Neurocomputing 588 (2024) 127654

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning visual stimulus-evoked EEG manifold for neural image classification

Salvatore Falciglia [a,c], Filippo Betello [a], Samuele Russo [b], Christian Napoli [a,d,e,*]

[a] Department of Computer, Control and Management Engineering, Via Ariosto 25, Rome, 00185, Italy
[b] Department of Psychology, Sapienza University of Rome, Via dei Marsi 78, Rome, 00185, Italy
[c] The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, 56025, Italy
[d] Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19, Rome, 00185, Italy
[e] Department of Intelligent Computer Systems, Czestochowa University of Technology, al.Armii Krajowej 36, Czestochowa, 42-200, Poland

ARTICLE INFO

Communicated by X. Li

Keywords:
Neural manifold learning
Visual neural decoding
Brain computer interfaces
Neural image classification
Riemann manifold
Spectrogram analysis
Uniform manifold approximation and projection
Variational auto-encoders
Convolutional neural networks

ABSTRACT

Visual neural decoding, namely the ability to interpret external visual stimuli from patterns of brain activity, is a challenging task in neuroscience research. Recent studies have focused on characterizing patterns of activity across multiple neurons that can be described in terms of population-level features. In this study, we combine spatial, spectral, and temporal features to achieve neural manifold classification capable to characterize visual perception and to simulate the working memory activity in the human brain. We treat spatio-temporal and spectral information separately by means of custom deep learning architectures based on Riemann manifold and the two-dimensional EEG spectrogram representation. In addition, a CNN-based classification model is used to classify visual stimulus-evoked EEG signals while viewing the 11-class (i.e., all-black plus 0-9 digit images) MindBigData Visual MNIST dataset. The effectiveness of the proposed integration strategy is evaluated on the stimulus-evoked EEG signal classification task, achieving an overall accuracy of 86%, comparable to state-of-the-art benchmarks.

1. Introduction

Understanding how the human brain works, therefore devising the neural, bioelectrical and biochemical patterns that constitutes the human ability to think, as well as accessing visual memories, has become an emerging field of study both for scientific, diagnostic and clinical purposes [1,2]. Human visual perception (i.e., the collection of light photons and the related electrochemical reactions on the retina) causes a cascade of neural signals to fire from the retina, traveling via the optical nerve to the brain, and propagate through the visual system, eventually reaching the visual cortex [3]. Holding a mental image of something you have seen, even when you are not currently looking at it (specifically using the visual working memory), involves a series of processes occurring outside the visual system, e.g. in the frontal and prefrontal (PFC) association cortices [4]. Previous studies regarding brain lesions allowed us to assert that neurons of PFC encode working memory representations via persistent firing [5]; however, some recent studies have reported that neurons in some areas of the parietal and temporal lobes, classically associated with visual perception, also encode working memory representations via persistent firing [6].

Brain Computer Interfaces (BCIs) represent the last frontier in neuroengineering research, enabling a new level of communication through the brain, based on decoding, learning, and interpreting brain activity [7]. With the term Neural Decoding [8] we refer to the mapping of brain response to stimuli, therefore dealing also with the reconstruction of a stimulus. In particular, Visual Decoding, namely determining the external visual stimulus from brain activity patterns [9], is a challenging task in neuroscience research that is taking great advantage of the recent advances in generative deep learning models that allow us to generate the realistic images by learning the intrinsic statistic distribution of the training data [10,11].

By using visual neural decoding techniques, it is possible to reconstruct and model the visual stimuli, that a person is perceiving [12]. In this scenario the decoding approach can reveal how visual information is represented and processed in the brain cortex, contributing to our understanding of visual perception [13,14].

The significant clinical applications of this study are vast, seeking to expand our medical knowledge and ability to help various categories of mentally and physically disabled people. Extremely important areas of application include communication with people unable to express themselves verbally, such as patients suffering from locked-in syndrome [15] or from aphasia [16], and the treatment of cerebral paralysis and degenerative diseases such as amyotrophic lateral sclerosis [17]. Decoding and translating the brain activity of such patients

* Corresponding author at: Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19, Rome, 00185, Italy.
E-mail address: cnapoli@diag.uniroma1.it (C. Napoli).

https://doi.org/10.1016/j.neucom.2024.127654
Received 21 July 2023; Received in revised form 5 February 2024; Accepted 2 April 2024
Available online 15 April 2024

could help them communicate their thoughts, preferences, or needs. Specifically, severe or progressive aphasia refers to a condition in which individuals have significant difficulties with language production and comprehension [16]. Nonetheless, if their higher cognitive processes and vision remain intact, visual neural decoding may offer a way to establish alternative communication channels or support existing communication methods. The application of visual neural decoding in severe aphasia is still largely speculative [18,19]. However, we propose a number of theoretical scenarios in which studies and techniques, such as the decoding architecture presented in this work, could help this case typology. First, visual neural decoding could help determine what visual stimuli or cues patients are attending to, even if they cannot express it verbally. Deep learning systems could be used to develop non-verbal communication systems that rely on the patient's visual focus as a means of expressing their intentions. In addition, patients with severe aphasia often rely on alternative approaches to communication, such as gestures, writing, or the use of augmentative and alternative communication (AAC) devices. Visual neural decoding could complement these methods by incorporating visual cues or feedback. For example, an AAC device could be designed to provide visual prompts or suggestions based on the patient's ongoing neural responses to help with speech production and comprehension.

Relatively to cognitive function, the model of single neurons alone is not sufficient to gain a complete understanding of the involved processes [20]. In addition to what can be studied at the level of a single neuron, task-relevant information can be represented as patterns of activity across multiple neurons [21]. Unfortunately, in the worst case, characterizing such patterns can require collecting an exponential number of measurements (the curse of dimensionality [22]). However, in most cases the observed number of neurons, or even the number of patterns of neural population activity, can be described in terms of fewer population-level features [23–25]. Since the spatiotemporal dynamics of brain activity is low-dimensional, or at least much lower-dimensional than pattern space, then it stands to reason that such activity can be characterized within reasonable experimental time scales. The main difficulty of such approaches is to identify *neural ensembles* by grouping together neurons with sufficiently highly correlated activity during the same behaviors or in response to the same stimuli. A number of techniques, commonly referred to as Neural Manifold Learning (NML), have been used to accomplish this task [26,27].

Studies in neuroscience and neuroimaging [28] demonstrate that non-invasive imaging techniques like Functional Magnetic Resonance Imaging (fMRI), EEG, and Magnetoencephalography (MEG) can decode human cognitive processes, particularly those related to perception and visual perception. By applying standard notions of information theory, it is reasonable to assume that since our brains function as transmitters and receivers of electrical signals, it is clear that by modifying our system through reliable measures we can decode these signals. However, it remains uncertain how reliable non-invasive measures are. The EEG technique achieves excellent time-resolution but poor space-resolution. It is still debated the possibility to successfully achieve acceptable accuracy in visual classification tasks, and, especially for reconstruction tasks. The literature indicates that fMRI produces outstanding results in this area. In contrast, we have very little information concerning EEG data.

Nishimoto et al. [29] used fMRI: to overcome the limitation of the slowness of signal measured with fMRI [30] they built a new motion-energy [31] encoder. Their results show that dynamic brain activity detected under realistic situations can be decoded using the fMRI technologies now available. More recent studies have used fMRI signals from early visual areas to perform visual neural decoding, using the voxel as the basic unit of measurement, the analog of a pixel on a regular grid in three-dimensional space. Kay et al.'s basic idea presented in [32], together with their own dataset of natural images on which they achieved 90% accuracy, was to model the responses to each individual image as a weighted sum of simple image filters responses, called Gabor Wavelet Pyramids, and then to evaluate the trained model by asking whether it is possible to identify a novel image that a subject is looking at from a set of possible images. Wavelets are a powerful tool for analyzing a signal: they can be used embedded in a neural network [33,34], for subspace decomposition [35,36], and more. Most recent deep learning methods for natural image reconstruction from fMRI [37–39] employed GAN architecture with the assumption that there is a linear relationship between brain activity and the latent features of GANs [40]. A group of studies on reconstructing natural images from brain activity patterns [41,42] incorporated probabilistic inference using VAE-GAN. Specifically, [41] presented a combined network called D-VAE/GAN. Nonetheless, fMRI is too expensive, both in terms of costs and technical requirements.

Mengxi et al. [43] explored EEG data evoked by visual object stimuli to perform visual decoding. Inspired by the recent breakthrough via convolutional neural networks (CNNs) in classifying mental load, several improved CNNs methods were introduced: they proposed a CNN-VAE architecture to perform the classification of visual stimuli. Zhicheng et al. [44] tried to simplify the integrated LSTM-CNNs model by proposing variations of CNNs with feature selection and fusion units.

A different type of dataset used for EEG classification comes from a rapid serial visual presentation (RSVP): [45] presents work based on stimuli from indoor and outdoor images and applies a CNN directly to the EEG data, achieving an Area Under the Curve (AUC) of 72%. Manor et al. [46] build their own RSVP dataset showing a set of five categories every 90–110 ms, achieving an accuracy value that varies between 70% and 81% depending on the subject.

Another less recent version of the MindBigData dataset [47] is used in [48]: they use the spectrogram signal alone with a simple CNN, achieving an accuracy of 91%. However, this dataset differs from ours in that it consists of 'digital' digits, as opposed to MNIST's handwritten digits. Other types of datasets have been created specifically for classification: Spampinato et al. [49] used 40 ImageNet classes and achieved an accuracy of 83%, conditioned by the fact that, unlike ours, features of the original image are also given as input to the classifier. Yang et al. [50] build a dataset by showing images of sinusoidal gratings, showing the difference between considering all subjects to an experiment and only one, achieving 60% accuracy.

The novel contribution of this paper concerning visual NML consists of the following proposed model for EEG data analysis, named Riemann Manifold Spectrogram Network (RieManiSpectraNet), designed for modeling and classifying electroencephalograms. To the best of our knowledge, this is the first study presenting a DL model that integrates spatio-temporal and spectral features from EEG signals to reveal the latent representation of visually evoked brain signals recorded by EEG. By doing so, we have established a benchmark for the CustomCap64-v0.016 MindBigData dataset and are the first, as far as we know, to use it.

In this paper, we focus on visual Neural Manifold Learning (see Fig. 1), proposing a novel model, called *RieManiSpectraNet* (Riemann Manifold Spectrogram Network), constituted by a deep learning architecture for modeling and classifying electroencephalograms that integrates spatio-temporal and spectral features from EEG signals, by revealing the inner latent representation of visually evoked brain signals recorded by EEG. First of all, the system deals separately with spatial information and temporal information by extracting Spatial Covariance Matrices and Feature Maps, respectively. Spatial features are drawn out through Riemann geometry, while Temporal features are brought out through LSTM layers. Both extracted features are then combined to reach a proper embedding to integrate with the spectral features coming from the spectrogram images of each EEG channel computed according to the Short-Time Fourier Transform (STFT) algorithm [51]. In this way, by combining a spatio-temporal- and a spectral-processing stream, we are able to describe the overall activity of the entire brain, delegating the system itself to focus on the correct weighting of features coming from different brain areas.

Table 1
Custom Cap 64 Channels device. Details about the dataset under investigation, including the number of EEG recordings, the number and order of channels used, the number of samples for each channel, the recording interval, and the recording frequency.

Channels N.	64
Samples N. (each)	400 floating point (for each channel)
Samples N. (total)	25,600 floating point (total)
Recording interval	2 s
Recording frequency	200 Hz
EEG recordings	148,736
Channels order	FP1, FPz, FP2, AF3, AFz, AF4, F7, F5, F3, F1, Fz, F2, F4, F6, F8, FT7, FC5, FC3, FC1, FCz, FC2, FC4, FC6, FT8, T7, C5, C3, C1, Cz, CCPz, C2, C4, C6, T8, TP7, CP5, CP3, CP1, CPz, CP2, CP4, CP6, TP8, P7, P5, P3, P1, Pz, P2, P4, P6, P8, PO7, PO5, PO3, POz, PO4, PO6, PO8, CB1, O1, Oz, O2, CB2.

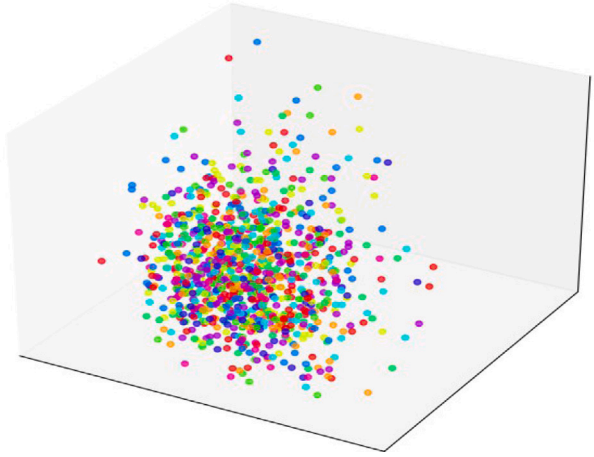


Fig. 1. RieManiSpectraNet Neural Manifold. A 3D projection of the neural manifold created by the suggested architecture is presented. Each individual point corresponds to a recording's representation within the feature space, obtained from the CustomCap64-v.0.016-MindBigData dataset. The 11-classes identified are represented by color coding, comprising "black-screen", "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9". The RieManiSpectraNet Neural Manifold and the Neural Manifold derived from its sub-variant SpeNet architecture facilitate the dataset's linear separability in the feature space. This enables us to successfully carry out the Neural Image Classification task, achieving an accuracy of 55% for the RieManiSpectraNet Neural Manifold and 78% for the SpeNet architecture. RieManiSpectraNet+, the improved version of the complete proposed architecture, increases accuracy to 86%.

