IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 20, 2021, accepted July 29, 2021, date of publication August 3, 2021, date of current version August 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102183

Speed Classification of Upper Limb Movements Through EEG Signal for BCI Application

SEPIDEH ZOLFAGHARI[1], TOHID YOUSEFI REZAII[1], SAEED MESHGINI[1], ALI FARZAMNIA[2], (Senior Member, IEEE), AND LIAU CHUNG FAN[2]

[1]Biomedical Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 5166616471, Iran
[2]Faculty of Engineering, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia

Corresponding authors: Ali Farzamnia (alifarzamnia@ums.edu.my) and Tohid Yousefi Rezaii (yousefi@tabrizu.ac.ir)

This work was supported by the Research Management Center (PPP) and the Faculty of Engineering, Universiti Malaysia Sabah (UMS).

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of the Faculty of Electrical and Computer Engineering, University of Tabriz under Application No. IR.TBZ-REC.1399.12.

ABSTRACT Brain-Computer Interface (BCI) systems have obtained remarkable results in rehabilitation and robot control processes by converting brain signals into control commands. The quantity of movement speed is the fundamental issue in BCI that requires additional research. This paper investigated the classification of the slow and fast speeds of eight different upper limb movements through electroencephalogram (EEG) signals and information about the values of speed and maximum angle of movements from the MPU6050 module. Datasets were obtained by recording the EEG signals from 10 subjects and the module information connected on their right hand during movements. This study used Filter Bank Common Spatial Pattern (FBCSP) and Wavelet-Common Spatial Pattern (W-CSP) methods to extract speed features of movements. In both methods, features selected by the Mutual Information (MI) were sent to the Convolution Neural Network (CNN) and various machine learning classifiers. Due to the results of subject-independent speed classification, the FBCSP-CNN method obtained the highest accuracy of 90% with a Kappa coefficient of 0.8 for flexion/extension of the shoulder. Results from our proposed method demonstrate the ability to introduce a refined set of control commands into the BCI system by recognizing the features associated with the speed of movement parameters.

INDEX TERMS Brain-computer interface (BCI), convolution neural network (CNN), electroencephalogram (EEG), filter bank common spatial pattern (FBCSP), movement speed, wavelet-common spatial pattern (W-CSP).

I. INTRODUCTION

The human brain is a significant part of the central nervous system and the most complex structure known within the human body [1]. The root of all thoughts, feelings, and behaviors is the link between neurons in the brain [2]. The sum of changes in the electric field of neurons over time generates an electrical signal [3]. Electroencephalogram (EEG) is one of the most commonly used non-invasive methods for recording brain signals [4]. It is possible to identify neurophysiological activities related to evoked potentials and rhythmic brain activities by analyzing the EEG signals [5]. Event-related desynchronization (ERD) and event-related synchronization (ERS) patterns appear in the Mu and Beta frequency bands during imagination or performing movement [6]. EEG-based Brain-Computer Interface (BCI) systems are a field of research that has been used for different purposes [7], [8], particularly in the area of rehabilitation to assist patients with disabilities or motor injuries [9], [10]. There are five steps in the BCI system, including signal acquisition, pre-processing, feature extraction, classification, and control of interface devices [11]. The pre-processing purpose is to eliminate noise, reduce artifacts, and get rid of unwanted information. The feature extraction step maximizes the differences between the feature vectors for the various classes, leading to higher classification accuracy [12]. Finally, training data is given to the classification algorithms to construct a model based on features and assess the test data

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

accuracy [13]. Bandara *et al.* [14] proposed a new method for controlling a wearable robot with several degrees of freedom by predicting a person's intention to move in real-time via EEG signal. The experimental stages included moving an object, drinking, and resting. A time delay feature matrix was used to provide input for the neural network and Support Vector Machine (SVM) classifiers. Hortal *et al.* [15] developed a way of detecting the beginning and end of walking with the ERD phenomenon. They used the Motion Capture system, which was a wireless motion analysis system. Jin *et al.* [16] proposed a spars Bayesian ELM (SBELM)-based method that improved motor imagery classification performance on a dataset from BCI Competition IV IIb by controlling model complexity and removing additional neurons.

The Common Spatial Pattern (CSP) is one of the most popular methods for separating data between two classes [17]. Ang *et al.* [18] used both the CSP and Filter Bank Common Spatial Pattern (FBCSP) methods to separate the 4-class motor imagery data. They could extend the 2-class CSP method into a multi-class through One-Versus-Rest (OVR) and Divide-and-Conquer (DC) approaches. Suwannarat *et al.* [19] examined the classification of Motor imagery tasks that included wrist flexion/extension, hand opening/closing, and forearm pronation/supination of both hands. In this study, features were obtained from the CSP method, once for the whole band and second for five filter banks. Then, the classification was performed by SVM and LDA classifiers. López-Larraz *et al.* [20] studied proximal to distal articulate movements of the shoulder, elbow, and wrist of patients and healthy subjects. Then, they analyzed the ERD and Movement-Related Cortical Potential (MRCP) patterns. Bhattacharyya *et al.* [21] suggested an interval type-2 fuzzy classifier to decode the wrist movements (flexion and extension) and the finger (opening and closing the fist), which dealt with the uncertainty of the EEG signal during different sessions. The first level of classification was about moving or not moving. The second level specified the movement of the wrist or finger, and the third level focused on the type of movement performed. Robinson *et al.* [22] investigated the detection of direction and speed of movements. They recorded EEG signals as subjects moved their hands at two slow and fast speeds in four different directions: up, down, left, and right. Wavelet-Common Spatial Pattern (W-CSP) and Fisher Linear Discriminant (FLD) were used to extract features and classification, respectively. Bhattacharyya *et al.* [23] proposed a two-level classification method to differentiate between fast and slow hand movement execution. The feature vectors were obtained through the Welch method and tested on the SVM, Naive Bayesian (NB), Linear Discriminant Analysis (LDA), and K-Nearest Neighbor (KNN) classifiers.

Deep learning is a branch of machine learning, artificial intelligence, and a set of algorithms that attempt to model high-level abstract concepts using different levels and layers of learning [24]. Due to the power and accuracy of Deep learning in real-world issues, it has achieved remarkable success in many areas such as speech recognition [25], language processing [26], and computer vision [27]. Cheng *et al.* [28] suggested the feature extraction method based on Principal Component Analysis (PCA) and a Deep Belief Network (DBN). The use of the PCA-DBN algorithm on BCI competition data reduced the algorithm complexity quickly and eliminated additional information. Chen *et al.* [29] recognized the human movement intention from EEG signals of the BCI2000 dataset. A particular kind of Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), created temporary features after separating the signals into different frequencies. Finally, a layer exploited the temporal correlation between signal properties. They found that Multi-task RNN could increase the correlation between different EEG frequencies by learning differentiated signals.

