
GRAPH-BASED CLASSIFICATION OF EEG SIGNALS

Armin Navardi

Electrical Engineering Department
Sharif University of Technology
navardiarmin@gmail.com

Sepideh Hajipour Sardouie

Electrical Engineering Department
Sharif University of Technology
hajipour@sharif.edu

October 8, 2025

1 Introduction

Brain-Computer Interface (BCI) systems have gained significant attention in recent years due to their potential applications in assistive technology, neurorehabilitation, and human-computer interaction. These systems aim to establish a direct communication pathway between the human brain and external devices by decoding brain activity, often recorded using electroencephalography (EEG). EEG signals, being non-invasive and high-temporal-resolution data sources, provide valuable insights into brain dynamics but also pose significant challenges due to their non-stationary nature, low signal-to-noise ratio, and inter-subject variability.

Traditional EEG classification methods, such as machine learning and deep learning models, rely heavily on feature extraction and handcrafted techniques to capture the discriminative characteristics of brain signals. However, these approaches often struggle with the complex connectivity patterns and spatial-temporal dependencies inherent in EEG signals. Graph-based methods have recently emerged as a powerful alternative, leveraging graph structures to model the relationships between EEG channels and enhance classification performance.

In this work, we propose a graph-based approach for EEG signal classification in a BCI system. By representing EEG signals as graphs, we can effectively capture the spatial and temporal relationships between different brain regions, leading to a more robust classification framework. Our approach utilizes graph signal processing techniques and machine learning models to improve EEG classification accuracy.

The main contributions of this paper are as follows:

We introduce a graph representation of EEG signals that incorporates both spatial and temporal dependencies. We employ graph-based learning models to enhance classification accuracy in a BCI system. We evaluate the proposed method on benchmark EEG datasets and compare it with state-of-the-art approaches. The remainder of this paper is organized as follows: Section 2 discusses related work on EEG classification and graph-based learning. Section 3 presents our methodology, including data preprocessing, graph construction, and classification models. Section 4 describes the experimental setup and performance evaluation. Section 5 discusses the results, and Section 6 concludes with insights and future directions. [2]

1.1 Graph-Based EEG

Graph-Based methods tend to capture spatial dependencies better and detect interaction of each pair of nodes and the whole network states. These aspects make Graph-Based methods suitable for representing neural interactions in brain and

2 Preliminaries

2.1 Graph Signal Processing

Adjacency Matrix of a Graph: In graph signal processing, an adjacency matrix is a fundamental graph representation that encodes the relationships between its nodes. Given a graph $G = (V, E)$, where V is the set of N nodes and E

is the set of edges connecting them, the adjacency matrix A is defined as an $N \times N$ matrix where each element A_{ij} represents the weight of the edge between node i and node j .

In the context of EEG signal processing, the adjacency matrix is used to represent functional or structural connectivity between EEG channels. The functional connectivities may be defined based on measures such as correlation, coherence, mutual information, or graph-theoretic distances between EEG signals. This representation allows for the application of graph-based techniques to analyze and classify EEG signals in a BCI system.

Graph Fourier Transform: The Graph Fourier Transform decomposes a graph signal into topological components. These components are the eigenvectors of the Laplacian matrix ($L = D - A$) that serve as the graph's frequency modes. The eigenvectors associated with small eigenvalues are considered low-frequency modes that vary slowly across connected nodes, while large eigenvalues represent high-frequency modes with rapid variation between neighboring nodes. Notably, any graph signal could be represented as a linear combination of all the graph modes. The GFT thus reveals how a graph signal distributes across smooth and oscillatory patterns relative to the underlying topology, enabling the design of spectral filters, the study of localized versus global dynamics, and the extension of classical signal processing and machine learning techniques such as filtering, neural networks, etc. to graph-structured data [15, 1].

$$\hat{X}_i = V^T X_i$$

The columns of \hat{X}_i thus correspond to the GFT coefficients for each frequency mode at a given instant.

