

Article

Enhancing Motor Imagery Classification in Brain–Computer Interfaces Using Deep Learning and Continuous Wavelet Transform

Yu Xie ¹  and Stefan Oniga ^{1,2,*} ¹ Faculty of Informatics, University of Debrecen, 4032 Debrecen, Hungary; yu.xie@inf.unideb.hu² North University Center of Baia Mare, Technical University of Cluj-Napoca, 400114 Cluj-Napoca, Romania

* Correspondence: stefan.oniga@ieec.utcluj.ro

Abstract: In brain–computer interface (BCI) systems, motor imagery (MI) electroencephalogram (EEG) is widely used to interpret the human brain. However, MI classification is challenging due to weak signals and a lack of high-quality data. While deep learning (DL) methods have shown significant success in pattern recognition, their application to MI-based BCI systems remains limited. To address these challenges, we propose a novel deep learning algorithm that leverages EEG signal features through a two-branch parallel convolutional neural network (CNN). Our approach incorporates different input signals, such as continuous wavelet transform, short-time Fourier transform, and common spatial patterns, and employs various classifiers, including support vector machines and decision trees, to enhance system performance. We evaluate our algorithm using the BCI Competition IV dataset 2B, comparing it with other state-of-the-art methods. Our results demonstrate that the proposed method excels in classification accuracy, offering improvements for MI-based BCI systems.

Keywords: EEG signal analysis; continuous wavelet transform; convolutional neural networks; feature extraction



Citation: Xie, Y.; Oniga, S. Enhancing Motor Imagery Classification in Brain–Computer Interfaces Using Deep Learning and Continuous Wavelet Transform. *Appl. Sci.* **2024**, *14*, 8828. <https://doi.org/10.3390/app14198828>

Academic Editors: Andoni Carrasco-Uribarren, Xavier Marimon, Portela Alejandro, Oliver Millan and Lucia Billeci

Received: 23 July 2024

Revised: 11 September 2024

Accepted: 27 September 2024

Published: 1 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Electroencephalography (EEG), a technique used to monitor brain activity via electrodes attached to the scalp [1], has become an indispensable tool in the field of neuroscience. EEG is particularly useful in brain–computer interface (BCI) research, where it is capable of detecting electrical signals from the brain, including those related to imaginative movement. Moreover, its utility extends to the evaluation of brain function, rendering invaluable insights for the diagnosis and treatment of various neurological disorders. However, the manual analysis of EEG data remains a laborious and intricate task, fraught with complexities. In recent years, the advent of Convolutional Neural Networks (CNNs) has revolutionized EEG analysis [2], providing researchers with a powerful tool for feature extraction without the need to understand the complexities of EEG. CNNs offer an end-to-end learning capability that helps reduce the blindness and waste of time associated with manually searching for suitable features. Notably, EEG signals exhibit diverse features in both the time and frequency domains. While some researchers focus solely on either time or frequency domain signals as CNN inputs, potentially overlooking crucial information, others seek to integrate various features to enhance classification performance. Continuous Wavelet Transform (CWT), a widely employed technique for time–frequency signal analysis, presents a promising approach for representing EEG signals as time–frequency images [3]. By mapping EEG data onto a time–frequency plane, CWT enables the visualization of signal characteristics that may be imperceptible in the time or frequency domains alone. In our study, we propose a novel multi-input time–frequency CNN model that combines CWT with CNN, leveraging the complementary strengths of both approaches.

Since the inception of BCI competitions in 2001, these events have served as invaluable platforms for benchmarking algorithm performance and fostering collaboration among

researchers. The work of [4] combines CEMD preprocessing, ERS/ERD encoding, and a 1D-MS-CNN model for effective classification of EEG signals related to MI tasks. The authors of [5] propose a Covariate Shift Estimation and Unsupervised Adaptive Ensemble Learning (CSE-UAEL) method to address non-stationarity in EEG-based BCI systems. It integrates covariate shift detection with dynamic ensemble updating to adapt to changing input data distributions. Being active scheme-based, it adds classifiers based on detected shifts, enhancing MI classification. The work of [6] integrates features from multiple domains using sparse representation (SR) to enhance classification capability. Initial features including time–frequency energy, maximum entropy power spectral estimation, and Hjorth parameters are extracted, then fused via SR to obtain discriminative low-dimensional features. The authors of [7] proposed a time-scale CNN method for EEG MI classification. They decomposed the raw signal into signals in different frequency bands, namely 4–7 Hz, 8–13 Hz, and 13–32 Hz. Then, they input three small CNNs with the same parameters, and concatenated all of the results in the fully connected layer. The authors of [8] proposed the idea of mapping EEG signals using CWT as the input and used a layer of CNN with a convolution kernel of size 30×30 to verify it on public data sets. The authors of [9] designed two CNN branch networks (TBTF-CNN) to extract the original CNN signal and CWT signal, respectively. The obtained feature images were then connected to form a new feature map. Like the previously mentioned article, we used the 2008 BCI Competition IV Database 2B [10] data set to validate our methodology and compared our results with theirs. Some studies utilize deep machine learning algorithms, increasing the number of network layers to improve accuracy. Others employ complex data augmentation techniques, such as FastGAN [11], to enhance the network’s generalization capabilities. These approaches often obtain good results by assuming unlimited computational resources. This paper aims to design a lightweight, hardware-friendly CNN architecture that can be applied to resource-constrained devices like small FPGAs for real-time BCI. Given the constraints of resource-constrained devices, we adopted a shallow CNN architecture. Compared with other models using parallel CNN [7,12,13], we replaced the raw signal input with extracted feature images and separated the alpha and beta rhythms. Additionally, different preprocessing methods and classifiers were explored to further analyze their impact on the branch CNN structure. Finally, our results will be compared and analyzed with other state-of-the-art models.

2. Methods

2.1. Preprocessing

2.1.1. Common Spatial Patterns

Common Spatial Patterns (CSPs) are derived from the common spatial subspace decomposition (CSSD) algorithm, which is a spatial filtering method for the analysis of multichannel EEG data. The main goal of CSP is to find spatial filters that maximize the variance of EEG signals in one category and minimize the variance of EEG signals in another category. This is achieved by calculating the covariance matrices of EEG signals of different categories and then applying eigenvalue decomposition to find the spatial filter. The resulting filter is called the CSP and is used to transform the original EEG signal into a new eigenspace where the classes are more easily separable [14]. The CSP is shown as follows:

$$W = \frac{\omega^T X_1^T X_1 \omega}{\omega^T X_2^T X_2 \omega} = \frac{\omega^T M_1 \omega}{\omega^T M_2 \omega} \quad (1)$$

where W is the spatial filter matrix, T is the transpose of the matrix, X represents the data matrix of the class, and M_1 and M_2 are the covariance matrices of the two classes of EEG signals, respectively.