Convolution Neural Network (CNN) is one of the most important deep learning methods in which multiple layers are trained powerfully [30]. Sakhavi *et al.* [31] introduced a CNN architecture for Motor imagery classification by presenting a temporal representation of the data generated by modifying the FBCSP method. Tang *et al.* [32] used a five-layer CNN architecture for feature extraction and motor imagery tasks classification. Kumar *et al.* [33] proposed a classification method using a Deep Neural Network (DNN) with four layers. The results showed that using the CSP-DNN method on several filter banks can reduce the number of calculations and errors. Taheri *et al.* [34] introduced a method for separating the imagery of hand and foot movements. Several signal representations include discrete cosine transform, empirical mode decomposition, Fast Fourier Transform (FFT), and CSP, were combined in a matrix and were sent into the AlexNet [35] architecture.

By studying the EEG-based BCI field, we found a few articles about classifying movement speed through EEG signals. None of them has examined the speed of movement of the upper limb joints by considering the direction of movement. For more variety and investigation on the movements a person needs in life, this article examines the speed classification of the wrist, shoulder, and elbow movements. It also offers an idea for obtaining speed and angle values of movements using the module. In the proposed algorithm, FBCSP and W-CSP methods individually extract speed features from EEG signals, and the MI method selects the distinctive features. Because of the importance and effectiveness of deep learning in signal analysis, the CNN classifier is used, and its performance is compared with other classifiers.

The rest of the paper includes the following: Section II describes the steps of data acquisition. Section III introduces the proposed algorithms and explains their subsections. Section IV shows the results, followed by the conclusion in Section V.

II. EXPERIMENT

A. DATA ACQUISITION

In this experiment, 10 healthy right-handed people participated, including 7 women and 3 men between the ages of 20 and 29. EEG signals were recorded with a 250 Hz

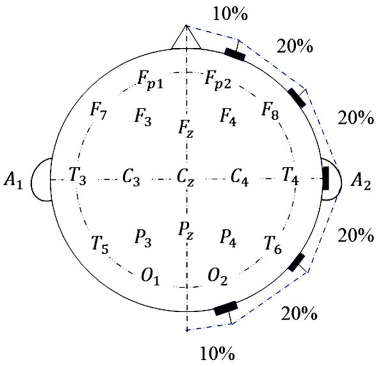


FIGURE 1. Location of electrodes according to the 10-20 system.

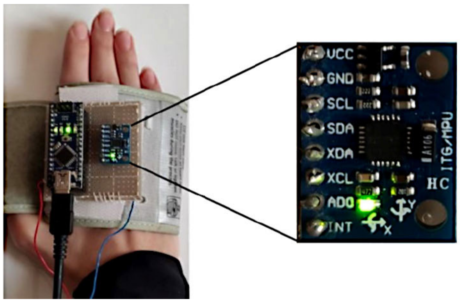


FIGURE 2. Location of the module on hand.

sampling rate and by 21 electrodes, according to the 10-20 system as shown in Fig. 1. Two electrodes near the right ear (A_2) and near the left ear (A_1) were considered as a reference, and 19 other electrodes were placed at different positions. Due to the elimination of noise, artifacts, and use of important information, two frequency filters, including a 50 Hz notch filter and bandpass filter in the frequency range of 0.5 to 70 Hz, were adjusted in the software of the Encephalan EEG recorder [36]. As shown in Fig. 2, the MPU6050 module was attached to the subject's right hand with the wristband to obtain movement acceleration in three directions of X, Y, and Z axes and movement angle around X and Y axes. In each scenario, the subject sat in front of the computer screen with the module on hand and performed eight upper limb movements, including M1 (flexion/extension of the wrist), M2 (radial/ulnar deviation of the wrist), M3 (pronation/supination of the wrist), M4 (flexion/extension of the shoulder), M5 (vertical abduction/adduction of the shoulder), M6 (horizontal abduction/adduction of the shoulder), M7 (lateral/medial rotation of the shoulder), and M8 (flexion/extension of the elbow).

Fig. 3 shows the timing scheme and visual cues of a scenario that consists of 346 s. The cues for each movement are as follows: First, a cross is indicated in the middle of the laptop screen for 3.5 s. In this case, the person looks at the center and refuses to move. Then, the image of the type of movement is displayed for 12 s, and the participants adjust their initial hand's position by seeing it. By displaying the white circle for 15.5 s, the subject performs the movement shown in the previous step. Finally, the "rest" text is displayed for 11 s, which is a sign of returning to the desired state and rest. After the third movement, the text of the "starting new movements" is demonstrated for 10 s. Then, the subject makes the necessary preparations for the subsequent movements. Fig. 4 shows subject number 9 preparing to perform M2 movement. The designed scenario was displayed 16 times for each subject. In 8 of them, the subject performed the movements at fast speed, and in the other 8 times, the subject performed the movements at slow speed. Therefore, by considering all subjects, each type of movement was repeated 80 times slowly and 80 times rapidly.

B. MODULE CONNECTIONS

For designing the accelerometer and goniometer, we used the Arduino NANO V3 with ATmega328p CPU and GY521-MPU6050 sensor module with 3-axis Gyroscope and 3-axis accelerometer. Fig. 5 shows the connections between the Arduino, module, and the laptop.

The Arduino Nano board was connected to the laptop by two cables. The first cable is a USB to TTL serial converter to apply the trigger while performing movements. The second USB was used to power the Arduino and send acceleration and angle information to the PLX-DAQ software. We wrote the codes related to finding acceleration and angle values in Arduino IDE software and sent them to the Arduino Nano board.

III. PROPOSED METHODS

Two proposed methods were used to analyze EEG signals and classify the speed of movements. They were simulated using MATLAB 2017a software on a laptop with 8 GB of RAM and a 3 GHz Core i7 CPU. Fig. 6 shows the block diagram of the proposed methods. First, movement intervals were separated from the EEG signals recorded in each scenario. Then, in the subject-independent model, identical movements based on speed were allocated into two groups, fast and slow. For each type of movement, 72 movements equal to 90% for training and 8 movements equal to 10% for testing were randomly selected in each speed group. In addition, 10-fold cross-validation was considered for evaluating the performance of the models. Subsequently, the feature extraction process was performed in the first proposed method by FBCSP and in the second proposed method by W-CSP. The filter bank in the FBCSP method and the wavelet in the W-CSP method also removed the artifacts. After selecting the distinctive features by the MI method, feature matrices were labeled and then mixed. Labeled features were sent to KNN, LDA, NB, SVM, and CNN classifiers. Finally, the accuracies of subject-independent speed classification for each type of movement were obtained by following the steps of Fig. 6.

