Fast and Accurate Multiclass Inference for MI-BCIs Using Large Multiscale Temporal and Spectral Features

Michael Hersche*, Tino Rellstab*, Pasquale Davide Schiavone*, Lukas Cavigelli*, Luca Benini*‡, Abbas Rahimi*†

*Integrated System Laboratory, ETH Zurich, Switzerland. ‡DEI, University of Bologna, Italy.
†EECS Department, University of California, Berkeley.
Emails: {herschmi, tinor}@ethz.ch, {pschiavo, cavigelli, lbenini, abbas}@iis.ee.ethz.ch

Abstract—Accurate, fast, and reliable multiclass classification of electroencephalography (EEG) signals is a challenging task towards the development of motor imagery brain–computer interface (MI-BCI) systems. We propose enhancements to different feature extractors, along with a support vector machine (SVM) classifier, to simultaneously improve classification accuracy and execution time during training and testing. We focus on the well-known common spatial pattern (CSP) and Riemannian covariance methods, and significantly extend these two feature extractors to multiscale temporal and spectral cases. The multiscale CSP features achieve 73.70±15.90% (mean± standard deviation across 9 subjects) classification accuracy that surpasses the state-of-the-art method [1], 70.6±14.70%, on the 4-class BCI competition IV-2a dataset. The Riemannian covariance features outperform the CSP by achieving 74.27±15.5% accuracy and executing 9× faster in training and 4× faster in testing. Using more temporal windows for Riemannian features results in 75.47±12.8% accuracy with 1.6× faster testing than CSP.

Index Terms—EEG, motor imagery, brain–computer interfaces, multiclass classification, multiscale features, SVM.

I. INTRODUCTION

A brain–computer interface (BCI) is a system which enables communication and control between the brain and an external device. The system aims to recognize human intentions from spatiotemporal neural activity, typically recorded non-invasively by a large set of electroencephalogram (EEG) electrodes. There exist various applications for BCI such as motor imagery (MI), which is the cognitive process of thinking of a motion, e.g. of the left or right hand, without actually performing it. MI-BCI systems are designed to find patterns in the EEG signals and match the signal to the motion that was thought of. Such information could enable communication for severely paralyzed users or the control of a prosthesis [2].

Recognizing these patterns, however, is still susceptible to errors. The non-stationarity and high inter-subject variance of the EEG signal requires large amounts of labeled data, which mostly is not available. The lack of sufficient labeled data makes training of complex classifiers with large numbers of parameters difficult. However, the knowledge of the structure of the EEG signal allows us to design comprehensive feature extractors, which can be combined with simpler and robust classifiers.

For extracting the most discriminative features, different approaches have been suggested. The well-known common spatial patterns (CSP) algorithm learns spatial filters which maximize the discriminability between two classes [3]. Its performance is highly dependent on the considered operational frequency bands. Hence, in most applications the data is first split into several frequency bands and then spatially filtered. This is better known as filter bank common spatial pattern (FBCSP) [4].

Augmented CSP features originating from a multilevel frequency decomposition have been used in Convolutional Neural Networks (CNNs) for classification [5] and achieved an average classification accuracy of 69.27% on the 4-class data set of the BCI competition IV-2a [6]. This approach is further extended in [1], where multiple CNNs are fed with dynamic energy features and combined with a static energy network. A multilevel spectral and temporal decomposition in connection with CSPs is used as feature extractor. The best frequency bands and spatial filters are selected using mutual information as feature selection measure. With this architecture, an average classification accuracy of 70.60% has been achieved.

Recently, Riemannian approaches [7] allow the direct manipulation of spatial EEG signal covariance matrices using the dedicated Riemannian geometry. In contrast to the Euclidean geometry, it introduces a more accurate approximation of the distance on smoothly curved spaces. The Riemannian approach yields a large number of features as opposed to the CSP that induces a reduction of the feature space susceptible to loss of important spatial information. Furthermore, Riemannian kernel features do not require labeled data for training, in contrast to CSPs, and therefore allow unsupervised feature calibration. Riemannian features have been classified with multi-kernel relevance vector machine achieving an accuracy of 70.30% [8].

