ITC 3/52
Information Technology and Control
Vol. 52 / No. 3 / 2023
pp. 744-760
DOI 10.5755/j01.itc.52.3.33320

Optimized Deep Learning Model Using Modified Whale's Optimization Algorithm for EEG Signal Classification

Received 2023/01/31 | Accepted after revision 2023/05/31

HOW TO CITE: Venu, K., Natesan, P. (2023). Optimized Deep Learning Model Using Modified Whale's Optimization Algorithm for EEG Signal Classification. *Information Technology and Control*, 52(3), 744-760. https://doi.org/10.5755/j01.itc.52.3.33320

Optimized Deep Learning Model Using Modified Whale's Optimization Algorithm for EEG Signal Classification

K. Venu, P. Natesan

Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, 638060, India; e-mail: kvenuresearch22@outlook.com

Corresponding author: kvenuresearch22@outlook.com

Brain-Computer Interface (BCI) is a technology in which Electroencephalogram (EEG) signals are utilized to create a link between a person's mental state and a computer-based signal processing system that decodes the signals without needing muscle movement. The mental process of picturing the movement of a body component without actually moving that body part is known as Motor Imagery (MI). MI BCI is a Motor Imagery-based Brain-Computer Interface that allows patients with motor impairments to interact with their environment by operating robotic prostheses, wheelchairs, and other equipment. Feature extraction and classification are essential parts of the EEG signal processing for MI BCI. In this work, Whales Optimization Algorithm with an Improved Mutualism Phase is proposed to find the optimal Convolutional Neural Network architecture for the classification of motor imagery tasks with high accuracy and less computational complexity. The Neurosky and BCI IV 2a datasets were used to evaluate the proposed methodology. Experiments demonstrate that the suggested technique outperforms other competing methods regarding classification accuracy values at 94.1% and 87.7% for the Neurosky and BCI datasets, respectively.

KEYWORDS: Brain-Computer Interface, Convolution Neural Network, CNN architecture, Motor Imagery, Whale's Optimization.

1. Introduction

Several people worldwide suffer from neural diseases that result in body paralysis. Those kinds of people cannot be able to communicate directly, so Brain-Computer Interface System (BCI) is used to estimate the voltage deviations from the electrodes fixed on the patient's scalp and to classify them as commands. A Brain-computer Interface (BCI) is a system that allows direct communication between the brain and an external device, such as a computer or a prosthetic limb, without requiring any muscle movement or peripheral nerves. Some electronic indicators transform brain signals to outfit classes [36]. This is useful for identifying the patient's intentions, controlling robots, wheelchair operation, and alphabet selections, and used in neuro-prosthesis for the movement of paralyzed legs [6, 19, 22, 24]. BCI uses electroencephalogram (EEG) to record collectively and analyze neuron activity approximately for mobile devices, so it should be analyzed in depth. EEG is applied everywhere because of its simplicity and low cost.

BCI construction has three stages. The first is to accession data from EEG, which is derived from brain signals [36]. The second thing is extracting features from the result of the previous step. The final one is to classify the signals and circulate them to different devices. Several classification techniques are availed to predict the category of motor imagery tasks from EEG signals [18, 23]. The outcomes are helpful in BCI applications for producing command controls. Several devices like robotic devices, intelligent home appliances, and the movement of wheelchairs [15] avail above output. Several classification techniques, such as Random Forest, Naive Bayes, SVM, KNN, and CNN, are availed to classify the EEG data.

Electronic indicators are devices used to measure and detect brain signals, also known as neural activity. These indicators are used to transform the electrical signals produced by the brain into a digital format that can be analyzed by computer software or other electronic devices

Current Scholars are mostly applying CNN for classifications such as object detection [25], audio classification [9], motion prediction [17, 4], disease analysis [29, 17], sentiment analysis [1], and also modern agriculture [27]. Exceeding the other classification technique, Deep CNN results in better milestones [12, 13]. Still, it remains significant to achieve a perfect CNN framework. While creating a CNN framework, several parameters need to be in mind, like the levels of the network, type of each layer, the count and size of the convolution layer, and the relation between layers. The above parameters affect the overall results of CNN, so it is considered to be a difficult task to design the best CNN. Note that trying the same CNN framework for different jobs does not provide the optimal solution, so CNN should be redesigned accordingly for every task. When experts are trying to design a CNN manually for a dataset specifically, it requires several amounts of time and more research knowledge [30]. Developing the CNN framework uses evolutionary algorithms, which becomes an optimization problem. Most researchers use the same evolutionary techniques to obtain optimal neural networks. In this work, a modified Whale Optimization algorithm is used to design CNN architecture.