Despite the limitations imposed by our specific input signals and the technique used to collect them, our proposed work is the first one on the CustomCap64-v0.016 MindBigData Visual MNIST dataset [47] to achieve comparable outcomes to state-of-the-art results on previous EEG dataset benchmarks.

The rest of the paper is organized as follows. In Section 2 we describe the material and methods of this study, presenting the dataset in which we evaluated our proposed framework, providing data acquisition information and data analysis. Section 2.2 introduces our proposed architecture, providing a detailed description of all the experiments performed. Section 3 provides in detail the theoretical background on Neural Manifold Learning. In Section 4 we present our results, followed by further discussion in Section 5. Finally, in Section 6 we draw the conclusions of the work and highlight some possible future research directions.

2. Material and methods

Leveraging brain signals, data collection plays a crucial role. If the input data is poor or incomplete, the output of the model will also be poor or incomplete. This is known as the "garbage in, garbage out" principle [52], which is particularly important in deep learning, where models require large amounts of high-quality data in order to learn and generalize effectively.

Being EEG data gathering beyond our scope, we had to rely on datasets already presented in the literature. The dataset was chosen to cover a number of different brain regions, collecting raw signals acquired from several EEG channels. We used the Visual MNIST of Brain Digits [47] (from MindBigData). All recordings would eventually be arranged in tuples (signal, image), associating each visual stimulus to the evoked brain signal. In the following, we report the necessary parameters for our deep learning framework.

2.1. MindBigData2022_VisMNIST_Cap64 Dataset

CustomCap64-v0.016 is the last version of the MindBigData Visual MNIST database, updated on 12/27/2022, as reported in [53].

This dataset comes from the home-built Custom Cap 64 Channels device (all parameters are listed also in Table 1), where a subset of the well-known Yann LeCun MNIST digits dataset [54] is displayed while the EEG signals are being captured. It displays 59,699,200 EEG data points, for a total of 2,332 frames. This gives us 25,600 floating points per frame. Since we are working with 64 channels, we have 400 floating points for each channel, representing a single recording of 2 s at a sampling rate of 200 Hz. The 64 EEG channels used are the following: FP1, FPz, FP2, AF3, AFz, AF4, F7, F5, F3, F1, Fz, F2, F4, F6, F8, FT7, FC5, FC3, FC1, FCz, FC2, FC4, FC6, FT8, T7, C5, C3, C1, Cz, CCPz, C2, C4, C6, T8, TP7, CP5, CP3, CP1, CPz, CP2, CP4, CP6, TP8, P7, P5, P3, P1, Pz, P2, P4, P6, P8, PO7, PO5, PO3, POz, PO4, PO6, PO8, CB1, O1, Oz, O2 and CB2. According to the 10–20 system [Fig. 2(a)], the A1 earlobe clip channel was used as a reference for the left hemisphere and the A2 earlobe clip channel was used as a reference for the right hemisphere; the center channels CCPz, CPz, Pz, POz, and Oz were referenced to A1 and FPz, AFz, Fz, FCz, Cz to A2. While evaluating the performance of our proposed system, we also investigated the relationships that might exist between our input data and the behavior of the system. However, the results of our analysis did not reveal any strong evidence of any dominance that could be attributed to any particular channel, asserting our idea that visual recognition is more a high-cognitive task than simply a visual task. Specifically, we ran two trials with a reduced number of electrodes: the first considering only the occipital and parietal electrodes (covering the visual cortex), and the second considering selected occipital, parietal, and frontal electrodes. In both cases, we did not find any electrode to be redundant for the task.

Since the visual stimuli were presented by alternating 28 × 28 MNIST digits with a black screen while always recording from the brain [Fig. 2(c)], the VisMNIST_Cap64 dataset consists of 1,166 black screen images and 1,166 digit events, observing a fairly balanced number of samples for each digit from 0 to 9 [Fig. 2(b)]. For each image, it is possible to reconstruct the corresponding EEG recording consisting of 64 segments, one from each EEG channel [Fig. 2(d)].

Moreover, for each image, it is possible to retrieve the spectral content of each channel [Fig. 2(e)]. This is the result of applying a 50 Hz notch filter and a band-pass Butterworth filter between 14 Hz and 71 Hz, including the Beta (15–31 Hz) and Gamma (32–70 Hz) frequencies.

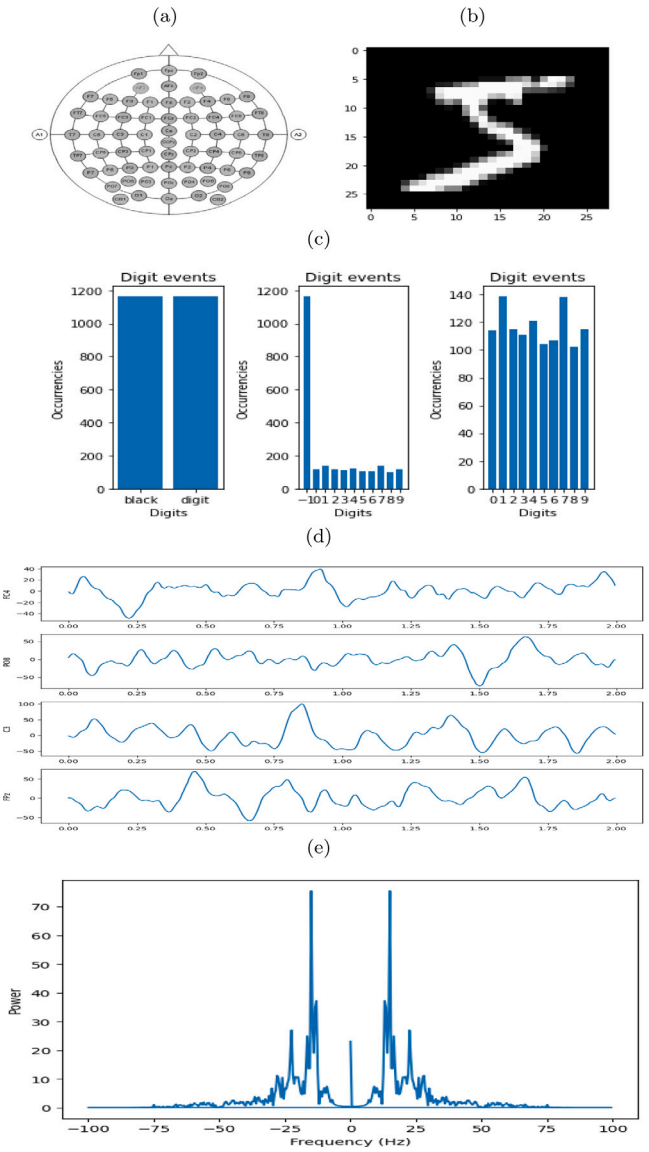


Fig. 2. MindBigData CustomCap64-v0.016 Dataset Analysis. (a) 10–20 International system. (b) Example of a MNIST digit. (c) Digit events occurrences. (d) Example of four EEG channels recording. (d) Example of one channel spectral content.

2.2. RieManiSpectraNet architecture

In this section, a more detailed description is reported for the developed Deep Learning Architecture, called *RieManiSpectraNet*, implemented to learn a Neural Manifold from our EEG Data. This architecture, shown in Fig. 3, consists of two processing streams, the spatio-temporal stream and the spectral stream. The former consists of three subsequent blocks, where the Riemann Manifold technique together with LSTMs and VAEs is used to reach an inner low-dimensional representation for our EEG signals taking into account spatio-temporal features. The latter is based on the STFT Algorithm, applied to our EEG signals to extract spectral features. On top of that, there is a CNN classifier, aiming to classify visual stimulus-evoked EEG signals. The architecture relies on three distinct turning points: (i) the locality principle, fundamental to CNNs; (ii) the correlations between pairs of electrodes which is imperative in recognizing highly cognitive tasks undertaken by subjects; and (iii) the soft attention mechanism, the initial step to realize an architecture that mirrors the structure of LLMs.

2.2.1. Spatial–temporal stream

The first stream of our proposed architecture consists of a pre-processing pipeline, which is shown in Fig. 5, that from raw EEG data would lead us to a proper features-embedding, integrating spatio-temporal EEG representation. This system is inspired by the framework presented in [55], allowing us to propose a novel architecture for EEG neural manifold learning.

Starting from EEG recordings, a stream involving all EEG channels is processed by learning its own spatio-temporal representation through a spatial/temporal extractor, a spatial/temporal processor, a spatial/temporal Variational Auto-Encoders and a final fusion block aiming to obtain a suitable embedding from the latent spatial and temporal representations of that stream. Instead of analyzing a stream consisting of just those channels covering the occipital lobe, being the visual processing hub of our brain, we decided to involve information from all EEG channels, allocating image recognition as a high-level cognitive task.

In the following, we referred to all input and output signals as PyTorch [56] tensors, to keep consistent with our implementation.

Data pre-processing. Given a single EEG recording, a tensor of size $[N, T]$, where N is the number of channels and T is the number of samples for each segment, we applied a 5th order Butterworth filter bank as a bandpass filter between 14–71 Hz, splitting the signal into H frequency sub-bands. This allowed us to treat the eye-blink artifacts contaminating the low-frequency EEG bands (0–12 Hz) [57]. Then, a notch filter at 50 Hz was applied to reduce power line noise. Eventually, signal amplitudes were re-scaled to the range of $[-1, 1]$ through min–max normalization, so that data discrepancy across different recording sessions was decreased. In this way, the module output is a tensor of size $[H, N, T]$.

Temporal feature processing. We distinguish here a Temporal Feature Extraction Module and a proper Temporal Feature Processing Module, shown in Fig. 4, the latter implementing artificial neural network layers to learn latent representations.

Given as input a tensor of size $[H, N, T]$, the extraction module returns a tensor of size $[H, L, N, F]$, actually splitting the time interval T in L windows and extracting F exemplifying features for each channel within each window. Being this tensor the input to the processing module, we eventually obtain a monodimensional tensor of size *hidden_temporal_fc*.

Spatial feature processing. In this section, we distinguish a Spatial Feature Extraction Module and a proper Spatial Feature Processing Module, again with the latter implementing Artificial Neural Networks to learn latent representations.

Given a tensor of size $[H, N, T]$ coming from data pre-processing, the extraction module returns here a tensor of size $[H, P, R(R+1)/2]$, basically splitting the EEG recording into P segments and representing each of them through $R(R+1)/2$ features, coming from the computation of Spatial Covariance Matrices (see Section 3.2), the subsequent application of dimensionality reduction and tangent space learning, ending up with the implementation of a half-vectorization step. Indeed, in order to obtain an appropriate input for the succeeding layer of our architecture, we vectorize our features to form a one-dimensional array. This process ensures that no information is lost. Following the feature extraction, through the processing module, we eventually obtain a monodimensional tensor of size *hidden_spatial_fc*. This general procedure is represented in Fig. 5.

Fusion strategy. The strategy for the fusion of spatial and temporal information, shown in Fig. 6, plays an essential role in dealing with multimodal or multi-learning approaches of one modality in order to perform classification. Attention mechanisms have been successfully implemented for refining fusion weights applied to different modalities.

In our architecture, variational auto-encoders (VAEs) are used to learn embedding-specific features, in place of the fully-connected (FC)

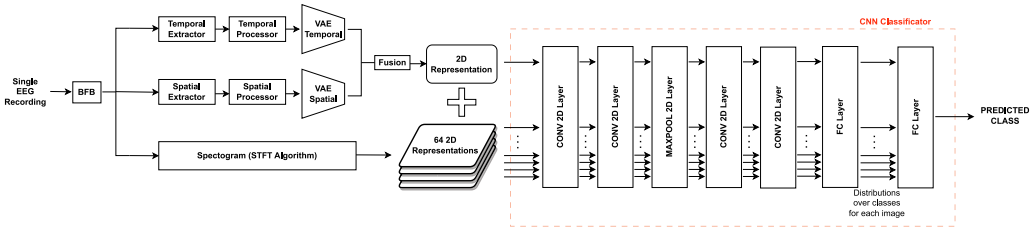


Fig. 3. RieManiSpectraNet Architecture. Block diagram of our proposed architecture. A 5th-order Butterworth filter bank is used as a band-pass filter, along with a Notch filter at 50 Hz. The signal is filtered between frequencies 14–71 Hz, followed by the spatio-temporal processing stream and spectral processing stream. A fusion unit is used to perform multi-learning through the application of attention mechanisms, dealing with both spatial and temporal information. The N+1 features extracted from the two processing streams, where N represents the number of electrodes used, are combined and input into our CNN module. The module is composed of four Conv2D Layers and a FC Layer, generating N+1 distributions over the 11 presented classes. Finally, a second FC Layer produces a single distribution by weighting its inputs, enabling the classification of EEG signals evoked by visual stimuli.