In this paper, CSP and Riemannian methods are enhanced to multiscale spectral and temporal features capturing the dynamic nature of the EEG signals. Therefore, we introduce a new architecture which includes four stages: temporal division, spectral division, CSP or Riemannian feature generation, and classification with a support vector machine (SVM). This vastly increases the number of features, introduces redundancy, yet increases the classification accuracy on average by $\approx 5\%$ compared to the state-of-the-art method [1]. Using almost the equal size of features, the Riemannian covariance features perform slightly better than multiscale CSP features in accu-
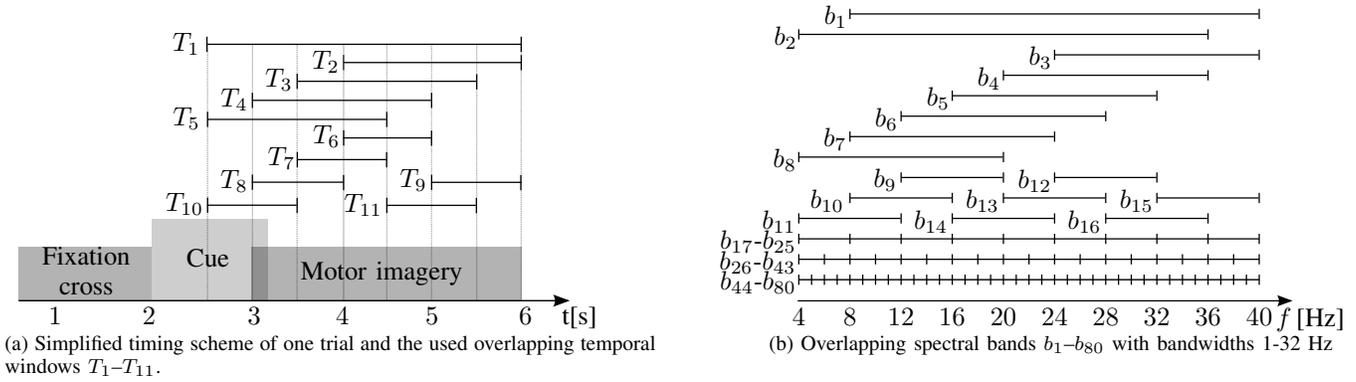
(a) Simplified timing scheme of one trial and the used overlapping temporal windows T_1–T_{11}.

(b) Overlapping spectral bands b_1–b_{80} with bandwidths 1-32 Hz

Fig. 1: Multiscale temporal (a) and spectral (b) decomposition of the EEG signal.

racy (74.27±15.5% vs. 73.70±15.90%) and execute 9× faster in training and 4× faster in testing. Using more temporal windows increases Riemannian covariance features by 3× resulting in 75.47±12.8% accuracy and 1.6× faster testing than CSP. This improvement in compute time is particularly important when targeting an online, real-time implementation.

In the following sections, we briefly introduce the CSP and Riemannian frameworks. The different feature extraction methods are tested on the BCI competition IV-2a data using a support vector machine (SVM) as classifier.

II. DATASET DESCRIPTION

The BCI Competition IV 2a dataset [6] consists of EEG data from 9 different subjects. The subjects were requested to carry out four different MI tasks, namely the imagination of the movement of the left hand, right hand, both feet and tongue. Two sessions were recorded on two different days. For each subject a session consists of 72 trials per class yielding 288 trials in total. One session is used for training and the other for testing exclusively. The signal was recorded using 22 EEG electrodes according to the 10-20 system. It is bandpass filtered between 0.5 Hz and 100 Hz and sampled with 250 Hz. In addition to the 22 EEG channels, three Electrooculography (EOG) channels give information about the eye movement. An expert marked the trials containing artifacts based on the EOG signal. This way 9.41% of the trials were excluded from the dataset. The number of trials per class remains balanced.

III. COMMON SPATIAL PATTERN FEATURES

Common spatial pattern was first introduced for discriminating between two different kinds of populations, one with a neurological disorder and one without [9]. The basic idea of CSP is to combine the spatial channels such that the average variance between two classes is maximized. We first introduce CSP for the basic binary class case.