2.2 Graph Learning

One of the common approaches to learning graph from observed data is based on graph smoothness assumption. This method originates in graph signal processing and its main assumption is that the resulting graphs must be smooth with respect to the observed signals. In this context total variation of the signal x with respect to a graph g is defined as:

$$TV_g(x) = \sum A_{ij}(x_i - x_j)^2 = x^T L x$$

which $L = D - A$ is the Laplacian matrix of the graph. This term measures how close the neighboring nodes of the graph are. In other words, a lower total variation means that neighboring nodes have more similar values and the graph is smoother with respect to the signal. In general we have:

$$\text{tr}(X^T L X) = \frac{1}{2} \text{tr}(W Z) = \frac{1}{2} \|W \circ Z\|_{1,1}$$

Kalefolias introduced a general model for learning a sparse graph with no isolated vertex when no prior information is available. Later on, sabokseyr utilized this framework for learning task-specific graphs. The objective introduced by Kalefolias is eq 3. This objective contains the total variation of the graph and two regularization terms. The logarithmic barrier prevents two things: 1) prevents the edges from becoming negative and 2) prevents isolation of vertices. The other term forces the graph to be sparser and have fewer edges.

$$\min_{W \in \mathcal{W}_m} TV_g(X) + f_{reg}(W) = \min_{W \in \mathcal{W}_m} \|W \circ Z\|_{1,1} - \alpha \mathbf{1}^\top \log(W \mathbf{1}) + \beta \|W\|_F^2.$$

In order to learn a graph from smooth signals, we propose, as explained in Section 2, to rewrite problem (2) using the weighted adjacency matrix W and the pairwise distance matrix Z instead of X

Since W is positive we could replace the first term by $\text{tr}(W Z)$, but we prefer this notation to keep in mind that our problem already has a sparsity term on W . This means that $f(W)$ has to play two important roles: (1) prevent W from going to the trivial solution $W = 0$ and (2) impose further structure using prior information on W . This said, depending on f the solution is expected to be sparse, that is important for large scale applications.

Based on our framework (10) our goal is to give a general purpose model for learning graphs, when no prior information is available. In order to obtain meaningful

graphs, we want to make sure that each node has at least one edge with another node. It is also desirable to have control of how sparse is the resulting graph. To meet these expectations, we propose the following model with parameters $\alpha > 0$ and $\beta > 0$ controlling the shape of the edges:

The logarithmic barrier acts on the node degree vector \mathbf{W}_1 , unlike the model of Proposition 1 that has a similar barrier on the edges. This means that it forces the degrees to be positive, but does not prevent edges from becoming zero. This improves the overall connectivity of the graph, without compromising sparsity. Note however, that adding solely a logarithmic term ($= 0$) leads to very sparse graphs, and changing γ only changes the scale of the solution and not the sparsity pattern (Proposition 2 for $\gamma = 0$). For this reason, we add the third term. We showed in eq. (4) that adding an ‘ ℓ_1 ’ norm to control sparsity is not very useful. On the other hand, adding a Frobenius norm we penalize the formation of big edges but do not penalize smaller ones. This leads to more dense edge patterns for bigger values of γ . An interesting property of our model is that even if it has two terms shaping the weights, if we fix the scale we then need to search for only one parameter:

2.3 Fukunaga-Koontz Transform

The Fukunaga-Koontz Transform (FKT) is a supervised subspace projection method that aims to enhance class separability by identifying orthogonal directions where the statistical properties of two classes differ the most. This can be formally described using generalized eigenvalue decomposition (GEVD) of class-specific covariance matrices.