CSP does not require the selection of a specific frequency band. Therefore, it is suitable for all types of EEG signals and provides a feature vector that captures signal components relevant to the task, while reducing noise and irrelevant components. CSP is sensitive to

noise, but EEG signals typically have low signal-to-noise ratios. CSP typically requires EEG recordings from multiple electrodes for an effective spatial filter, which may cause it to have a low performance on signals with only a few channels.

2.1.2. Fast Fourier Transform

Fast Fourier Transform (FFT) is commonly used for calculating the Power Spectral Density (PSD), which is the power distribution of a signal in the frequency domain. The calculation process is as follows: For a discrete-time signal $x[n]$, its FFT is denoted as $X[k]$.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}} \quad (2)$$

The power spectrum $P[k]$ is usually calculated as the amplitude squared of the FFT result:

$$P[k] = |X[k]|^2 \quad (3)$$

PSD is the normalized form of the power spectrum, which is usually normalized taking into account the signal length N and the sampling frequency f_s .

$$\text{PSD}(f) = \frac{1}{Nf_s} |X[k]|^2 \quad (4)$$

The EEG signal itself contains features in both time and frequency domains. However, it cannot effectively solve the problem of information correspondence in these two domains. While some researchers focus only on time- or frequency-domain signals, which may overlook important information, others work on combining various features to improve classification performance. For this limitation, the FFT was extended to various time–frequency domain analysis methods, such as the Short-Time Fourier Transform (STFT) and the CWT, which were used in subsequent studies.

2.1.3. Short-Time Fourier Transform

In 1946, Gabor Dennis first proposed STFT, which introduced the concept of using a time window to identify the frequency information at a specific moment, effectively solving the problem of information localization in the time–frequency domain. STFT is expressed as follows:

$$\text{STFT}\{x(t)\} = X(m, \omega) = \int_{-\infty}^{\infty} x(t) w(t - m) e^{-j\omega t} dt \quad (5)$$

where $x(t)$ represents the EEG signal, $w(t)$ represents a temporal window, and $w(t - m)$ represents the complex conjugate of $w(t)$. The size of the time window determines the relationship between the frequency and time domains. A shorter time window improves the time resolution, while a narrower frequency window improves the frequency resolution [15].

2.1.4. Continuous Wavelet Transforms

CWT stands as widely adopted approach for analyzing signals in the time and frequency domain [3,16,17]. Initially introduced by Morlet and Grossman in 1987, CWT involves the dissection of a signal into wavelets across various components. This decomposition enables the examination of diverse frequency scales embedded within the data. Unlike traditional frequency analysis methods, the CWT affords a comprehensive analysis. Consequently, it provides researchers with a time–frequency representation of data, facilitating a deeper comprehension of their intricacies and temporal dynamics. Equation (6) presents the formulation for the CWT.

$$w_s(a, \tau) = a^{\frac{1}{2}} \int s(t) \phi\left(\frac{t - \tau}{a}\right) dt \quad (6)$$

where the scaling of the wavelet transform is denoted as a , while the input signal is represented as $s(t)$, the wavelet basis function is denoted as ϕ , and the time offset is denoted as τ .

Among the array of five wavelet basis functions commonly employed in contemporary signal processing, the Morlet wavelet holds a prominent position [3]. We have selected the Morlet wavelet as the basis wavelet for our study. Its expression in the time domain is given in Equation (7).

$$\phi(t) = \left(\frac{2}{\pi T^2}\right)^{\frac{1}{4}} \exp\left(-\frac{t^2}{T^2} + jw_c t\right) \quad (7)$$

The expression of frequency is shown in Equation (8).

$$\phi(w) = \left(\frac{T^2}{2\pi}\right)^{\frac{1}{4}} \exp\left(-\frac{(w - w_c)^2}{4T^2}\right) \quad (8)$$

From the above equation, T and w_c are determined. The resulting time–frequency domain image, generated from the CWT mapping process, is employed as the input for our model.

2.2. Convolutional Neural Networks

CNN, a pivotal component of machine learning, remains at the forefront of emerging fields that continue to push the boundaries of innovation. As a pioneering force in numerous areas, it has garnered significant attention from scholars and researchers alike [17–19]. Simultaneously, significant advancements have also been made in the research and development of BCI systems [20,21]. These parallel efforts reflect the increasing recognition of the potential of deep learning and BCI systems to revolutionize and transform numerous domains of human activity.

Within the core of CNN lies its indispensable convolutional layer, pivotal for executing convolutional operations on input signals. This process necessitates the utilization of convolutional kernels, commonly known as filters. A single convolutional layer has the capacity to accommodate multiple filters, with their respective weight parameters and biases subject to adjustment throughout the neural network's training phase. Through the application of matrix multiplication principles, convolutional operations yield feature mappings crucial for facilitating the transition from input to output. The positional coordinates of neural elements within the feature map, generated by the k th convolution kernel, are denoted as (m, n) . The resulting output is represented in Equation (9).

$$y_{m,n} = f(w(i)_k * I(m, n) + b) \quad (9)$$

where $I(m, n)$ represents the input data, b stands for the bias, and $w(i)_k$ is denoted as the k th convolution kernel of the i th layer. The activation function, denoted as f , plays a crucial role in neural networks. Commonly used activation functions include the hyperbolic tangent (tanh), sigmoid, and rectified linear unit (ReLU). Beyond merely performing convolution operations, the convolutional layer of a neural network encompasses additional intricate processes such as padding and stride. These operations significantly enhance the complexity of the computational process associated with the convolutional layer.

