Image-Based Motor Imagery EEG Classification With Deep Learning for Brain Computer Interfaces

Ward Fadel

Pázmány Péter Catholic University

Faculty of Information Technology and Bionics

Roska Tamás Doctoral School of Sciences and Technology

Supervisors

Dr. István Ulbert, DSc

Dr. Lucia Wittner, DSc

A thesis submitted for the degree of Doctor of Philosophy

Budapest, 2025

1 Introduction

1.1 Motivation and Goals

Electroencephalography (EEG)-based Motor Imagery (MI) classification forms the backbone of non-invasive Brain–Computer Interface (BCI) systems, offering users the ability to interact with external devices solely through imagined movements. This has profound implications for assistive technologies, rehabilitation, and human–computer interaction. Despite decades of research, however, robust, generalizable, and high-performance MI-EEG classifiers remain elusive. The difficulty arises from the intrinsic characteristics of EEG: its low signal-to-noise ratio, coarse spatial resolution, and strong inter- and intra-subject variability. These limitations make reliable decoding of motor imagery highly challenging in practical, real-world settings.

Traditional BCI pipelines have largely relied on handcrafted features—such as bandpower modulations in Mu (8–13 Hz) and Beta (13–30 Hz) rhythms at electrodes C3, C4, and Cz, typically associated with event-related desynchronization (ERD) and synchronization (ERS). While such features capture the canonical sensorimotor rhythm, they fail to consistently generalize. Inter- and intra-session variability, deviations from expected ERD/ERS patterns, and the inherently dynamic and distributed nature of cortical activations mean that C3/C4/Cz-centric approaches are insufficient. Moreover, these pipelines frequently employ shallow classifiers such as Linear Discriminant Analysis (LDA) or Support Vector Machines (SVM), which cannot fully exploit the complex spatiotemporal structure of EEG.

Recent advances in Deep Learning offer a powerful alternative: rather than manually crafting features, neural networks can learn directly from data, provided the input representation is rich and structured. However, EEG signals are not naturally image-like or spatially uniform, which poses a barrier to leveraging state-of-the-art deep learning architectures from computer vision. This motivates a rethinking of how EEG should be represented before classification.

The central motivation of this thesis is therefore twofold:

1. To overcome the limitations of traditional electrode- and feature-centric methods by moving beyond sparse sensorimotor channels and embracing the full 64-electrode montage, thereby capturing richer spatial patterns across the cortex.

2. To enable deep learning to operate effectively on EEG data by transforming irregular, noisy signals into structured 2D image representations that preserve neurophysiological meaning while aligning with convolutional neural network (CNN) inductive biases.

Guided by these motivations, this work pursues the following goals:

- Design novel EEG-to-image transformations. We propose two complementary methods:

- Chessboard Projection, introduced here as a novel discrete grid-aligned encoding, which maps electrodes onto a regularized 2D lattice optimized for compatibility with CNNs.

- Interpolated Azimuthal Projection, deployed here for MI-EEG, which preserves cortical topology through geodesic projection and interpolation, producing smooth topographic maps.

- Develop a hybrid spatiotemporal deep learning framework. A CNN branch is designed to extract spatial and spectral features from the transformed images, while an LSTM branch captures the temporal dynamics of cortical activity across motor imagery epochs. This architecture is co-designed with the input representation, scaled to exploit the distinct properties of Chessboard and Azimuthal encodings.

- Systematically evaluate the role of spectral bands. Beyond Mu and Beta, we investigate the contribution of the Delta band (0.5–4 Hz). Our analysis challenges the prevailing assumption of Delta's irrelevance in motor imagery, demonstrating that its decodability is contingent on spatial encoding.

- Validate on a large-scale dataset. We employ a 64-channel, multi-subject dataset with multiple MI classes, ensuring that findings are not limited to small, constrained experimental setups. This allows us to assess both within-subject and cross-subject generalizability, a critical step toward practical BCI deployment.

- Reframe representation as a decisive factor in EEG decoding. By contrasting Chessboard and Azimuthal projections, we show that spatial encoding is not a neutral preprocessing step but a design choice that governs what information the brain makes accessible to machine learning.

In summary, the motivation driving this thesis is the inadequacy of conventional MI-EEG approaches in handling variability, complexity, and distributed cortical dynamics. The goal is to establish image transformation as a principled and powerful paradigm for EEG representation, enabling deep learning models to achieve more robust, interpretable, and generalizable motor imagery classification.

1.2 Functional Organization of the Sensorimotor Cortex

The synchronized activity of neuronal populations provides the foundation for understanding how the cortex encodes sensory and motor functions at a macroscopic level. Large populations of pyramidal neurons, organized into cortical columns and aligned across layers, generate the population-level signals measurable by EEG. These neuronal populations are also topographically organized across distinct cortical regions, often delineated by Brodmann areas, which correspond to primary sensory, motor, and associative functions (Figure 1). For example, areas 1–3 correspond to the primary somatosensory cortex, area 4 to the primary motor cortex, area 17 to the primary visual cortex, and areas 44/45 to Broca's language region.

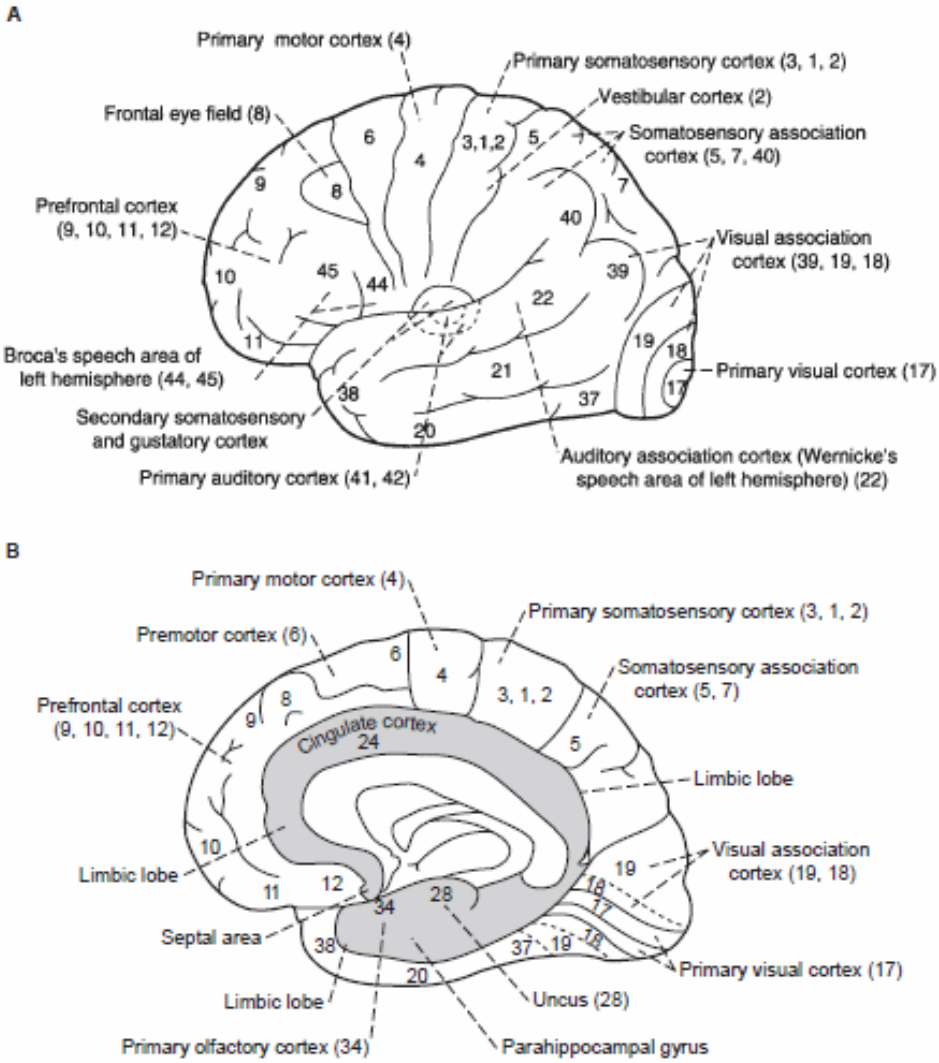

Figure 1: Brodmann Areas of the Human Cortex. Schematic map of the cerebral cortex showing selected Brodmann areas and their approximate functional assignments, with (a) lateral and (b) medial views of the cortical surface and inner structures [1].

Building on this anatomical framework, Penfield's cortical homunculi provide a functional illustration of how body regions are represented across the sensorimotor cortex. The somatosensory homunculus (A) on the postcentral gyrus highlights regions with high sensory acuity, such as the lips, tongue, and fingertips, whereas the motor homunculus (B) on the precentral gyrus emphasizes areas requiring fine motor control, notably the hands and face

(Figure 2). These representations reflect the principle that cortical allocation is determined by functional importance rather than physical size. Recent neuroimaging and electrophysiological studies indicate that cortical boundaries and proportional representations are more complex and variable across individuals, with overlapping somatotopic patterns.