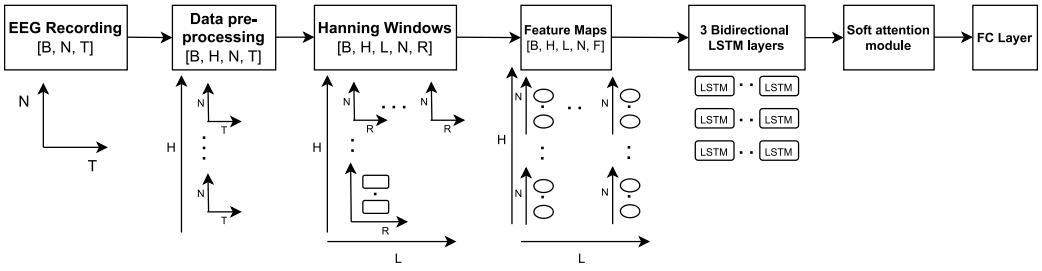


Fig. 4. Temporal Feature (TF) Processing sub-stream. Block diagram showing the TF Extraction module followed by the TF Processing module. Temporal features are extracted from a batch of EEG recordings and then processed through ANN layers to learn latent representations. Workflow inspired by [55].

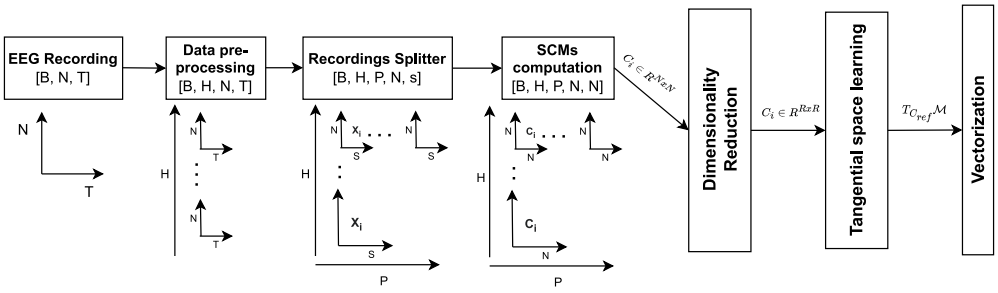


Fig. 5. Spatial Feature (SF) Processing sub-stream. Block diagram showing the SF Extraction module. Spatial Covariance Matrices are extracted from a batch of EEG recordings in order to apply the Riemann metric learning method. Workflow inspired by [55].

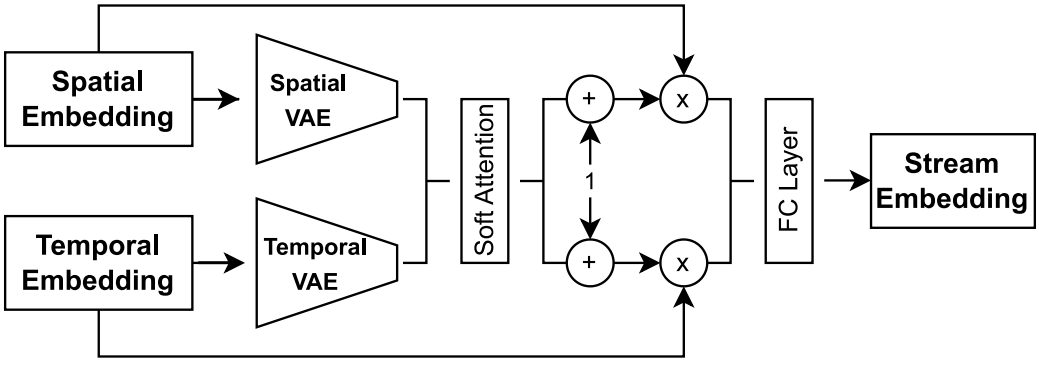


Fig. 6. Spatio-Temporal Fusion Module. Block diagram showing the strategy for combining spatial and temporal features to achieve a single stream embedding. By introducing the VAE architecture, we improve the fusion strategy presented in [55].

layers encoders implemented in [55]. Then, we employ soft attention to learn the weight (α) applied to each embedding-specific feature. Next, we compute the new weighted embedding by multiplying the weight score with the original individual learning embedding. Finally, we perform decision-level fusion on the concatenation of the two new embeddings using an FC layer.

Eventually, the monodimensional output tensor of the stream has size *hidden_fusion_fc*. This concludes the description of the *RieManiSpectraNet*'s first processing stream.

2.2.2. Spectrogram stream

The second processing stream of our proposed architecture involves again all EEG channels, learning their own spectral content to convert them into image-like representations.

We applied a 5th order Butterworth filter bank, band-pass filtering it between 14–71 Hz, with a Notch filter applied at 50 Hz to reduce power line noise. In this way, we aim to extract the spectral content of the recorded EEG signals at the Beta (15–31 Hz) and Gamma (32–70 Hz) frequencies, as they carry information about the psychological cycles involved in visual recognition [58]. To do that, we used the STFT, a widely used technique that provides a 2D representation of a signal by dividing it into short, overlapping segments and calculating the Fourier Transform for each segment. The STFT results in a 2D matrix often referred to as the spectrogram of the signal, where the x-axis represents time, the y-axis represents frequency, and the value at each point represents the magnitude or power of the corresponding frequency component at that time.

STFT algorithm. The STFT has parameters that affect its behavior, such as the size of the Fourier Transform window (*n_fft*), which determines the number of frequencies bins in the resulting STFT, the number of samples between successive STFT columns (*hop_length*), the choice of the window function to be applied (*window*) and its size.

Given a single EEG recording as a tensor of size $[N, T]$, for each of the N channels we applied the PyTorch built-in version of the STFT algorithm, selecting such parameters accordingly with the required time–frequency resolution. Specifically, we set *n_fft*=256, *h_l*=64, choosing the Hanning window with size *w_l*=256. Since the STFT is normalized by the sum of the window function, for each channel, we reached a $[129, 7]$ 2D tensor representation, that we reshaped into an RGB image-like tensor of size $[43, 7, 3]$.

2.2.3. CNN classifier

In order to perform the classification of visual stimulus-evoked EEG signals, both outputs coming from the spatio-temporal and spectral processing streams serve as input to our CNN classifier.

Here, the classifier architecture consists of four Conv2D Layers, with a MaxPool2D Layer between the first and the last two of them, followed by two FC Layers that lead to a single prediction for the initial input to the entire architecture. All four Conv2D Layers use ReLU as a non-linear activation function, whereas the two FC Layers are interleaved by Softmax as non-linearity in order to deal with distributions over classes. Further details are specified in Table 2, as well as represented in Fig. 3.

Fusion spatial–temporal–spectral. To combine the features coming from both processing streams, the output had to be reshaped of the spatio-temporal stream into an RGB image-like tensor of size $[43, 7, 3]$. In this way, since the new representation is compliant with the principle of locality, it is possible to process these features with the proposed CNN classifier.

Given a single EEG recording as a tensor of size $[N, T]$, according to the workflow of our architecture, the CNN classifier would receive N images coming from the spectral processing stream, plus one more image coming from the spatio-temporal stream. Therefore the overall input can be represented as a four-dimensional tensor of size $[N + 1, 43, 7, 3]$.

Table 2
Summary of all the layers of our RieManiSpectraNet architecture.

Type	Num Params
ButterworthFilterBank	0
Spectrogram_Module	0
TemporalFeatureProcessing_Module	22.6 M
SpatialFeatureProcessing_Module	205 K
VAE_spatial	624 K
VAE_temporal	624 K
FeatureFusion_Module	529 K
CNN_classifier	2.8 M
Total parameters	27.3 M

Table 3
Convolution parameter size. The kernel size, stride, and padding are set to five, one, and two, respectively. Between each convolution is a ReLu activation function. The kernel size of the MaxPool is four.

Layer	In features	Out features
Conv1	3	256
Conv2	256	256
MaxPool	–	–
Conv3	256	128
Conv4	128	64

To achieve a single prediction for the initial input, our CNN classifier's strategy is to process every single image through the four Conv2D Layers plus the first FC Layer, achieving a distribution over the 11 presented classes. In this way, the last FC Layer of our architecture would receive as input $N+1$ different distributions over classes, leading to a final single distribution coming from the weighted sum of its inputs (see Tables 3 and 4).

2.3. Experiments

The proposed *RieManiSpectraNet* architecture belongs to the family of Hierarchical Architectures [59], a class of Deep Learning models structured in a layered manner, where each layer represents a different level of abstraction. In this kind of architecture, higher-level features are built upon lower-level ones, allowing our model to learn complex patterns and relationships.

The training process in Hierarchical Architectures often involves a combination of supervised and unsupervised learning. Specifically, our training strategy implies distinguishing among three layers, each one with its own loss function and optimizer.

On the one hand, we have the Temporal Feature Processing layer and the Spatial Feature Processing layer, both involving their own VAE to train in an unsupervised fashion. During training, the VAE optimizes a loss function that balances two objectives: the reconstruction loss, which measures the difference between the generated sample and the original input, and the Kullback–Leibler (KL) divergence [60], which measures the difference between the distribution of the learned representation (i.e., the posterior) and the prior distribution. This divergence is used as a regularization term in the VAE loss, ensuring that the posterior distribution allows for the generation of meaningful and useful data points.

On the other hand, the Fusion Strategy Layer (shown in Fig. 6) and the subsequent CNN classifier are trained in a supervised fashion, trying to minimize the Cross-Entropy Loss as a loss function.

After conducting several experiments, using a V100 GPU with 32 GB of Virtual RAM (VRAM), we have decided to fix the parameter values of the two layers trained in an unsupervised fashion as soon as the threshold of 0.15 on their training losses is reached. In this way, we allow the last layer of our architecture to learn complex representations from a fixed latent space, whose final version has been optimized during the initial phase of the training.

Table 4
Hyperparameters from RieManiSpectraNet.

Parameter	Value
BFB Hyperparameters	
Order	5
Frequency Ranges	[14, 71]
Notch Frequency	50 Hz
Notch Q-factor	0.01
Sampling Frequency	200 Hz
Temporal Module	
Sampling Frequency	200 Hz
Window Width	1 s
Input Size	$1 \times 64 \times 102$
L (LSTM Layers)	3
Hidden Size (LSTM)	256
Hidden Size (Fully Connected)	256
Dropout Rate	0.5
Slope	0.3
Spatial Module	
R (Radius)	100
S (Stride)	8
Input Size	$1 \times 4 \times 36$
Hidden Size (Fully Connected 1)	512
Hidden Size (Fully Connected 2)	256
Dropout Rate	0.5
Spatial VAE	
Input Dimension	256
Hidden Dimension	512
Latent Dimension	64
Number of Layers	3
Temporal VAE	
Input Dimension	256
Hidden Dimension	512
Latent Dimension	64
Number of Layers	3
Fusion Module	
Input Size	128
Hidden Size (Fully Connected)	129 × 7
Dropout Rate	0.5

We performed the three final experiments on a single NVIDIA RTX 3090 with 10496 CUDA cores and 24 GB of VRAM, aiming to highlight the contribution of our proposed architecture in combining spatial–temporal–spectral features. Specifically, we decomposed our architecture, dealing in the first place with the two streams (see Sections 2.2.1 and 2.2.2) independently to then compare the results achieved with the ones obtained by our full proposed architecture. The batch size is fixed to 16, while additional details about the specific losses and optimizers used, with their hyperparameter values, are specified in Table 5 for all three case studies. We split our dataset into three separated subsets containing 70%, 20% and 10% of the data respectively for training, validation, and testing. The number of epochs has been set to 500, but in order to avoid overfitting, the system has been provided with an early-stopping callback in order to end the training loop if the validation loss of the classifier was not improving for 50 epochs consecutively.

2.4. Evaluation metrics

Several studies in neural manifold analysis of brain circuit dynamics [27] apply internal measures to evaluate the lower-dimensional space reached, demonstrating the applicability of these methods to the study of neurological disorders. In evaluating the manifold realized by our proposed architecture, we opted for external measures while implementing a Neural Image Classification task with it.

To be consistent with studies in the literature, we assess the performance of our architecture by computing the average accuracy, which measures the overall correctness of the model's predictions across all classes in a multi-class classification problem. In order to offer a more detailed view of the model's performance, we also provide the confusion matrix, a table displaying the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions for each class.