Let \mathbf{R}_1 and \mathbf{R}_2 denote the covariance matrices of two distinct classes. The FKT seeks a projection matrix \mathbf{W} that diagonalizes both matrices simultaneously. This is achieved by solving the generalized eigenvalue problem:

$$\mathbf{R}_1 \mathbf{w} = \lambda \mathbf{R}_2 \mathbf{w}, \quad (1)$$

where $\lambda \in \mathbb{R}$ is a generalized eigenvalue and $\mathbf{w} \in \mathbb{R}^n$ is the corresponding eigenvector. Each eigenvector \mathbf{w}_i defines a projection axis along which the ratio of variances between class 1 and class 2 is equal to λ_i .

The set of eigenvectors $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_d\}$ forms the columns of the transformation matrix \mathbf{W} , which projects the original input \mathbf{x} into a subspace:

$$\mathbf{z} = \mathbf{W}^\top \mathbf{x}. \quad (2)$$

Directions corresponding to eigenvalues $\lambda_i > 1$ indicate greater variance in class 1, while $\lambda_i < 1$ indicates greater variance in class 2. By selecting eigenvectors associated with the most discriminative eigenvalues (i.e., farthest from 1), FKT produces a low-dimensional embedding optimized for classification.

This GEVD-based interpretation shows that FKT generalizes linear discriminant analysis (LDA) while maintaining orthogonality in the projected subspace. It has proven effective in domains such as brain-computer interface (BCI) systems, where capturing discriminative subspaces from multichannel EEG signals is crucial.

3 Discriminative Graph Learning

Saboseyr adopted a discriminative graph learning framework that takes task-specific information into account when learning the underlying graphs from the observed signals. In this method the goal is to learn smooth and sparse graphs for each class such that the signals of each class are smooth to its own graph and non-smooth to the graph of other classes. Later on, we demonstrate that this method improves performance of the graph learning model.

To address classification problems involving network-structured data, we adopt a discriminative graph learning framework that leverages graph signal processing (GSP) principles to learn graph representations that are both expressive of the class structure and discriminative for classification [8, 9]. The central idea is to learn class-specific graphs such that the graph Fourier transform (GFT) of signals from each class is smooth on its respective graph, while signals from other classes exhibit non-smooth behavior.

3.1 Problem Formulation

Let the dataset $\mathcal{X} = \bigcup_{c=1}^C \mathcal{X}_c$ comprise labeled graph signals from C classes, where each subset $\mathcal{X}_c = \{\mathbf{x}_p^{(c)}\}_{p=1}^{P_c}$ represents signals from class c . Each class is assumed to be associated with a distinct underlying graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c, \mathbf{W}_c)$, where $\mathbf{W}_c \in \mathbb{R}^{N \times N}$ is the adjacency matrix. The main assumption of this method is that the signals of each class are smooth with respect to their own graph while non-smooth to the graph of the other classes.

This setup is analogous to a multiple subspace model, where signals in each class are assumed to be spanned by a few Laplacian eigenvectors of their corresponding graph, thus forming a low-dimensional subspace [10].

3.2 Optimization Objective

The goal is to learn each class-specific adjacency matrix \mathbf{W}_c such that signals in class c are smooth on \mathcal{G}_c , while signals from other classes are not. To this end, the following convex optimization problem is solved for each class:

$$\min_{\mathbf{W}_c \in \mathcal{W}_m} \|\mathbf{W}_c \circ \mathbf{Z}_c\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{W}_c \mathbf{1}) + \beta \|\mathbf{W}_c\|_F^2 - \gamma \sum_{k \neq c} \|\mathbf{W}_c \circ \mathbf{Z}_k\|_1, \quad (3)$$

where:

- \mathbf{W}_c is the symmetric adjacency matrix for class c , constrained to the set $\mathcal{W}_m = \{\mathbf{W} \in \mathbb{R}_+^{N \times N} \mid \mathbf{W} = \mathbf{W}^\top, \text{diag}(\mathbf{W}) = 0\}$.
- \mathbf{Z}_c is a pairwise distance matrix constructed from signals in \mathcal{X}_c .
- \circ denotes the Hadamard (elementwise) product.
- α , β , and γ are positive regularization parameters.