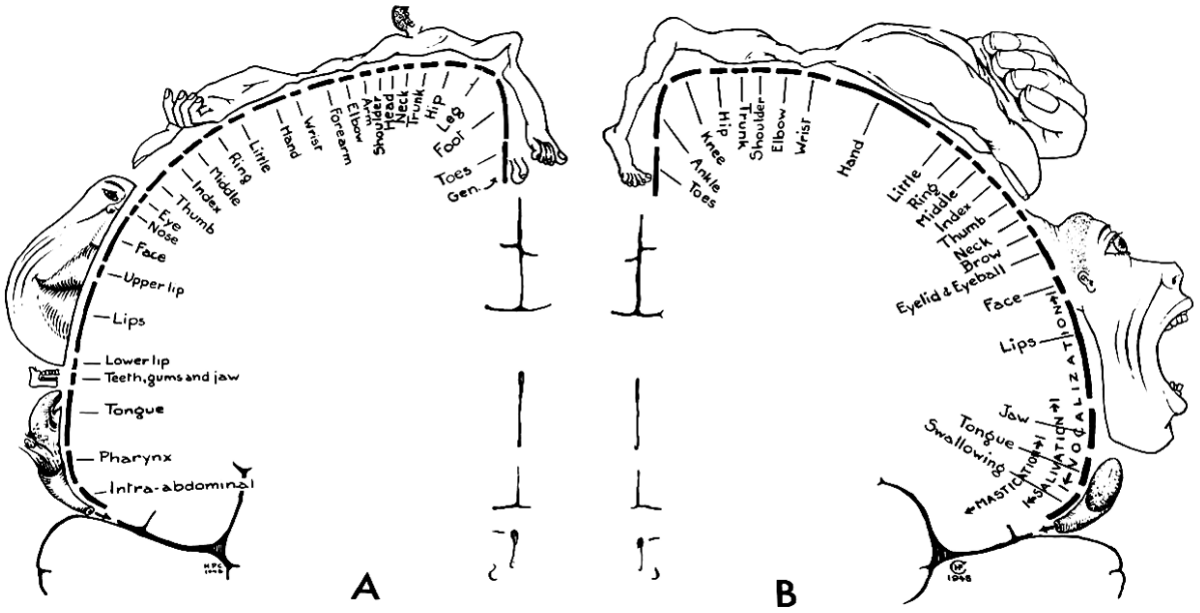

Figure 2: The cortical homunculi originally mapped by Wilder Penfield in 1948. A: the somatosensory homunculus on the postcentral gyrus of the parietal lobe, illustrating cortical representation of touch and sensation. B: the motor homunculus on the precentral gyrus of the frontal lobe, illustrating cortical representation of voluntary movement. Disproportionately large hands, lips, and tongue reflect their greater cortical allocation.

1.3 Convolutional Neural Network (CNN)

The concept of convolutional neural networks (CNNs) originated from the biological inspiration of the visual cortex, particularly the discovery of receptive fields in the work of Hubel and Wiesel [2]. Early computational models, such as Neocognitron [3], introduced the idea of local connectivity and weight sharing to mimic hierarchical feature extraction. Modern CNNs were popularized by LeCun et al. [4] through the LeNet architecture, which successfully recognized handwritten digits. Later, the breakthrough AlexNet [5] demonstrated the effectiveness of deep CNNs for large-scale image classification, significantly outperforming traditional handcrafted feature approaches. Building on these advances, deeper architectures such as VGGNet [6] and ResNet [7] refined convolutional designs and demonstrated that depth, with careful architectural choices, could yield superior generalization. CNNs have since become the dominant architecture for computer vision and related tasks, including medical imaging, speech processing, and EEG-based classification. Their strength lies in automatic feature extraction through hierarchical filters, enabling models to learn low-level features (edges, textures) and high-level representations (objects, patterns) directly from raw data. A CNN learns hierarchical representations through a series of learnable filters and non-linear transformations. The core components include convolutional layers, activation functions, pooling operations, and fully connected layers.

1. Convolutional Layer

Given an input image (or feature map) $X \in R^{H \times W \times C_{\text{in}}}$, where H is the height, W is the width, and C_{in} is the number of input channels, a convolutional layer applies a set of learnable kernels (filters) $R^{k \times k \times C_{\text{in}} \times C_{\text{out}}}$ where k is the kernel size and C_{out} is the number of output channels.

The convolution operation at spatial location (i, j) for output channel mmm is:

$$Y_{i,j,m} = \sum_{c=1}^{C_{\text{in}}} \sum_{u=1}^{k} \sum_{v=1}^{k} K_{u,v,c,m} \cdot X_{i+u,j+v,c} + b_m \quad (1)$$

where b_m is the bias term for channel m. Stride s: controls the step size of the convolution. Padding p: adds zeros around the input to preserve spatial resolution

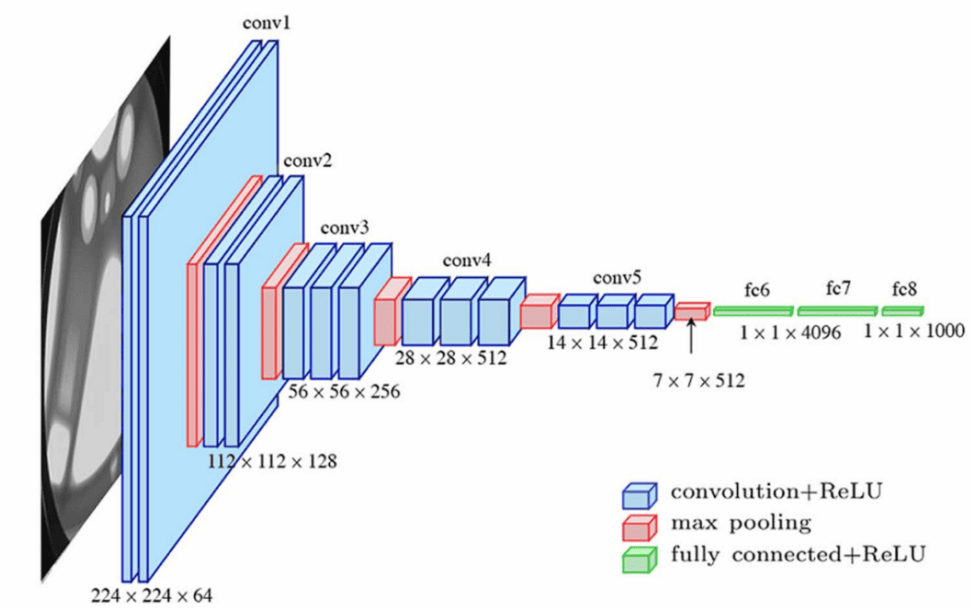

Figure 3: Architecture of the VGG Convolutional Neural Network. Schematic representation of the VGG network illustrating the sequence of convolutional layers, pooling layers, and fully connected layers. The model is characterized by its use of small 3×3 convolutional filters, stacked in increasing depth across the network, with max-pooling layers for spatial downsampling and fully connected layers for classification at the output.

2. Activation Function

After convolution, a non-linear activation is applied. The most common is the Rectified Linear Unit (ReLU):

ReLU introduces non-linearity while being computationally efficient and alleviating the vanishing gradient problem compared to sigmoid or tanh activations.

3. Pooling Layer

Pooling reduces the spatial resolution, enabling translation invariance and reducing computational complexity. The most common is MaxPooling, defined as:

$$Y_{i,j,c} = \max_{(u,v) \in \mathcal{R}} X_{s \cdot i+u,\, s \cdot j+v,\, c} \quad (2)$$

where R is the pooling region (e.g., 2×2), and s is the stride.

4. Fully Connected Layer

A fully connected (dense) layer flattens the feature maps and applies a linear transformation:

$$y = Wx + by = Wx + b \tag{3}$$

where:

$$W \in R^{d_{\text{out}} \times d_{\text{in}}}, x \in R^{d_{\text{in}}}, y \in R^{d_{\text{out}}}$$

The final layer for classification applies the softmax function to output probabilities over C classes:

$$\widehat{y_c} = \frac{\exp(z_c)}{\sum_{j=1}^{C} \exp(z_j)}, \quad c = 1, \ldots, C. \tag{4}$$

5. Training

CNN parameters are optimized by minimizing a loss function using gradient descent. For classification, the cross-entropy loss is standard:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log(\widehat{y_c}) \tag{5}$$

where y_c is the one-hot ground truth label and $\widehat{y_c}$ is the predicted probability.

Optimization is performed using Stochastic Gradient Descent (SGD):

$$\theta \leftarrow \theta - \eta \nabla_{\theta \mathcal{L}} \tag{6}$$

where θ are the learnable parameters, η is the learning rate, and $\nabla_{\theta \mathcal{L}}$ is the gradient computed by backpropagation.