Table 5
Training process information for all three case studies, including the complete RieManiSpectraNet architecture and its SpeNet and STNet variants. For each Deep Learning layer, we specified the loss function to minimize, and the adopted optimizer with the associated learning rate (lr) and weight decay (wd) hyperparameters. Note that we work with the same multiple optimizers simultaneously for both the RieManiSpectraNet and the STNet architectures, due to the independent optimization of the parameters of the temporal sub-stream, the spatial sub-stream, and the parameters of the fusion unit plus the CNN-classifier. As the SpeNet architecture does not have a spatio-temporal sub-stream, we only use one optimizer for the CNN-classifier parameters.

Architecture	Layers	Loss & Optimizer
RieManiSpectraNet	Temporal_feature+ VAE_temporal	MSE + Adam $lr = 10^{-5}$
	Spatial_feature+ VAE_spatial	MSE + Adam $lr = 10^{-3}$
	Feature_Fusion+ CNN_classifier	CrossEntropy + SGD $lr = 6 \cdot 10^{-2}$ + $wd = 10^{-6}$
SpeNet	CNN_classifier	CrossEntropy+RMSProp $lr = 10^{-4}$ + $wd = 10^{-6}$
STNet	Temporal_feature+ VAE_temporal	MSE + Adam $lr = 10^{-5}$
	Spatial_feature+ VAE_spatial	MSE + Adam $lr = 10^{-3}$
	Feature_Fusion+ CNN_classifier	CrossEntropy + SGD $lr = 6 \cdot 10^{-2}$+$wd = 10^{-6}$

3. Neural manifold learning

Underpinning the term Neural Manifold Learning is an ensemble of different algorithms designed to embed in low-dimensional matrices $\boldsymbol{Y}$ the information that represents higher-dimensional activity matrices $\boldsymbol{X}$. Such high-dimensional matrices are typically used to describe the activity of a set of N neurons at a given time interval of T samples. By performing a topological projection into a lower dimensional space, it is possible to observe the emergence of topological manifolds Y, originating from a few constrained topological structures. Such manifolds represent specific patterns of neural activity, which can be recognized and classified because they retain a local linear geometry despite the globally curved topology of the space.

Several different systems have taken advantage of manifold learning [61]. Such systems can be generally classified into two distinct groups: linear methods and nonlinear methods. The choice between such methods depends on the feature extraction process, in order to understand which strategy best suits the type of neural information under investigation, as this may fundamentally affect the subsequent interpretation.

To achieve Linear manifold learning, the solution requires the execution of a set of linear transformations that are properly designed in order to verify a set of properties that guarantees the optimality conditions and, in particular, to achieve a sufficiently low-dimensional embedding that preserves the possibility of giving an interpretation of the neurological and physiological phenomena. Principal component analysis (PCA) and Multi-layer scaling (MDS) have been commonly used for manifold learning: the former aims to reduce large datasets with high-dimensional recordings to a lower-dimensional representation of the information without altering the statistical variability of the dataset, while the latter attempts to determine a low-dimensional map without altering the pairwise distance between the data points in the original space. Linear techniques such as PCA and MDS can only describe linear relationships between electrodes, which are too weak to describe high cognitive functions since it is known that more than just a linear combination of individual neurons is involved. This shifts our

attention to non-linear techniques, which are able to reveal a broader non-linear approximation of the activity of the neural population.

Non-linear manifold learning is used to determine whether the action matrix manifests emergent non-linear structures. Non-linear manifold learning often embeds in the analysis an approximated calculus of the manifold $Upsilon$ from the reduced space $\boldsymbol{Y}$ in order to identify population-wide variables capable of describing the local relationship between points (representing neural states), while neglecting any effect that would potentially take into account distant points, thus preserving a locality principle in its application. The most widely used solutions of this kind are, namely, Locally Linear embedding (LLE), Laplacian eigenmaps (LEM), t-distributed stochastic neighbor embedding (t-SNE) [62] and Uniform manifold approximation and projection (UMAP).

3.1. ML algorithms

The state-of-the-art of Machine Learning Algorithms in Neural Manifold Learning is reached through t-SNE and UMAP. Both aim to match local distances in the high-dimensional space $\boldsymbol{X}$ to the low-dimensional embedding $\boldsymbol{Y}$.

For instance, through t-SNE, this is obtained by first constructing a probability distribution over pairs of high-dimensional points $\boldsymbol{x}_i, \boldsymbol{x}_j$ in such a way that nearby points are assigned a higher probability while dissimilar points are assigned a lower probability. Then t-SNE defines a similar probability distribution over the points $\boldsymbol{y}_i, \boldsymbol{y}_j$ in the low-dimensional space, and it minimizes the Kullback–Leibler divergence between the two distributions [63].

Artificial Neural Networks (ANNs) can also be employed for manifold learning as they have the potential to extract complex non-linear structures in high-dimensional data. Autoencoders exemplify this approach as they are designed to find an optimal encoding between a high-dimensional input and a low-dimensional representation stored in their "bottleneck" code layer, which preserves the information necessary to then reconstruct the original input from it. Within a more technical level, Variational Auto-Encoders (VAE) [64] promises great potential at extracting low-dimensional structure in varied high-dimensional data, by constructing a stochastic model of the low-dimensional data, and by constructing a stochastic model of the low-dimensional dynamics underlying the neural activity.

The basis of our architecture is discussed in the following, including the reported solution to deal with spatio-temporal and spectral information. The first applied technique (Riemann Manifold) is inspired by Zhang and Etemad's Riemann approach [55], outperforming UMAP on several datasets, as it enhances separability in feature space. The second technique (Image-like EEG Manifold) provides an intuitive and visually understandable representation of complex EEG data, making it easier to identify and interpret patterns or anomalies.

3.2. Riemann manifold

Many approaches use Spatial Covariance Matrices (SCMs) to extract spatial information from raw multi-channel EEG, applying then Euclidian metric learning, such as the average of SCMs and the distance between two SCMs. These matrices are of size $R \times R$ and allow us to observe the level of correlation between the signals and noise in the environment as received by each pair of sensors. As the covariance matrix is symmetrical, it is only necessary to consider the unique pairs. Therefore, to represent it, we require only R(R+1)/2 values. On the other hand, Euclidean metric learning suffers from two problems: the linear mixing effect of EEG due to volume conduction, and the inaccuracy of the Euclidian mean computation of SCMs, since SCMs are Symmetric Positive Definite (SPD) matrices and the determinant of this kind of matrices can be strictly larger than the determinant of any of the SPD matrices used to compute the mean itself (also known as swelling effect [65]).

To overcome these issues, Riemann metric learning method has been applied to SCMs enhancing the EEG classification performance [55]. Riemann distance between any two SCMs, unlike the Euclidian distance, is affine-invariant. If any linear transformation is applied to EEG signals, this affine-invariance property will allow the distance between the two SCMs of EEG signals to remain unchanged. As a result, the linear mixing effect of EEG will be minimized and no swelling effect would exist during the estimation of the Riemann mean.

So, the assumption is: SCMs of raw EEG are SPD matrices on Riemann manifold [55]. As a consequence, Riemann geometry is employed to better learn and manipulate the SPD matrices, in order to capture spatial information. To fully understand the implemented architecture, basic concepts of Riemann geometry are introduced in the following Appendix A.

3.3. Image-like EEG manifold

The time–frequency resolution of EEG data can achieve two-dimensional EEG representation. Signals can be converted to a spectrogram image using STFT [66] (further details on this process are given in Section 2.2.2). The image-like representation of EEG data as a neural manifold allows us to apply various analysis and machine learning techniques for interpretation and analysis. For instance, CNNs can be employed to extract features and patterns from image-like EEG data. Additionally, clustering algorithms and dimensionality reduction techniques can be used respectively to identify distinct groups and reduce the dimensionality of the data while preserving essential information.

4. Results

Conducting a baseline to validate the results of our proposed architecture, we performed the neural image classification task through the STNet (Spatio-Temporal Network) model, including only spatial and temporal features. The accuracy levels obtained do not deviate from the statistics of the problem, remaining constant at the 14% level (see Appendix C for further details on the ablation studies). Reasons for such an expected result, such as the low spatial resolution of the EEG, are discussed in detail in Section 5. This outlines the significance of our results, which are discussed below.

Fig. 8 shows the three losses (VAE temporal, VAE spatial, and training loss) for each model used: specifically Fig. 8(a) shows the losses of the *RieManiSpectraNet* model while Fig. 8(b) shows the losses of the SpeNet (Spectrogram Network).

In Fig. 8(a), we can see how the training loss starts to decrease soon after the two VAE losses reach the threshold of 0.15 (see Section 2.3), when the model enters the second training phase where the VAE weights are frozen. Interestingly, a few steps after reaching the relative optimum value of loss and accuracy (shown in Figs. 8(a) and 9(a)), the loss value increases again until the early stopping function interrupts the training process. The *RieManiSpectraNet* accuracy value on the test set is **55%**. From its confusion matrix (Fig. 10(a)), it can be seen that the architecture performs poorly on several images depicting the digits 1, 7, and 9, while performing remarkably for the remaining digits. Since each class is a handwritten digit, it is reasonable that an architecture that integrates spatial features of brain signals would be confused by similar digits (see Fig. 11). In an attempt to enhance the integration of spatio-temporal features acquired from brain signals with spectral features, an attention layer was added before concatenating the outcomes of the two processing streams depicted in Fig. 3 in order to modulate the attention that the system needs to pay to each stream. This revised architecture, called RieManiSpectraNet+, attained an accuracy value of **86%** on the test set. Refer to Appendix C for a comprehensive analysis of the performance of the RieManiSpectraNet+.

On the other hand, Fig. 8(b) shows the training and validation losses for the SpeNet architecture. It can be seen that the training loss achieves values close to 0, yielding a training accuracy (Fig. 9(b))

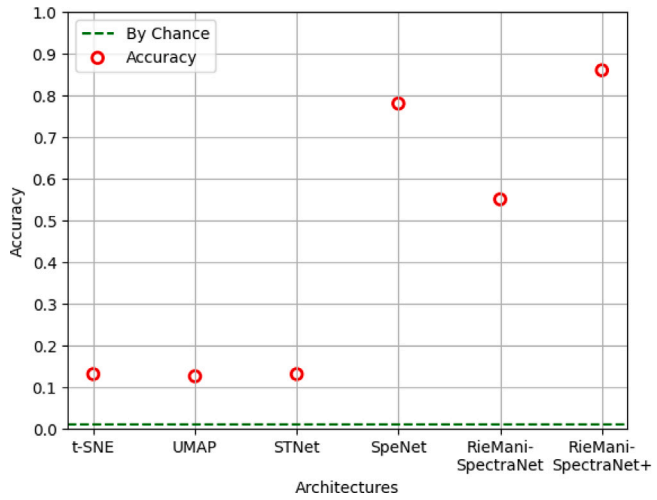


Fig. 7. Ablation Studies accuracies. The accuracy levels of all tested architectures are presented and compared against the established 'by chance' level, as indicated by the green dashed line. As shown in Fig. 2(c), the calculation of the 'by chance' level takes into account the unbalanced dataset. This reduces the 'by chance' level from 9% (1/11) to approximately 1%. Our findings reveal that the t-SNE (13%) and UMAP (12%) accuracies resemble the reference level (1%). Meanwhile, the spatio-temporal (STNet, with 14%) and spectral (SpeNet, with 78%) individual processing streams impart an increase in accuracy. Additionally, RieManiSpectraNet (55%) and RieManiSpectraNet+ (86%) exhibit the accuracy levels reached through the integration of both processing streams, resulting in a successful outcome for the latter enhanced architecture.

Table 6
Comparison with previous studies, with the methodology used and the accuracy associated with it.

Research	Methodology	Acc. (%)
Yang et al. [50]	Personal datasets Different spatial frequency Sinusoidal images	60%
Spampinato et al. [49]	40 Image-Net classes Use features from the original images	83%
Kumari et al. [48]	"Real Digital" digits Only spectrogram	91%
Our Approach	**Handwritten MNIST** **Only spectrogram**	**78%**
Our Approach	**Handwritten MNIST** **RieManiSpectraNet** **Temporal + Spatial + Spectrogram**	**55%**
Our Approach	**Handwritten MNIST** **RieManiSpectraNet+** **Temporal + Spatial + Spectrogram**	**86%**

of approximately 99%, with a corresponding validation accuracy of around 70%. For the SpeNet model, the achieved test accuracy is **78%**. Its confusion matrix is shown in Fig. 10(b).

Table 6 shows a comparison with previous studies, as well as the approach and accuracy involved. Fig. 7 displays the accuracy values achieved by the implemented architectures in our ablation studies (discussed in Appendix C) compared to the baseline random accuracy. These results reasonably support our thesis that the integration of spatial, temporal, and spectral features is a successful strategy for decoding brain signals. In the following, we will discuss our results in comparison with the others in literature.

5. Discussion

In the light of what has already been presented in Section 1, Kumari et al. presents a work which, at first glance, may seem very similar to ours. First of all, they use a less recent version of the MindBigData [47] dataset (2015), completely different from our version. The main difference is that the images shown are "real digital" digits, not those from the handwritten MNIST dataset, and the subjects are not shown the black screen between the images. Please refer to Appendix B for a deeper explanation. Moreover, our dataset had only 148,736 brain signals, while their work uses 1,207,293 brain signals.