The pooling layer, also referred to as the sampling layer, plays a critical role that reduces the dimensions of the input images. Positioned after the convolutional layer, this layer effectively performs downsampling operations to extract local features. By decreasing the number of network parameters, the pooling layer reduces the computational complexity of the model. Additionally, it has been shown to enhance the model's overall robustness by minimizing minor errors in the data and combating overfitting issues. Two types of pooling operations are usually performed: maximum pooling, in which the largest element

value determines the feature value of the target region within the region. Conversely, average pooling calculates the feature value of the target region based on the average value of its constituent elements. The mathematical expressions for two pooling are delineated in Equations (10) and (11), respectively.

$$f(x) = \max(x_{[m,m+N],[n,n+N]}) \quad (10)$$

$$f(x) = \frac{1}{N * N} \sum_{m=m_1, n=n_1}^{m=m_1+N, n=n_1+N} X_{m,n} \quad (11)$$

where $(N * N)$ is the size of the pooling kernel.

In a traditional CNN architecture, the fully connected layer is commonly considered the last component. It receives features from upper layers, which are converted to vector format before passing through. Here, previously extracted features are merged for matrix multiplication. The fully connected process plays a critical role in transmuting the high-dimensional spatial features received from upper layers, thereby using non-linear mapping finalizing the entire CNN architecture. The robustness and generalization capabilities of the CNN network models owe much to design principles. These principles enable CNN models to automatically learn intrinsic signal features through convolutional and related operations, ensuring efficient processing and extraction of meaningful information from input data.

2.3. Classifier

Support Vector Machine (SVM) methods are based on the principles of statistical VC dimensionality and structural risk minimization. They are widely used for classification (binary/multiclassification), regression, and outlier detection [22]. SVMs are robust to unknown data and often outperform other traditional machine learning algorithms, especially when dealing with small datasets. The main goal of SVMs is to compute optimal decision boundaries or hyperplanes to distinguish different classes of data in the feature space. SVMs can be categorized into three types [23]: Linear Support Vector Machines for linearly differentiable data (hard margin maximization), Linear Support Vector Machines for non-linearly divisible data (soft margin maximization), and Nonlinear Support Vector Machines for non-linearly divisible data (soft margin maximization).

SoftMax is a classifier that is used as a multi-classification task and is often located in the last layer of the neural network architecture. It transforms a set of raw scores, also known as logits, into probabilities for each class [24]. Given a vector of logits $X = [x_1, x_2, \dots, x_n]$, the softmax function is defined as follows:

$$f(x)_i = \frac{e^{x_i}}{\sum_n^j e^{x_j}} \quad (12)$$

where $f(x)_i$ represents the probability of the i -th class.

Decision trees (DTs) are widely used for classification and regression tasks. They utilize a recursive algorithm to split the feature spaces and arrange them in a tree-like structure. Each node in the DT represents a decision, and each leaf node represents an outcome or prediction [25]. The concept of DT was first introduced by J. Ross Quinlan and has undergone several optimizations over various versions [26]. This algorithm breaks down tasks into subtasks from root to leaf, calculating the optimal result for each subtask without necessarily achieving the best overall result [27].

2.4. Proposed CNN Structure

To enhance the precision of EEG signal classification for MI, the authors of [7] designed a parallel CNN structure. Specifically, they combined three separate frequency bands as the input and constructed a composite convolutional scale by allocating each frequency band to a corresponding convolutional layer with three varying scales of the convolutional kernel. This method improves the performance of feature extraction from high-dimension information in the input data. To enhance the classification of MI, the authors of [13] presented a CNN model with a parallel multi-scale filter bank. They adopted time-scale convolutional kernels to heighten the model's efficiency. Subsequently, they utilized four distinct time-domain images as the input to effectively extract features.

CNNs have been widely used in processing only time- or frequency-domain images as the input. However, given the continuity and complexity of the EEG signal itself, it is difficult to extract sufficient features through only the frequency- or time-domain dimensions. Therefore, we propose a novel multi-input time–frequency CNN structure for this situation.

Event-related desynchronization (ERD) and event-related synchronization (ERS) ordinarily appear inside the scope of alpha rhythms (7–14 Hz) and beta rhythms (14–30 Hz), with slight variations noted across various articles [28,29].

Therefore, we focused our study on these two rhythms.

Our parallel CNN is displayed in Figure 1. We first converted the signals into time–frequency images of alpha and beta rhythms utilizing CWT. These images were then taken care of as the input through two comparative minor CNN structures. The convolutional neural network architecture utilized in this study includes two layers, each comprised of 32 and 64 convolutional kernels measuring 5×5 , with stride step sizes of 2×2 , respectively. The proposed model incorporates Rectified Linear Units (ReLU) as its activation function, which enhances classification accuracy and accelerates learning in our model. To avoid overfitting, the model is regularized using L2 regularization. Throughout the preparation, we set the L2 regularization boundary to 0.01 and utilized the Adam optimizer. The learning rate was laid out at 0.1 at the beginning, with an automatically adjusted learning rate during the training. The model was trained for 100 epochs; more epochs would make the model learn better, but would easily lead to overfitting. Batch size was used to estimate the gradient in each iteration. A larger batch size leads to a more accurate estimate of the gradient and faster computation due to parallel processing, but requires more memory. On the other hand, it can offer regularization benefits but might result in noisy gradient estimates and slower training with smaller batch sizes. The batch size for this study was 64. An early stopping mechanism was employed in the training phase to moderate the overfitting. Specifically, the training blocked if the validation loss failed to decrease for five consecutive epochs. This ensured that a well performing weights were retained, which would work well with future signals. We considered the limitation of data size, optimized the network parameters, and deleted the pooling layer to avoid the loss of effective features.

The resulting feature image from both minor regions were processed by a fully connected layer to convert them into a 1D vector. Subsequently, the 1D features extracted from both branches were concatenated into a single 1D vector, which was used for the classifier as the input. We used three normally utilized classifiers—specifically DT, softmax, and SVM. Furthermore, the pooling layer was wiped out to streamline the model system and forestall the deficiency of sampling [30].