1.4 Long Short-Term Memory (LSTM) Networks

Recurrent Neural Networks (RNNs) were originally designed to process sequential data by maintaining a hidden state that evolves over time. However, traditional RNNs suffer from the vanishing and exploding gradient problem [8], which hinders their ability to capture long-term dependencies. To address this limitation, Sepp Hochreiter and Jürgen Schmidhuber [9] introduced the Long Short-Term Memory (LSTM) architecture. LSTMs augment the RNN structure with a memory cell and gating mechanisms that regulate information flow, enabling stable learning over long sequences. Today, LSTMs are widely used in natural language processing, time series forecasting, and EEG/biomedical signal analysis.

Recurrent Neural Networks (RNNs) are designed to model sequential and time-dependent data, addressing a key limitation of feed-forward neural networks. In standard feed-forward architectures, the prediction $\mathrm{p}(\mathrm{y_t} \mid \mathrm{x_t})$ depends only on the current input $\mathrm{x_t}$, with all prior information encoded implicitly in the learned weights after training. This lack of explicit memory makes it difficult to capture sequential or temporal dependencies.

RNNs extend this paradigm by introducing a hidden state h_t that recurs across time:

$$h_t = f(W_x x_t + W_h h_{t-1} + b) \tag{8}$$

where: x_t is the input at time t, h_t is the hidden state, W_x, W_h are weight matrices, b is a bias term, $f(\cdot)$ is a nonlinear activation (e.g., $tanh$ or $ReLU$).

The unfolded representation of an RNN shows how the hidden state is passed along a temporal chain, enabling the network to condition predictions on both current and past information:

$$p(y_t \mid x_t, h_{t-1}) \tag{9}$$

This recurrence allows RNNs to retain information from earlier time steps, making them suitable for sequence modeling in language, speech, and EEG signals.

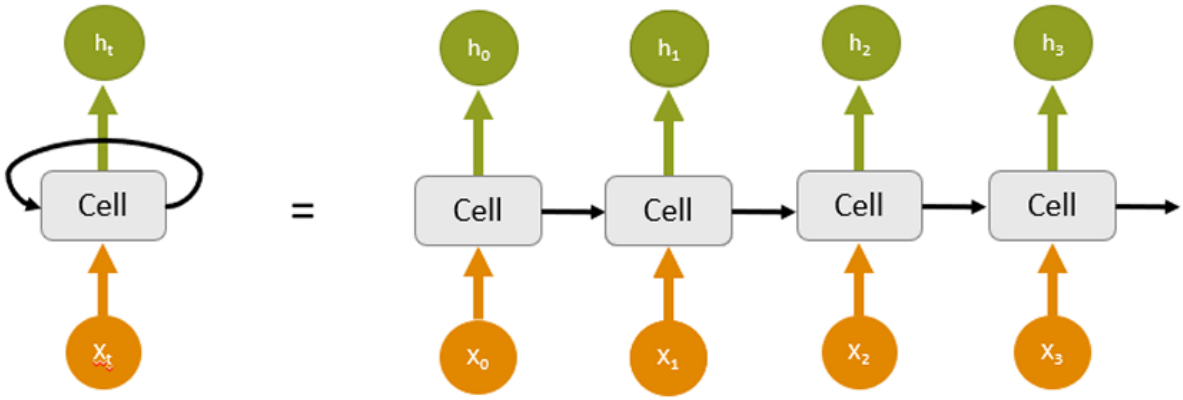

Figure 4: Unfolded Representation of a Recurrent Neural Network (RNN). Schematic illustration of an RNN unfolded through time, showing how the hidden state is passed recurrently from one time step to the next. Each unit receives input at time t, updates its hidden state based on both the new input and the previous hidden state, and produces an output. This unfolded view highlights the temporal dependencies modeled by the network.

The vanishing gradient problem is a major challenge in training RNNs**.** In theory, the hidden state h_t can carry information across an arbitrary number of time steps, enabling long-term dependencies to be modeled. In practice, however, training relies on *Backpropagation Through Time (BPTT)*, where loss gradients must be propagated not only across the depth of the network but also through the temporal dimension. The recursive update $h_t = f(W_x x_t + W_h h_{t-1} + b)$ causes gradients to involve repeated multiplication by Jacobian matrices. Over many time steps, these multiplications tend to either: Vanish (gradients approach 0) leading to long-term dependencies cannot be learned, or Explode (gradients grow uncontrollably) leading to unstable training. This is known as the *vanishing/exploding gradient problem* [8] It is especially problematic for tasks where relevant dependencies span tens or hundreds of time steps, such as EEG signal analysis.

To overcome the vanishing gradient problem, Hochreiter and Schmidhuber [9] proposed the *Long Short-Term Memory (LSTM)* architecture. An LSTM cell enhances the RNN cell by introducing a *cell state* c_t, which serves as long-term memory. Unlike the hidden state, which is updated at every time step, the cell state allows information to flow along a conveyor belt-like pathway, with only minor linear interactions. Thus, the LSTM has access to both *short-term memory*, represented by the hidden state h_t, and *long-term memory*, maintained through the cell state c_t. Carefully designed *gates* (forget, input, output) regulate how information enters, persists, or leaves the cell. These gates ensure that essential information and error gradients can pass unchanged across long temporal distances, solving the vanishing gradient problem.

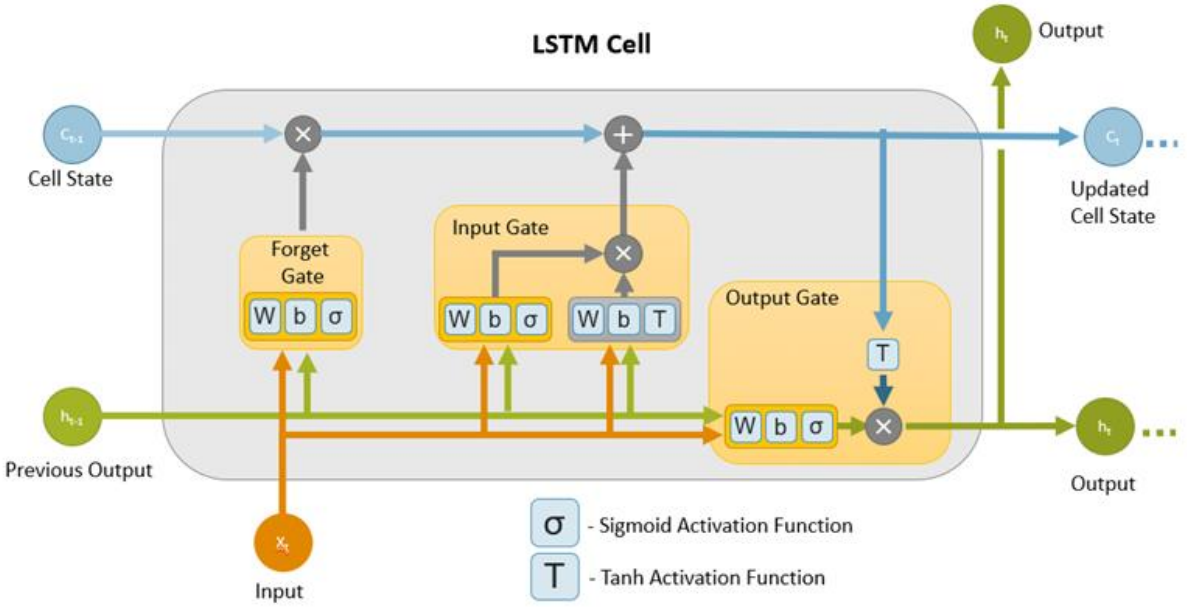

Figure 5: Long Short-Term Memory (LSTM) Cell Structure. Schematic diagram of a single LSTM cell showing the flow of information through the cell state and hidden state. The input gate, forget gate, and output gate regulate how new information is added, how past information is retained or discarded, and how the final hidden state is produced. This gating mechanism enables LSTMs to capture both short- and long-term dependencies in sequential data.

Consider an input sequence: $x_1, x_2, \dots, x_T$, where $x_t \in R^{d_x}$ with hidden state dimension d_h.

At each time step t, the LSTM maintains:

- A hidden state: $h_t \in R^{d_h}$
- A cell state (memory): $c_t \in R^{d_h}$

The dynamics are governed by gates, each represented as a function of the current input x_t ,and previous hidden state h_{t-1}.

1. Forget Gate: Controls how much of the previous memory c_{t-1} should be retained.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{10}$$

where: $f_t \in [0,1]^{d_h}$ (elementwise values close to 0 → forget, close to 1 → keep), $W_f \in R^{d_h \times d_x}$, $U_f \in R^{d_h \times d_h}$, $b_f \in R^{d_h}$.

2. Input Gate: Regulates how much new information enters the celland is defined as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{11}$$

a candidate cell update is then computed as:

$$\widetilde{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{12}$$

where $\widetilde{c_t} \in R^{d_h}$ represents new candidate memory content.

3.Cell State Update: The new cell state combines retained old memory and candidate information:

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t} \tag{13}$$

where $\odot$ denotes elementwise multiplication.