The contribution of the proposed work is given as follows:

- An optimal convolutional neural network architecture search algorithm WOAIMCNN is proposed.
- A mutual vector is generated in the proposed work to provide an exploitation search for the local information.
- A novel difference and addition operator with a mutualism phase is provided. It enhances the exploration capability of the algorithm without trapping into optimum local value.
- This study involves the utilization of the Whale Optimization Algorithm, which includes an Improved Mutualism Phase.
- The goal of this approach is to efficiently identify the optimal Convolutional Neural Network architecture that can accurately classify motor imagery tasks with minimal computational complexity.

The paper is organized as follows: Section 3 explains the proposed methodology and the WOAIMCNN algorithm. The dataset description, results, and discussion are provided in Section 4. Section 5 concludes the proposed work.

2. Related Works

In 2020, Zhang et al. [41] plan to provide an improved ELM-based technique for EEG categorization in MI-based BCI. Here, the PSO-ELM considerably increases the classification accuracy for both two-class and four-class MI classifications compared to SVM, ELM, and PSO-SVM. In 2020, Chen [7] introduced the FBSF-TSCNN method, which combines filter bank spatial filtering with deep temporal-spatial feature learning to generate a quick and accurate method. Experimental findings on the BCI IV 2a and SMR-BCI datasets, respectively. Demonstrate the usefulness of the recommended approach.

Fan [39] invented QNet in the year 2020, which uses 3D-AM to learn attention weights for channels, time points, and feature maps. On average, the highest level of accuracy was 68%, followed by 74.75 percent and 82.88 percent.

In 2021 [34] Varsehi proposed a unique motor imagery EEG data channel selection technique based on Granger causality analysis. It outperforms the previous correlation-based method. Xiao [28] presented a novel EWT-based technique that outperforms the previous correlation-based method for improving MI task EEG signal categorization accuracy in the context of a motor imagery task. The proposed EWT technique achieves higher classification accuracy than current techniques that use 118 electrode channels, with the IA and IF components achieving average classification accuracy of up to 95.19 percent and 94.60 percent, respectively, for dataset IVa.

For designing Neural Networks, genetic algorithms are used in Neuro Evolution of Augmenting Technology (NEAT) [32]. HyperNEAT was designed by combining NEAT with secondary encoded topology using connected Compositional Pattern Producing Networks (CPPN) [31]. Many researchers have discussed that because of several parameters required for the CNN framework, evolutionary algorithms are not compatible with the computational model of CNN [2]. Google team proposed an algorithm based on GA for the CNN framework, which is applied for all tasks until 2017 and is called Large scale Evaluation of Image Classifier (LEIC) [26]. Another algorithm called EvoCNN based on GA is designed to show CNN construct with an evolutionary algorithm within optimal time [33]. Another scholar proposed an algorithm called IPPSO, Particle Swarm Optimization (PSO) on CNN for simulating IPs for encoding CNN [35]. Francisco designed a search algorithm for the CNN framework called PSO CNN based on PSO [14]. Still, the researchers are more interested in introducing and proving their knowledge by designing an optimal algorithm that provides the best on the CNN framework with the evolutionary network.

The attraction with the performance of the algorithms, this novel work introduced an enhanced method called WOAIMCNN, which is presented by combining formal WOA with a modified mutualism phase. Even though the formal technique is optimal, it lacks less exploration facility, delays convergence speed, and is efficiently confined to a communal result. Formal WOA uses a coefficient vector as a parameter (AC) in the interval between -2 and 2. AC value reduces to 0 from 2 in each iteration; when AC>=1, it is selected as the exploration phase. This guarantees the execution of the exploitation segment inside the second half of the iteration and weakens the exploration phase. Another novel method is a modified SOS algorithm to get optimal solutions by balancing the search process.

In the modern era, SOS updates independently by random independents instead of the overall optimal independents, which increases the heterogeneity of the result. Since the global efficiency of the algorithm is increased, this is useful to jump to the next level in the optimal communal solution. This way, WOA exploration capacity increases, investigations are eventually adjusted, and coupling speed increases.

One of the significant challenges with BCI systems is inter-subject variability. EEG signals can vary significantly between individuals due to differences in brain structure, skull thickness, and other factors. This variability makes it challenging to create a one-size-fits-all BCI system that works equally well for everyone. To address this challenge, researchers have been exploring approaches that can adapt to individual differences in EEG signals, such as personalized training and calibration procedures, and using machine learning algorithms to learn individual-specific features of EEG signals.

Another major challenge with BCI systems is the low signal-to-noise ratio of EEG signals. EEG signals are often contaminated by various types of noise, including muscle artifacts, electrode movement, and environmental interference. This noise can make

it difficult to decode the user's intentions from the EEG signal accurately. To address this challenge, researchers have been developing signal processing techniques, such as filtering and artifact removal methods, to improve the quality of the EEG signal. Additionally, researchers have been exploring novel electrode designs and placement strategies to reduce noise and improve the quality of the EEG signal.