During the scenario execution, the last output column of the MPU6050 module changed from value one to zero by displaying white circles. Thus, the separation of acceleration and angle values of each movement occurred between the

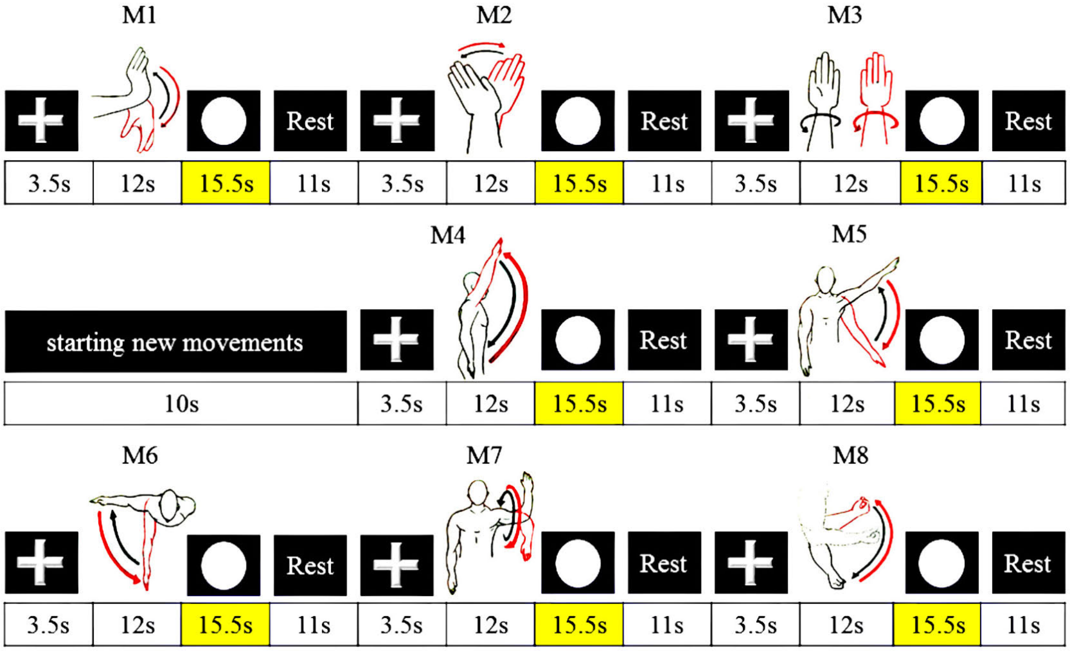


FIGURE 3. Timing scheme and visual cues of a scenario.

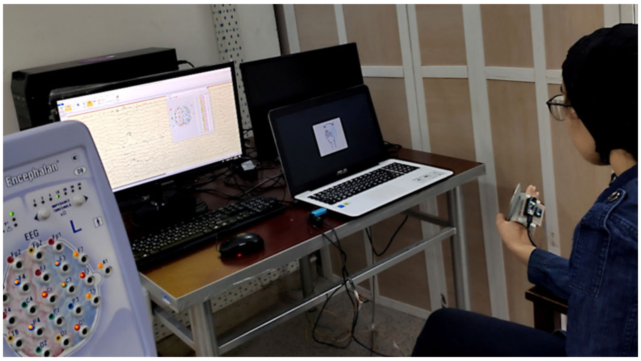
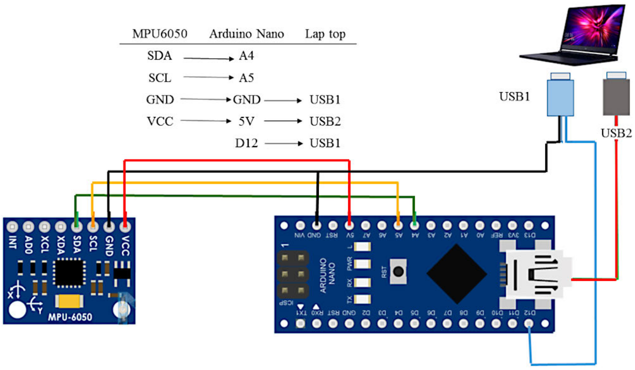


FIGURE 4. Image of subject number 9 preparing to perform M2 movement during the experiment.



MPU6050	Arduino Nano	Lap top
SDA	A4	
SCL	A5	
GND	GND	USB1
VCC	5V	USB2
	D12	USB1

FIGURE 5. Connections of arduino nano board to MPU6050 module and laptop.

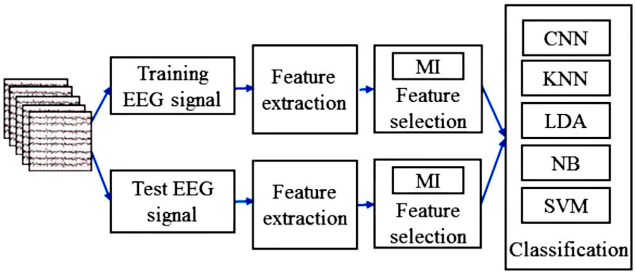


FIGURE 6. Block diagram of the proposed methods.

beginning of zero and 15.5 s later. After applying the second-order Butterworth high pass filter, the speed of movements in three directions was obtained by integrating from the surface below the acceleration diagrams. For each type of movement, mean speed and maximum angle values were calculated.

A. FEATURE EXTRACTION USING THE FBCSP

The FBCSP method uses EEG frequency sub-band information to obtain the optimally weighted combination of filtered channels. The FBCSP algorithm includes a filter bank that decomposes the EEG signal into nine bands of 4 to 40 Hz with a frequency interval of 4 Hz using a bandpass filter [37]. Then, for each sub-band, the CSP filter is calculated. The CSP algorithm is used extensively for distinguishing between two classes. The purpose of CSP is to design spatial filters that lead to new time series. By applying spatial filters to inputs, the variance of the signal is maximized in the first class and minimized in the second class [38]. The method of designing filters is based on the simultaneous diagonalization of two covariance matrices for two classes [39]. For analyzing the CSP method, consider the raw EEG data as a matrix X with $N \times T$ dimensions. Where N is the number of channels, and

T is the number of samples per channel. The normalized spatial covariance of the EEG for the two classes are as follows:

$$C_a = \frac{X_a X_a'}{trace(X_a X_a')} \tag{1}$$

$$C_b = \frac{X_b X_b'}{trace(X_b X_b')} \tag{2}$$

where C_a and C_b respectively are the covariance matrix for X_a and X_b. "Trace" represents its sum of the diagonal elements of the matrix and X' denotes the matrix transposition. The mean of the normalized covariance matrix $\overline{C}_a$ and $\overline{C}_b$ are added together to form the composite covariance matrix C.