4. Output Gate: Determines which parts of the memory are exposed to the hidden state:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{14}$$

The hidden state is then updated as:

$$h_t = o_t \odot \tanh(c_t) \tag{15}$$

Here, $\sigma(\cdot)$ denotes the sigmoid activation function, $\sigma(z) = \frac{1}{1+e^{-z}}$, which squashes values into $(0,1)$ and is used in the gates. The function $tanh(\cdot)$ denotes the hyperbolic tangent activation, $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, which outputs values in $(-1,1)$ and is used to regulate candidate updates and hidden states.

1.5 Azimuthal Equidistant Projection for EEG Electrode Layouts

To visualize and analyze scalp EEG data, electrode positions obtained from the realistic head model were projected from three-dimensional head coordinates onto a two-dimensional plane using an azimuthal equidistant projection. This method is well suited for EEG topography because it preserves the relative distances of electrodes from a central reference point. In this study, the vertex electrode (Cz) was selected as the projection center, as it lies near the geometric midpoint of the electrode array and minimizes distortion across the scalp surface. Each electrode position (x, y, z) was first expressed in spherical coordinates, where the radius is defined as $r = \sqrt{x^2 + y^2 + z^2}$, the colatitude as $\arccos(z/r)$, and the azimuth as $\phi = \arctan 2\,(y, x)$. The azimuthal projection then maps these spherical coordinates to two-dimensional Cartesian coordinates through $\rho = k\theta, X = \rho\cos(\phi)$, and $Y = \rho\sin(\phi)$, where ρ represents the distance from Cz in the 2D plane and k is a scaling factor proportional to the head radius. This formulation ensures that distances from Cz are preserved, thereby maintaining local spatial relationships between electrodes. By applying this projection, the 64-electrode configuration derived from the realistic head model was transformed into a two-dimensional representation suitable for generating interpretable electrode layouts and scalp potential maps [10]. The resulting projection is illustrated in Figure. 9.

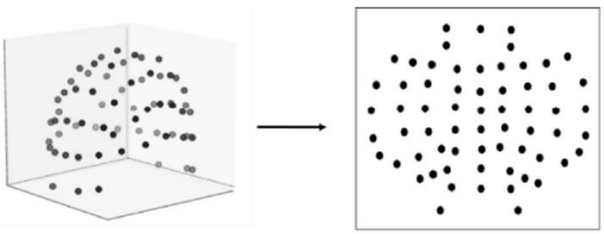

Figure 6: Electrode positions over the realistic head model (left) and their azimuthal equidistant projection onto a 2D plane using Cz as the center (right).

1.6 Interpolation Methods for Scattered Spatial Data

Interpolation is a fundamental technique for estimating values at unsampled locations based on known data points. In EEG and other spatially distributed measurements, electrode positions are typically irregularly spaced, requiring interpolation to generate continuous maps. Several

commonly used interpolation methods exist, including nearest-neighbor interpolation, which assigns the value of the closest known point; linear or barycentric interpolation, which constructs piecewise linear surfaces; polynomial and spline-based interpolation, which fit continuous functions with varying degrees of smoothness; and radial basis function (RBF) or Kriging interpolation, which generate smooth surfaces based on distance or spatial correlation [11–12]. While these methods each have advantages, they also have limitations such as blocky discontinuities, non-smooth gradients, over-smoothing, or computational complexity. One approach that addresses many of these challenges is Clough–Tocher interpolation, a piecewise cubic method on triangular meshes that naturally adapts to the convex shape of the electrode array [11]. The following subsections provide a brief overview of nearest-neighbor and linear interpolation, followed by a detailed description of Clough–Tocher interpolation.

Clough–Tocher Interpolation

Clough–Tocher interpolation is a piecewise cubic interpolation method designed to construct smooth surfaces over scattered data points, particularly on triangular meshes. It is widely used in applications such as EEG scalp topography due to its ability to generate continuous and smooth surfaces from irregularly spaced data. The method begins by performing a Delaunay triangulation of the input points, dividing the domain into non-overlapping triangles. Each triangle is then subdivided into three sub-triangles by connecting the centroid of the triangle to its three vertices. Over each sub-triangle, a cubic polynomial $P(x, y)$ is defined as:

$$P(x,y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + a_6x^3 + a_7x^2y + a_8xy^2 + a_9y^3 \qquad (16)$$

where the coefficients are determined to match the data values at the vertices while ensuring that the first derivatives $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$ are continuous across triangle edges, producing a C^1-continuous surface. The derivatives at the centroids are also chosen to optimize smoothness across the sub-triangles. This approach produces a mathematically smooth surface that avoids artificial sharp edges or discontinuities, making it particularly suitable for spatially continuous data such as EEG measurements. Moreover, because it is defined on triangular meshes, Clough–Tocher interpolation naturally adapts to the convex shape of the electrode array, ensuring that no artificial extrapolation is introduced outside the scalp boundary. When applied to EEG electrode layouts, this method allows discrete electrode measurements to be interpolated into smooth two-dimensional topographic maps that preserve local spatial features, facilitating visualization and further analysis [11].

2 Materials and Methods

2.1 PhysioNet Motor Movement/Imagery Dataset

We utilized the PhysioNet EEG Motor Movement/Imagery Dataset [13], which contains recordings from 109 subjects collected with the BCI2000 system. EEG signals were acquired from 64 electrodes following the 10–10 system, sampled at 160 Hz.

Each subject performed 14 runs:

- 2 baseline runs (one with eyes open and one with eyes closed), each lasting one minute.
- 12 task runs, corresponding to 4 different motor tasks repeated three times, each run lasting two minutes.

The four task types were:

- Real movement: opening and closing the left or right fist depending on the target location (left or right side of the screen).
- Imagined movement: imagining opening and closing the left or right fist.
- Real movement: opening and closing both fists or both feet.
- Imagined movement: imagining opening and closing both fists or both feet.

Each task run contained 15 trials, each lasting approximately 8 seconds and consisting of two distinct epochs: a nearly 4-second Motor Imagery (MI) epoch, followed immediately by a nearly 4-second Rest epoch. Trials were annotated as:

- T0: Rest epoch (no movement or imagery),
- T1: Motor imagery or real epoch corresponding to either left fist or both fists, depending on the run condition.
- T2: Motor imagery or real epoch corresponding to either right fist or both feet.

After every T1 or T2 epoch, a rest epoch (T0) followed.

Our study focuses on classifying five distinct classes derived from this dataset: imagined left fist movement, imagined right fist movement, imagined both fists movement, imagined both feet movement, and Rest.

Following data quality assessment, subjects S043, S088, S089, S092, S100, and S104 were excluded [14] leaving 103 subjects for the analysis.

This setup enables a comprehensive evaluation of the PhysioNet dataset in terms of MI classification.

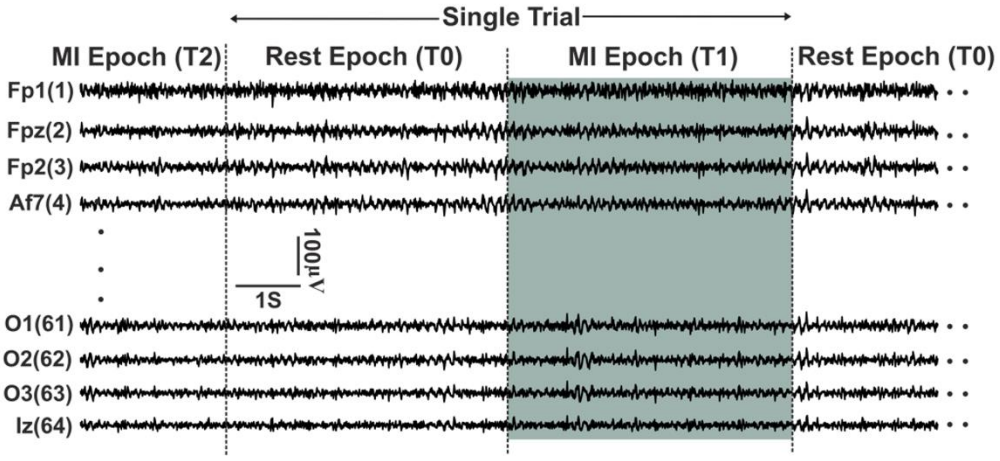

Figure 7: EEG recordings from the PhysioNet Motor Imagery dataset. The vertical axis corresponds to the 64 electrodes arranged according to the international 10–10 system, while the horizontal axis represents time. An illustrative 8-second motor imagery (MI) single trial is shown, consisting of a T0 rest epoch followed by a highlighted T1 MI epoch.

2.2 Motor Imagery EEG Signal to Image Transformation

2.2.1 Chessboard Projection

The EEG data were bandpass filtered using a zero-phase 5th-order Butterworth filter into three frequency bands: Delta (0.5–4 Hz), Mu (8–13 Hz), and Beta (13–30 Hz).

Each trial performed by a subject consisted of two epochs: a rest epoch and a motor imagery epoch, each approximately 4 seconds long. Each epoch was further segmented into 10 segments of 0.4 seconds each.