In addition to these challenges, other issues need to be addressed in any BCI-related work, such as real-time processing of EEG signals, ensuring user safety and privacy, and developing user-friendly BCI interfaces. Researchers in the field of BCI are continuously working on developing new techniques and approaches to address these challenges and make BCI systems more practical and accessible for everyday use.

3. Proposed Methodology

3.1. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm copies the Bryde whale's behavior. Like standard methods, WOA optimization also begins with initializing the population. The overall searching technique is split into 3 phases: Search for target prey, surround the prey, and rounded feeding by bubble-net manner. The efficiency of every is based on the stability of local and global techniques; WOA avails the whale technique for exploitation and exploration phase balancing based on some criteria. Final terminations are based on the requirements already defined by fitness value or several iterations.

a Searching the Prey

At the current location, randomly, the whale searches for prey. The same behavior of the Bryde whale is applied to magnify the exploration potential of the algorithm. Here, new CNN architecture generation process is done by using Equations (1)-(2), as is described in Figure 5. The mathematical representation of the equations is as follows:

$$Diff = \left| CV.p_{rand}^{(x)} - p^{(x)} \right| \quad (1)$$

$$p^{(x+1)} = p_{rand}^{(x)} - Q.Diff\,, \quad (2)$$

where P represents the population's position vector which is CNN architecture, $_{\text{Brand}}$ is a population vector randomly chosen from the current population, x denotes current iteration, Diff represents space between the population's current and random individuals, and the operator dot (.) represents multiplication process element by element, | | represents the absolute result. Two coefficient vectors, CV and Q, are defined as in Equations (3)-(4).

$$Q = 2r * rand - r \quad (3)$$

$$CV = 2 * rand\,, \quad (4)$$

where r represents a number that linearly decreases by 2 to 0 in iteration, rand is a random number chosen between intervals.

b Encircling the Prey

In this second phase, the most optimal solution determined in the previous iteration is nearer to the best value. The remaining individuals and inhabitants update their locations close to the current optimal solution. Here, location refers to the layers used in the CNN architecture. New CNN architecture designed using Equations (5)-(6) is described in Figure 5.

$$Diff = \left| CV.p_{best}^{(x)} - p^{(x)} \right| \quad (5)$$

$$p^{(x+1)} = p_{best}^{(x)} - Q.Diff\,, \quad (6)$$

where $_{\text{best}}$ represents the present optimal solution.

c Bubble-net Attacking Strategy

Humpback or Bryde whales stick to bubble-net raids on the helix structure path to attack the prey. New CNN architecture is generated by using Equations (7)-(8), which is described in Figure 4. Same bubble-net technique is followed in WOA while searching, which is represented as:

$$Diff^{*} = p_{best}^{(x)} - p^{(x)} \quad (7)$$

$$p^{(x+1)} = Diff^{*}.e^{sr}.\cos\cos\left(2\pi r\right) + p_{best}^{(x)}\,, \quad (8)$$

where s represents the logarithmic shape of the spiral and it is also a constant value, r is chosen as a random number evaluated by the below given mathematical representation as Equation (9):

$$r = \left(r2 - 1\right) rand + 1\,. \quad (9)$$

The r2 value is linearly decreased between the range -1 to -2 during the iteration process, and the rand is several arbitrary values in the span [0, 1].

During the process of searching, depending on the Q value exploration process is switched to the exploitation process. When Q is positive (Q>=1), the exploration process is activated, which starts global searching based on Equations (1)-(2). Suppose Q is negative (Q<1), alteration of location started by individuals based on Equation (6) or Equation (8). Based on the AC value probability, WOA jumps the process between bubble-net raids for attacking and encircling prey technique. For every strategy, the probability value will be 0.5. The mathematical representation for the above method is represented as Equation (10) as follows:

$$\begin{cases} p^{(x+1)} = p_{rand}^{(x)} - Q.Diff \ \ if \ \ AC \le 0.5 \\ p^{(x+1)} = Diff^{*}.e^{sr}.\cos\cos(2\pi r) + p_{best}^{(x)} if \ \ AC > 0.5 \end{cases}. \quad (10)$$

3.2. Proposed Whale Optimization Algorithm with Improved Mutualism Phase of SOA Optimization Algorithm (WOAIMCNN)

Based on the current new process of searching, populations are initialized, and then every individual's fitness is estimated to determine the optimal global solution. Later in all iterations, the mutation phase is processed initially after the absolute values of the population are updated. For every individual (Pq), choose an individual (Ps) as random from the overall population where Pq is not equal to Ps to interact with Pq. The new CNN architecture is generated using Equations (11)-(12), as described in Figures 3-4. The above mutation is mathematically defined as follows:

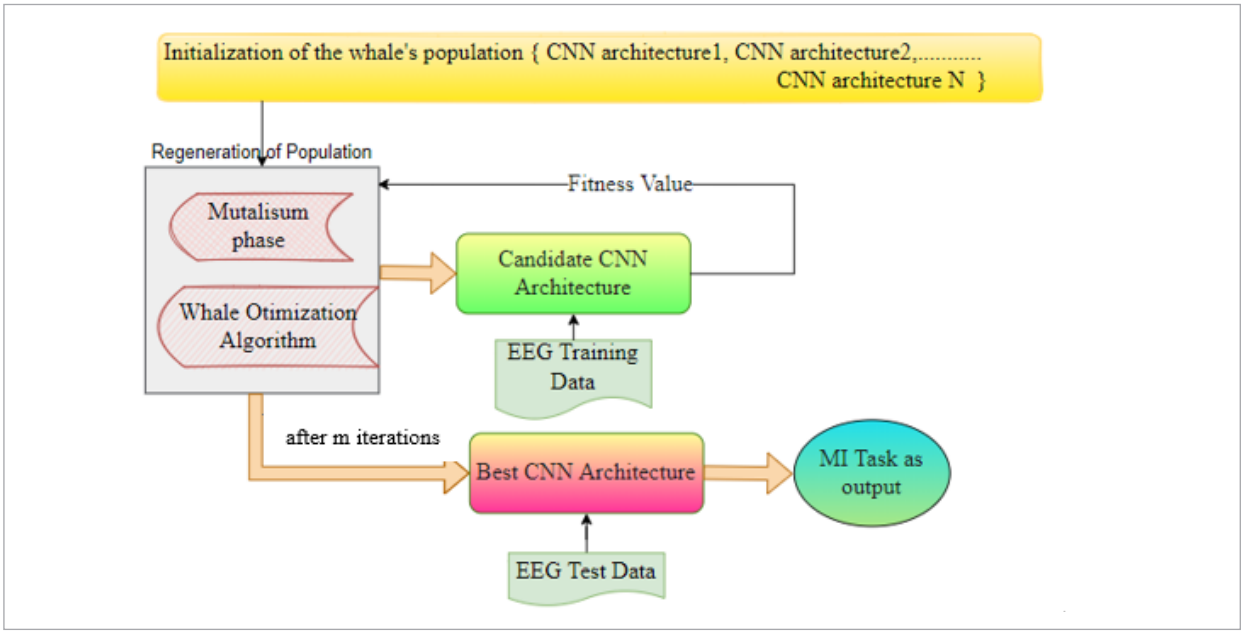
$$P_{qnew} = P_q + rand\ X(P_{Nbest} - MUV\ X\ B1 \quad (11)$$

$$P_{snew} = P_s + rand\ X\left(P_{Nbest} - MUV\ X\ B2\right) \quad (12)$$

$$MUV = P_q + P, \quad (13)$$

where $_{best}$ represents any one population Ps or Pq with high fitness value, MUV shows the advantages both individuals attain from everyone in inaugurating a mutually optimal symbiotic correlation. It is defined by calculating the average result of the above two individuals; B1 & B2 ranges between 1 and 2 is a proportion coefficient by comparing how individuals are obtaining benefits from each other. It is shown in Equation (13).

Figure 1
Overall Architecture Diagram



The changes are done on two individuals among the overall population at a particular time using the optimal individual selected randomly among the two individuals already selected randomly, which enables high diversity in the result. After the enhanced mutation phase global best optimal solution is evaluated and reassigned if it gives the best than the already defined best results; by integrating the enhanced mutation phase, the requirement of exploration capacity of WOA is controlled here. The convergence speed of the algorithm increases by selecting optimal individuals during the current phase and the globally best optimal individuals during the local search. This above process is continued until the best result is obtained according to the termination criteria. The pseudo-code for the proposed method is defined in Algorithm1, and the complete architecture model is drawn in Figure 1. Here, CNN architecture is provided as the whale's population. Based on the population, it is regenerated. From the population the candidate CNN is trained and the fitness value is calculated. WOA computes for 'm' iterations and testing data are performed in the best CNN.

a Candidate CNN Architecture Representation for WOAIMCNN Algorithm

While dealing with most complex architecture models, the important thing is the representation; here, it is CNN architecture Representation for EEG data classification. This novel method introduces a direct encoding method for the CNN model. This work only focuses on the CNN model, which contains three layer types pooling, fully connected, and convolutional. Every individual in a population of WOAIMCNN is represented as shown in Figure 3, which is the candidate model of CNN architecture. The number of layers in candidate CNN architecture varies in the WOAIMCNN process, which may increase individual diversity in the population space.