Given the signal $\boldsymbol{X} \in \mathbb{R}^{N_c \times N_s}$, we can estimate the covariance matrix of the signals as

$$\boldsymbol{C} = \frac{1}{N_s - 1}(\boldsymbol{X}\boldsymbol{X}^T), \tag{1}$$

where N_c denotes the number of channels, N_s the number of time samples and $\boldsymbol{A}^T$ the transpose of a matrix $\boldsymbol{A}$. The

signal has zero-mean component since it is bandpass filtered. We calculate the arithmetic average covariance matrix over all occurrences of class $j \in \{0, 1\}$ by

$$\overline{\boldsymbol{C}}_j = \frac{1}{N_j} \sum_{k=1}^{N_j} \boldsymbol{C}_j^{(k)}, \tag{2}$$

where $\boldsymbol{C}_j^{(k)}$ corresponds to the covariance matrix where class j occurred and N_j to the total number of occurrences of class j. The goal of CSP is to find a spatial filter $\boldsymbol{w} \in \mathbb{R}^{N_c}$ which maximizes the Rayleigh quotient

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \overline{\boldsymbol{C}}_1 \boldsymbol{w}}{\boldsymbol{w}^T \overline{\boldsymbol{C}}_2 \boldsymbol{w}}. \tag{3}$$

This is achieved by solving the generalized eigenvalue decomposition (GEVD) problem

$$\overline{\boldsymbol{C}}_1 \boldsymbol{U} = \boldsymbol{\Sigma} \overline{\boldsymbol{C}}_2 \boldsymbol{U} \tag{4}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix containing the eigenvalues in descending order and $\boldsymbol{U}$ contains the corresponding eigenvectors as columns. We then take pairs of eigenvectors with the corresponding largest and smallest eigenvalues. They build a set of spatial filters. Finally, the feature f_l is the logarithm of the spatial filtered and normalized variances.

$$f_l = \log\left(\frac{\boldsymbol{w}_l^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{w}_l}{\sum_k \boldsymbol{w}_k^T \boldsymbol{X}\boldsymbol{X}^T \boldsymbol{w}_k}\right) \tag{5}$$

Various approaches to extend CSP to the multiclass problem have been proposed. A heuristic but most successful technique is to perform two-class CSP on all possible combinations of classes [10]. In the four class case, this results in at least 12 spatial filters, namely two for each pair of classes:

$$\{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\} \tag{6}$$

Multiscale Temporal and Spectral CSP Features

MI activities cause brain oscillations mainly within the μ (8–14 Hz) and β (14–30 Hz) bands [11]. However, which bands reveal the most discriminative information is highly subject-dependent. Furthermore, not every subject needs the same amount of time to initiate MI after the cue. Therefore,
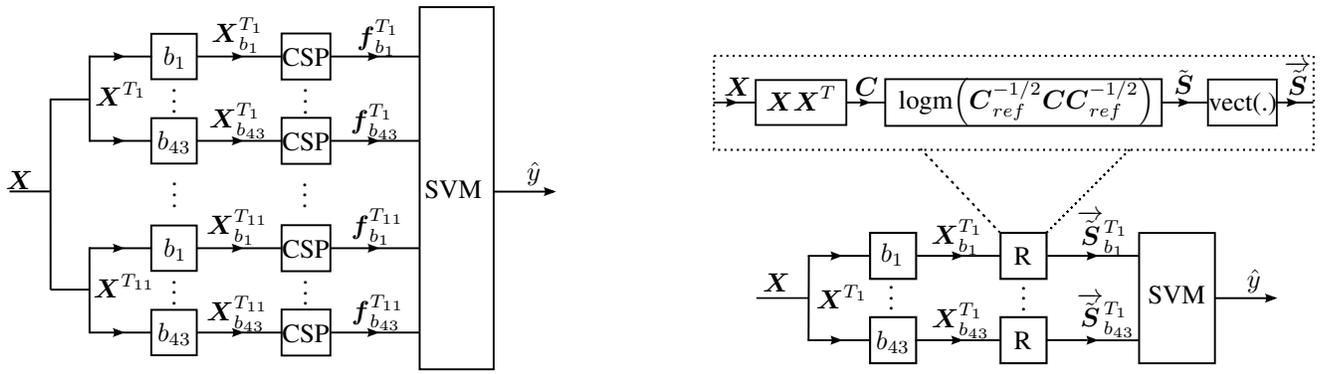
(a) Multiscale CSP features using 11 temporal windows T_1–T_{11} and 43 frequency bands b_1–b_{43}. Every CSP block uses 24 separate spatial filters and calculates the features according to (5) which gives a total of 11352 features.