Spampinato et al. [49] use EEG data from observing 40 Image-Net classes, using a 128-channel EEG cap. However, their results are also achieved by using a CNN-based approach to extract features directly from the images since they map such images to the corresponding EEG feature vectors. In this case, using original images as prior knowledge in input to the model helps to achieve high accuracy.

On the other hand, our approach is based on only 148,736 brain signals with a 64-channel EEG cap: having so few signals makes it easier to train the classifier, but much harder to obtain acceptable results. Despite these limitations, the proposed architecture was able to achieve satisfying results with an accuracy comparable to the others.

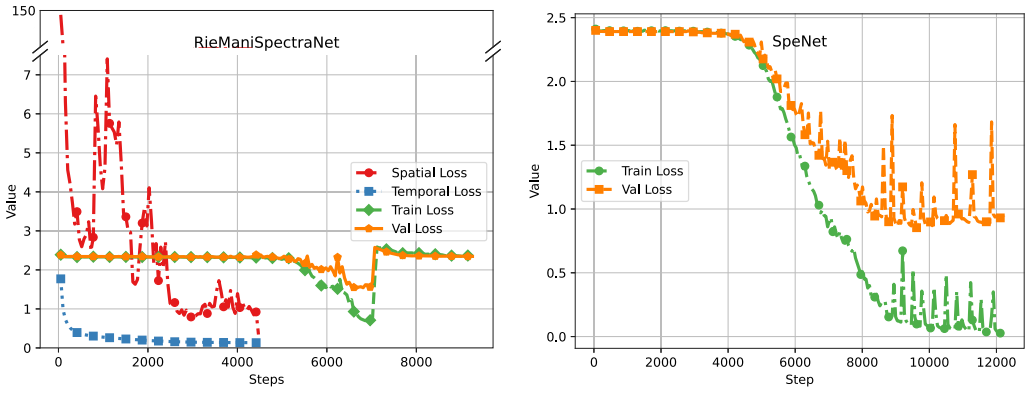
The possibility to combine spatial–temporal–spectral features derived from EEG recordings is of great importance. By integrating information about the spatial distribution, temporal dynamics, and spectral characteristics of brain signals, our approach aims to help us to gain a comprehensive understanding of neural activity and its underlying processes. Our proposed model for decoding purposes seems to be underperforming when combining the spatial–temporal–spectral features (RieManiSpectraNet) with respect to using only spectral features (SpeNet). However, this effect is connected to the intrinsic difficulty of extracting stronger relationships from spatio-temporal dynamics in a small-scale machine learning experiment. This can be seen in Fig. 10(a), which shows the confusion matrix of RiemaniSpectraNet: the MNIST digits 1, 7, and 9 (shown in Fig. 11) are very similar to each other, and an architecture that incorporates spatial features of the brain signals would have a hard time distinguishing similar digits, resulting in poor performance. In addition, we improved the integration of spatial, temporal, and spectral features in our RieManiSpectraNet+ C by incorporating an attention layer prior to concatenating the two processing streams, resulting in a more segregated final neural manifold.

On the other hand, the proposed approach is unique due to its ability to effectively capture local relationships effectively. Moreover, EEG recordings have inherent limitations in spatial resolution, which further impact the effectiveness of combining spatial–temporal–spectral information, and have been shown to have individual specificity [46], leading to variable effectiveness in extracting features. This is an initial, very promising attempt to create a technique that could be crucial not only for patients with verbal difficulties but also for individuals in their daily lives, using non-invasive and handy techniques to collect EEG data. To ensure its broad applicability, it is crucial to incorporate an Explainable AI module into our architecture. This will help overcome the individual idiosyncrasies associated with EEG signals and enable us to conduct larger-scale experiments with more extensive datasets featuring brain signals from multiple subjects. Future research directions should explore complementary DL modules, such as transformers or Variational Auto-Encoders (VAE), to efficiently extract features from the spatio-temporal stream. These advanced modules may offer enhanced capabilities in capturing complex relationships within EEG data and improving the performance of decoding tasks.

6. Conclusions

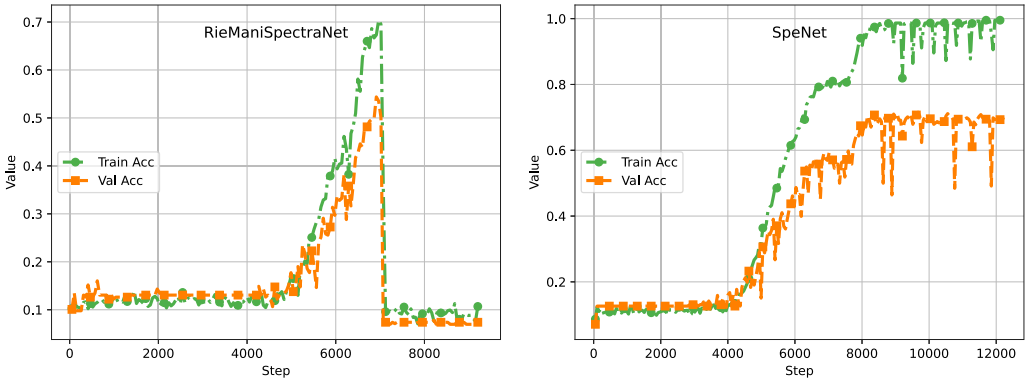
Anchored in this work, the *RieManiSpectraNet* model has been presented, a deep learning architecture for learning and classifying electroencephalograms that integrates spatial–temporal–spectral features from EEG signals. Our findings align with the initial research objectives and shed light on Neural Manifold Learning of visually evoked brain signals recorded by EEG.

The significance of our study lies in its contribution to the field of Visual Systems Neuroscience. By proposing our deep learning approach to combine information about the spatial distribution, temporal



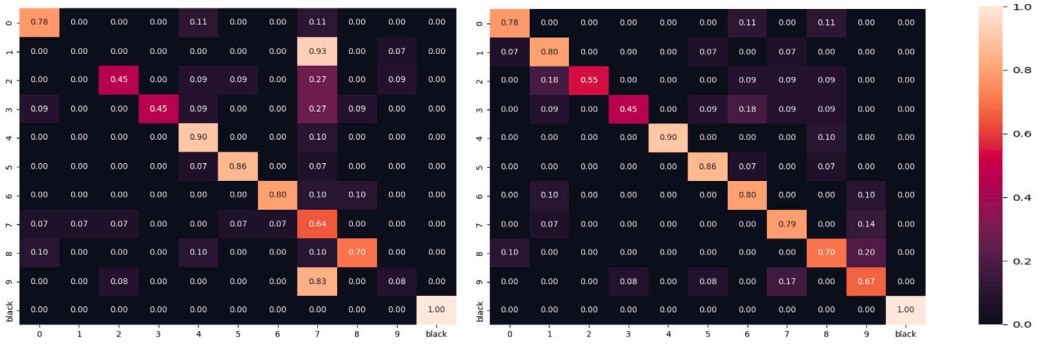
(a) RieManiSpectraNet Losses (b) SpeNet Losses

Fig. 8. Plots of the Losses profiles during the training process. Case study (a) shows the loss functions optimized by VAEs in the spatial (Spatial Loss) and temporal (Temporal Loss) layers, in addition to the training (Train Loss) and validation (Val Loss) losses associated with the final classifier for the RieManiSpectraNet. Note that in the upper part of the case study (a) there is the initial value of the spatial loss plot (~130) and then we cut the plot until smaller values to facilitate the overall view. Case study (b) shows the training (Train Loss) and validation (Val Loss) losses associated with the final classifier in the SpeNet.



(a) RieManiSpectraNet Accuracies (b) SpeNet Accuracies

Fig. 9. Plots of the average accuracy profiles during the training process. Both case studies (a) and (b) show the accuracy calculated during training on the training set (Train Acc) and the validation set (Val Acc). It can be seen, as in Fig. 8, that the performance of RieManiSpectraNet collapses quickly after reaching the optimum value, unlike the performance of SpeNet. This is taken as an index of the difficulty of training the proposed RieManiSpectraNet architecture.



(a) RieManiSpectraNet Confusion Matrix (b) SpeNet Confusion Matrix

Fig. 10. Confusion Matrices on Test Set. The decoding performance for each class is calculated for both case studies (a) and (b). It can be seen that both architectures perform equally well in decoding most of the classes, with the exception of classes '1', '7', and '9', where RieManiSpectraNet is completely fooled.

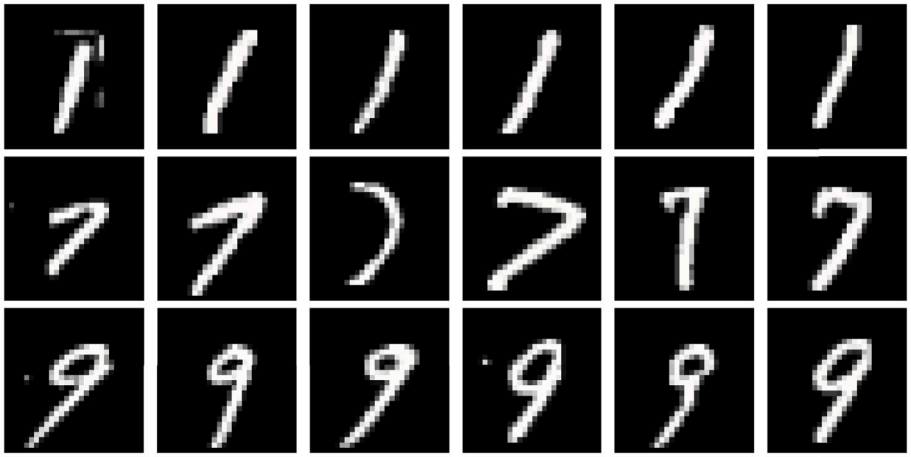


Fig. 11. Examples of the three interested MNIST classes, digits 1, 7 and 9. The first row (top) shows examples of the digit 1, the second row (middle) displays examples of the digit 7, and the third row (bottom) presents examples of the digit 9. It is understandable, given that each class represents a handwritten digit, that an architecture incorporating spatial features of the brain signals would struggle to distinguish similar digits, resulting in poor performance specifically on images depicting the digits 1, 7 and 9. However, the architecture excels at recognizing the remaining digits.

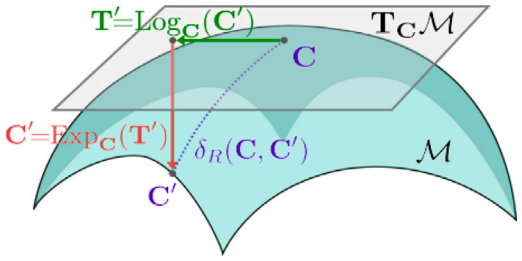


Fig. A.12. Basics of Riemann Geometry.
Source: Courtesy of [55].

dynamics, and spectral characteristics of brain signals, we have provided new insights into how Deep Learning systems could enhance Neural Decoding. Furthermore, our study has practical implications for BCI and neuroengineering research, enabling a new level of communication through the brain. The RieManiSpectraNet architecture aims to be a tool for dimensionality reduction of neural population activity, improving the separability of features in the analysis feature space. This enhanced separability of features implies that the reduced dimensional representations may better capture the essential information encoded by neural populations, providing valuable insights into the neural mechanisms underlying brain functioning and visual perception. Further directions of this study relate to the neural correlates of specific visual perceptual features or categories. By mapping the reduced-dimensional representations back to the original neural population activity, it is possible to investigate how different visual features are represented and processed in the brain. This can shed light on the underlying neural mechanisms involved in visual perception, contribute to our understanding of how the brain constructs visual representations, and reveal hierarchical organization within the brain.

Future research should explore variations on the proposed architecture, possibly based on attention-based mechanisms. Moving forward, several promising directions for future research emerge from our findings, e.g. in order to explore Neural Visual Decoding from non-invasive fMRI imaging techniques, working with natural scenes and natural images, and exploiting deep generative methods to perform Neural Image Reconstruction.

CRediT authorship contribution statement

Salvatore Falciglia: Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Filippo Betello:** Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Samuele Russo:** Writing – review & editing, Visualization, Validation, Resources, Investigation, Formal analysis, Conceptualization. **Christian Napoli:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data are openly available since this is a public dataset.

Acknowledgments

This work was developed at the is.Lab() Intelligent Systems Laboratory of the Department of Computer, Control and Management Engineering, Sapienza University of Rome, and partially supported by the Age-It: Ageing Well in an Ageing Society project, task 9.4.1 work package 4 spoke 9, within topic 8 extended partnership 8, under the National Recovery and Resilience Plan (PNRR), Mission 4 Component 2 Investment 1.3- Call for tender No. 1557 of 11/10/2022 of the Italian Ministry of University and Research funded by the European Union–NextGenerationEU, CUP B53C22004090006.