$$C = \overline{C}_a + \overline{C}_b \tag{3}$$

In (3), C can be displayed as $C = U\lambda U'$. Where U is the eigenvectors matrix and λ is diagonal matrix. The eigenvalues are arranged in descending order and the whitening transformation matrix P is calculated as follows:

$$P = \lambda^{-\frac{1}{2}} U' \tag{4}$$

If P is applied to the average covariance matrix $\overline{C}_a$ and $\overline{C}_b$, all eigenvalues of PCP' are equal to one. Therefore, it can be said that $S_a = P\overline{C}_a P'$ and $S_b = P\overline{C}_b P'$ have common eigenvectors. If $S_a = B\lambda_a B'$ and $S_b = B\lambda_b B'$ then $\lambda_a + \lambda_b = I$ where I is a unique matrix. The common eigenvector is B, which can simultaneously produce the largest eigenvalues for $\overline{S}_a$ and the smallest eigenvalues for $\overline{S}_b$, and vice versa. The spatial filter matrix W with dimensions $N \times N$ for transmitting the EEG signal to another space is obtained by:

$$W = B'P \tag{5}$$

The matrix Z is calculated by multiplying $2m$ lines of W with the largest difference in variance and the EEG signal as follows:

$$Z = WX \tag{6}$$

Finally, the feature vectors are computed as follows:

$$f_p = \log \left(\frac{\mathrm{var}(Z_p)}{\sum_{i=1}^{2m} \mathrm{var}(Z_i)} \right) \tag{7}$$

where p indicates the row number of matrix Z and Z_p indicates the corresponding row vector.

Fig. 7 shows the performance of our FBCSP algorithm. In this feature extraction process, the training and test data of fast and slow groups were filtered in nine frequency bands by a fifth-order Butterworth bandpass filter. For each subband, training data were entered into the CSP algorithm and created a spatial filter matrix. Each spatial filter was applied to the training and test data. Then, for each movement, feature vectors were obtained.

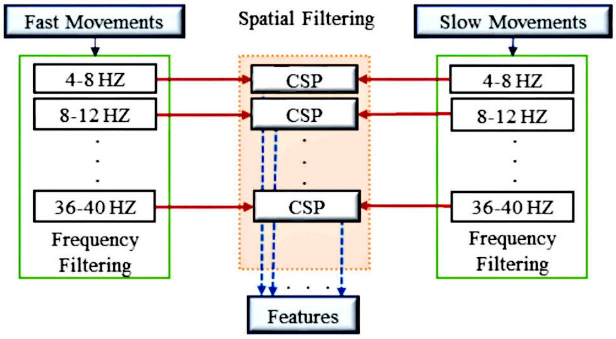


FIGURE 7. Feature extraction process using the FBCSP algorithm.

B. FEATURE EXTRACTION USING THE W-CSP

The W-CSP is consists of Wavelet and CSP algorithms. The Discrete Wavelet Transform method eliminates the drawbacks of the FFT algorithm. FFT-based signal analysis works well when its frequency spectrum is not time-dependent. Displaying the time-frequency of time-series signals are an attractive way to capture frequency information at low frequencies and time information at high frequencies. The wavelet method is one such technique that uses multiple resolution analysis [40]. For continuous-time signal $x(t)$, the wavelet transform is defined as follows:

$$WT_x(j,k) = \frac{1}{\sqrt{a_0^j}} \int x(t)\psi^* \left(\frac{t - ka_0^j b_0}{a_0^j} \right) dt \tag{8}$$

where ψ^* indicates conjugate of mother wavelet, $ka_0^j b_0$ is the transmission parameter, a_0^j is the scale parameter and j represents the wavelet level. At each decomposition level during a dyadic sequence, the approximation coefficients are created by passing the signal through the low pass filter and the detail coefficients are generated by passing the signal through the high pass filter as follows:

$$CA_j(k) = \sum_{n=-\infty}^{\infty} x_{j-1}(n)\, l(n-2k) \tag{9}$$

$$CD_j(k) = \sum_{n=-\infty}^{\infty} x_{j-1}(n)\, h(n-2k) \tag{10}$$

where CA and CD represent approximation coefficients and detail coefficients, respectively. h denotes the high pass filter and l denotes the low pass filter. After separating the signal into high and low frequencies in the first level, the wavelet transform in the next steps is taken only from the low-frequency part [41].

Fig. 8 shows the overall performance of our W-CSP algorithm. In this feature extraction process, the Daubechies wavelet of order six (db6) was used. The training and test data related to fast and slow groups in 8 levels were decomposed into high and low frequencies. The detail coefficients of the training data of two classes were entered into the CSP algorithm and created a spatial filter matrix at each level.

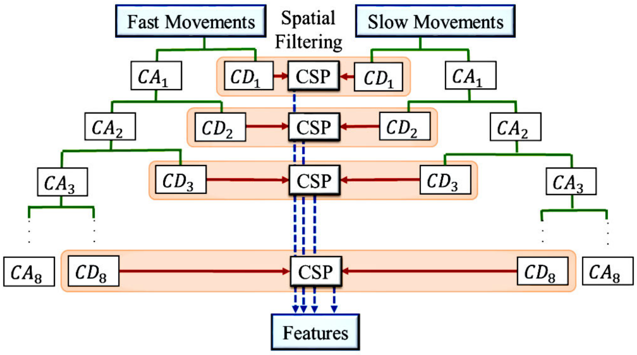


FIGURE 8. Feature extraction process using the W-CSP algorithm.

Each spatial filter was applied to the detail coefficient of training data and test data, depending on the wavelet conversion level. Finally, the feature vectors were obtained for each movement.

C. MUTUAL INFORMATION

The MI method measures the extent of nonlinear dependencies of two random variables [42]. This method is closely related to the concept of entropy of a random variable. For discrete arbitrary variables X and Y, the MI is calculated using the following equation:

$$I(X, Y) = \sum_{y\in Y}\sum_{x\in X} P(x, y)\log\left(\frac{P(x, y)}{P(x), P(y)}\right) \tag{11}$$

where $p(x, y)$ is the joint probability distribution function of X and Y. $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively. The MI method can also be computed as:

$$I(X, Y) = H(X, Y) - H(X|Y) - H(Y|X) \tag{12}$$

where $H(X|Y)$ and $H(Y|X)$ are the conditional entropies and $H(X, Y)$ is the joint entropy of X and Y.

In this study, feature selection was executed by the MI method. The obtained feature vectors were arranged in descending order based on the degree of independence. For each movement, 150 features were selected and sent to the classifiers as input.

D. CONVOLUTION NEURAL NETWORK

CNN is a type of artificial neural network that is usually trained with multi-layered architectures. Two forward and backward propagation steps are used for network training. The primary purpose of the forward step is to move the input information with weight and bias parameters to the layers and generate the output. In the backward step, network parameters are updated to reduce network error calculated by the loss function [43]. The standard CNN model consists of an input layer, a hidden layer, and an output layer. The hidden layer includes the convolution layer, the pooling layer, the fully connected layer, and activation functions.