The first term encourages edge weights to align with similarity between class c signals. The second term avoids trivial solutions by enforcing non-zero node degrees. The third term imposes sparsity control via the Frobenius norm. The fourth term enforces discriminability by penalizing smoothness of signals from other classes on the learned graph.

3.3 Interpretation and Use

Minimizing $\|\mathbf{W}_c \circ \mathbf{Z}_c\|_1$ promotes graphs where class c signals are smooth (i.e., their energy is concentrated in low-frequency components of the GFT). Simultaneously, maximizing $\|\mathbf{W}_c \circ \mathbf{Z}_k\|_1$ for $k \neq c$ encourages signals from other classes to be non-smooth, pushing their spectral energy into higher frequencies. This induces a discriminative spectral structure, akin to Fisher's Linear Discriminant Analysis (LDA) [11, 12], where inter-class scatter is maximized while intra-class scatter is minimized.

The resulting graphs can be used to compute graph Fourier transforms that are tailored for classification, enabling the extraction of class-discriminative features even from previously unseen or unlabeled graph signals.

3.4 Optimization Algorithm

We simply prove that the objective in eq (3) could be turned into the objective (2) by introducing a new variable \tilde{X} . In this manner, we simply use the algorithm proposed by Kalefolias to learn a graph for each class. Additionally, this format makes us able to use better measures of distance such as correlation making it suitable for cases with inconsistent channel scales.

$$\begin{aligned} \tilde{X} &= [X_c, i\sqrt{\gamma}X_1, i\sqrt{\gamma}X_2, \dots] \\ \Rightarrow \frac{1}{2} \|\tilde{\mathbf{W}} \circ \tilde{\mathbf{Z}}\|_1 &= \text{trace}(\tilde{X}^T L_c \tilde{X}) = \text{trace} \left(\begin{bmatrix} X_c^T \\ i\sqrt{\gamma}X_1^T \\ \vdots \end{bmatrix} L_c \begin{bmatrix} X_c & i\sqrt{\gamma}X_1 & \cdots \end{bmatrix} \right) \\ &= \text{trace} \left(\begin{bmatrix} X_c^T L_c X_c & i\sqrt{\gamma}X_c^T L_c X_1 & \cdots & i\sqrt{\gamma}X_c^T L_c X_N \\ i\sqrt{\gamma}X_1^T L_c X_c & -\gamma X_1^T L_c X_1 & \cdots & -\gamma X_1^T L_c X_N \\ \vdots & \vdots & \ddots & \vdots \\ i\sqrt{\gamma}X_N^T L_c X_c & -\gamma X_N^T L_c X_1 & \cdots & -\gamma X_N^T L_c X_N \end{bmatrix} \right) \\ &= \text{trace}(X_c^T L_c X_c) - \gamma \sum_{k \neq c} \text{trace}(X_k^T L_c X_k) = \frac{1}{2} \|\mathbf{W}_c \circ \mathbf{Z}_c\|_1 - \frac{\gamma}{2} \sum_{k \neq c} \|\mathbf{W}_c \circ \mathbf{Z}_k\|_1 \end{aligned}$$

4 Methodology

4.1 Experimental Setup

Experimental Paradigm: The experiment consists of two sessions conducted on two different days with 9 participants who suffered from disability caused by spinal cord injury and stroke. During each session subjects performed 40 trials for each of the 5 tasks while the EEG signals were recorded. At the beginning of each trial of the experiment, a cross cue is shown to the subjects to fixate on it for relaxation and avoiding eye movement. At $t = 3$ s, a beep was sounded and one of the five cues displayed in Fig 3 appeared right after the beep and the subject had to perform the mental or motor imagery task right related to the cue until another beep was sounded at $t = 10$ s. After the inter-trial-interval (about 2.5–3.5 s), the next trial began. The cues were ordered randomly and each corresponds to a specific mental or motor imagery task described as below [13]:

- **Word Association (WORD):** Participants were instructed to generate as many words as possible starting with a given letter in Spanish. Letters were presented in a pseudo-randomized sequence.
 - **Mental Subtraction (SUB):** Participants performed successive elementary subtractions, subtracting a randomly chosen single-digit number from a randomly selected number between 15 and 30.
 - **Spatial Navigation (NAV):** Participants were asked to mentally imagine navigating through a familiar home environment, focusing on spatial orientation.
 - **Motor Imagery of the Right Hand (HAND):** Participants engaged in kinesthetic imagery of repetitively squeezing a hand-sized ball with their right hand.
- Motor Imagery of Both Feet (FEET): Participants performed kinesthetic imagery of repetitive, self-paced movements of both feet without actual physical motion.

Data Acquisition The EEG data was recorded using 30 electrodes according to the international 10–20 system at a sampling rate of 256 Hz. The electrodes include channels AFz, F7, F3, Fz, F4, F8, FC3, FCz, FC4, T3, C3, Cz, C4, T4, CP3, CPz, CP4, P7, P5, P3, P1, Pz, P2, P4, P6, P8, PO3, PO4, O1, and O2. For recording the signals, the g.tec GAMMAsys system with g.LADYbird active electrodes and two g.USBamp biosignal amplifiers (Guger Technologies, Graz, Austria) was used. In addition, the EEG signals were filtered using a notch filter and a band pass filter with 0.5 and 100 Hz cutoff frequencies [13].

4.2 Data Preprocessing

We only used data from 4 subjects, since the performance of other subjects and the data acquired from them was extremely poor [13]. For enhancement of signals, we first normalized the EEG channels by their variance. Subsequently, the trials were extracted by cropping signals from 4.5 to 7.5 s after each trial start. In order to show our robustness of our method, no further processing, filtering, or trial removal was performed.

4.3 Proposed Method

Learning Class-Specific Graphs The first step in our method is to learn class-specific graphs using Discriminative Graph Learning. The reason for choosing this graph learning algorithm is its ability to focus on task-related brain networks and maintain more informative features. To run this algorithm, we calculate channel-wise distances of each class separately and pass the distance matrices to the graph learning algorithm in eq ?. Additionally, we used a correlation-based distance metric as an alternative to Euclidean distance. This is because EEG channels have different scales and artifacts that make the Euclidean distance unsuitable.

Graph Fourier Filtering Subsequently, we transform the EEG signals from channel domain to graph Fourier domain. The graph Fourier transform reveals insights into how brain regions interact over time and how different graph modes are combined at each moment. More specifically, each GFT coefficient is the projection of the graph signal onto one Laplacian eigenvector and tells you how much of that graph frequency mode is present in the signal, just like Fourier coefficients tell you how much of each sinusoid is present in a time signal. [15] Since Laplacian eigenvectors corresponding to lower eigenvalues are low-frequency modes of the graph, these eigenvectors indicate smooth modes of the graph. In other words, lower-frequency graph modes, associated with small Laplacian eigenvalues, correspond to smooth patterns that are spread across the brain and often align with large-scale sub-networks. In contrast, higher-frequency modes, associated with larger eigenvalues, capture localized variations with strong differences between neighboring brain regions. [15] [16] Therefore, we only keep first N_{gft} frequencies of the graph which is equivalent to applying a low-pass filter in the graph Fourier domain.