(b) Riemannian covariance features using 43 frequency bands b_1–b_{43} and one temporal window T_1. Every Riemannian block R includes the calculation of $\bar{S}$ (23) and its vectorization resulting in 253 features per block and a total of 10879 features.

Fig. 2: Multiscale feature generation for (a) CSP, and (b) Riemannian features. The signal is split into temporal windows as in Figure 1a and then band pass filtered using bands according to Figure 1b. All features generated either by CSP or Riemannian are stacked into one vector and fed to a SVM for classification.

we divide the signal into multiscale temporal and spectral components before applying CSP. Figure 2a shows the tree-like structure of feature generation. We add temporal information by dividing the data into temporal windows, shown in Figure 1a. Every temporal window is split again into various spectral bands, shown in Figure 1b, using second order Butterworth filters. Both the temporal windows and the spectral bands induce redundant information which help to increase the robustness of the feature extractor. For every leaf we use a separate set of 24 spatial filters and apply (5). This results in a total number of features:

$$N = N_{temp} \times N_{spec} \times N_{spat} = 11 \times 43 \times 24 = 11352 \quad (7)$$

All the features are stacked into one vector and fed to a ℓ_2-regularized SVM.

IV. RIEMANNIAN COVARIANCE FEATURES

This section describes briefly the Riemannian geometry in the context of covariance matrices. Then, a Riemannian based kernel is introduced which can be applied on a linear SVM using the kernel trick.

1) Introduction to Riemannian geometry: Roughly speaking, Riemannian geometry studies smoothly curved spaces that locally behave like Euclidian spaces [7]. The idea is to locally approximate a smooth curved space on a tangent space. This concept is not new, since we approximate the Earth locally as a flat space.

The space of real symmetric positive definite $N_c \times N_c$ covariance matrices P_{N_c} forms a smooth *differential manifold*, which allows the mapping between the manifolds and their corresponding local tangent space. If the number of time samples for estimating the covariance matrices is large enough, they are in P_{N_c}.

Let us fix a reference point $C_{ref} \in P_{N_c}$. The corresponding tangent space is then denoted as $T_{C_{ref}} P_{N_c}$. As we are working

with a smooth space, slightly changing the reference point C_{ref} should not affect the calculations in the tangent space too much. The logarithmic map is used to project the vectors from the submanifold P_{N_c} to its tangent space $T_{C_{ref}} P_{N_c}$. Conversely, the exponential map projects the point on the tangent space back to the submanifold:

$$S = \text{Log}_{C_{ref}}(C) = C_{ref}^{1/2} \text{logm} \left(C_{ref}^{-1/2} C C_{ref}^{-1/2} \right) C_{ref}^{1/2}, \quad (8)$$

$$C = \text{Exp}_{C_{ref}}(S) = C_{ref}^{1/2} \text{expm} \left(C_{ref}^{-1/2} S C_{ref}^{-1/2} \right) C_{ref}^{1/2}, \quad (9)$$

where logm() and expm() denote the matrix logarithm and matrix exponential function, respectively. Next, we introduce various geometric measures in the Riemannian sense and compare them to the Euclidean counterparts. The Euclidean matrix inner product in the space of $N_c \times N_c$ real matrices is the Frobenius inner product defined as

$$\langle C_A, C_B \rangle_F = \text{Tr} \left(C_A^T C_B \right), \quad (10)$$

where $\text{Tr}(\cdot)$ stands for the trace. Let S_1 and S_2 be two elements of the tangent space $T_{C_{ref}} P_{N_c}$. Then, we define the Frobenius inner product on the tangent space $T_{C_{ref}} P_{N_c}$ as

$$\langle S_1, S_2 \rangle_{C_{ref}} = \text{Tr} \left(C_{ref}^{-1} S_1 C_{ref}^{-1} S_2 \right). \quad (11)$$