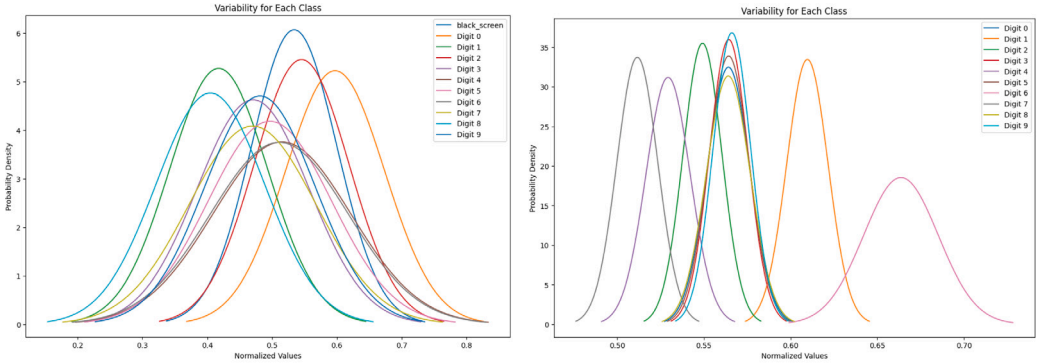
Appendix A. Basics of Riemann geometry

We briefly introduce basic concepts of Riemann geometry (Fig. A.12), in order to fully understand the architecture developed.

Let $\mathcal{M}$ be a differentiable manifold with G dimensions. Let us denote with $\boldsymbol{T}_{\boldsymbol{C}}\mathcal{M}$ the tangent space (also called derivative) of $\mathcal{M}$ at $\boldsymbol{C} \in \mathcal{M}$. Given a tangent vector $\boldsymbol{T} \in \boldsymbol{T}_{\boldsymbol{C}}\mathcal{M}$, its norm is given by the inner product operator:

$$\|\boldsymbol{T}\|_{\boldsymbol{C}} = [\langle \boldsymbol{T}, \boldsymbol{T} \rangle_{\boldsymbol{C}}]^{\frac{1}{2}} = Tr(\boldsymbol{T}\boldsymbol{C}^{-1}\boldsymbol{T}\boldsymbol{C}^{-1})^{\frac{1}{2}} \tag{A.1}$$

In order to rely on specific tangent vectors $\boldsymbol{T}'$, that are the projections of other matrices $\boldsymbol{C}' \in \mathcal{M}$ in $\boldsymbol{T}_{\boldsymbol{C}}\mathcal{M}$, as shown in Fig. A.12. In order to project $\boldsymbol{C}'$ to $\boldsymbol{T}'$, and then to project $\boldsymbol{T}'$ back to $\boldsymbol{C}'$, we introduce two



(a) Current study (b) Kumari et al.'s work

Fig. B.13. Dataset variations between the current study and Kumari et al.'s work. Each Gaussian represents the variation of a class from the two datasets under examination. They are defined by the average and the standard deviation of all brain signals belonging to the respective class. It can be seen that the Gaussians from our dataset have greater overlap and wider spread in comparison with those from Kumari's dataset..

Table B.7
Dataset variations between the current study and Kumari et al.'s work. The Gaussians (mean, std) from our dataset have a larger standard deviation (std) and an interquartile range (IQR) that is one or even two orders larger than those from Kumari's dataset.

Classes	Our dataset			Kumari et al. dataset		
	mean	std	IQR	mean	std	IQR
black_screen	0.5341	0.0657	0.0817	–	–	–
0	0.5972	0.0763	0.0954	0.5642	0.0123	0.0082
1	0.4176	0.0756	0.0944	0.6097	0.0119	0.0085
2	0.5455	0.0731	0.0907	0.5492	0.0112	0.0083
3	0.4707	0.0861	0.1066	0.5644	0.0110	0.0081
4	0.5152	0.1061	0.1327	0.5293	0.0127	0.0088
5	0.4966	0.0953	0.1179	0.5643	0.0117	0.0081
6	0.5111	0.1063	0.1328	0.6637	0.0215	0.0165
7	0.4700	0.0976	0.1216	0.5126	0.0118	0.0087
8	0.4048	0.0837	0.1040	0.5641	0.0127	0.0082
9	0.4814	0.0847	0.1038	0.5662	0.0108	0.0079

operators, the Logarithm mapping Log and the Exponential mapping Exp respectively:

$$\boldsymbol{T}' = Log_{\boldsymbol{C}}(\boldsymbol{C}') = \boldsymbol{C}^{\frac{1}{2}} log(\boldsymbol{C}^{-\frac{1}{2}} \boldsymbol{C}' \boldsymbol{C}^{-\frac{1}{2}}) \boldsymbol{C}^{\frac{1}{2}} \tag{A.2}$$

$$\boldsymbol{C}' = Exp_{\boldsymbol{C}}(\boldsymbol{T}') = \boldsymbol{C}^{\frac{1}{2}} exp(\boldsymbol{C}^{-\frac{1}{2}} \boldsymbol{T}' \boldsymbol{C}^{-\frac{1}{2}}) \boldsymbol{C}^{\frac{1}{2}} \tag{A.3}$$

Where $\boldsymbol{C}, \boldsymbol{C\prime} \in \mathcal{M}$, $\boldsymbol{T}' \in \boldsymbol{T}_{\boldsymbol{C}}\mathcal{M}$, $log(.), exp(.)$ are logarithm and exponential operations applied on a matrix. Let us define Riemann distance (also called geodesic distance) δ_R as the shortest path between $\boldsymbol{C}$ and $\boldsymbol{C\prime}$, thus equivalent to the length of the tangent vector from $\boldsymbol{C}$ to $\boldsymbol{C}'$:

$$\delta_R(\boldsymbol{C}, \boldsymbol{C}') = \|\boldsymbol{T}'\|_{\boldsymbol{C}} = \|Log_{\boldsymbol{C}}(\boldsymbol{C}')\|_{\boldsymbol{C}} \tag{A.4}$$

Finally, let us define the four spaces that are used in this work:

1. $\boldsymbol{S}_N = \{\boldsymbol{M} \in \mathbb{R}^{N \times N}, \boldsymbol{M}^T = \boldsymbol{M}, \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \geq 0 \ \forall \boldsymbol{x} \in \mathbb{R}^N \setminus 0\}$ is the space of Symmetric Positive Semi-Definite (SPSD) matrices;
2. $\boldsymbol{S}_N^+ = \{\boldsymbol{M} \in \mathbb{R}^{N \times N}, \boldsymbol{M}^T = \boldsymbol{M}, \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} > 0 \ \forall \boldsymbol{x} \in \mathbb{R}^N \setminus 0\}$ is the space of SPD matrices;
3. $\boldsymbol{S}_R = \{\boldsymbol{M} \in \boldsymbol{S}_N, rank(\boldsymbol{M}) = R, R < N\}$ is the space of SPSD reduced matrices with rank R;
4. $\boldsymbol{S}_R^+ = \{\boldsymbol{M} \in \mathbb{R}^{R \times R}, \boldsymbol{M}^T = \boldsymbol{M}, \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} > 0 \ \forall \boldsymbol{x} \in \mathbb{R}^R \setminus 0\}$ is the subspace of SPD matrices with full rank R.

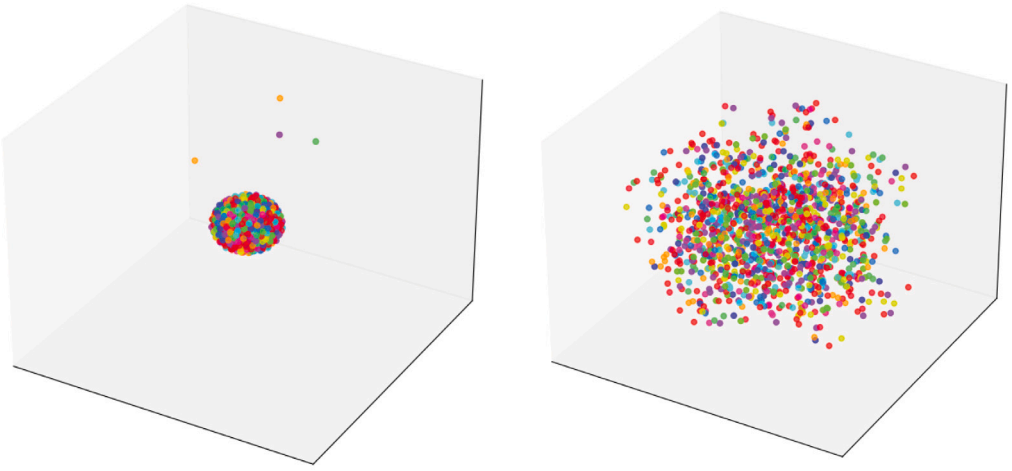
Appendix B. Dataset variations between the current study and Kumari et al.'s work

In this Section we are going to explain in detail the dataset variations between the current study and Kumary et al.'s work. Our dataset comprises 11 classes, including ten digits from 0 to 9, as well as a 'black screen' category. In contrast, Kumari's dataset comprises 10 categories that encompass the ten digits from 0 to 9 exclusively. This forms a consistent distinction between the two datasets, which makes our decoding task more engaging. Technically, including a "null case" category in a machine learning model, particularly in digit recognition or comparable activities, can be a beneficial tactic for enhancing the overall performance and stability of the model. This technique is commonly referred to as managing "negative" or "background" samples. It involves training the model to identify not only positive instances (such as EEG signals associated with digits), but also cases in which the target is absent or irrelevant.

Furthermore, Kumari's dataset contains brain signals obtained from the observation of 'digital' digits, while our dataset contains brain signals recorded during the observation of handwritten digits. Reasonably, images of a handwritten digit exhibit some variability in comparison to the image of the same 'digital' digit. To establish the discrepancy between our dataset and Kumari's dataset, it is necessary to show that the brain signals of a handwritten-digit class exhibit a greater variability than those of the corresponding 'digital'-digit class. As Kumari's dataset includes signals from 5 EEG channels ('AF3', 'AF4', 'T7', 'T8', 'Pz'), we limited our analysis to these same channels in our dataset to enable a fair comparison.

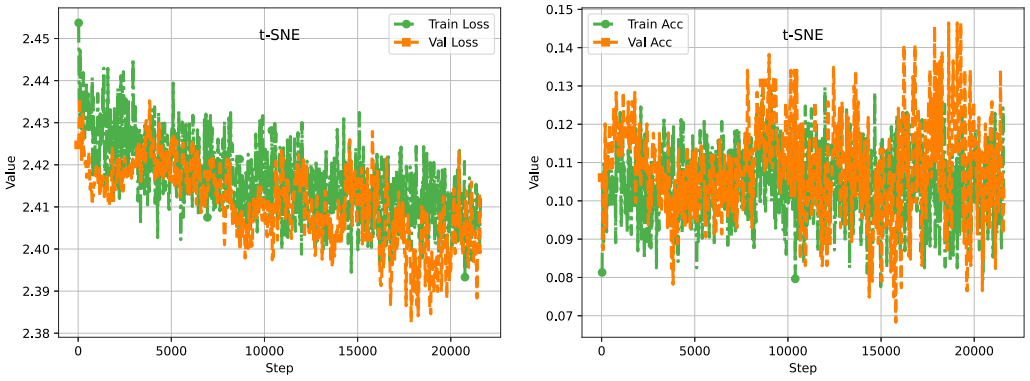
We represented the variability of a class by using a Gaussian, computing the mean and standard deviation of all brain signals referred to the corresponding digit. We replicated this process for both datasets. As a result, Fig. B.13 shows how Gaussians from our dataset are more overlapped than those from Kumari's dataset. Additionally, Gaussians from our dataset display a greater standard deviation and one- or even two-orders greater interquartile range (IQR) than those from Kumari's dataset, as shown in Table B.7.

In summary, the larger standard deviation and interquartile range suggest increased variability in our dataset compared to Kumari's



(a) t-SNE Neural Manifold (b) UMAP Neural Manifold

Fig. C.14. Neural Manifolds resulting from state-of-the-art machine learning algorithms. 3D projections of the neural manifolds, generated with t-SNE (a) and UMAP (b), are displayed. Each point relates to the representation of a recording within the feature space derived from the Cus-tomCap64-v.0.016-MindBigData dataset. Color coding demonstrates the 11 identified classes, including "black-screen", "0", "1", "2", "3", "4", "5", "6", "7", "8", and "9". Due to the high computational expense of both algorithms, manifolds of three and ten dimensions have been accomplished, respectively. Upon comparison with the manifold of our recommended architecture in Fig. 1, it is noticeable that they are more compact, resulting in an accuracy of approximately 10% in both cases, not so far from the by chance level.



(a) t-SNE Losses (b) t-SNE Accuracies

Fig. C.15. Performance of the t-SNE decoding algorithm. The Train Loss and Val Loss for training and validation are presented in figure (a). The Average accuracy profiles of Train Acc and Val Acc for the respective sets are depicted in (b). Throughout the training process (800 epochs), both losses and accuracy show a plateau, indicating that the CNN classifier is unable to learn dependencies from the manifold constructed through the t-SNE algorithm.

dataset. This suggests that the data points within each class in our dataset exhibit more dispersion and variability around the mean than in Kumari's dataset, leading to a more challenging decoding task.