In the convolution layer, a series of filters or kernels extract local features from the input and produce feature maps [44]. By considering the ith feature map of the lth layer as x_i^l and the kth feature map of the layer $l-1$ as x_k^{l-1}, the mathematical definition of convolution layer is described as follows:

$$x_i^l = f\left(\sum_k w_{k,i}^l * x_k^{l-1} + b_i^l\right) \tag{13}$$

where b_i^l is the bias for the ith feature map of the current layer, $w_{k,i}^l$ represents the connecting weight from the kth feature map of the previous layer to the ith feature map of the current layer, $*$ indicates the convolution operation, and $f(.)$ is the nonlinear activation function [45]. Applying a nonlinear function after layers of weight is very important because it allows a neural network to learn nonlinear mapping [46]. In this study, the Relu activation function was used that is expressed as follows:

$$f(x) = \max(0, x) \tag{14}$$

After the convolution layer, the principal aim of the pooling layer is to compress the dimension of feature maps without changing their numbers [47]. The common types of pooling are max pooling and average pooling that reduce the computational complexity and increase training speed. By considering x_i^l and x_i^{l-1} as the ith feature map of the current layer and previous layer, respectively, the pooling can be defined as follows:

$$x_i^l = f\left(B_i^l down\left(x_i^{l-1}\right) + b_i^l\right) \tag{15}$$

where $down(.)$ denotes a down-sampling function, b_i^l and B_i^l are additive and multiplicative biases for ith feature map of the lth layer, respectively. Final output feature maps are flattened into a one-dimension array and joined to one or several fully connected layers [48]. Afterward, the probability distribution of output classes are calculated by the Softmax function as follows:

$$\sigma(Z)_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \quad \text{for } i = 1, 2, \ldots, K \tag{16}$$

where $Z \in R^K$ is the input vector. The output values are between 0 and 1 and their sum is equal to 1. In the back-propagation stage, for evaluating the network performance, the loss function calculates the distance between the outputs predicted by the network and the desired output [49]. Then, weights are updated with the help of the gradient descent approach.

Fig. 9 shows our proposed CNN architecture. Each input contains selected features of a movement that are arranged in a 150 × 1 matrix. A kernel with a size of 30 × 1, stride 2 × 1, and zero-padding 0 × 1 was moved on the inputs in the convolution layer. In this layer 16 kernels were applied, which created feature maps with a depth of 16 and dimensions of 62 × 1. Following the convolution layer, the

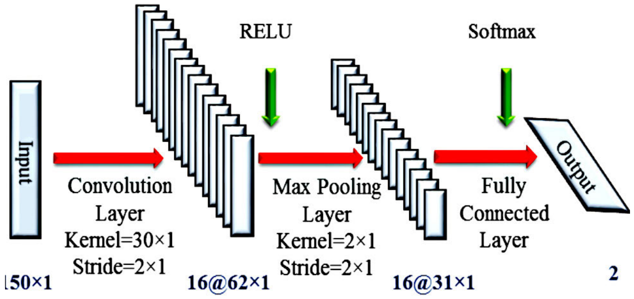


FIGURE 9. Proposed CNN network architecture.

Relu activation function was used. Then, the max-pooling layer with kernel size 2 × 1 and stride 2 × 1 was applied to extract the features, which changed the dimension of feature maps to 31 × 1. Finally, the fully connected layer brought the features together, and in the output layer, a Softmax function was utilized to classify the speed of movements. Also, cross-entropy was used as a loss function. In the network training, 150 epochs with an initial learning rate of 0.4 and a mini-batch size of 72 were tuned as hyper-parameters. Epoch is a full pass over the training set, the batch size denotes the number of processed samples before updating weights, and the learning rate determines the step size of the adjustments made to the weights.

E. MACHINE LEARNING HYPER-PARAMETERS

The hyper-parameters need to be set correctly to increase the performance of models. In this section, we discuss the hyper-parameters of machine learning classifiers that were tuned. KNN Classifier is a non-parametric learning algorithm that in the classification settings, the distance between the point that wants to be labeled is calculated with the k nearest points. The decision on the label of the desired point is made according to the maximum votes of neighboring points [50]. In this classifier, the Euclidean distance metric was used, and $K = 1$ was considered as the nearest number of neighbors. LDA classifier is utilized for dimensionality reduction problems aiming to maximize the between-classes variance and minimize the within-classes variance [51]. In the LDA classifier, the value of Delta, which is a linear coefficient threshold, was considered zero. NB classifier is based on the Bayes theorem, a probabilistic machine learning model that assumes features are independent of each other. In this classifier, Gaussian distribution function was used to estimate the distribution of the data. The SVM classifier aims to create a hyperplane to maximize the margin between the support vectors of the two classes [52]. When the data of classes are not separable, their separation by a linear boundary is always accompanied by an error. In this case, the Kernel function is used to transfer training data from the original space to a high-dimensional space [53]. Also, the regularization coefficient (C) strikes a balance between maximizing margin and minimizing classification error. In the SVM classifier, linear kernel function and $C = 1$ were tuned as hyper-parameters.

TABLE 1. Accuracy of classifiers using thE FBCSP algorithm.

Type of movement	KNN	LDA	NB	SVM	CNN
M1	81.25	71.25	75.62	80.62	83.13
M2	77.50	69.37	74.37	78.75	80.63
M3	79.37	70.62	70.00	80.00	82.50
M4	83.75	75.00	80.00	86.87	90.00
M5	83.12	73.12	76.25	83.12	86.25
M6	78.75	68.12	71.87	79.37	80.00
M7	81.87	71.25	70.00	79.37	81.88
M8	80.62	73.75	71.25	80.00	81.25
Average	80.77	71.56	73.67	81.01	83.20

TABLE 2. Accuracy of classifiers using the W-CSP algorithm.

Type of movement	KNN	LDA	NB	SVM	CNN
M1	78.75	71.87	76.25	78.12	81.88
M2	76.25	68.12	71.87	78.12	79.38
M3	77.50	70.62	72.50	79.37	79.38
M4	83.75	77.50	75.62	83.75	85.00
M5	80.62	70.62	68.75	79.37	81.88
M6	77.50	66.87	71.25	78.12	75.00
M7	76.25	65.62	68.12	77.50	75.63
M8	79.37	70.62	73.75	79.37	78.13
Average	78.75	70.23	72.26	79.21	79.53

IV. RESULTS AND DISCUSSION

This section presents the simulation results of the proposed algorithms. Table 1 and Table 2 show the accuracy of KNN, LDA, NB, SVM and, CNN classifiers for classifying fast and slow speeds of movements (M1, M2, M3, M4, M5, M6, M7, and M8) in the subject-independent model using FBCSP and W-CSP feature extraction methods, respectively. The last row of the tables shows the average accuracy of all movements based on the type of classifiers. Overall, it is clear that the classification accuracy of CNN by the FBCSP method is higher than other classifiers in all movements. However, the most remarkable accuracies by the W-CSP method in M1, M2, M3, M4, and M5 movements are related to CNN classifier, and in M6, M7, and M8 movements are related to the SVM classifier. Moreover, the NB and LDA classifiers using both FBCSP and W-CSP methods comprise the lowest classification accuracies.