When using two elements C_1 and C_2 directly out of P_{N_c} they first need to be transformed into the tangent space using the logarithmic mapping. The inner product can be computed according to (11). Depending on the chosen inner product we can also define different distance metrics. Let us first recall the Euclidean distance between two matrices:

$$\delta_E(C_A, C_B) = \| C_A - C_B \|_F \quad (12)$$

Conversely, the Riemannian distance between C_A and $C_B \in P_{N_c}$ is

$$\delta_R(C_A, C_B) = \|\text{logm}\left(C_A^{-1}C_B\right)\|_F. \quad (13)$$

The difference between these two distances is that the Euclidean distance determines the shortest distance along direct paths, whereas the Riemannian distance rather searches the shortest path along geodesics [12].

Finally, the last measure we encounter often in BCI are means of covariance matrices. Lets say we have a set $\{C_i\}_{i=1}^n$ including n covariance matrices. The arithmetic mean which is associated with the Euclidean distance metric is

$$\mathfrak{U}(C_1, C_2, ..., C_n) = \frac{1}{n}\sum_{i=1}^n C_i. \quad (14)$$

In the Riemannian framework, we use the geometric mean (also called Fréchet mean or Karcher mean), which tries to find a point C that minimizes the sum of all squared Riemannian distances [12]

$$\mathfrak{G}(C_1, C_2, ..., C_n) = \underset{C}{\arg\min}\sum_{i=1}^n \delta_R(C, C_i)^2. \quad (15)$$

This mean is not straightforward to calculate, as no closed-form solution exists. However, numerous iterative algorithms solve this problem numerically [12].

2) Vectorization of symmetric matrices: To use a classifier like SVM in connection with covariance matrices, we vectorize the matrix as

$$\overrightarrow{C} := \text{vect}(C) = [C_{1,1}; \sqrt{2}C_{1,2}; ...C_{N_c,N_c}] \in \mathbb{R}^{(N_c+1)N_c/2}, \quad (16)$$

since covariance matrices are symmetric. The off-diagonal elements are scaled by $\sqrt{2}$ to preserve the norm, i.e.

$$\|C\|_F = \|\text{vect}(C)\|_2,. \quad (17)$$

3) Kernel approach: Now we build a matrix kernel which takes the Riemannian distance into account [13]. The kernel formulation of the SVM is

$$f(C) = \beta + \sum_{i=1}^n \alpha_i y_i k(C_i, C; C_{ref}). \quad (18)$$

Here, $k(.,.; C_{ref})$ is the kernel function given the reference C_{ref}, β and α_i trainable coefficients and C_i the support vector. The kernel function can be written as

$$k(C_i, C; C_{ref}) = \langle \Phi(C_i), \Phi(C) \rangle_{C_{ref}} \quad (19)$$
$$= \text{Tr}\left(C_{ref}^{-1}\text{Log}_{C_{ref}}(C_i)C_{ref}^{-1}\text{Log}_{C_{ref}}(C_i)\right) \quad (20)$$
$$= \langle \tilde{S}_i, \tilde{S} \rangle_F \quad (21)$$
$$= \overrightarrow{S}_i^T \overrightarrow{S}_i. \quad (22)$$

We use the Riemannian inner product $\langle ., . \rangle_{C_{ref}}$ with respect to the reference point C_{ref} defined in (11). Therefore, the matrices C_i and C are projected from the submanifold to the

tangent space $T_{C_{ref}}P_{N_c}$ using the logarithmic map $\Phi(C) = \text{Log}_{C_{ref}}(C)$. We introduce a new matrix

$$\tilde{S_i} = C_{ref}^{-1/2}\text{Log}_{C_{ref}}(C_i)C_{ref}^{-1/2} = \text{logm}\left(C_{ref}^{-1/2}C_iC_{ref}^{-1/2}\right). \quad (23)$$

Together with the definition of the Frobenius norm on the Euclidean space, we obtain (21). Due to the norm conversation (17) of the vectorization, we get (22) [13].