Algorithm 1. Pseudo-code of the WOAIM algorithm.

Input: whale population S_i (i=1,2,..... Np)
Output: P_{best} – Optimal population

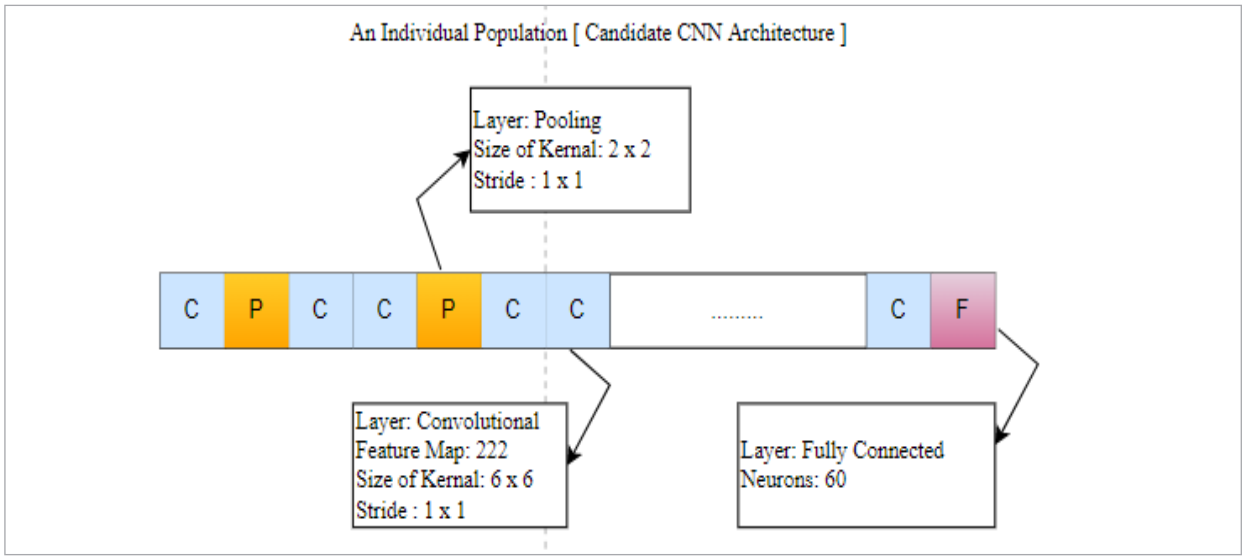
Initialize the whale population S_i (i=1,2,..... Np)
For all searches, the agent calculates the fitness value.
P_{best} -> Search agent with the highest fitness value.
Assign k = 0 and miteration = 20
While (k <= miteration) do
For each Search agent P_q
Choose another search agent P_s randomly where $P_q \neq P_s$
If fitness(P_q) > fitness(P_s) then
$P_{Nbest} = P_q$
Else
$P_{Nbest} = P_s$
Calculate the new value for P_q and P_s using Equation 11 and Equation. 12, respectively.
Update the fitness value for P_q and P_s
End For
Update P_{best}
For each Search agent
Calculate the value for Q, CV,r2, and AC
If (AC< 0.5)
If(|Q| >= 1)
Update the value for the current search agent using Equation 2
Else
Update the value for the current search agent using Equation 6
Else
Update the value for the current search agent using Equation 8
End For
k=k+1
End while
Return P_{best}

b The Initialization of the WOAIMCNN Population

To start the WOAIMCNN process, the input parameters are initialized with the original population value, which is CNN architecture. In Algorithm 2, the initialization process is explained in detail. The initialization of the population is done by using the method followed in [24] and with some improvement to obtain a correct process. The generated CNN model will be an effective architecture with the first layer as a convolutional layer and the final last layer as a fully connected layer with the middle as a pooled layer, convolution layer and fully connected layer provided randomly.

In an integrated coding update model, while generating the CNN model automatically, the position of the fully connected layer should be located after the convolutional and pooling layer. When the fully connected layer is inserted before the other two layers, the final result will have extreme weight in connec-

Figure 2
Representation of the Individual Population



tion. It will lead CNN architecture into a detrimental stage, and two different stages were added to the CNN model calculation. The resultant architecture should be arranged by adjusting the fully connected layer after all other pooling and convolutional layers. Due to consecutive down-sampling stages in the continuous pooling layer, the convolutional layer is added in between the continuous pooling layers. In the integrated update method adaptive span encoding model is used, which will not limit the length of architecture in the CNN model.

The lmax is the maximum count of layers, set for the count of layers on the network during the initialization. Side by side, the network model architecture is generated and re-initialized by considering the WOAIMCNN. Here, the algorithm finds the optimal structure by considering the requirements of the original dataset. An individual from the WOAIMCNN population is an architecture model for a CNN candidate. The best individuals are obtained from the population of WOAIMCNN and reframed as the optimal CNN architecture.