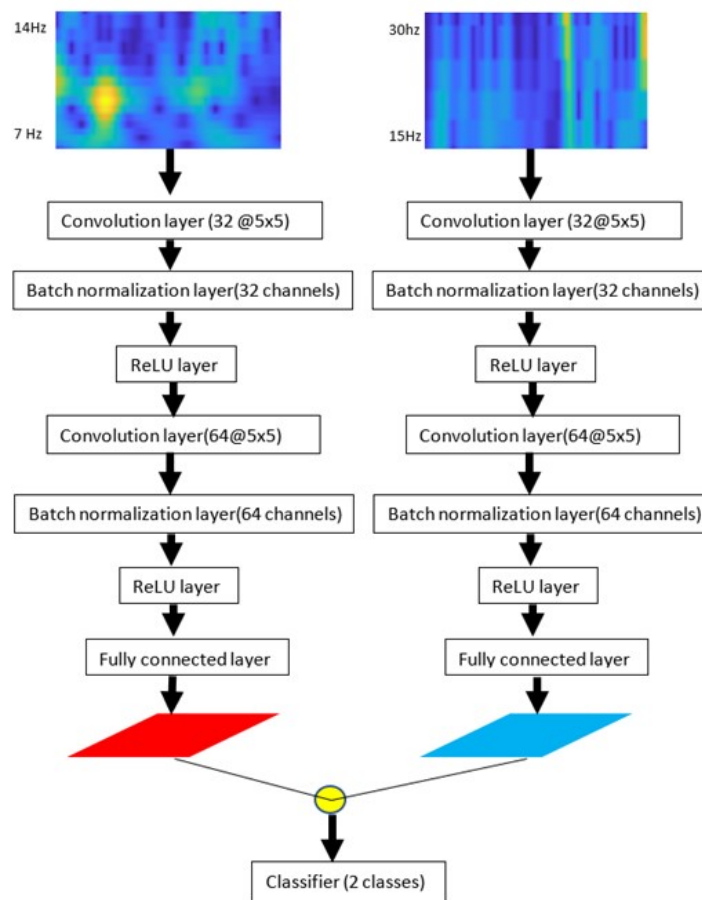


Figure 1. The proposed CNN framework.

3. Result

3.1. Database

Many BCI competitions have been conducted to provide researchers in EEG with standardized detection algorithms and reliable sources of data since 2001. To assess the efficacy of our work, we chose a publicly available dataset, known as 2008 BCI Competition IV Data Set 2B. The dataset utilized in this study comprises three bipolar EEG channels (C3, Cz, and C4) acquired from nine participants, each performing two classes of MI (left-hand and right-hand). Each subject contains five sections. We used the first three sections (...01T,...02T,...03T) as the training datasets and the last two sections (...04E,...05E) as the validation datasets. All data encodings of this dataset are shown in Figure 2.

Subject	Training			Evaluation	
1	B0101T	B0102T	B0103T	B0104E	B0105E
2	B0201T	B0202T	B0203T	B0204E	B0205E
3	B0301T	B0302T	B0303T	B0304E	B0305E
4	B0401T	B0402T	B0403T	B0404E	B0405E
5	B0501T	B0502T	B0503T	B0504E	B0505E
6	B0601T	B0602T	B0603T	B0604E	B0605E
7	B0701T	B0702T	B0703T	B0704E	B0705E
8	B0801T	B0802T	B0803T	B0804E	B0805E
9	B0901T	B0902T	B0903T	B0904E	B0905E

Figure 2. All datasets from 9 subjects.

Figure 3 depicts the acquisition process. The process of acquiring a solitary EEG episode consists of three stages. At the beginning, a 3.5s preparation phase, during which directional arrows show up on the displayer, is joined by a discernible alert. During the second stage, which spans from the 3.5 s to the 7 s, subjects are instructed to engage in MI based on the on-screen prompts. Following this, the volunteers are refreshed and wait for the sign to begin the next recording. We captured data from 4 s to 7 s, taking into consideration the possibility that the subject may not have responded when the cue first appeared. In the experiment, this obtained a total of 750 data points. All raw data are formatted as 750×3 , that is 750 represents 3 s of signal points (250 Hz), and 3 denotes the number of electrodes used. The raw data go through CWT to separately obtain the alpha and beta wave images, both sized at (40, 750). Figure 4 displays the CWT feature maps of the two rhythms for three electrodes. To reduce the computational load on the CNN, we calculate the average of every five sample points along the time axis. As a result, each training sample is sized (40, 150, 3).

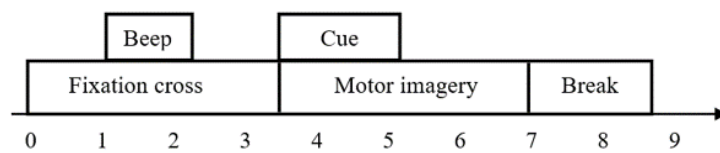


Figure 3. Timing scheme of the paradigm.

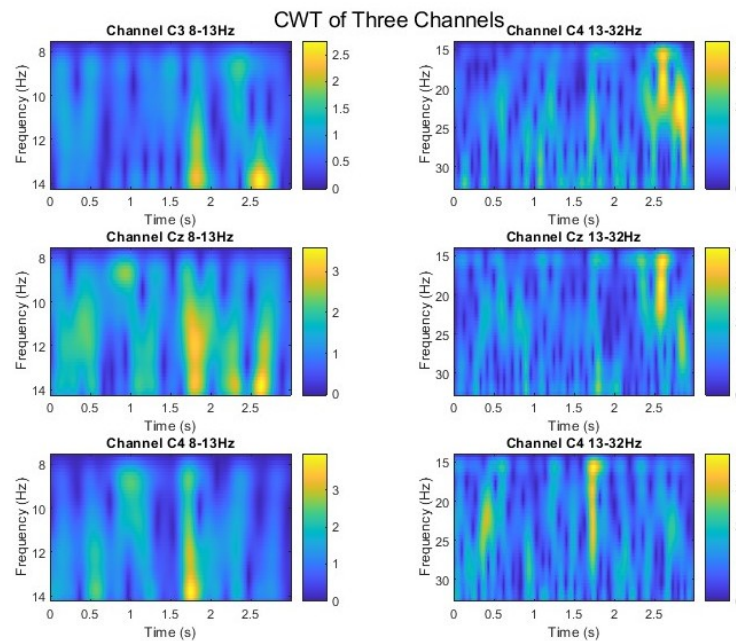


Figure 4. CWT of three channels.

3.2. Performance Evaluation Metrics

This section outlines a number of practical measurements that were utilized to assess the effectiveness of our work. The precision was determined using Equation (13).

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{13}$$

where T_P , T_N , F_P , and F_N represent true positives, true negatives, false positives, and false negatives, respectively.