As Table 1 shows, the accuracies of the CNN classifier using FBCSP are between 80% and 90%, which is the lowest value for M6 movement and the highest value for M4 movement. At the same time, the accuracies gained by the SVM classifier are approximately 79-87% and by the KNN classifier are nearly 77.5-84%. The significant difference between them and the CNN classifier's accuracies is in M4 movement. Meanwhile, the results of LDA and NB classifiers with notable differences of 7.5-15% and 6-12%, respectively, are less than the results of the CNN classifier. As Table 2 shows, classification accuracies of the CNN from M1 to M5 movements with 79.5-85% are slightly more than the SVM and KNN classifiers. Whereas, in M6, M7, and M8 movements, the SVM classifier performed approximately 1-3% better than the CNN classifier. Additionally, the LDA

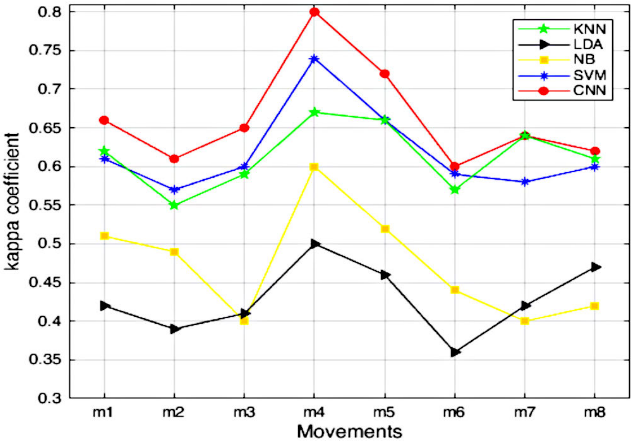


FIGURE 10. Kappa coefficient plot of the FBCSP method for classifying the speed of eight upper limb movements.

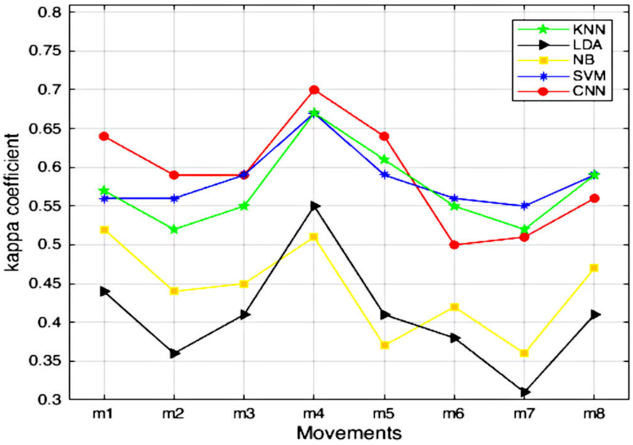


FIGURE 11. Kappa coefficient plot of the W-CSP method for classifying the speed of eight upper limb movements.

classifier with 65.5-77.5% and the NB classifier with 68-76.5% possess low accuracies.

Due to Table 1 and Table 2, the average speed classification accuracy of the KNN, LDA, NB, SVM, and CNN for all movements with the W-CSP method is 2%, 1.3%, 1.4%, 1%, and 3.7% lower than using the FBCSP method, respectively. It is obvious that the performance of the FBCSP algorithm is effective compared to the W-CSP algorithm in extracting distinctive features. Consequently, we can introduce CNN, KNN, and SVM as efficient classifiers in this study, which the performance of FBCSP-CNN is the best between our proposed methods. By this algorithm, the speed of M4, M5, and M1 movements with 90%, 86.25%, and 83.13%, respectively, are better classifiable than other movements. For detailed analysis, Cohen's Kappa coefficients of the proposed algorithms for classifying the speed of movements were drawn in Fig. 10 and Fig. 11. Kappa coefficients were achieved between 0.6 and 0.8 using the FBCSP-CNN method and between 0.5 and 0.7 using the W-CSP-CNN method.

Table 3 shows the average speed of fast and slow movements for all subjects after filtering and averaging from below the acceleration diagram in X, Y, and Z directions. Also, Table 4 shows the average maximum angle of movements around the X and Y axes for all subjects. We analyzed the values obtained from the MPU6050 module based on Fig. 2 and Fig. 3 because they specify the kind of movement, module position, axes, and rotate direction. The speed values indicate the amount of movement in centimeters per second. On the side of performing movements, the speed values for fast mode are higher than in slow mode. It is remarkable that the difference between the speed values of the movements M1 and M2 is less than other movements. Consequently, it can be said that the speed difference increases in movements that require longer distances to perform. According to the results, acceptable values for the angle and speed of movements have been obtained, but these values are not very accurate. By modifying the module location and using a module that can directly measure movement speed, the results can be improved.

TABLE 3. The average speed of fast and slow movements for all subjects in X, Y and Z directions.

Movement	v_x (cm/s)		v_y (cm/s)		v_z (cm/s)	
	Fast	Slow	Fast	Slow	Fast	Slow
M1	2.09	2.18	**13.26**	13.00	2.68	3.79
M2	**8.13**	7.54	7.76	9.29	5.19	5.49
M3	2.04	3.28	2.07	1.80	**14.42**	12.73
M4	1.84	1.97	**16.52**	13.99	**19.68**	16.67
M5	1.83	1.91	**16.46**	14.29	**18.37**	13.97
M6	14.69	14.41	2.36	2.67	**13.59**	12.78
M7	2.84	3.08	**16.02**	15.03	**26.00**	21.47
M8	3.39	2.21	**13.92**	12.04	**10.71**	9.32

TABLE 4. The average of the maximum angle of movements for all subjects around X and Y aces.

Movement	A_x (º)	A_y (º)
M1	74.79	24.18
M2	51.69	47.50
M3	23.48	91.84
M4	102.50	30.81
M5	98.30	38.18
M6	83.41	88.85
M7	101.06	89.35
M8	124.28	43.81

V. CONCLUSION

In this study, our goal was to design an algorithm that can extract information about the speed of various movements from the subjects' EEG signals and classify them into fast and slow classes. Also, we tried to find the speed and angle values of movements from the MPU6050 module attached to the subject's hand. The FBCSP algorithm in the first proposed method and the W-CSP algorithm in the second proposed method was used for extracting speed features. After selecting the features using the MI method, KNN, LDA, NB, SVM, and CNN classifiers calculated the speed classification accuracy of movements. Comparing the subject-independent classification results showed that the speed of flexion/extension of the shoulder and vertical abduction/adduction of the shoulder are more separable and achieved an accuracy of 90% and 86%, respectively, with the FBCSP-CNN method. Determining the values of speed and position of movements and their relationship with EEG signals can help design wearable robots. In this study, the speed and angle values obtained from the module were not very accurate. These values can be improved using a device with high accuracy in obtaining speed values directly, sending motion information via Wi-Fi to the computer, and connecting the module directly to the subject's hand. Also, to enhance the accuracy of the classifiers, the repetition of slow and fast movements can be increased. In the direction of this study, we can consider motor imagery instead of doing movements, investigate the speed of feet and left-hand movements, and increase two classes to three classes by recording EEG signals when performing movements at normal speed. The future work includes the classification of eight movements using the expansion of the two-class CSP method to multi-class.