This equivalence allows a kernel-free linear SVM implementation. The reference matrix C_{ref} can be selected in various ways. For example, one could use a mean covariance matrix (either the geometric $\mathfrak{G}$ or arithmetic $\mathfrak{U}$ mean) from the training data or simply a $N_c \times N_c$ identity matrix $\mathfrak{I}$. Using the identity matrix as reference is equivalent to calculating the matrix logarithm and applying the vectorization (16). Finally, we apply the same multiscale division as in the previous CSP section. However, since the number of features per leaf is 253, we use only the largest temporal window. The structure is shown in Figure 2b. A separate reference point C_{ref} is used for every frequency band and is averaged over all temporal windows.

V. EXPERIMENTAL RESULTS

The aforementioned approaches are evaluated on the dataset 2a of the BCI competition IV. The classification accuracy is measured as

$$\text{classification accuracy} = \left(\frac{N_{\text{correct}}}{N_{\text{total}}}\right) \times 100\%, \quad (24)$$

where N_{correct} is the number of correct classified trials and N_{total} the total number of trials in the test set per subject. To evaluate the computational cost the average training and testing time per subject over all trials is measured. The training time includes the preprocessing and the training of the classifier, whereas the testing time covers the calculation of the features as well as the classification itself. The experiments were conducted on an Intel Core i7-4790 3.6 GHz processor with 16 GB RAM. The code is available[1].

The features are classified with a ℓ_2-regularized SVM. The regularization hyperparameter C is determined in 5-fold cross-validation using grid search within the range $[10^{-2}, 10^3]$. Moreover the temporal windows and frequency bands were selected in cross-validation as well. Table I shows the results on the test set.

The proposed Riemannian covariance features achieve higher accuracies than CSPs with a similar number of features. Moreover, the testing time of the Riemannian features is 4× lower than those of the CSP. The Riemannian feature extractor exploits 253 features per frequency band and temporal window instead of 24 in the CSP case. A single temporal window is sufficient for the Riemannian features to achieve the same accuracy as the CSPs.

We evaluate different SVM kernels for the multiscale 11352 CSP features. The linear kernel achieves 73.70% classification accuracy. The nonlinear radial basis kernel function classifies

[1]https://github.com/MultiScale-BCI/IV-2a

TABLE I: Classification accuracy (%), training and testing time for the common spatial patterns (CSPs) and Riemannian features tested with different SVM kernels.

	CSP				Riemannian			
No. features	11352	11352	11352	20856	10879	10879	10879	32637
SVM kernel	linear	rbf	poly	linear	linear	linear	linear	linear
Riemannian kernel	N/A	N/A	N/A	N/A	$\mathfrak{G}$	$\mathfrak{U}$	$\mathfrak{I}$	$\mathfrak{G}$
Spectral bands	b_1-b_{43}	b_1-b_{43}	b_1-b_{43}	b_1-b_{80}	b_1-b_{43}	b_1-b_{43}	b_1-b_{43}	b_1-b_{43}
Temporal windows	T_1-T_{11}	T_1-T_{11}	T_1-T_{11}	T_1-T_{11}	T_1	T_1	T_1	T_1,T_2,T_5
Subject 1	86.83	85.41	83.27	84.70	91.81	90.75	84.70	90.04
Subject 2	57.24	57.24	49.47	57.60	51.59	47.70	48.76	55.48
Subject 3	86.45	80.95	77.66	84.98	83.52	85.35	84.25	81.32
Subject 4	61.40	62.28	56.58	58.33	73.25	63.16	58.33	71.93
Subject 5	61.23	67.03	60.14	63.77	63.41	67.39	62.32	69.57
Subject 6	50.70	50.23	44.65	48.84	58.60	58.60	56.74	56.74
Subject 7	92.42	86.28	70.40	88.45	86.64	89.89	81.59	85.56
Subject 8	87.82	87.45	86.72	87.08	81.55	85.24	79.34	83.76
Subject 9	79.17	85.98	79.92	84.85	82.58	80.30	80.68	84.85
Avg. accuracy	**73.70**±15.9	73.65±14.5	67.65±15.4	73.18±15.7	**74.77**±13.9	74.27±15.5	70.75±14.0	**75.47**±12.8
Avg. training time [s]	49.46	42.19	43.65	97.77	17.08	**5.45**	13.79	50.93
Avg. testing time [s]	23.19	24.24	25.40	47.35	**5.53**	**5.33**	5.56	13.88

on par with the linear kernel while the polynomial kernel was outperformed.