Algorithm 2. Pseudocode for the initialization phase of WOAIMCNN

Input: Np-> Population size, $layers_{max}$ -> Max number of layers, $feature_{max}$ -> Max number of output feature maps, ker_{max} -> Max value for the kernel size, N_{max}-> Max number of Fully connected layer neurons, O_{out} -> Number of output neurons

Output: A population which is the representation of optimized CNN architecture

For i = 1 to Np do
S_i.length = rand(3,length);
Fix the value for max number of pooling layers ($pool_{mzxi}$);
For j = 1 to S_i.length do
If j ==1 then
Layer[j] = addcon(ker_{max},$feature_{max}$);
Else if j== S_i.length then
Layer[j] = addFullC(O_{out});
Else if layer[j-1].typ == "FC" then
Layer[j] = addFullc(N_{max});
Else
Layer_typ = rand(1,3)
If layer_typ ==1 then
Layer[j] = addcon(ker_{max},$feature_{max}$);
Else if Layer_typ == 2 then
Layer[j] = addpooling();
Else
Layer[j] = addFullc(N_{max});
End

For k=2 till s_i do
If ((layer[k].typ ==”pooling”)
&& (layer[k-1].typ==”pooling”))
then
Layer[k] =insert.addcon(ker_{max},$feature_{max}$);
End
S_i.layer = layer
End
Return tpop = {S_1,..... S_{NP}}

c Calculating Mutual Vector

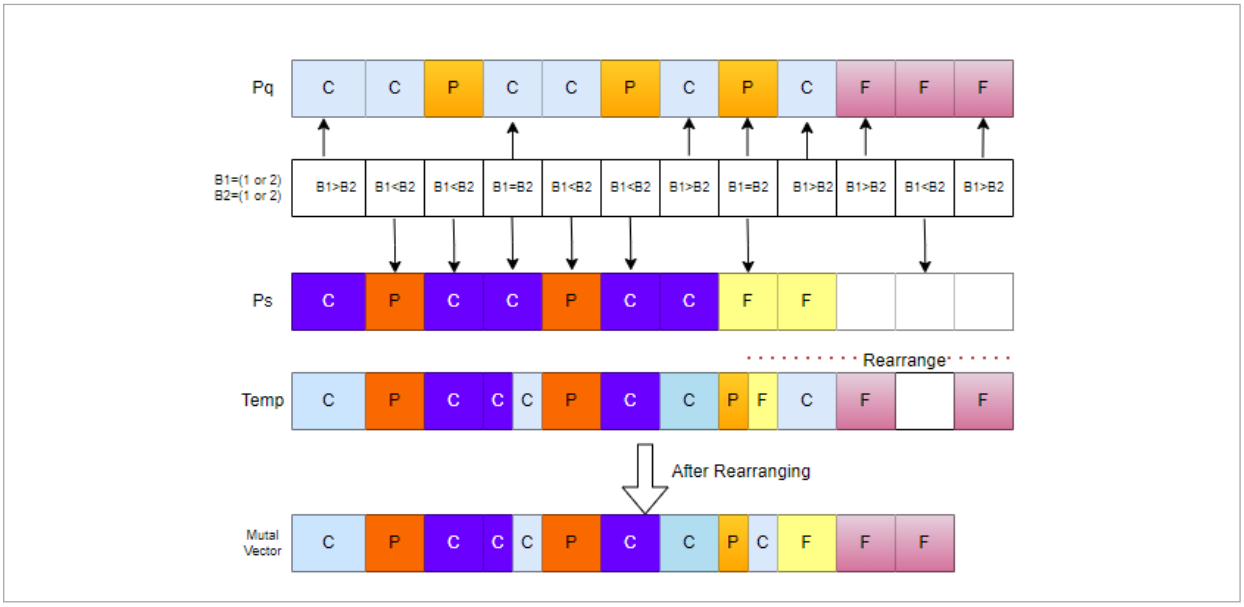
The mutualism technique of SOS mainly focuses on the average of two non-numeric variables. Every individual calculation process is defined in Algorithm 3. This operation is represented pictorially in Figure 3.