The *precision*, *recall*, and *F1* score are shown in Equations (14)–(16), respectively.

$$Precision = \frac{T_P}{T_P + F_P} \quad (14)$$

$$Recall = \frac{T_P}{T_P + F_N} \quad (15)$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (16)$$

3.3. Performance of the Proposed Model

The experiment results are based on the following environment: CPU: Intel(R) Core(TM) i5-7300HQ @2.50 GHz, GPU: NVIDIA GeForce GTX 1050 Ti, Operating System: Windows 10 64 bits, Matlab: 2022a. Figure 5 displays the comparative results of three classifiers, with SVM achieving the highest accuracy of 84.1%, while softmax performed the worst with only 75.3%. This illustrates that SVM showed a better generalization ability in the small sample data sets. A notable limitation of this study was the relatively small size of the dataset, as deep learning models typically require a larger training dataset to fully demonstrate their capabilities. In future work, we aim to address this limitation by either collecting our own dataset or applying appropriate data augmentation methods to increase the dataset size. Another potential limitation is the complexity of hyperparameters, as this study employed numerous parameters. Thus, further research is necessary to determine how best to appropriately tune these hyperparameters.

subject	Accuracy (%)		
	DT	Softmax	SVM
S1	71.2	77.8	75.1
S2	60.7	65.4	69.8
S3	66.2	70.7	70.5
S4	89.5	97.1	92.6
S5	85.2	86.4	91.7
S6	80.4	72.1	87.2
S7	70.1	80.5	81.5
S8	73.9	89.6	93.3
S9	80.5	86.2	95.2
Average	75.3	80.6	84.1

Figure 5. Comparison of our work and different classifiers.

In addition, alternative evaluation metrics could be used to assess the effectiveness of the proposed model. Figure 6 provides an ROC curve of three classifiers. Figure 7 and Table 1 present the confusion matrices and the overall results of the precision and recall values for the proposed CNN.

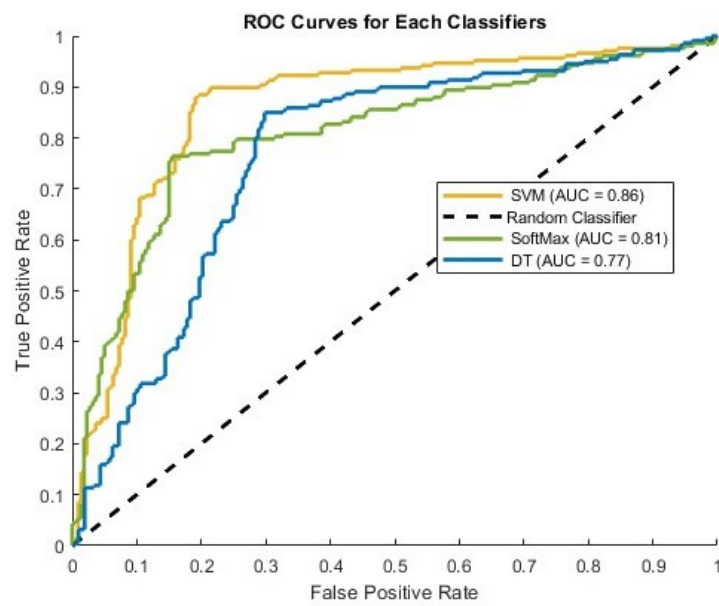


Figure 6. ROC curve of the Support Vector Machines (SVMs), SoftMax, and Decision trees (DT).

Table 1. Performance metrics in our CNN.

Metrics	Our
Precision (%)	85.71
Recall (%)	81.41
F1-Score (%)	83.5

To showcase the effectiveness of the proposed method, we conducted a comparative analysis with several state-of-the-art models. The presentation of this comparative literature is detailed in Table 2.

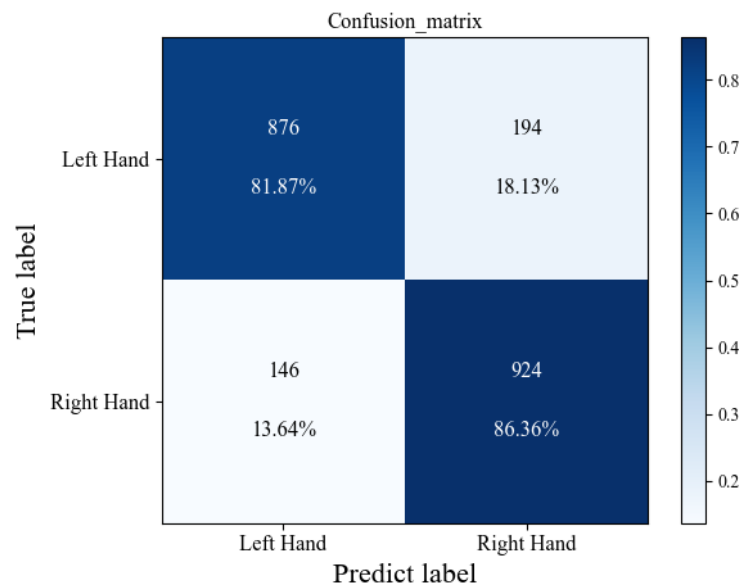


Figure 7. Confusion matrix for the proposed model with SVM.

Table 2. Compare the model structures to the comparative literature.

Literature	Contribution	Hardware Complexity
[4]	<ul style="list-style-type: none"> • Conditional empirical mode decomposition (CEMD) • 1D multi-scale CNN: consists of three 1D CNN layers of different sizes. 	<ul style="list-style-type: none"> • CEMD involves iterative processes that do not fully leverage the hardware feature
[5]	<ul style="list-style-type: none"> • Covariate shift estimation (CSE) • unsupervised adaptive ensemble learning(UAEL) 	<ul style="list-style-type: none"> • Requires moderate computational resources • Integrating multiple classifiers increases the memory requirement
[6]	<ul style="list-style-type: none"> • Multi-domain feature extraction and sparse feature fusion (MFESF), such as the power spectrum estimation, time-frequency energy 	<ul style="list-style-type: none"> • Mentioned feature extraction techniques require a relatively low hardware implementation complexity
[7]	<ul style="list-style-type: none"> • Temporal Two 1D Convolutional layers CNN: three different kernel sizes (1×45, 1×65 and 1×85) • Data Augmentation(DA) 	<ul style="list-style-type: none"> • 1D structure reduces computational parameters • 9 branches increases implementation complexity
[8]	<ul style="list-style-type: none"> • CWT • 30×30 kernel CNN 	<ul style="list-style-type: none"> • 30×30 significantly increases computational complexity
[9]	<ul style="list-style-type: none"> • Two-Branch CNN fusing temporal and frequency features 	<ul style="list-style-type: none"> • Single Convolutional Layer: Facilitates hardware implementation • 64×1 Kernel: Too large for efficient hardware implementation
Our	<ul style="list-style-type: none"> • Two-branch lightweight CNN • CWT 	<ul style="list-style-type: none"> • 5×5 kernel CNN small kernel good for reduce hardware memory and computing resource usage • Many hardware acceleration techniques for CNNs and CWTs