Appendix C. Ablation studies

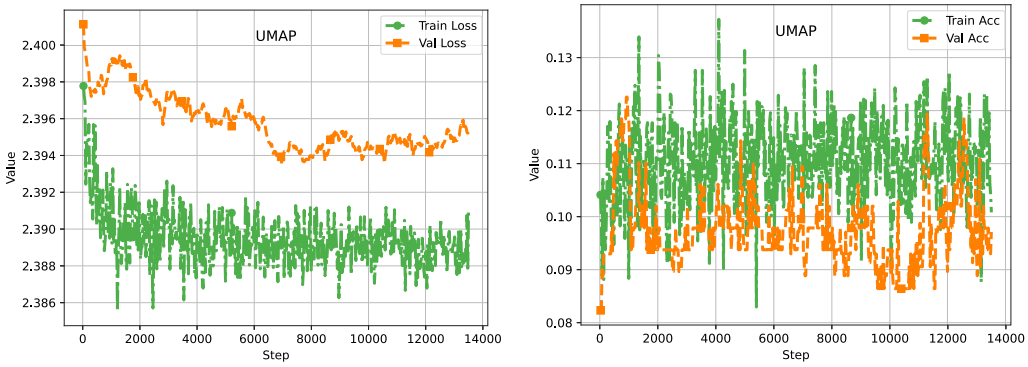
In this section, we delve into the explanation of the ablation studies carried out for this work.

To showcase the enhancement of our proposed model in separating data within the feature space, we conducted comparative experimental analysis with the two state-of-the-art algorithms from machine learning, t-SNE and UMAP. The neural manifolds attained after the application of the algorithms are presented in Fig. C.14. The experimental results are displayed in Figs. C.15 and C.16. By examining the confusion matrices depicted in Fig. C.17, it is evident that the architectures are entirely misled, as all the digits, including the black screen, are mistakenly identified as a single digit.

Not surprisingly, as discussed in Section 4, the STNet shows an expected problem in recognizing all digits as one due to its over-reliance on spatio-temporal features. This is demonstrated by the corresponding confusion matrix in Fig. C.18.

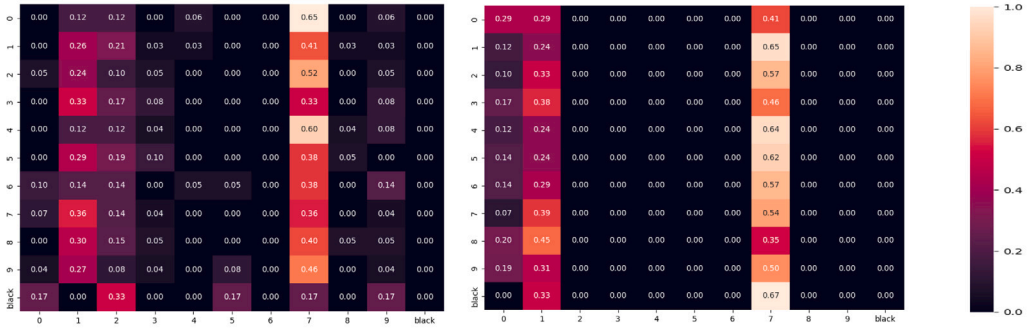
In an effort to refine the method for merging spatio-temporal features derived from brain signals with spectral features, we incorporated an attention layer prior to concatenating the results of the two processing streams illustrated in Fig. 3. This augmented version of the proposed architecture is denoted as RieManiSpectraNet+. The architectural model was trained using the same loss and optimizer parameters as the VAE in the other models. In contrast, the classifier was trained using SGD optimizer with the lr set to 10^{-2}. After observing a decreasing trend in the loss function after 500 epochs, it was reasonable to extend the training period by an additional 300 epochs to assess potential convergence. Fig. C.19 displays the experimental results. The confusion matrix in Fig. C.20 illustrates the improved integration of spatio-temporal features with spectral ones through the new attention layer, resulting in an accuracy of 86% on the Neural Image Classification task.

The significance and efficacy of combining spatial, temporal, and spectral features from brain signals are underlined by the results of the different ablation studies conducted, which are summarized in Fig. 7 and Table C.8. Further technical tests have been carried out on the suggested structure, utilizing various optimizers, including Adam and its different variants, as well as diverse learning rates. Nonetheless, they have not been documented for the aim of clarity, and the best results have been the preferred choice.



(a) UMAP Losses (b) UMAP Accuracies

Fig. C.16. Performance of the UMAP decoding algorithm. Figure (a) presents the Train Loss and Val Loss for both training and validation, while Figure (b) depicts the Average Accuracy profiles of Train Acc and Val Acc for their respective sets. The CNN classifier was unable to learn dependencies from the manifold constructed via the UMAP algorithm, as demonstrated by a plateau observed throughout the training process (800 epochs) in both losses and accuracy.



(a) t-SNE Confusion Matrix (b) UMAP Confusion Matrix

Fig. C.17. Confusion matrices resulting from state-of-the-art machine learning algorithms. The decoding performance for every class has been assessed for both architectures relying on t-SNE (a) and UMAP (b) manifolds. Confirming the limited ability demonstrated in Figs. C.15 and C.16 of the models to learn dependencies, the confusion matrices show that these architectures are completely fooled, since all the digits, even the black screen, are confused with digit '7'. This issue is entirely resolved with the proposed architecture, RieManiSpectraNet+.

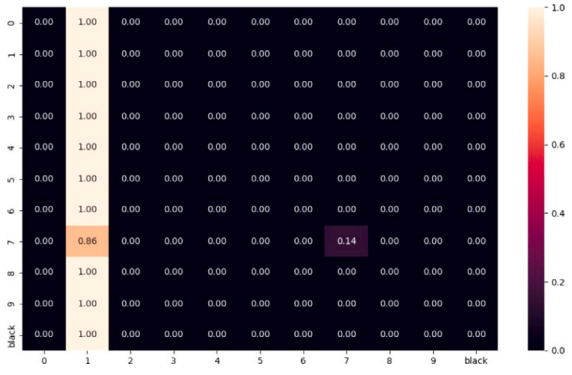
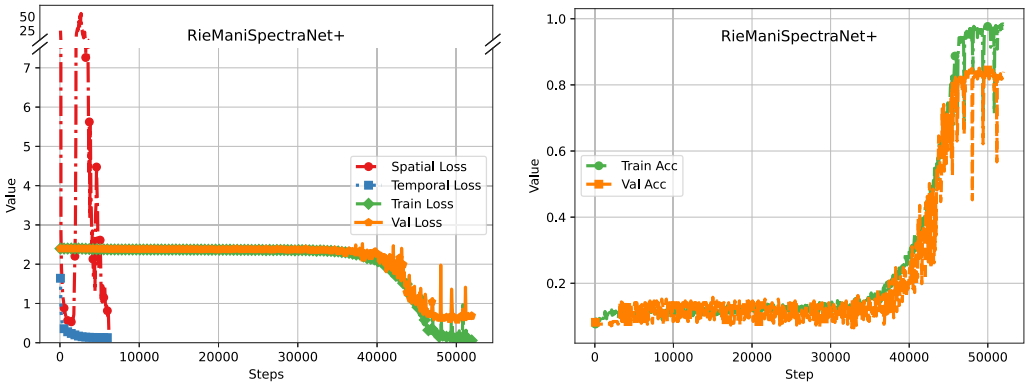


Fig. C.18. Confusion matrix resulting from STNet architecture. The decoding performance for each class has been computed. Similar to what happens for t-SNE and UMAP, the STNet architecture is entirely misled because all digits, including the black screen, are misinterpreted as digit '1'. However, this problem is completely solved by merging spatial, temporal, and spectral features under the proposed RieManiSpectraNet+ architecture.



(a) RieManiSpectraNet+ Losses (b) RieManiSpectraNet+ Accuracies

Fig. C.19. Performance of the RieManiSpectraNet+ decoding architecture. The training and validation losses (Train Loss and Val Loss, respectively), along with losses incurred through the spatio-temporal processing stream (Spatial Loss and Temporal Loss), are presented in Figure (a). It is worth noting that the graph exhibits fluctuations with peak values (around 60) that surpass those observed near convergence. Average accuracy profiles of the training set (Train Acc) and the validation set (Val Acc) are depicted in (b). Based on the data presented in 8(a) and 9(a), it can be observed that RieManiSpectraNet+ exhibits greater stability than the original RieManiSpectraNet architecture, as it maintains convergence during the final epochs.

Table C.8
Ablation studies were carried out in this study. The training process of all tested architectures, along with details of the used optimizers and losses, are documented. The cumulative accuracies achieved and displayed in Fig. 7 have been presented in a summary table.

Architectures	Accuracies (%)	Losses	Optimizers
t-SNE algorithm	13%	CrossEntropy	RMSprop (lr = 10^{-4}, wd = 10^{-6})
UMAP algorithm	12%	CrossEntropy	RMSprop (lr = 10^{-4}, wd = 10^{-6})
STNet	14%	MSE MSE CrossEntropy	Adam (lr = 10^{-5}) Adam (lr = 10^{-3}) SGD (lr = $6 * 10^{-2}$, wd = 10^{-6})
SpeNet	78%	CrossEntropy	RMSProp (lr = 10^{-4}, wd = 10^{-6})
RieManiSpectraNet	55%	MSE MSE CrossEntropy	Adam (lr = 10^{-5}) Adam (lr = 10^{-3}) SGD (lr = $6 * 10^{-2}$), wd = 10^{-6})
RieManiSpectraNet+	86%	MSE MSE CrossEntropy	Adam (lr = 10^{-5}) Adam (lr = 10^{-5}) SGD (lr = 10^{-2}, wd = 10^{-6})

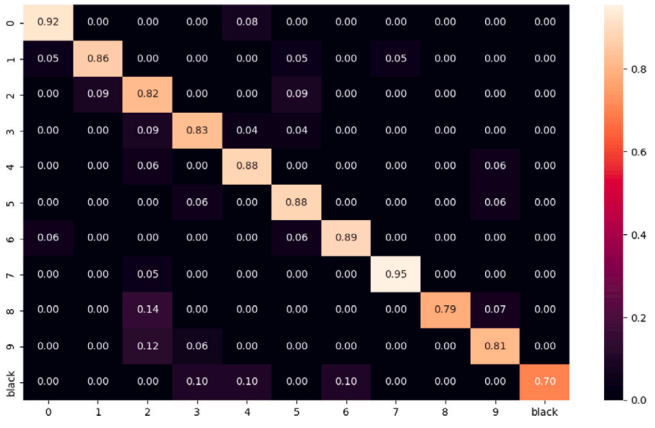


Fig. C.20. Confusion matrix resulting from RieManiSpectraNet+ architecture. The decoding performance for each class has been computed. When compared to Fig. 10(a), it is evident that all previous difficulties in distinguishing the classes have been resolved, thereby confirming that the RieManiSpectraNet+ architecture has achieved better integration of both spatio-temporal and spectral features.

References

[1] V. Gallese, A. Goldman, Mirror neurons and the simulation theory of mind-reading, Trends Cogn. Sci. 2 (12) (1998) 493–501.

[2] C.M. Heyes, C.D. Frith, The cultural evolution of mind reading, Science 344 (6190) (2014) 1243091.

[3] M. Bear, B. Connors, M.A. Paradiso, Neuroscience: Exploring the Brain, Enhanced Edition: Exploring the Brain, Jones & Bartlett Learning, 2020.

[4] S. Funahashi, Prefrontal cortex and working memory processes, Neuroscience 139 (1) (2006) 251–261.

[5] J.M. Fuster, Unit activity in prefrontal cortex during delayed-response performance: neuronal correlates of transient memory, J. Neurophysiol. 36 (1) (1973) 61–78.

[6] M. Roussy, D. Mendoza-Halliday, J.C. Martinez-Trujillo, Neural substrates of visual perception and working memory: two sides of the same coin or two different coins? Front. Neural Circuits (2021) 131.

[7] J.R. Wolpaw, N. Birbaumer, W.J. Heetderks, D.J. McFarland, P.H. Peckham, G. Schalk, E. Donchin, L.A. Quatrano, C.J. Robinson, T.M. Vaughan, et al., Brain-computer interface technology: A review of the first international meeting, IEEE Trans. Rehabil. Eng. 8 (2) (2000) 164–173.

[8] K.O. Johnson, Neural coding, Neuron 26 (3) (2000) 563–566.

[9] Y. Kamitani, F. Tong, Decoding the visual and subjective contents of the human brain, Nature Neurosci. 8 (5) (2005) 679–685.

[10] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, M. Shah, Generative adversarial networks conditioned by brain signals, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3410–3418.

[11] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, M. Shah, Brain2image: Converting brain signals into images, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 1809–1817.

[12] D.L. Yamins, J.J. DiCarlo, Using goal-driven deep learning models to understand sensory cortex, Nature Neurosci. 19 (3) (2016) 356–365.