Algorithm 3. For calculating Mutual Vector

Input: Population P_q and P_s
Output: Mutual Vector(M), which is similar to the new population

MV_layers = maxlayers(P_q,P_s)
For i in MV_layers, do
B1,B2 <- rand_int(1,2)
If P_q.layer[i] != None and P_s.layer[i]!=None then
If B1 ==B2 Then
Add P_q.layer[i] and P_s.layer[i] to M
Else if B1 < B2 then
Add P_s.layer[i] to M
Else if B1 > B2 then
Add P_q.layer[i] to M
End
Else if P_q.layer[i] != None and P_s.layer[i]=None then
If B1 ==B2 Then
Add P_q.layer[i] twice to M
Else if B1 < B2 then
Add delete flag -1 to M
Else if B1 > B2 then
Add P_q.layer[i] to M
End
Else if P_q.layer[i] = None and P_s.layer[i] !=None then
If B1 ==B2 Then
Add P_s.layer[i] twice to M
Else if B1 < B2 then
Add P_s.layer[i] to M
Else if B1 > B2 then
Add delete flag -1 to M or
do not add anything to M
End
End
Return M

Figure 3
Representation of Mutual Vector Calculation



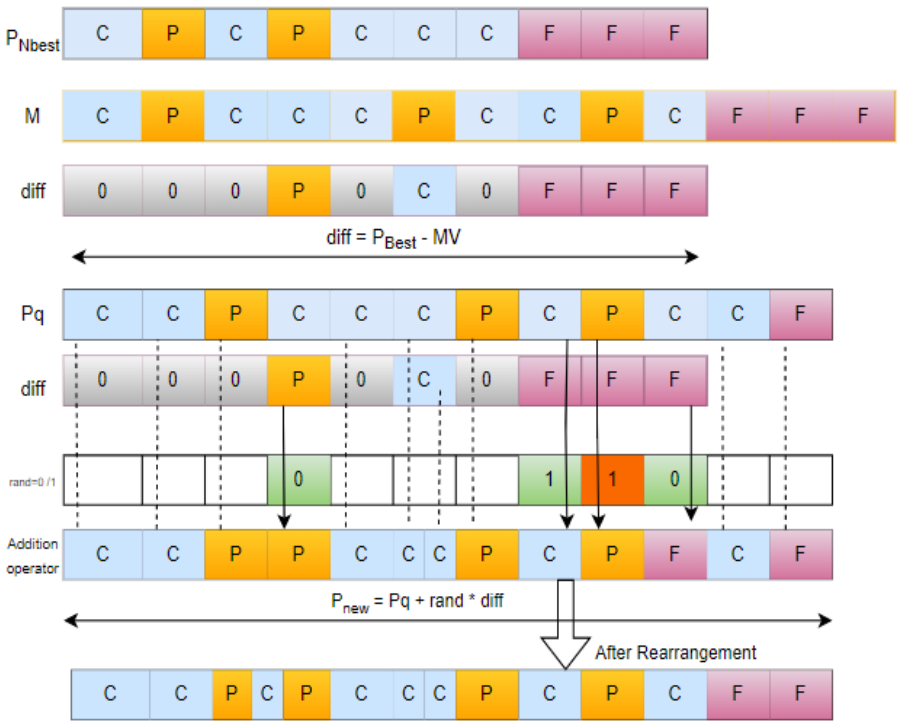
d Steps to Implement the Mutualism phase include Difference Operator and Addition Operator.

To find the difference in Pq value to be changed, the calculation needs to be done to find the difference between the Mutual Vector (M) and the best architecture selected $P_{Nbest.}$ Differential operation is described in Figure 7. The main key point of this proposed novel methodology is dimensions of corresponding block types of two different candidate models are equal, then their difference is determined as 0. When both are different, then the optimal architecture P_{Nbest} dimension is used. On the other hand, when the P_{Nbest} architecture model block is empty, the architecture block difference's corresponding block dimension value will also be empty. Normal logical addition process is done among the candidate Pq and value diff, which is the difference between Pq and P_{Nbest}. The addition operation of calculations done on non-numerical architectural blocks also retains the architectural block's dimension in Pq and Diff on corresponding blocks.

In this way, the diversity of candidate architecture is increased, which is very effective in selecting the optimal architectural model. Algorithm 4 explains the addition and difference operation optimally. To control the gain selector, a threshold value alpha α is chosen and set as 0.5 Figure 4 describes the above operations' addition and the difference clearly.

Figure 4

Examples of the Mutualism phase include difference operator and addition operator



Algorithm 4. Implements the Mutualism phase, including the difference operator and addition operator.