REFERENCES

[1] T. Alotaiby, F. E. A. El-Samie, S. A. Alshebeili, and I. Ahmad, "A review of channel selection algorithms for EEG signal processing," *EURASIP J. Adv. Signal Process.*, vol. 2015, no. 1, p. 66, 2015.

[2] J. V. Tranquillo, "Quantitative neurophysiology," *Synth. Lect. Biomed. Eng.*, vol. 3, no. 1, pp. 1–142, Jan. 2008.

[3] D. Purves, *Neuroscience*, 6th ed. New York, NY, USA: Univ. Oxford Press, 2018. [Online]. Available: https://learninglink.oup.com/access/purves-6e

[4] S. Sheykhivand, Z. Mousavi, T. Y. Rezaii, and A. Farzamnia, "Recognizing emotions evoked by music using CNN-LSTM networks on EEG signals," *IEEE Access*, vol. 8, pp. 139332–139345, 2020.

[5] J. Britton, L. Frey, J. Hopp, P. Korb, M. Koubeissi, W. Lievens, E. Pestana-Knight, and E. Louis, *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. Chicago, IL, USA: American Epilepsy Society, 2016.

[6] Z. Tang, H. Yu, C. Lu, P. Liu, and X. Jin, "Single-trial classification of different movements on one arm based on ERD/ERS and corticomuscular coherence," *IEEE Access*, vol. 7, pp. 128185–128197, 2019.

[7] A. Kreilinger, H. Hiebel, and G. R. Muller-Putz, "Single versus multiple events error potential detection in a BCI-controlled car game with continuous and discrete feedback," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 519–529, Mar. 2016.

[8] S. He, R. Zhang, Q. Wang, Y. Chen, T. Yang, Z. Feng, Y. Zhang, M. Shao, and Y. Li, "A P300-based threshold-free brain switch and its application in wheelchair control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 6, pp. 715–725, Jun. 2017.

[9] B. K. Kang, J. S. Kim, S. Ryun, and C. K. Chung, "Prediction of movement intention using connectivity within motor-related network: An electrocorticography study," *PLoS ONE*, vol. 13, no. 1, Jan. 2018, Art. no. e0191480.

[10] F. Cincotti, F. Pichiorri, P. Arico, F. Aloise, F. Leotta, F. de Vico Fallani, J. del R Millan, M. Molinari, and D. Mattia, "EEG-based brain-computer interface to support post-stroke motor rehabilitation of the upper limb," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2012, pp. 4112–4115.

[11] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.

[12] F. Babiloni, A. Cichocki, and S. Gao, "Brain-computer interfaces: Towards practical implementations and potential applications," *Comput. Intell. Neurosci.*, vol. 2007, pp. 1–2, Feb. 2007.

[13] U. Chaudhary, N. Birbaumer, and A. Ramos-murguialday, "Brain-computer interfaces for communication and rehabilitation," *Nature Rev. Neurol.*, vol. 12, no. 9, pp. 513–525, 2016.

[14] D. Bandara, J. Arata, and K. Kiguchi, "A noninvasive brain–computer interface approach for predicting motion intention of activities of daily living tasks for an upper-limb wearable robot," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 2, Apr. 2018, Art. no. 172988141876731.

[15] E. Hortal, A. Úbeda, E. Iáñez, E. Fernández, and J. M. Azorín, "Using EEG signals to detect the intention of walking initiation and stop," in *Proc. Int. Work-Conf. Interplay Between Natural Artif. Comput.* Cham, Switzerland: Springer, 2015, pp. 278–287.

[16] Z. Jin, G. Zhou, D. Gao, and Y. Zhang, "EEG classification using sparse Bayesian extreme learning machine for brain-computer interface," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 6601–6609, Oct. 2018.

[17] V. Mishuhina and X. Jiang, "Feature weighting and regularization of common spatial patterns in EEG-based motor imagery BCI," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 783–787, Jun. 2018.

[18] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Frontiers Neurosci.*, vol. 6, no. 1, p. 39, 2012.

[19] A. Suwannarat, S. Pan-Ngum, and P. Israsena, "Comparison of EEG measurement of upper limb movement in motor imagery training system," *Biomed. Eng. OnLine*, vol. 17, no. 1, pp. 1–22, Dec. 2018.

[20] E. López-Larraz, L. Montesano, Á. Gil-Agudo, and J. Minguez, "Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement EEG correlates," *J. NeuroEng. Rehabil.*, vol. 11, no. 1, p. 153, 2014.

[21] S. Bhattacharyya, M. Pal, A. Konar, and D. N. Tibarewala, "An interval type-2 fuzzy approach for real-time EEG-based control of wrist and finger movement," *Biomed. Signal Process. Control*, vol. 21, pp. 90–98, Aug. 2015.

[22] N. Robinson, A. P. Vinod, K. K. Ang, K. P. Tee, and C. T. Guan, "EEG-based classification of fast and slow hand movements using wavelet-CSP algorithm," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2123–2132, Aug. 2013.

[23] S. Bhattacharyya, M. A. Hossain, A. Konar, D. N. Tibarewala, and J. Ramadoss, "Detection of fast and slow hand movements from motor imagery EEG signals," in *Advanced Computing, Networking and Informatics*. Cham, Switzerland: Springer, 2014, pp. 645–652.

[24] E. Alpaydin, "Neural networks and deep learning," in *Machine Learning New AI*. Cambridge, MA, USA: MIT Press, 2016, pp. 85–109.

[25] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[26] B. Xu, R. Cai, Z. Zhang, X. Yang, Z. Hao, Z. Li, and Z. Liang, "NADAQ: Natural language database querying based on deep learning," *IEEE Access*, vol. 7, pp. 35012–35017, 2019.

[27] T. Dobhal, V. Shitole, G. Thomas, and G. Navada, "Human activity recognition using binary motion image and deep learning," *Procedia Comput. Sci.*, vol. 58, pp. 178–185, Dec. 2015.

[28] L. Cheng, D. Li, G. Yu, Z. Zhang, X. Li, and S. Yu, "A motor imagery EEG feature extraction method based on energy principal component analysis and deep belief networks," *IEEE Access*, vol. 8, pp. 21453–21472, 2020.

[29] W. Chen, S. Wang, X. Zhang, L. Yao, L. Yue, B. Qian, and X. Li, "EEG-based motion intention recognition via multi-task RNNs," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 279–287.