The contrasted performance with different results are presented in Table 3. We demonstrate the notable classification accuracy of our approach, surpassing most of the current state-of-the-art models. Impressively, we achieved a remarkable accuracy rate of 95.2% for subject 9, which is higher than for the other models. Additionally, our study boasts a minimal average accuracy gap of 25.4% across the nine subjects, indicating that our method effectively avoids personal irregularity and learns distinct action models associated with various MI states automatically in EEG. These findings highlight the potential of our approach to improve EEG-based classification in various research domains.

Table 4 shows the classification accuracy with different feature maps as the input, CSP,FFT, and STFT, and a CWT test using BCI Competition IV Dataset 2B. The input image signal is trained and tested by the proposed CNN using 10×10 -fold cross-validation. Among the evaluated models, CWT-CNN demonstrates a competitive performance, which outperforms the CSP-CNN 4.1%, and FFT-CNN 4.5%, and STFT-CNN 2.2%, respectively, highlighting the efficacy of utilizing CWT for enhanced time-frequency feature extraction in EEG signal analysis. Although the average accuracy of CWT-CNN is the highest, however, for subjects 4, 7, and 9, the accuracy of CWT-CNN is lower than other models, suggesting a potential influence of individual variability on model performance.

Table 3. Compare the result with advanced CNNs.

Accuracy (%)					
Subject No.	CEMD-CNN [4]	CSE-UAEL [5]	MFESF [6]	DA-CNN [7]	Our
1	80.56	78.13	75.3	80.5	75.1
2	65.44	54.69	65.4	70.6	69.8
3	65.97	53.13	62.2	85.6	70.5
4	99.32	94.38	97.8	94.6	92.6
5	89.19	85.31	88.8	86.6	87.2
6	86.11	80.31	87.2	87.6	87.2
7	81.25	72.81	73.4	89.6	81.5
8	88.82	78.75	91.9	95.6	93.3
9	86.81	74.38	87.5	87.4	95.2
Mean	82.61	74.65	81.1	87.6	84.1
Gap	33.8	41.25	32.4	27.7	25.4

CEMD: conditional empirical mode decomposition; CSE: covariate shift estimation; UAEL: unsupervised adaptive ensemble learning; MFESF: multi-domain feature extraction and sparse feature fusion.

Table 4. Compare the result with different input.

Accuracy (%)				
Subject No.	CSP-CNN	FFT-CNN	STFT-CNN	CWT-CNN
1	65.9	68.2	73.5	75.1
2	65.1	63.5	68.1	69.8
3	65.2	67.9	65.2	70.5
4	91.8	92.1	93.1	92.6
5	81.7	84.2	85.7	91.7
6	80.31	78.4	84.3	87.2
7	82.1	73.7	78.1	81.5
8	92.4	90.0	91.9	93.3
9	95.3	90.1	92.7	95.2
Mean	79.9	79.6	81.9	84.1

Table 5 compares the models including TBTF-CNN [9], CWT-CNN [8], and our proposed model. On average, our proposed model achieves an average accuracy of 84.1%, achieving the highest accuracy compared with TBTF-CNN (81.3%) and CWT-CNN (83.0%). It is worth noting that our model achieved the highest accuracy for subjects 5, 6, 8, and 9, which were 91.7%, 87.2%, 93.3%, and 95.2%, respectively.

Table 5. Compare the results with other models using CWT and CNN.

Accuracy (%)			
Subject No.	TBTF-CNN [9]	CWT-CNN [8]	Our
1	84.5	85.6	75.1
2	63.3	72.8	69.8
3	62.3	78.0	70.5
4	98.1	95.4	92.6
5	89.7	82.6	91.7
6	85.0	79.8	87.2
7	79.5	82.9	81.5
8	84.7	85.0	93.3
9	84.5	85.3	95.2
Mean	81.3	83.0	84.1

TBTF: The name of a two-branch CNN model.

4. Discussion

To enhance BCI system performance, we proposed a parallel novel CNN architecture that inputs EEG signal maps transformed by CWT. Results on the left and right hand

MI-EEG signals, the mean classification Precision, Recall and F1-Score obtained by CWT-CNN algorithm are 85.71%, 81.41%, 0.835, respectively. Basically, the proposed CNN model is able to process these time–frequency images and classify MI tasks with superior performance than other traditional classification models, which demonstrates the potential of combining CWT with deep learning techniques.

As illustrated in Tables 3 and 5, our method exhibits superior classification accuracy across subjects compared with traditional algorithms like CSE-UAEL or MFESF. Furthermore, when contrasted with deep learning methods like CEMD-CNN or DA-CNN, our approach demonstrates a higher average classification accuracy. Among these, DA-CNN has the highest average accuracy because it uses data augmentation techniques to triple the number of training samples. For deep learning networks, increasing the number of samples is one of the effective means to improve the training accuracy, and is widely used in the field of engineering. However, EEG data are biological data, and we must carefully verify the use of biological data and its reliability, especially artificially generated biological data. In this context, the applicability of conventional data enhancement methods, such as inversion, resizing, and adding noise, to biological data needs to be further explored. Therefore, future works will develop data augmentation methods suitable for EEG data that meet both the engineering and biological requirements to optimize the performance of our model. Notably, individual subject analysis unveils diverse responses to different models, implying the influence of individual variability on model efficacy. In particular, our method showcases minimal accuracy discrepancies among subjects compared with other techniques. Although CWT and CNN have been applied to the MI field in previous work, our network still has an advantage in accuracy compared with models using similar CWT-CNN technology.