The power spectral density (PSD) for each electrode was estimated using the Fast Fourier Transform (FFT) over each segment. The squared absolute values of the Fourier coefficients within each frequency band were summed to produce a single band-power value per electrode per frequency band:

$$P_{\text{band}} = \sum_{k=k_{\min}}^{k_{\max}} |X_e[k]|^2 \tag{17}$$

where $X_e(f_k)$ is the FFT of the signal at electrode e.

In this work, we propose the Chessboard EEG signal-to-image transformation to convert motor imagery EEG signals into images. The 3D electrode positions are projected onto a 2D chessboard-like grid, where each electrode occupies a 4 × 4 pixel square. This arrangement preserves the relative spatial organization of the electrodes in a simplified 2D grid, making it suitable for image-based analysis. The color intensity of each square encodes the band-power of the corresponding electrode: green shades for Mu band activity and red shades for Beta band activity, allowing quantitative visualization of neuronal activity.

Each unicolor chessboard has a size of 32 × 32 pixels, representing the activity of 64 electrodes for a single frequency band. By combining the Mu and Beta chessboards, 2-channel images are obtained. For each subject, the total number of images was calculated as 30 tasks (epochs) × 10 segments per task × 6 motor imagery runs, resulting in 1,800 images per subject. Across all 103 subjects, this amounted to 1,800 × 103 = 185,400 images in total. The transformation of EEG signals into chessboard-like images for each subject is illustrated in Figure 11.

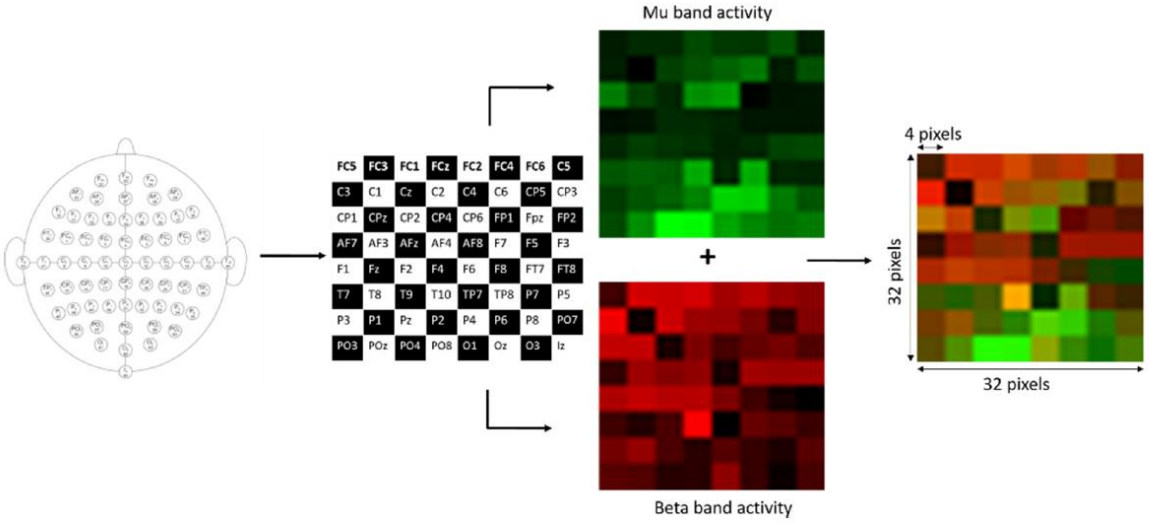

Figure 8: Chessboard EEG signal-to-image transformation method. The 64 electrodes are projected onto a 2D chessboard-like grid, where each electrode occupies a 4 × 4 pixel square, forming a 32 × 32 2-channel image. Mu band activity is represented in green shades and Beta band activity in red shades, with the color intensity of each square reflecting the band-power of the corresponding electrode.

Deep Learning Model

For classifying the Chessboard projection images, we designed a hybrid CNN–LSTM network. The CNN branch, inspired by VGGNet [6], extracts spatial features, while the LSTM branch [9] captures temporal dependencies in the EEG sequences.

The input consists of 32 × 32 RGB images (δ, μ, β) or 32 × 32 × 2 images (μ, β, with the blue channel zeroed), totaling 185,400 images (103 subjects × 1800 images). The CNN begins with three convolutional layers (3×3, 32 filters each) with ReLU activations, followed by max pooling. This is followed by one convolutional layer (3×3, 64 filters, ReLU), max pooling, and two convolutional layers (3×3, 128 filters, ReLU), ending with a final max pooling layer. The resulting feature map is reduced from 2048 to a 512-dimensional vector, which serves as the input to a single-layer LSTM with 128 cells. The LSTM output is connected to a fully connected layer (128 → 5), followed by a softmax activation that outputs the five classes (left fist, right fist, both fists, both feet, rest). Training was performed with a batch size of 16 over 20 epochs, using cross-entropy loss and the Adam optimizer. The overall classification workflow for the Chessboard projection approach is illustrated in Figure 12.

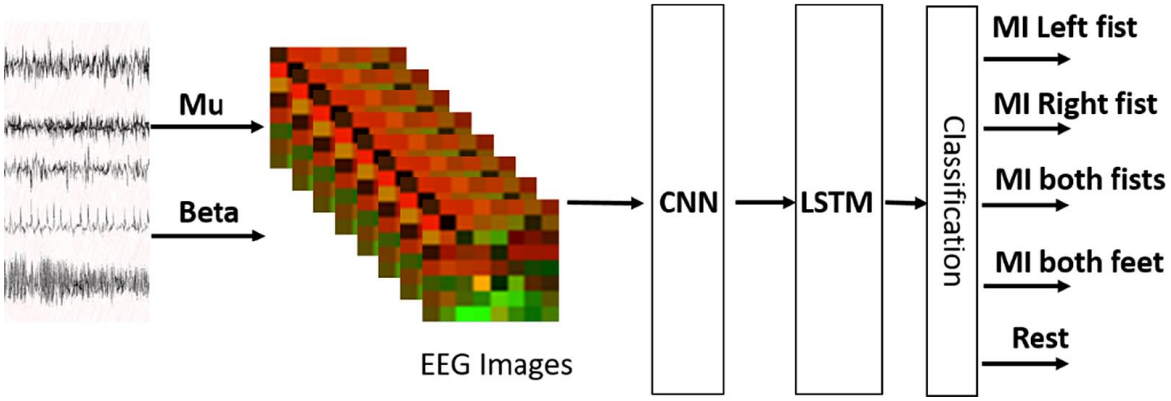

Figure 9: Classification pipeline using the Chessboard projection images representing the Mu and Beta bands. The images are processed by a CNN–LSTM architecture, which outputs five classes corresponding to motor imagery tasks (left fist, right fist, both fists, both feet) and rest.

The network architecture was developed incrementally through an iterative process. Layers were added sequentially, with performance evaluated on an initial subset of 20 subjects due to the high computational cost of LOOCV. Batch Normalization (BN) layers were tested but did not consistently improve classification accuracy, so they were omitted in the final design to maintain simplicity and avoid over-regularization. This process resulted in an efficient architecture that balances model complexity and generalization, minimizing the risk of overfitting associated with excessively deep networks. For comparison, Table 3 summarizes the network architectures, input sizes, and key training parameters for both the Chessboard and Azimuthal projection approaches.

Results

The Leave-One-Out Cross-Validation (LOOCV) method was used for performance evaluation. For each iteration, the 1,800 2-channel images (either chessboard-transformed or azimuthal-projection images) of one subject were used for testing, while the images of another randomly chosen subject were used for validation. The images of the remaining 101 subjects were used as the training set.

The average test classification accuracy using these 2-channel images for five classes (one 'rest' and four motor imagery classes) was 68.72% for the chessboard approach and 68.13% for the azimuthal projection approach. The subject-wise classification accuracy is presented in Figure.13. For some subjects, such as S004 and S098, accuracy was high for both approaches. For others, one approach performed significantly better than the other, as shown in Table 1, Table 2. These results highlight the importance of the input data representation, even when using the same number of channels.

Adding the Delta band (0.5–4 Hz) to form 3-channel images improved the performance of the azimuthal projection approach but did not improve the Chessboard projection results. This further emphasizes that the structure of the input images, not just the number of channels, strongly affects model performance.