[13] T. Naselaris, R.J. Prenger, K.N. Kay, M. Oliver, J.L. Gallant, Bayesian reconstruction of natural images from human brain activity, Neuron 63 (6) (2009) 902–915.

[14] J.V. Haxby, J.S. Guntupalli, A.C. Connolly, Y.O. Halchenko, B.R. Conroy, M.I. Gobbini, M. Hanke, P.J. Ramadge, A common, high-dimensional model of the representational space in human ventral temporal cortex, Neuron 72 (2) (2011) 404–416.

[15] D. Bacher, B. Jarosiewicz, N.Y. Masse, S.D. Stavisky, J.D. Simeral, K. Newell, E.M. Oakley, S.S. Cash, G. Friehs, L.R. Hochberg, Neural point-and-click communication by a person with incomplete locked-in syndrome, Neurorehabil. Neural Repair 29 (5) (2015) 462–471.

[16] M.L. Gorno-Tempini, A.E. Hillis, S. Weintraub, A. Kertesz, M. Mendez, S.F. Cappa, J.M. Ogar, J.D. Rohrer, S. Black, B.F. Boeve, et al., Classification of primary progressive aphasia and its variants, Neurology 76 (11) (2011) 1006–1014.

[17] D. Dash, P. Ferrari, A.W. Hernandez-Mulero, D. Heitzman, S.G. Austin, J. Wang, Neural speech decoding for amyotrophic lateral sclerosis, in: INTERSPEECH, 2020, pp. 2782–2786.

[18] M.J. Russo, V. Prodan, N.N. Meda, L. Carcavallo, A. Muracioli, L. Sabe, L. Bonamico, R.F. Allegri, L. Olmos, High-technology augmentative communication for adults with post-stroke aphasia: A systematic review, Expert Rev. Med. Devices 14 (5) (2017) 355–370.

[19] S.C. Kleih, L. Gottschalt, E. Teichlein, F.X. Weilbach, Toward a P300 based brain-computer interface for aphasia rehabilitation after stroke: presentation of theoretical considerations and a pilot feasibility study, Front. Hum. Neurosci. 10 (2016) 547.

[20] E.K. Miller, T.J. Buschman, Cortical circuits for the control of attention, Curr. Opin. Neurobiol. 23 (2) (2013) 216–222.

[21] S. Panzeri, J.H. Macke, J. Gross, C. Kayser, Neural population coding: combining insights from microscopic and mass signals, Trends Cogn. Sci. 19 (3) (2015) 162–172.

[22] R.E. Bellman, Dynamic Programming, Princeton University Press, 2010.

[23] M.M. Churchland, J.P. Cunningham, M.T. Kaufman, J.D. Foster, P. Nuyujukian, S.I. Ryu, K.V. Shenoy, Neural population dynamics during reaching, Nature 487 (7405) (2012) 51–56.

[24] E.H. Nieh, M. Schottdorf, N.W. Freeman, R.J. Low, S. Lewallen, S.A. Koay, L. Pinto, J.L. Gauthier, C.D. Brody, D.W. Tank, Geometry of abstract learned knowledge in the hippocampus, Nature 595 (7865) (2021) 80–84.

[25] V. Mante, D. Sussillo, K.V. Shenoy, W.T. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex, Nature 503 (7474) (2013) 78–84.

[26] M. Rigotti, O. Barak, M.R. Warden, X.-J. Wang, N.D. Daw, E.K. Miller, S. Fusi, The importance of mixed selectivity in complex cognitive tasks, Nature 497 (7451) (2013) 585–590.

[27] R. Mitchell-Heggs, S. Prado, G.P. Gava, M.A. Go, S.R. Schultz, Neural manifold analysis of brain circuit dynamics in health and disease, J. Comput. Neurosci. 51 (1) (2023) 1–21.

[28] J.-D. Haynes, G. Rees, Decoding mental states from brain activity in humans, Nature Rev. Neurosci. 7 (7) (2006) 523–534.

[29] S. Nishimoto, A.T. Vu, T. Naselaris, Y. Benjamini, B. Yu, J.L. Gallant, Reconstructing visual experiences from brain activity evoked by natural movies, Curr. Biol. 21 (19) (2011) 1641–1646.

[30] K.J. Friston, P. Jezzard, R. Turner, Analysis of functional MRI time-series, Hum. Brain Mapp. 1 (2) (1994) 153–171.

[31] E.H. Adelson, J.R. Bergen, Spatiotemporal energy models for the perception of motion, Josa A 2 (2) (1985) 284–299.

[32] K.N. Kay, T. Naselaris, R.J. Prenger, J.L. Gallant, Identifying natural images from human brain activity, Nature 452 (7185) (2008) 352–355.

[33] M. Ahmadi, F. Dashti Ahangar, N. Astaraki, M. Abbasi, B. Babaei, et al., FWNNet: presentation of a new classifier of brain tumor diagnosis based on fuzzy logic and the wavelet-based neural network using machine-learning methods, Comput. Intell. Neurosci. 2021 (2021).

[34] M. Ahmadi, M. Soofiabadi, M. Nikpour, H. Naderi, L. Abdullah, B. Arandian, Developing a deep neural network with fuzzy wavelets and integrating an inline PSO to predict energy consumption patterns in urban buildings, Mathematics 10 (8) (2022) 1270.

[35] M. Zangeneh Soroush, P. Tahvilian, M.H. Nasirpour, K. Maghooli, K. Sadeghniiat-Haghighi, S. Vahid Harandi, Z. Abdollahi, A. Ghazizadeh, N. Jafarnia Dabanloo, EEG artifact removal using sub-space decomposition, nonlinear dynamics, stationary wavelet transform and machine learning algorithms, Front. Physiol. 13 (2022) 910368.

[36] M.Z. Ahmad, A.A. Khan, S. Mezghani, E. Perrin, K. Mouhoubi, J.-L. Bodnar, V. Vrabie, Wavelet subspace decomposition of thermal infrared images for defect detection in artworks, 2015, arXiv:1508.06010.

[37] K. Seeliger, U. Güçlü, L. Ambrogioni, Y. Güçlütürk, M.A. van Gerven, Generative adversarial networks for reconstructing natural images from brain activity, NeuroImage 181 (2018) 775–785.

[38] M. Mozafari, L. Reddy, R. VanRullen, Reconstructing natural scenes from fmri patterns using bigbigan, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–8.

[39] K. Qiao, J. Chen, L. Wang, C. Zhang, L. Tong, B. Yan, Biggan-based bayesian reconstruction of natural images from human brain activity, Neuroscience 444 (2020) 92–105.

[40] Z. Rakhimberdina, Q. Jodelet, X. Liu, T. Murata, Natural image reconstruction from fmri using deep learning: A survey, Front. Neurosci. 15 (2021) 795488.

[41] Z. Ren, J. Li, X. Xue, X. Li, F. Yang, Z. Jiao, X. Gao, Reconstructing seen image from brain activity by visually-guided cognitive representation and adversarial learning, NeuroImage 228 (2021) 117602.

[42] R. VanRullen, L. Reddy, Reconstructing faces from fMRI patterns using deep generative neural networks, Commun. Biol. 2 (1) (2019) 193.

[43] M. Dai, D. Zheng, R. Na, S. Wang, S. Zhang, EEG classification of motor imagery using a novel deep learning framework, Sensors 19 (3) (2019) http://dx.doi.org/10.3390/s19030551, URL https://www.mdpi.com/1424-8220/19/3/551.

[44] Z. Jiao, X. Gao, Y. Wang, J. Li, H. Xu, Deep convolutional neural networks for mental load classification based on EEG data, Pattern Recognit. 76 (2018) 582–595.

[45] J. Shamwell, H. Lee, H. Kwon, A.R. Marathe, V. Lawhern, W. Nothwang, Single-trial EEG RSVP classification using convolutional neural networks, in: Micro-and Nanotechnology Sensors, Systems, and Applications VIII, vol. 9836, SPIE, 2016, pp. 373–382.

[46] R. Manor, A.B. Geva, Convolutional neural network for multi-category rapid serial visual presentation BCI, Front. Comput. Neurosci. 9 (2015) 146.

[47] D. Vivancos, F. Cuesta, MindBigData 2022 a large dataset of brain signals, 2022, arXiv preprint arXiv:2212.14746.

[48] N. Kumari, S. Anwar, V. Bhattacharjee, Convolutional neural network-based visually evoked EEG classification model on MindBigData, in: Proceedings of Research and Applications in Artificial Intelligence: RAAI 2020, Springer, 2021, pp. 233–241.

[49] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, M. Shah, Deep learning human mind for automated visual classification, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 4503–4511, http://dx.doi.org/10.1109/CVPR.2017.479.

[50] L. Yang, L.L.H. Chan, Y. Lu, Decoding of visual-related information from the human EEG using an end-to-end deep learning approach, 2019, arXiv:1911.00550.

[51] J.B. Allen, L.R. Rabiner, A unified approach to short-time Fourier analysis and synthesis, Proc. IEEE 65 (11) (1977) 1558–1564.

[52] W.D. Mellin, Work with new electronic 'brains' opens field for army math experts, Hammond Times 10 (1957) 66.

[53] D. Vivancos, F. Cuesta, MindBigData website, 2021, URL http://mindbigdata.com/opendb/visualmnist.html. (Accessed 7 May 2023),

[54] Y. LeCun, The MNIST database of handwritten digits, 1998, http://yann.lecun.com/exdb/mnist/.

[55] G. Zhang, A. Etemad, Spatio-temporal EEG representation learning on Riemannian manifold and euclidean space, 2020, arXiv e-prints, arXiv–2008.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, 2017.

[57] A.K. Maddirala, K.C. Veluvolu, Eye-blink artifact removal from single channel EEG with k-means and SSA, Sci. Rep. 11 (1) (2021) 11043.

[58] E. Niedermeyer, F.L. da Silva, Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, Lippincott Williams & Wilkins, 2005.

[59] I. Goodfellow, Y. Bengio, A. Courville, Deep feedforward networks, Deep Learn. (1) (2016).

[60] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.

[61] A.J. Izenman, Introduction to manifold learning, Wiley Interdiscip. Rev. Comput. Stat. 4 (5) (2012) 439–446.

[62] G.E. Hinton, S. Roweis, Stochastic neighbor embedding, in: Advances in Neural Information Processing Systems, vol. 15, 2002.

[63] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

[64] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, 2022, arXiv:1312.6114.

[65] Y. Thanwerdas, X. Pennec, O (n)-invariant Riemannian metrics on SPD matrices, Linear Algebra Appl. 661 (2023) 163–201.

[66] O. Tsinalis, P.M. Matthews, Y. Guo, S. Zafeiriou, Automatic sleep stage scoring with single-channel EEG using convolutional neural networks, 2016, arXiv:1610.01683.

Salvatore Falciglia is a pre-doctoral research fellow at the Biorobotics Institute of the Sant'Anna School of Advanced Studies — Pisa. He works in the Computational Neuro-engineering Laboratory of Professor Alberto Mazzoni. His main interests include information processing in the nervous system, decoding of neural signals through deep learning techniques, neuro-robotics and biorobotics applications. He received the B.Sc. degree cum laude in Electronic Engineering from the University of Catania, in 2021. He also earned the M.Sc. degree cum laude in Artificial Intelligence and Robotics from Sapienza University of Rome, in 2023, while attending the Superior School of Advanced Studies (SSAS) Sapienza.

Filippo Betello is a Ph.D. candidate at the Sapienza University of Rome. He works under the supervision of Professor Fabrizio Silvestri. His main interests are deep learning, information retrieval, robustness-fairness, and applications to medicine. He received his Master's degree in Artificial Intelligence and Robotics cum laude in October 2023 and his Bachelor's degree in Computer and Automatic Engineering cum laude in July 2021, both from Sapienza University of Rome.

Samuele Russo is a Psychotherapist, Clinical Psychologist, Pediatric Psychologist and EMDR Psychotherapist, Ph.D. candidate in Behavioral Neurosciences at the Department of Psychology and former research fellow of Sapienza University of Rome. He got a BSc. (2013) and a MSc. in Psychology (2015), a 2nd level Master in Pediatric Psychology (2019), a certification as OMS-recognized EMDR Psychotherapist (2022), a certification in bio-neurofeedback techniques (2022) and the specialization in Systemic-Relational Psychotherapy (2023). His research interests are relate to Vision, Navigation, Visual-spatial orientation, Mental Imagery, Executive Functions, Brain Signals Analysis and Models, and EMDR therapy.

Christian Napoli is Associate Professor at Sapienza University of Rome, habilitated as Full Professor, Director of the Intelligent Systems Laboratory, and Scientific Director of the International School of Advanced and Applied Computing (ISAAC). He received the B.Sc. in Physics in 2010, the M.Sc. in Astrophysics in 2012, and the Ph.D. in Computer Science in 2016. Formerly Research Associate and Research Fellow at the University of Catania. Several times Invited Professor at the Silesian and Czestochowa Universities of Technology, and Visiting Academic at the New York University. His current research interests include neural networks, artificial intelligence, human–computer interaction, and computational neuropsychology.