For comparing the geometric mean $\mathfrak{G}$, the arithmetic mean $\mathfrak{U}$ and the identity matrix $\mathfrak{I}$ as reference point in the Riemannian kernel, we use the calculated features of one temporal window (T_1) and 43 spectral bands (b_1-b_{43}) yielding 10879 features. All means are calculated on the training set in order to keep the feature extractor causal. A difference in the training time between the reference point calculation methods is observed: the geometric mean requires iterative numerical operations which results in $3\times$ higher training time than the arithmetic mean. In terms of accuracy, however, we gain 0.50% accuracy by using the geometric mean.

For the highly subject dependent nature of this classification task, we use large multiscale features. However, increasing the number of features to 20856 by adding 1 Hz frequency bands was not found to be beneficial for CSP. Adding more temporal windows (T_2 and T_5) to the Riemannian improves classification accuracy (75.47±12.8%) at the cost of $3\times$ higher computation time. Using even more temporal windows reduces the accuracy that can be due to the overfitting situation for more features with the limited training samples.

VI. Conclusion

We propose an enhanced feature extraction method based on multiscale temporal windows and overlapping spectral bands for multiclass classification of EEG signals. Our method significantly increases the number of features for the CSP and Riemannian covariance methods. In combination with a simple linear SVM classifier our method outperforms the state-of-the-art on average by $\approx 5\%$. This confirms the importance of large multiscale temporal and spectral features for the MI-BCIs. Besides, the Riemannian covariance method is unsupervised, and achieves $4\times$ faster execution time during testing (compared to CSP with almost equal number of features). These are particularly important for a real-time embedded implementation.

References

[1] S. Sakhavi, C. Guan, and S. Yan, "Parallel convolutional-linear neural network for motor imagery classification," in *Proc. IEEE EUSIPCO*, Aug 2015, pp. 2736–2740.

[2] G. Pfurtscheller, G. R. Müller-Putz, R. Scherer, and C. Neuper, "Rehabilitation with brain-computer interface systems," *Computer*, vol. 41, no. 10, pp. 58–65, Oct 2008.

[3] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms," *IEEE Trans. Biomed. Engineering*, vol. 58, no. 2, pp. 355–362, 2011.

[4] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter bank common spatial pattern (fbcsp) in brain-computer interface," in *Proc. IEEE Int. Joint Conference on Neural Networks*, June 2008, pp. 2390–2397.

[5] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented csp features for multi-class motor imagery of eeg signals classification," in *Proc. IEEE EMBC*, 2015, pp. 2620–2623.

[6] C. Brunner, R. Leeb, G. R. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI competition 2008 - graz data set a." [Online]. Available: http://bnci-horizon-2020.eu/database/data-sets

[7] F. Yger, M. Berar, and F. Lotte, "Riemannian approaches in brain-computer interfaces: A review," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1753–1762, Oct 2017.

[8] C. H. Nguyen and P. Artemiadis, "EEG feature descriptors and discriminant analysis under riemannian manifold perspective," *Neurocomputing*, vol. 275, pp. 1871 – 1883, 2018.

[9] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, "Spatial patterns underlying population differences in the background EEG," *Brain Topography*, vol. 2, no. 4, pp. 275–284, Jun 1990.

[10] M. Grosse-Wentrup and M. Buss, "Multiclass common spatial patterns and information theoretic feature extraction," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1991–2000, Aug 2008.

[11] G. Pfurtscheller and F. L. da Silva, "Event-related EEG/MEG synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842 – 1857, 1999.

[12] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 3, pp. 735–747, 2005.

[13] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172 – 178, 2013.