The results indicate that the CWT-CNN approach has the highest (84.1%) average classification accuracy when compared with the CSP-CNN (79.9%), FFT-CNN (79.6%), and STFT-CNN (81.9%) methods, as shown in Table 4. Prior research indicates that CSP, being a linear analysis method, may overlook short-term signal changes and lack precision when capturing signal details [31]. Moreover, FFT struggles to capture the local characteristics of MI-EEG signals effectively [32]. Additionally, due to the fixed window size of STFT, both overall and local features may not be adequately represented. Contrastingly, CWT offers a balance between global and local features by decomposing the signal and offering time-varying windows with a high temporal resolution [33].

Despite these promising results, several limitations are acknowledged, including the small dataset size and hyperparameter complexity. As shown in Figure 5, SVM exhibited the highest performance, achieving a prediction accuracy of 84.1%, followed by Softmax (80.6%), and lastly decision tree (75.3%). EEG signals are often affected by considerable noise and interference, yet CNN can effectively extract relevant data through convolution and pooling layers. These layers enable the abstraction of feature representations from the raw data, which are highly discriminative for classification tasks. In contrast, decision trees may be limited to utilizing original features or a restricted set of manual features, leading to comparatively weaker expressive capabilities [34]. Moreover, the size and quality of the dataset significantly impact the model performance. The 2008 BCI Competition IV Data Set 2B is relatively small, and the Softmax classifier is susceptible to overfitting when handling smaller datasets, particularly those with a higher complexity such as EEG. Overfitting results in a model that performs well on training data, but fails to generalize effectively to new data. Future research directions will involve addressing these limitations through data augmentation techniques and systematic hyperparameter tuning to improve model generalization and performance optimization.

5. Conclusions

The complexity of EEG signals, along with the limitations of hand-operated feature extraction, presents significant challenges to the classification of EEG signals. To overcome this challenge, we introduce a parallel structure that utilizes CNN with CWT technology,

thereby enabling more comprehensive feature extraction and improving classification accuracy. Notably, our work accomplishes a mean accuracy of 84.1% on the BCI Competition IV 2b dataset for MI purposes. Upon comparing the classification outcome values obtained from various pre-processing methods, it is evident that CWT is better suited for integration with CNN for analyzing MI-EEG signals compared with CSP, FFT, and STFT. We replace the softMax classifier commonly used by CNN with SVM and DT, and experiments prove that SVM is more suitable for small datasets such as BCI Competition IV 2b. Moreover, our method compares favorably with other state-of-the-art approaches. We envision that this technique will be of value in different AI and MI analytical undertakings. We are eager to explore its potential further. Our future plans involve the creation of a specialized inference accelerator tailored for purposed CNN that is seamlessly compatible with adaptable devices like field programmable gate arrays (FPGAs). Nevertheless, our objective raises numerous layout hurdles, specifically concerning the parallel minor CNN. Furthermore, the implementation of CNN and CWT demand considerable computational resources, which can lead to memory limitations or sluggish inference speed on the hardware employed. To overcome these obstacles, we must devise inventive solutions that enable efficient resource utilization while ensuring performance.

Author Contributions: Conceptualization, Y.X. and S.O.; methodology, Y.X.; software, Y.X.; validation, Y.X.; formal analysis, Y.X.; investigation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, Y.X. and S.O.; supervision, S.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CWT	Continuous Wavelet Transform
FPGA	Field-Programmable Gate Arrays
DL	Deep learning
MI	Motor Imagery
EEG	Electroencephalography
CSE-UAEL	Covariate Shift Estimation and Unsupervised Adaptive Ensemble Learning
SR	Sparse Representation
CSP	Common Spatial Pattern
FFT	Fast Fourier Transform
STFT	Short Time Fourier Transform
BCI	Brain-Computer Interface
CEMD	Conditional Empirical Mode Decomposition
ERD	Event-related Desynchronization
ERS	Event-related Synchronization
ReLUs	Rectified Linear Units
SVM	Support Vector Machine
DT	Choice Decision Tree
MFESF	Multi-domain feature extraction and sparse feature fusion
CSSD	common spatial subspace decomposition
PSD	Power Spectral Density