[30] B. Xu, L. Zhang, A. Song, C. Wu, W. Li, D. Zhang, G. Xu, H. Li, and H. Zeng, "Wavelet transform time-frequency image and convolutional network-based motor imagery EEG classification," *IEEE Access*, vol. 7, pp. 6084–6093, 2018.

[31] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018.

[32] Z. Tang, C. Li, and S. Sun, "Single-trial EEG classification of motor imagery using deep convolutional neural networks," *Optik*, vol. 130, pp. 11–18, Feb. 2017.

[33] S. Kumar, A. Sharma, K. Mamun, and T. Tsunoda, "A deep learning approach for motor imagery EEG signal classification," in *Proc. 3rd Asia–Pacific World Congr. Comput. Sci. Eng. (APWC CSE)*, Dec. 2016, pp. 34–39.

[34] S. Taheri, M. Ezoji, and S. M. Sakhaei, "Convolutional neural network based features for motor imagery EEG signals classification in brain-computer interface system," *Social Netw. Appl. Sci.*, vol. 2, no. 4, pp. 1–12, 2020.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[36] S. Zhao, Z. Li, R. Cui, Y. Kang, F. Sun, and R. Song, "Brain-machine interfacing-based teleoperation of multiple coordinated mobile robots," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5161–5170, Jun. 2017.

[37] J. Chen, Z. Yu, Z. Gu, and Y. Li, "Deep temporal-spatial feature learning for motor imagery-based brain–computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 11, pp. 2356–2366, Nov. 2020.

[38] Y. Zhang, T. Zhou, W. Wu, H. Xie, H. Zhu, G. Zhou, and A. Cichocki, "Improving EEG decoding via clustering-based multitask feature learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 8, 2021, doi: 10.1109/TNNLS.2021.3053576.

[39] P. Goel, R. Joshi, M. Sur, and H. A. Murthy, "A common spatial pattern approach for classification of mental counting and motor execution EEG," in *Proc. Int. Conf. Intell. Hum. Comput. Interact.* Cham, Switzerland: Springer, 2018, pp. 26–35.

[40] A. Sharmila and P. Geethanjali, "DWT based detection of epileptic seizure from EEG signals using naive Bayes and k-NN classifiers," *IEEE Access*, vol. 4, pp. 7716–7727, 2016.

[41] T. O. Juuso, *Discrete Wavelet Transforms—Theory and Applications*. London, U.K.: IntechOpen. 2011.

[42] S. Kumar, A. Sharma, and T. Tsunoda, "An improved discriminative filter bank selection approach for motor imagery EEG signal classification using mutual information," *BMC Bioinform.*, vol. 18, p. 545, Dec. 2017.

[43] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecolog. Informat.*, vol. 48, pp. 257–268, Nov. 2018.

[44] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks," *Synth. Lect. Comput. Archit.*, vol. 15, no. 2, pp. 1–341, 2020.

[45] W. Gong, H. Chen, Z. Zhang, M. Zhang, R. Wang, C. Guan, and Q. Wang, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," *Sensors*, vol. 19, no. 7, p. 1693, 2019.

[46] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synth. Lectures Comput. Vis.*, vol. 8, no. 1, pp. 1–207, 2018.

[47] S. Zolfaghari, T. Y. Rezaii, S. Meshgini, and A. Farzamnia, "Using convolution neural networks pattern for classification of motor imagery in BCI system," in *Proc. 11th Nat. Tech. Sem. Unmanned Syst. Technol.* Singapore: Springer, 2021, pp. 683–692.

[48] H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," *IEEE Access*, vol. 7, pp. 159081–159089, 2019.

[49] Z. Mousavi, T. Y. Rezaii, S. Sheykhivand, A. Farzamnia, and S. N. Razavi, "Deep convolutional neural network for classification of sleep stages from single-channel EEG signals," *J. Neurosci. Methods*, vol. 324, Aug. 2019, Art. no. 108312.

[50] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.

[51] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, pp. 169–190, 2017.

[52] Y. Jiao, T. Zhou, L. Yao, G. Zhou, X. Wang, and Y. Zhang, "Multi-view multi-scale optimization of feature representation for EEG classification improvement," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 2589–2597, Dec. 2020.

[53] X. Liang, R. Ghannam, and H. Heidari, "Wrist-worn gesture sensing with wearable intelligence," *IEEE Sensors J.*, vol. 19, no. 3, pp. 1082–1090, Feb. 2019.

SEPIDEH ZOLFAGHARI received the B.Sc. degree in biomedical engineering from the Sahand University of Technology, Tabriz, Iran, in 2017, and the M.Sc. degree in biomedical engineering from the University of Tabriz, Tabriz, in 2021. Her research interests include biomedical signal processing, machine learning, and neural networks.

TOHID YOUSEFI REZAII received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering (communication) from the University of Tabriz, Tabriz, Iran, in 2006, 2008, and 2012, respectively. He is currently with the Faculty of Electrical and Computer Engineering, University of Tabriz. His research interests include biomedical signal processing, data compression, compressed sensing, statistical signal processing, pattern recognition-statistical learning, and adaptive filters.

SAEED MESHGINI received the B.Sc. degree in electrical engineering (electronics) and the M.Sc. and Ph.D. degrees in electrical engineering (telecommunication systems) from the University of Tabriz, Tabriz, Iran, in 2005, 2007, and 2013, respectively. He is currently with the Faculty of Electrical and Computer Engineering, University of Tabriz. His research interests include digital signal processing, processing of biomedical signals and images, soft computing and machine learning, pattern recognition and biometrics, and information theory and coding.

ALI FARZAMNIA (Senior Member, IEEE) received the B.Eng. degree in electrical engineering (telecommunication engineering) from Islamic Azad University of Urmia, Iran, in 2005, the M.Sc. degree in electrical engineering (telecommunication engineering) from the University of Tabriz, in 2008, and the Ph.D. degree in electrical engineering (telecommunication engineering) from the Universiti Teknologi Malaysia (UTM), in 2014. Since 2014, he has been a Senior Lecturer (Assistant Professor) with the Electrical and Electronic Engineering Program, Faculty of Engineering, Universiti Malaysia Sabah (UMS). He has secured several research grants with numerous dedicated collaborative research partners. His research interests include wireless communications, signal processing, network coding, information theory, and bio-medical signal processing. He is a member of IET. He is a Chartered Engineer (C.Eng.), U.K.

LIAU CHUNG FAN received the B.Sc. degree in electrical and electronic engineering from the University of Manitoba, in 1992, and the M.Sc. degree in electrical and electronic engineering from the University of Glasgow, in 1998. Since 2000, he has been a Lecturer with the Universiti Malaysia Sabah. His industry experience includes working as an electrical maintenance engineer in a stainless steel factory and an M&E Engineer in a consultancy firm. His research interests include signal and image processing with a focus on its application to communication system and autonomous system perception.