained by this kernel, for the five classes, are 0.983, 0.817, 0.867, 0.767, and 0.933, while the precision is given by 0.894, 0.845, 0.912, 0.742, and 0.983. That leads to the following F1 scores: 0.468, 0.415, 0.444,0.377, and 0.479, which shows a good uniformity of the errors among the classes.

VI. CONCLUSION

The results showed that the proposed TF-TCK achieved the best results in terms of accuracy and processing time when compared with other state-of-the-art tensor kernel functions. Furthermore, the H-TCK provided the second highest accuracy, and the F-TCK showed the second smallest processing time.

In future works, other tensor kernel functions will be tested by exploiting alternative tensor decompositions. In addition, more sensors will be incorporated into the glove. Future research will also include experiments with other databases as well as an analysis of the conditions of Mercel's theorem.

REFERENCES

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000. [Online]. Available: https://doi.org/10.1017/CBO9780511801389

[2] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," in *2013 Int. Conf. Adv. Technol. Eng. (ICATE)*, 2013, pp. 1–9. [Online]. Available: https://doi.org/10.1109/ICAdTE.2013.6524746

[3] L. Ma, Y. Hu, and Y. Zhang, "Support tensor machines based bubble defect detection of lithium-ion polymer cell sheets," *Eng. Lett.*, vol. 25, no. 1, pp. 304–337, 2017. [Online]. Available: http://www.engineeringletters.com/issues_v25/issue_1/EL_25_1_32.pdf

[4] Y. Qin, Z. Tang, H. Wu, and G. Feng, "Flexible tensor learning for multi-view clustering with markov chain," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1552–1565, 2024. [Online]. Available: https://doi.org/10.1109/TKDE.2023.3241234

[5] F. Qian, H. Hua, Y. Wen, S. Pan, G. Zhang, and G. Hu, "Unsupervised 3-d seismic erratic noise attenuation with robust tensor deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–16, 2024. [Online]. Available: https://doi.org/10.1109/TGRS.2023.3284567

[6] Q. Xiao, L. Zhao, S. Chen, and X. Li, "Enhanced tensor low-rank representation learning for hyperspectral anomaly detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023. [Online]. Available: https://doi.org/10.1109/LGRS.2022.3187654

[7] L. He, X. Kong, P. S. Yu, X. Yang, A. B. Ragin, and Z. Hao, "Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 127–135. [Online]. Available: https://doi.org/10.1137/1.9781611973440.15

[8] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009. [Online]. Available: https://doi.org/10.1137/07070111X

[9] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki, "Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 137–148, 2013. [Online]. Available: https://doi.org/10.1109/MSP.2013.2246292

[10] M. Signoretto, L. De Lathauwer, and J. A. Suykens, "A kernel-based framework to tensorial data analysis," *Neural Netw.*, vol. 24, no. 8, pp. 861–874, 2011. [Online]. Available: https://doi.org/10.1016/j.neunet.2011.06.004

[11] 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. [Online]. Available: https://doi.org/10.1137/S0895479896305696

[12] O. Karmouda, J. Boulanger, and R. Boyer, "Speeding up of kernel-based learning for high-order tensors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 2905–2909. [Online]. Available: https://doi.org/10.1109/ICASSP39728.2021.9413709

[13] L. He, C.-T. Lu, H. Ding, S. Wang, L. Shen, P. S. Yu, and A. B. Ragin, "Multi-way multi-level kernel modeling for neuroimaging classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 356–364.

[14] F. Qi, J. Li, Y. Zhang, W. Huang, B. Hu, and H. Cai, "Tensorlized multi-kernel clustering via consensus tensor decomposition," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 9, no. 1, pp. 406–418, 2025.

[15] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011. [Online]. Available: https://doi.org/10.1137/090752286

[16] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, 2008. [Online]. Available: https://doi.org/10.1109/TNN.2007.900286

[17] F. Lima. (2023) Dataset hand movement. GitHub. [Online]. Available: https://github.com/eng-flavio/dataset_hand_movement.