Table 1: Classification Accuracies

Subject	Class	Accuracy (%)		
		Our Method	Structural	CSP
Subject F	WORD vs HAND	80.00	80.00	68.75
	SUB vs NAV	90.00	86.25	95.00
	NAV vs HAND	90.00	87.50	85.00
	SUB vs HAND	85.00	82.50	85.00
	SUB vs FEET	90.00	88.75	87.50
	WORD vs FEET	73.75	72.50	68.75
	NAV vs FEET	67.50	67.50	65.00
	WORD vs SUB	86.25	78.75	87.50
	WORD vs NAV	86.25	80.00	78.75
	HAND vs FEET	81.25	76.25	73.75
Subject D	WORD vs HAND	76.25	70.00	70.00
	SUB vs NAV	85.00	81.25	77.50
	NAV vs HAND	80.00	83.75	82.50
	SUB vs HAND	78.75	73.75	70.00
	SUB vs FEET	70.00	73.75	70.00
	WORD vs FEET	86.25	80.00	76.25
	NAV vs FEET	75.00	78.75	83.75
	WORD vs SUB	73.75	70.00	70.00
	WORD vs NAV	86.25	81.25	93.75
	HAND vs FEET	67.50	61.25	52.50
Subject E	WORD vs HAND	85.00	80.00	76.25
	SUB vs NAV	76.25	73.75	67.50
	NAV vs HAND	81.25	77.50	75.00
	SUB vs HAND	78.75	72.50	68.75
	SUB vs FEET	73.75	76.25	76.25
	WORD vs FEET	81.25	80.00	81.25
	NAV vs FEET	78.75	75.00	76.25
	WORD vs SUB	76.25	73.75	67.50
	WORD vs NAV	75.00	71.25	72.50
	HAND vs FEET	58.75	57.50	58.75

Finding a Discriminative Subspace Through FKT and Classification The next step is finding a subspace in which the data of each class is highly discriminated. This is done by applying a Fukunaga-Koontz transform on the graph filtered signals. Therefore, by calculating variance of the resulting matrix along the time dimension, we get greatly informative features for classification. Lastly, we train a random forest classifier on these features and perform classification. This procedure allows us to exploit the strength of Graph-Signal Processing and highly non-linear classifiers. Since the data acquired for each subject is too scarce and noisy, we performed 10-fold cross validation and calculated average validation accuracy of each fold to better make use of the available data. Additionally, we implemented and performed other algorithms used previously and compared them with our method [2] [14].