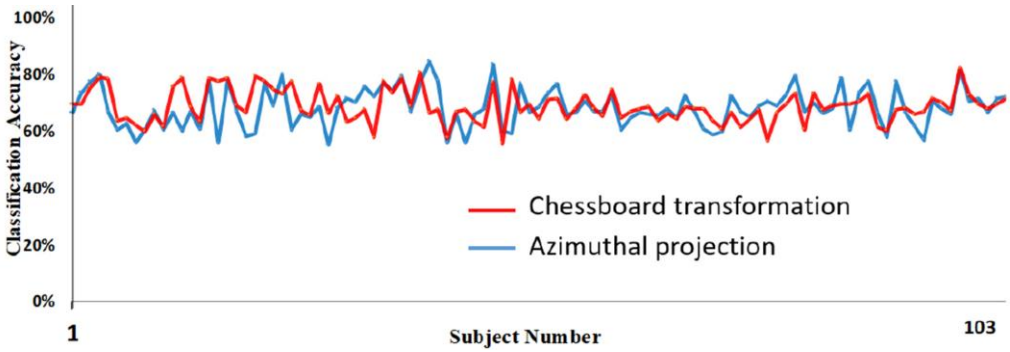

Figure 10: Classification accuracy for the (Mu and Beta) bands for each subject using leave-one-out cross-validation. The red line vertices represent the results obtained with the chessboard projection approach, while the blue line vertices represent the results obtained with the azimuthal projection approach.

The results are better compared with those of a famous baseline method for EEG classification which is Support Vector Machine (SVM) that had 64.64% classification accuracy.

Table 1. The highest performing subjects for Chessboard Projection

Subject Number	Chessboard Projection accuracy	Azimuthal Projection Accuracy
S098	**82.12 %**	81.17 %
S039	**80.60 %**	77.33 %
S021	**79.34 %**	59.13 %
S004	**78.80 %**	80.04 %
S037	**78.65 %**	79.50 %

Table 2. The highest performing subjects for Azimuthal Projection

Subject Number	Chessboard Projection accuracy	Azimuthal Projection Accuracy
S047	77.20 %	**83.50 %**
S098	82.12 %	**81.17 %**
S004	78.80 %	**80.04 %**
S024	73.13 %	**80.02 %**
S080	73.24 %	**79.53 %**

2.2.2 Interpolated Azimuthal Projection

The EEG data were bandpass filtered using a zero-phase 5th-order Butterworth filter into three frequency bands: Delta (0.5–4 Hz), Mu (8–13 Hz), and Beta (13–30 Hz).

Each trial performed by a subject consisted of two epochs: a rest epoch and a motor imagery epoch, each approximately 4 seconds long. Each epoch was further segmented into 10 segments of 0.4 seconds each.

For each segment, the Power Spectral Density (PSD) was estimated using Welch's method with a Hann window of 0.4 seconds. The PSD coefficients were then averaged within each frequency band, yielding a single band-power value per electrode per band.

To project the 3-D electrode coordinates onto a 2-D plane, we employed the Azimuthal Equidistant Projection (AEP), with the plane tangent to the realistic head model at the vertex electrode Cz. This choice preserves the relative spatial distances of the electrodes with respect to Cz when mapping from 3-D to 2-D space. The average PSD values for each of the 64 electrodes within each band were encoded as color intensity specific to the band and then interpolated over a 32×32 mesh using the Clough–Tocher algorithm, producing smooth, unicolor, topology-preserving topomaps for each frequency band. Each segment was subsequently converted into an RGB image by mapping the topomaps to the color channels: red for Delta, green for Mu, and blue for Beta. In addition to this 3-channel configuration, a 2-channel configuration was implemented by encoding Mu and Beta in the green and red channels, respectively, while setting the blue channel to zero. This approach, particularly the generation of 2D interpolated topomaps from raw EEG, was inspired by Bashivan et al. [15], who used these spatial representations to classify mental states. Both configurations produced the same number of images for training. For each subject, the total number of images was calculated as 30 tasks (epochs) × 10 segments per task × 6 motor imagery runs, resulting in 1,800 images per subject. Across all 103 subjects, this amounted to 1,800 × 103 = 185,400 images in total.

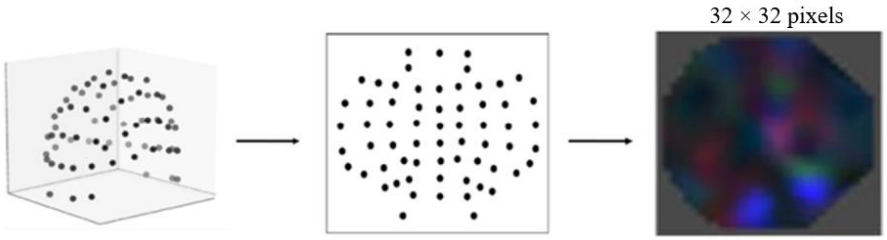

Figure 11: EEG signal-to-image transformation pipeline. Left: 3-D electrode positions on the realistic head model. Middle: 2-D electrode layout obtained using Azimuthal Equidistant Projection (AEP) with Clough–Tocher interpolation over a 32×32 mesh. Right: RGB image created by stacking the Delta (red), Mu (green), and Beta (blue) band topomaps.

Deep Learning Model

For the Azimuthally projected images, a similar CNN–LSTM architecture was employed. For the Azimuthal projection images, the input consisted of 32 × 32 × 3 RGB images (δ, μ, β) or 32 × 32 × 2 images (μ, β), totaling 185,400 images (103 subjects × 1,800 images). The CNN branch included three convolutional layers with 64 filters (3×3) each followed by ReLU activations and max pooling, two convolutional layers with 128 filters (3×3) followed by ReLU

and max pooling, and two convolutional layers with 256 filters (3×3) with ReLU and a final max pooling. The CNN output was flattened from 4,096 to a 512-dimensional vector, which served as input to a single LSTM layer with 256 cells. The LSTM output was connected to a fully connected layer mapping 256 units to 5 outputs, followed by a softmax activation for classification into the five motor imagery classes. Training was performed with a batch size of 16 over 20 epochs, using cross-entropy loss and the SGD optimizer. The network architecture was designed incrementally following the same logic as the Chessboard approach, with layers added sequentially and evaluated on a subset of subjects to efficiently balance model complexity and generalization. The overall classification workflow for the Azimuthal projection approach is illustrated in Figure. 15.

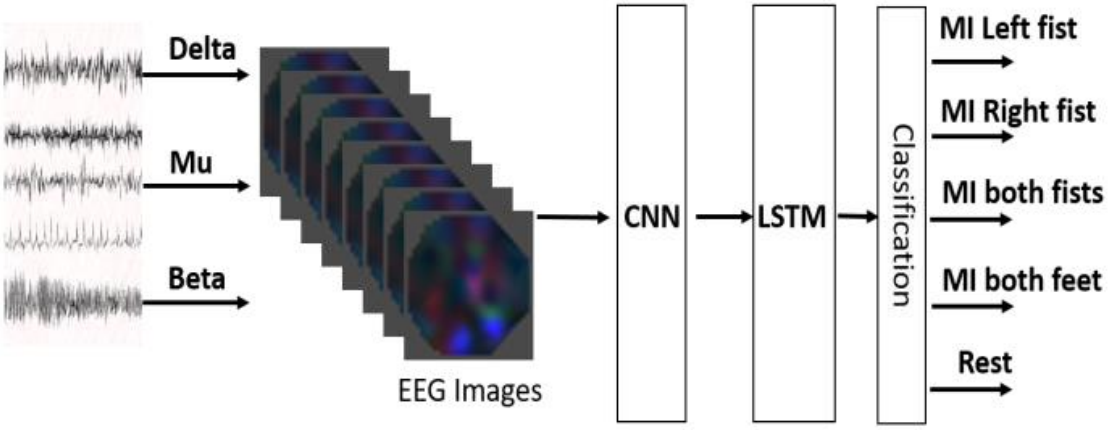

Figure 12: Classification pipeline using the Azimuthal projection images representing the Delta, Mu, and Beta bands. The images are processed by a CNN–LSTM architecture, which outputs five classes corresponding to motor imagery tasks (left fist, right fist, both fists, both feet) and rest.

For comparison, Table 3 summarizes the network architectures, input sizes, and key training parameters for both the Chessboard and Azimuthal projection approaches.

Table 3. Comparison of the Chessboard and Azimuthal projection network architectures. The table summarizes the input size, number of images, convolutional and LSTM layers, activation and pooling strategies, optimizer, loss function, batch size, and number of training epochs for both approaches.

Layer/Parameter	Chessboard Projection	Azimuthal Projection
Input Shape	32 × 32 × 3 RGB (δ, μ, β) or 32 × 32 × 2 (μ, β, Blue channel zeroed)	32 × 32 × 3 RGB (δ, μ, β) or 32 × 32 × 2 (μ, β)
Input Size	185,400 images (103 subjects x 1800 images)	185,400 images (103 subjects x 1800 images)
Convolutional Layers /Max Pooling	Conv (3×3, 32) → ReLU Conv (3×3, 32) → ReLU Conv (3×3, 32) → ReLU Max Pooling Conv (3×3, 64) → ReLU Max Pooling Conv (3×3, 128) → ReLU Conv (3×3, 128) → ReLU Max pooling	Conv (3×3, 64) → ReLU Conv (3×3, 64) → ReLU Conv (3×3, 64) → ReLU Max Pooling Conv (3×3, 128) → ReLU Conv (3×3, 128) → ReLU Max Pooling Conv (3×3, 256) → ReLU Conv (3×3, 256) → ReLU Max pooling

Flatten	2048 to 512	4096 to 512
LSTM	(input = 512) 1 layer, 128 cells	(input = 512) 1 layer, 256 cells
Fully Connected	128 to 5	256 to 5
Output Layer	Softmax (**5 classes**)	Softmax (**5 classes**)
Training Parameters	Batch size: 16, Epochs: 20, Loss: Cross-Entropy, Optimizer: Adam	Batch size: 16, Epochs: 20, Loss function: Cross-Entropy, Optimizer: SGD

Results

We used the Leave-One-Out-Cross-Validation (LOOCV) for evaluating the performance of the model. All the 180 trials for one subject was used for testing and the trials for another randomly chosen subject was used for validation and the rest 101 subjects trials were used as the training set.