References

1. Shih, J.J.; Krusienski, D.J.; Wolpaw, J.R. Brain-computer interfaces in medicine. In *Mayo Clinic Proceedings*; Elsevier: Amsterdam, The Netherlands, 2012; Volume 87, pp. 268–279.
2. Tabar, Y.R.; Halici, U. A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng.* **2016**, *14*, 016003. [[CrossRef](#)] [[PubMed](#)]
3. Kant, P.; Laskar, S.H.; Hazarika, J.; Mahamune, R. CWT Based transfer learning for motor imagery classification for brain computer interfaces. *J. Neurosci. Methods* **2020**, *345*, 108886. [[CrossRef](#)] [[PubMed](#)]
4. Tang, X.; Li, W.; Li, X.; Ma, W.; Dang, X. Motor imagery EEG recognition based on conditional optimization empirical mode decomposition and multi-scale convolutional neural network. *Expert Syst. Appl.* **2020**, *149*, 113285. [[CrossRef](#)]
5. Raza, H.; Rathee, D.; Zhou, S.M.; Cecotti, H.; Prasad, G. Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface. *Neurocomputing* **2019**, *343*, 154–166. [[CrossRef](#)] [[PubMed](#)]
6. Xu, C.; Sun, C.; Jiang, G.; Chen, X.; He, Q.; Xie, P. Two-level multi-domain feature extraction on sparse representation for motor imagery classification. *Biomed. Signal Process. Control* **2020**, *62*, 102160. [[CrossRef](#)]
7. Dai, G.; Zhou, J.; Huang, J.; Wang, N. HS-CNN: A CNN with hybrid convolution scale for EEG motor imagery classification. *J. Neural Eng.* **2020**, *17*, 016025. [[CrossRef](#)]
8. Lee, H.K.; Choi, Y.S. Application of continuous wavelet transform and convolutional neural network in decoding motor imagery brain-computer interface. *Entropy* **2019**, *21*, 1199. [[CrossRef](#)]
9. Yang, J.; Gao, S.; Shen, T. A two-branch CNN fusing temporal and frequency features for motor imagery EEG decoding. *Entropy* **2022**, *24*, 376. [[CrossRef](#)]
10. Tangermann, M.; Müller, K.R.; Aertsen, A.; Birbaumer, N.; Braun, C.; Brunner, C.; Leeb, R.; Mehring, C.; Miller, K.J.; Mueller-Putz, G.; et al. Review of the BCI competition IV. *Front. Neurosci.* **2012**, *6*, 55. [[CrossRef](#)]
11. Zhao, M.; Zhang, S.; Mao, X.; Sun, L. EEG Topography Amplification Using FastGAN-ASP Method. *Electronics* **2023**, *12*, 4944. [[CrossRef](#)]
12. Xie, Y.; Oniga, S. Classification of Motor Imagery EEG Signals Based on Data Augmentation and Convolutional Neural Networks. *Sensors* **2023**, *23*, 1932. [[CrossRef](#)] [[PubMed](#)]
13. Wu, H.; Niu, Y.; Li, F.; Li, Y.; Fu, B.; Shi, G.; Dong, M. A parallel multiscale filter bank convolutional neural networks for motor imagery EEG classification. *Front. Neurosci.* **2019**, *13*, 1275. [[CrossRef](#)] [[PubMed](#)]
14. Velásquez-Martínez, L.F.; Álvarez-Meza, A.M.; Castellanos-Domínguez, C.G. Motor imagery classification for BCI using common spatial patterns and feature relevance analysis. In Proceedings of the Natural and Artificial Computation in Engineering and Medical Applications: 5th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2013, Mallorca, Spain, 10–14 June 2013; Proceedings, Part II 5; Springer: Berlin/Heidelberg, Germany, 2013; pp. 365–374.
15. Kim, C.; Sun, J.; Liu, D.; Wang, Q.; Paek, S. An effective feature extraction method by power spectral density of EEG signal for 2-class motor imagery-based BCI. *Med. Biol. Eng. Comput.* **2018**, *56*, 1645–1658. [[CrossRef](#)] [[PubMed](#)]
16. Lee, H.K.; Choi, Y.S. A convolution neural networks scheme for classification of motor imagery EEG based on wavelet time-frequency image. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 906–909.
17. Chaudhary, S.; Taran, S.; Bajaj, V.; Sengur, A. Convolutional neural network based approach towards motor imagery tasks EEG signals classification. *IEEE Sensors J.* **2019**, *19*, 4494–4500. [[CrossRef](#)]
18. Roy, Y.; Banville, H.; Albuquerque, I.; Gramfort, A.; Falk, T.H.; Faubert, J. Deep learning-based electroencephalography analysis: A systematic review. *J. Neural Eng.* **2019**, *16*, 051001. [[CrossRef](#)]
19. Xie, Y.; Oniga, S. A review of processing methods and classification algorithm for EEG signal. *Carpathian J. Electron. Comput. Eng.* **2020**, *13*, 23–29. [[CrossRef](#)]
20. Lotte, F.; Congedo, M.; Lécuyer, A.; Lamarche, F.; Arnaldi, B. A review of classification algorithms for EEG-based brain-computer interfaces. *J. Neural Eng.* **2007**, *4*, R1. [[CrossRef](#)]
21. Xie, Y.; Oniga, S.; Majoros, T. Comparison of EEG Data Processing Using Feedforward and Convolutional Neural Network. In Proceedings of the Conference on Information Technology and Data Science, Debrecen, Hungary, 6–8 November 2020; pp. 279–289.
22. Li, Y. Recognition algorithm of driving fatigue related problems based on EEG signals. *NeuroQuantology* **2018**, *16*, 517–523. [[CrossRef](#)]
23. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
24. Qi, X.; Wang, T.; Liu, J. Comparison of support vector machine and softmax classifiers in computer vision. In Proceedings of the 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCC), Harbin, China, 8–10 December 2017; pp. 151–155.
25. Larose, D.T.; Larose, C.D. *Discovering Knowledge in Data: An Introduction to Data Mining*; John Wiley & Sons: Hoboken, NJ, USA, 2014; Volume 4.
26. Librelotto, S.R.; Mozzaquatro, P.M. Análise dos Algoritmos de Mineração J48 e Apriori Aplicados na detecção de Indicadores da Qualidade de vida e saúde. *Rev. Interdiscip. Ensino Pesqui. Extensão-RevInt* **2014**. Available online: <https://api.semanticscholar.org/CorpusID:170257329> (accessed on 26 September 2024).

27. Alvarenga, M.T. Utilização da Ferramenta j48 Para Descoberta do Conhecimento em Bases de Dados Fitossanitários, climáticos e Espectrais. Ph.D. Thesis, Universidade Federal de Lavras, Lavras, Brazil, 2014.
28. McFarland, D.J.; Miner, L.A.; Vaughan, T.M.; Wolpaw, J.R. Mu and beta rhythm topographies during motor imagery and actual movements. *Brain Topogr.* **2000**, *12*, 177–186. [[CrossRef](#)]
29. Shahid, S.; Sinha, R.K.; Prasad, G. Mu and beta rhythm modulations in motor imagery related post-stroke EEG: A study under BCI framework for post-stroke rehabilitation. *BMC Neurosci.* **2010**, *11*, 1. [[CrossRef](#)]
30. Li, F.; He, F.; Wang, F.; Zhang, D.; Xia, Y.; Li, X. A novel simplified convolutional neural network classification algorithm of motor imagery EEG signals based on deep learning. *Appl. Sci.* **2020**, *10*, 1605. [[CrossRef](#)]
31. Hsu, W.Y.; Sun, Y.N. EEG-based motor imagery analysis using weighted wavelet transform features. *J. Neurosci. Methods* **2009**, *176*, 310–318. [[CrossRef](#)] [[PubMed](#)]
32. Adeli, H.; Zhou, Z.; Dadmehr, N. Analysis of EEG records in an epileptic patient using wavelet transform. *J. Neurosci. Methods* **2003**, *123*, 69–87. [[CrossRef](#)]
33. Ma, L.; Stückler, J.; Wu, T.; Cremers, D. Detailed dense inference with convolutional neural networks via discrete wavelet transform. *arXiv* **2018**, arXiv:1808.01834.
34. Albaqami, H.; Hassan, G.M.; Subasi, A.; Datta, A. Automatic detection of abnormal EEG signals using wavelet feature extraction and gradient boosting decision tree. *Biomed. Signal Process. Control* **2021**, *70*, 102957. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.