5 Results

5.1 Accuracies

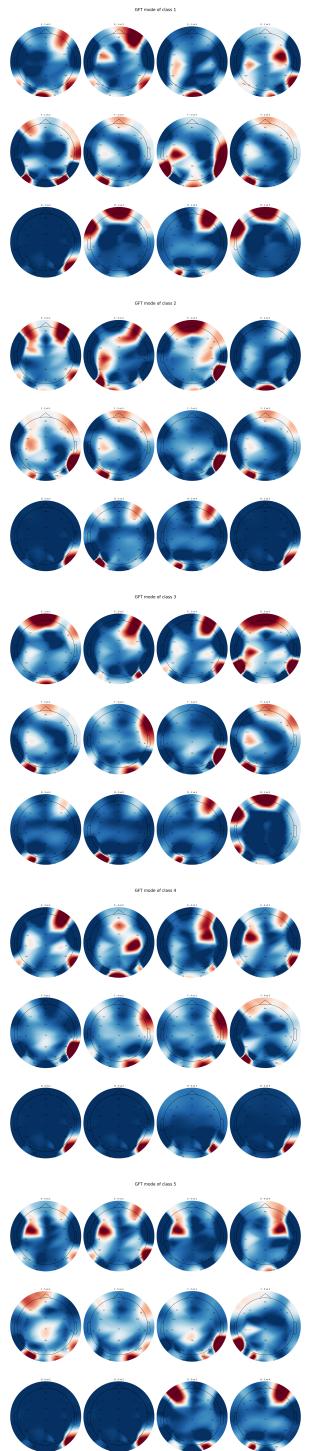


Figure 1: Graph Modes

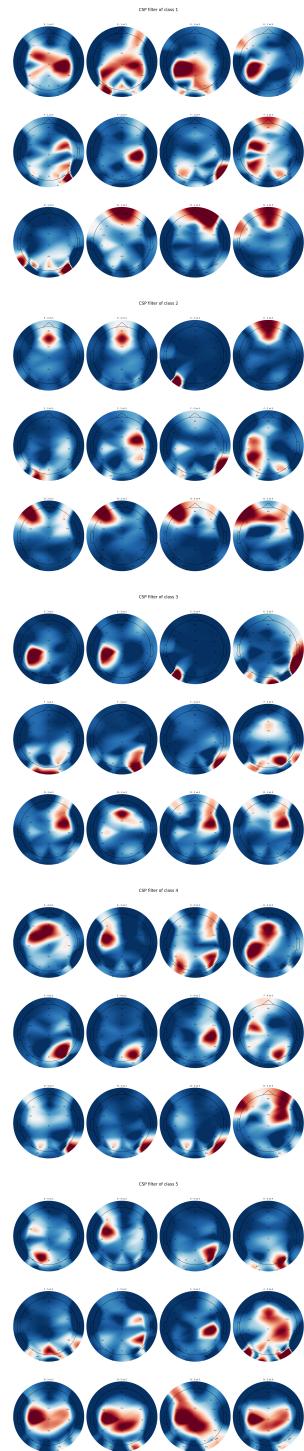


Figure 2: CSP Filters

6 Conclusion and Future Work

Although Graph Neural Networks need a large amount of data to perform properly and achieve considerable success on small datasets, several methods can be used to help tackle this challenge. One of the mostly use methods is using different loss functions and regulations such as contrastive learning which prevents the neural networks from overfitting. Another popular method is using deep generative models to increase the amount of datapoints in the training set. Current trending deep generative models are Generative Adversarial Networks and Diffusion-Based models which can be implemented on graph neural networks and graph signals to expand neural datasets.

References

- [1] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [2] S. Itani and D. Thanou, "A Graph Signal Processing Framework for the Classification of Temporal Brain Data," 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 2021, pp. 1180-1184, doi: 10.23919/Eusipco47968.2020.9287486. keywords: Autism;Signal processing algorithms;Signal processing;Tools;Prediction algorithms;Topology;Classification algorithms;Graph signal processing;machine learning;explainability;decision trees;functional MRI;autism spectrum disorder,
- [3] Fukunaga, K., and Koontz, W. L. G. (1970). *Application of the Karhunen–Loève expansion to feature selection and ordering*. IEEE Transactions on Computers, C-19(4), 311–318.
- [4] How to learn a graph from smooth signals
- [5] George Kour and Raid Saabne. Real-time segmentation of on-line handwritten arabic script. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 417–422. IEEE, 2014.
- [6] George Kour and Raid Saabne. Fast classification of handwritten on-line arabic characters. In *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*, pages 312–318. IEEE, 2014.
- [7] Guy Hadash, Einat Kermany, Boaz Carmeli, Ofer Lavi, George Kour, and Alon Jacovi. Estimate and replace: A novel approach to integrating deep neural networks with existing applications. *arXiv preprint arXiv:1804.09028*, 2018.
- [8] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, “Learning Laplacian matrix in smooth graph signal representations,” *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.
- [9] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 481–496, 2017.
- [10] H. Huang, Y. Wang, and L. Wang, “Graph learning based classification via multiple subspace analysis,” *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5309–5321, 2016.
- [11] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [12] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [13] Scherer R, Faller J, Friedrich EV, Opisso E, Costa U, Kübler A, Müller-Putz GR. Individually adapted imagery improves brain-computer interface performance in end-users with disability. *PLoS One*. 2015 May 18;10(5):e0123727. doi: 10.1371/journal.pone.0123727. PMID: 25992718; PMCID: PMC4436356.
- [14] Saboksayr Seyed, Mateos. Gonzalo, Cetin Mujdat. (2021). EEG-Based Emotion Classification Using Graph Signal Processing. 1065-1069. 10.1109/ICASSP39728.2021.9414342.
- [15] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [16] S. Atasoy, I. Donner, A. S. Fries, J. D. He, E. P. Van de Ville, and J. Pearl, “Human brain networks function in connectome-specific harmonic waves,” *Nature Communications*, vol. 7, article 10340, 2016.