The average test accuracy was 70.64% (using 3 bands scenario) and 68.13% (using 2 bands scenario). Figure.16 shows the classification accuracy for both scenarios with the subject number.

The results are promising for 5-class motor imagery EEG classification and can be compared with the state-of-the-art methods. We compared our results with a famous baseline

method for EEG classification which is Support Vector Machine (SVM) and our results showed 5% better classification accuracy.

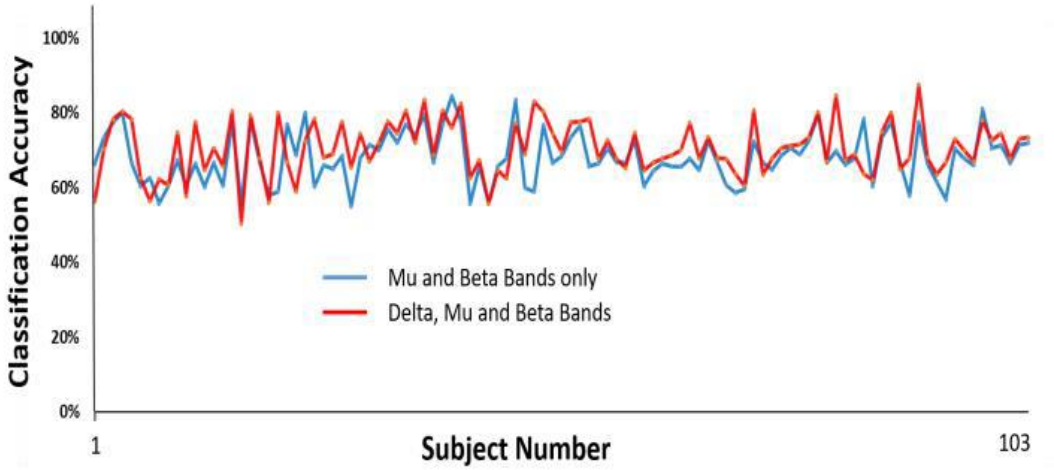

Figure 13: Classification accuracy across subjects using leave-one-out cross-validation with the Azimuthal projection approach. The red line vertices show the results for the 3-channel configuration (Delta, Mu, and Beta), while the blue line vertices show the results for the 2-channel configuration (Mu and Beta only).

The classification performance of different models on the PhysioNet motor imagery EEG dataset is summarized in Table 4. This table compares the input features, average accuracy, and evaluation protocols for CNN–LSTM architectures using topomap images (both Chessboard and Azimuthal projections), as well as conventional classifiers such as RBF-SVM, EEGNet, and Deep ConvNet.

Table 4. classification accuracy for motor imagery EEG on PhysioNet Dataset.

Model	Input Feature	Average Accuracy	Description
CNN + LSTM	3 Bands Topompas (Azimuthal)	70.64	103 subjects, Leave-One-Subject-Out Cross Validation (LH, RH, BH, BF, Rest)
	2 Bands Topompas (Azimuthal)	68.13	
	3 Bands Topompas (Chessboard)	68.81	-
	2 Bands Topompas (Chessboard)	68.72	-
(RBF)SVM	3 Band PSD Features	64.64	103 subjects, 5-fold cross validation (bad epochs dropped) (LH, RH, BH, BF, Rest) *
	2 Band PSD Features	60.48	
Accurate EEGNet [16]	Raw EEG, 3 s epochs	65.07	105 Subjects, Global 5-fold cross-validation (LH, RH, BH, BF)
I D ConvNet [17]	Raw EEG, 3 s epochs	58.58	109 Subjects, Global 5-fold cross-validation (LH, RH, BF, Rest)

* LH: Left Hand (Fist), RH: Right Hand, BH: Both Hands, BF: Both Feet, Rest: Rest condition

2.2.3 Discussion

While Mu (8–13 Hz) and Beta (13–30 Hz) rhythms dominate most motor imagery (MI) studies due to strong ERD/ERS effects, our results revealed a consistent improvement when Delta (0.5–4 Hz) features were included — but only under the azimuthal projection. This finding motivated a closer examination of slow cortical potentials and their interaction with spatial representation in cue-based MI decoding.

1. *Neurophysiological Basis of Delta Activity*: Delta oscillations are classically associated with deep non-REM sleep and slow-wave brain states. Beyond sleep, in awake, task-engaged participants, Delta rhythms encode attention, expectation, and motor preparatory states, supporting cortical coordination and the buildup of task-relevant activity. This provides a physiological rationale for the relevance of Delta in MI decoding.

2. *Capturing Slow Cortical Dynamics*: Although a high-pass filter at 0.5 Hz suppresses artifacts, it does not eliminate all slow cortical processes. Very slow drifts may still influence the lowest frequency bins. When spectral features are extracted in sliding windows (0.4 s in our study), extremely slow changes (<0.5 Hz) are not resolved within a single window but appear as gradual changes in Delta-band power across consecutive windows. Feeding these sequences into a unidirectional LSTM enables the network to capture slow trends, effectively modeling readiness- and anticipation-like dynamics.

3. *Readiness Potentials (RP) and CNV in Predictable Cue-Based MI*: Delta-band activity reflects slow cortical potentials such as the Readiness Potential (RP) and Contingent Negative Variation (CNV). In self-paced movement, the RP begins ~1–2 s before action (early RP) and steepens ~0.5 s before onset (late RP). In cue-based MI:

- The early RP is largely diminished, but in predictable repetitive paradigms (4 s MI / 4 s rest), participants anticipate the upcoming MI epoch. A small anticipatory RP may appear during the last ~1 s of the rest period.
- The late RP emerges during the imagery epoch, peaking ~0.5–1 s after cue onset.
- The CNV may begin near the end of the rest period and continues across the 4 s imagery epoch, reflecting sustained attention and preparation.

Even with high-pass filtering over 0.5 Hz, these effects are indirectly reflected in the temporal evolution of Delta-band power across consecutive windows. Since classification is based on 10 sliding windows per epoch with the final decision at the last segment, the LSTM can leverage cumulative Delta activity spanning anticipatory and imagery periods.

4. *Interaction with Mu and Beta Rhythms*: Slow cortical potentials are not independent of faster rhythms. *Cross-frequency coupling (CFC)* mechanisms, such as phase–amplitude modulation, allow Delta-phase dynamics to shape the amplitude and timing of Mu and Beta ERD/ERS. By including Delta, the CNN–LSTM captures both the direct contribution of slow drifts and their modulatory influence on sensorimotor rhythms, enhancing class discriminability.

5. *RGB Fusion of Frequency Bands*: Delta, Mu, and Beta are encoded as three channels of RGB topomaps. CNN filters can learn within-band spatial patterns as well as cross-band correlations, integrating slow cortical drifts with localized ERD/ERS patterns. This enables the network to capture cross-frequency interactions rather than treating each rhythm independently.

6. *Effect of Spatial Representation: Azimuthal Projection vs. Chessboard*: The benefit of Delta depends on how electrode activity is spatially represented:

- Azimuthal Projection with Clough–Tocher interpolation: Electrodes are projected onto a 2-D plane, generating smooth 32×32 maps. This preserves global spatial gradients and subtle low-frequency variations, making it ideal for identifying Delta-related RP/CNV drifts while retaining localized Mu/Beta ERD/ERS.
- Chessboard Layout: Electrodes are mapped discretely to a 4×4 grid, yielding block-like 32×32 images. While CNN filters can extract localized Mu/Beta features, smooth, widespread Delta patterns are poorly represented, limiting the capture of slow modulatory effects.

7. *Spatio-Temporal Modeling with CNN–LSTM*: Our compact VGGNet-inspired CNN extracts spatial features from topomaps. The LSTM models temporal dependencies across 10 windows per 4 s epoch, capturing:

1. Spatial topographies: readiness-like Delta distributions and localized Mu/Beta ERD/ERS.
2. Temporal dynamics: gradual buildup during the end of rest and imagery, reflecting late RP and CNV.

3. Epoch-level integration: final classification at the last segment leverages cumulative slow and fast dynamics.

Together, this hybrid CNN–LSTM framework explains why including Delta features significantly improved classification under the azimuthal projection while offering minimal gains with the chessboard layout. However, these conclusions require further validation.

3 Summary

3.1 Scientific Results

Thesis Group I: Signal to Image Transformation

We propose and evaluate two end-to-end EEG signal-to-image transformation methods—Chessboard Projection and Interpolated Azimuthal Projection—that convert spectral bandpower dynamics into spatially organized 2D representations optimized for deep learning. These transformations encode neurophysiologically relevant frequency bands (Mu, Beta, and optionally Delta) into color channels, allowing the first branch of the model (CNN) to capture spatial patterns of cortical activity during motor imagery tasks. Together, they establish a unified image-based paradigm for EEG classification.

Thesis Ia: The Chessboard Projection — A Discrete, Grid-Optimized EEG-to-Image Transformation

We introduce the Chessboard Projection: a novel signal-to-image transformation that maps 64 EEG electrodes onto a rigid 32×32 pixel grid, where each electrode occupies a fixed 4×4 pixel block. Band power for Delta (0.5–4 Hz), Mu (8–13 Hz), and Beta (13–30 Hz) — estimated via Welch's PSD after 5th-order Butterworth filtering — is encoded into image channels, yielding either 3-channel maps (red for Delta, green for Mu, blue for Beta) or 2-channel maps (red for Mu, green for Beta, with the blue channel set to zero). This approach prioritizes structural regularity, computational simplicity, and alignment with convolutional receptive fields.

Thesis Ib: The Interpolated Azimuthal Projection — A Continuous, Neuroanatomically informed EEG-to-Image Transformation

We deployed the Interpolated Azimuthal Equidistant Projection: a signal-to-image transformation that projects 3D electrode positions onto a 2D plane tangent at Cz using distance-preserving geodesic mapping, followed by Clough–Tocher interpolation over a 32×32 mesh. Band power for Delta (0.5–4 Hz), Mu (8–13 Hz), and Beta (13–30 Hz) — computed via FFT after 5th-order Butterworth filtering — is encoded as 3-channel maps (red for Delta, green for Mu, blue for Beta) or 2-channel maps (red for Mu, green for Beta, blue channel set to zero). This produces smooth, continuous topomap images that preserve cortical topology and spatial gradients for CNN feature extraction.

Thesis Group II: End-to-End Spatiotemporal Learning with Hybrid CNN-LSTM

We designed a hybrid CNN–LSTM architecture tailored to the spatial properties of Chessboard and Azimuthal projections, enabling joint learning of cortical topographies and their temporal dynamics. Notably, we show that the Delta band, traditionally dismissed in motor imagery,

improves accuracy only when spatially preserved through Azimuthal encoding, underscoring spatial representation as a decisive factor in EEG decoding.

Thesis IIa: The Architecture Co-Designed with Input Representation

Rather than applying a fixed deep learning template, we designed the CNN-LSTM architecture in response to the structure of the input images. The CNN branch was selected to capture spatial patterns of band power distribution across the 32×32 topographic grid, whether discrete (Chessboard) or smooth (Azimuthal), leveraging its inherent strength in learning local, translation-invariant features from image-like data. The LSTM branch was introduced to model the temporal evolution of these spatial patterns across the 10 segments (0.4s each) within a motor imagery epoch — crucial for capturing the dynamic nature of cortical activation. For the discrete, grid-aligned Chessboard projections, we employed a shallower, narrower CNN to avoid overfitting and match the input's structural simplicity. For the continuous, gradient-rich Azimuthal topomaps, we scaled depth and capacity to exploit their anatomical fidelity and spatial smoothness

Thesis IIb: The Delta Band Effect

Despite identical segmentation, filtering, temporal modeling, and architectural procedure, adding the Delta band (0.5–4 Hz) yielded a significant +2.5% accuracy gain — but only under Azimuthal encoding (70.64% vs. 68.13%), with no measurable improvement under Chessboard. This reveals that low-frequency cortical dynamics, long dismissed as non-informative in motor imagery, become decodable when spatially preserved through continuous, anatomy-aware interpolation. Importantly, such Delta activity overlaps with established slow cortical potentials, including the Readiness Potential (RP) and the Contingent Negative Variation (CNV), both of which reflect preparatory and anticipatory processes relevant to motor control. This result challenges the widespread assumption that low-frequency oscillations are irrelevant to motor imagery classification and suggests that even 'non-canonical' spectral components may carry latent, task-discriminative information. Future systems should treat spatial encoding not as a neutral preprocessing step, but as a decisive interface that gates what the brain reveals to the machine.

Use of AI Assistance

During the preparation of this dissertation, I used multiple generative AI tools, including the most recent versions available as of August 31, 2025, such as Grok, ChatGPT, Perplexity AI, DeepSeek, and Qwen, to assist with text generation and improve readability. All aspects of research design, analysis, and critical arguments are entirely my own. I have reviewed and edited the outputs where appropriate and take full responsibility for the content.

Publications

[1] W. Fadel, Cs. Kollod, M. Wahdow, Y. Ibrahim, and I. Ulbert, "*Multi-Class Classification of Motor Imagery EEG Signals Using Image-Based Deep Recurrent Convolutional Neural Network*", in 2020 8th International Winter Conference on Brain-Computer Interface (BCI), 2020, pp. 1–4. doi: 10.109/BCI48061.2020.061622.

[2] W. Fadel, M. Wahdow, Cs. Kollod, G. Marton, and I. Ulbert, "*Chessboard EEG Images Classification for BCI Systems Using Deep Neural Network*", in Bioinspired Information and Communication Technologies, Y. Chen, T. Nakano, L. Lin, M. U. Mahfuz, and W. Guo, Eds., Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering ser., vol. 329, Cham: Springer International Publishing, 2020, pp. 97–104. doi: 10.1007/978-3-030-57115-3_8.

[3] A. Adolf, Cs. Köllőd, G. Márton, W. Fadel, and I. Ulbert, *"The Effect of Processing Techniques on the Classification Accuracy of Brain-Computer Interface Systems,"* Brain Sciences, vol. 14, no. 12, p. 1272, 2024.

[4] Cs. Köllőd, A. Adolf, G. Márton, M. Wahdow, W. Fadel, and I. Ulbert, "*Closed loop BCI system for Cybathlon 2020*", Brain-Computer Interfaces, 10th vol., 2nd no., pp. 114–128, 2023. doi: 10.1080/2326263X.2023.2254463.

[5] Cs. Köllőd, A. Adolf, G. Márton, M. Wahdow, W. Fadel, and I. Ulbert, TTK dataset - 4 class Motor-Imagery EEG, 2022, https://repo.researchdata.hu/dataset.xhtml?persistentId=hdl:21.15109/CONCORDA/UOQQVK.

[6] M.Wahdow, M. Alnaanah, W. Fadel, A. Adolf, Cs. Kollod, and I. Ulbert, "*Multi frequency band fusion method for EEG signal classification*", Signal, Image and Video Processing, 2022. doi: 10.1007/s11760-022-02399-6.

[7] W. Fadel, R. Tóth, R. Fiáth, I. Ulbert, and Z. Somogyvári, "*Optimal inter-electrode distances for spike sorting in different brain regions*", (Poster, HunDoc 2024, Hungary)

[8] W. Fadel, I. Ulbert, "*Convolutional Neural Network Classifier for BCI systems using Motor Imagery EEG Signals-to-Image Transformation*" (Poster for CYBATHLON Symposium 2020, SWISS Arena, Switzerland).

Bibliography

[1] Epomedicine, “Brodmann Areas and Lesions,” Epomedicine, Jul. 31, 2016. [Online]. Available: https://epomedicine.com/medical-students/brodmann-areas-lesions/.

[2] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, Jan. 1962.

[3] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 7–9, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 770–778.

[8] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[10] D. F. Watson, *Contouring: A Guide to the Analysis and Display of Spatial Data*. Oxford, U.K.: Pergamon, 1992.

[11] R. Franke, "Smooth interpolation of scattered data by local polynomials," *ACM Trans. Math. Softw.*, vol. 8, no. 2, pp. 119–143, Jun. 1982.

[12] N. Cressie, *Statistics for Spatial Data*. New York, NY, USA: Wiley, 1993.

[13] J. Schalk et al., “BCI2000: A general-purpose brain-computer interface (BCI) system,” IEEE Trans. Biomed. Eng., vol. 51, no. 6, pp. 1034–1043, 2004.

[14] A. Loboda, A. Margineanu, G. Rotariu, and A. M. Lazar, "Discrimination of EEG-based motor imagery tasks by means of a simple phase information method," Int. J. Adv. Res. Artif. Intell., vol. 3, no. 10, 2014.

[15] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, “Learning representations from EEG with deep recurrent convolutional neural networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2016.

[16] X. Wang, Y. Yang, and Y. Zhang, “Accurate EEGNet-based motor imagery classification with minimal preprocessing,” *Journal of Neural Engineering*, vol. 20, no. 4, p. 046012, 2023.

[17] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, “An end-to-end deep learning approach to MI-EEG signal classification for BCI,” *Frontiers in Neuroscience*, vol. 12, p. 711, 2018.