Input : population P_{Nbest}, M, P_q threshold α
Output: new population NP
D_layers = maxlayers(P_{Nbest},M)
For i in D_layers, do
If P_{Nbest}.layer[i] != None then
If P_{Nbest}.layer[i]["typ"]!="remove" then
If M.layer[i]!= None then
If P_{Nbest}.layer[i]["typ"] == M.layer[i]["typ"] Then
Add "0" to D
Else
Add P_{Nbest}.layer[i] to D
End

Else
Add delete flag -1 to D or
Do not add anything to D
End
End
NP_layer= maxlayers(P_q,D)
For i in NP_layer, do
If P_q.layer[i] != None && D.layer[i]== (None/0) Then
Add P_q.layer[i] to NP
Else if P_q.layer[i] != None && D.layer[i] != None Then
If P_q.layer [i]["typ"] == D.layer[i]["typ"] Then
Add P_{Nbest}.layer[i] and D.layer[i] to NP
Else
If rand() < α then
Add D.layer[i] to NP
Else
Add P_{Nbest}.layer[i] to NP
End
Return NP

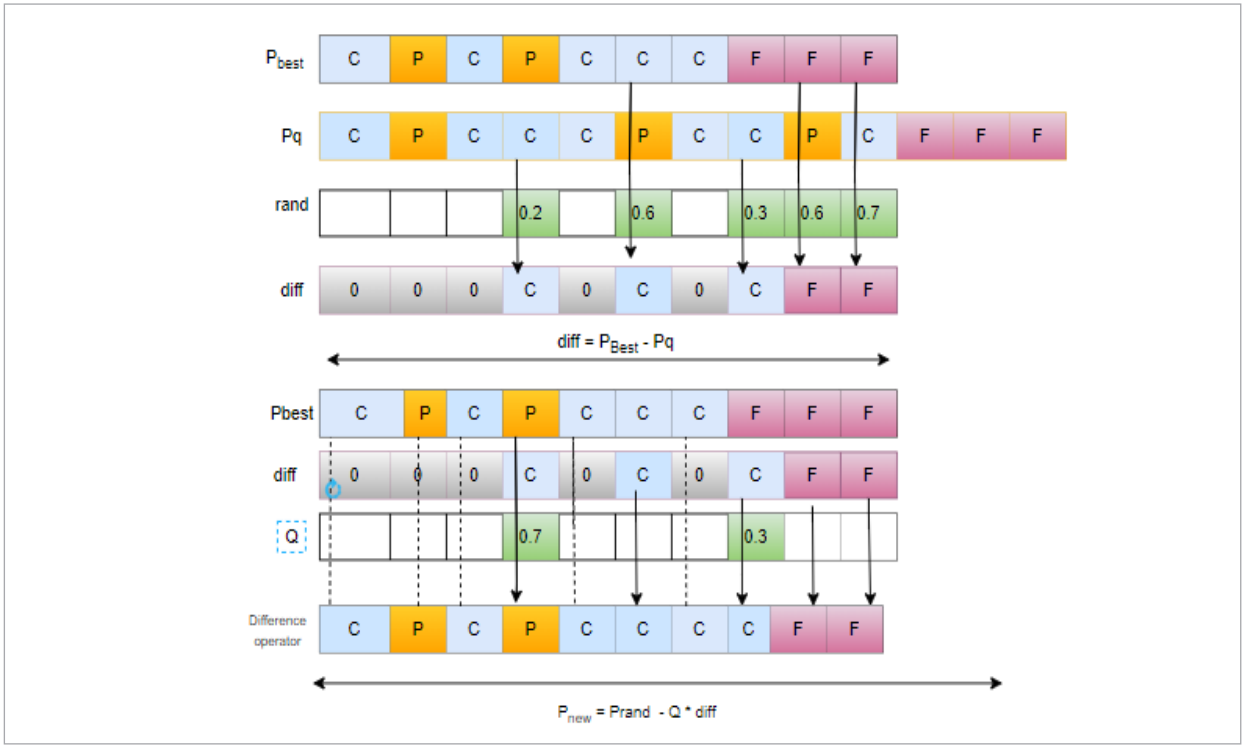
e Calculating the Difference in Architecture using Threshold

The difference between two optimal architectures is calculated by comparing the difference and the PBest architecture. The difference calculation is well defined in Figure 5 and Algorithm 5.

Algorithm 5. For Difference operator with the threshold value.

Input: population P_{Nbest}, P_q threshold α
Output: new population NP
D_layers = maxlayers(P_{Nbest}, P_q)
For i in D_layers, do
If P_{Nbest}.layer[i] != None
If P_{Nbest}.layer[i]["typ"]!="remove" then
If P_q.layer[i]!= None then
If P_{Nbest}.layer[i]["typ"] == P_q.layer[i]["typ"] Then
Add "0" to D
Else
If rand() < α then
Add P_q.layer[i] to D

Figure 5
Example for Difference Operator with Threshold



Else
Add P_{Nbest}.layer[i] to D
End
Add P_{Nbest}.layer[i] to D
End
Else
Add delete flag -1 to D,
End
End
NP_layer= max layers(P_{best},D)
For i in NP_layer, do
If P_{best}.layer[i] != None && D.layer[i]== (None|0) Then
Add P_{best}.layer[i] to NP
Else if P_{best}.layer[i] != None && D.layer[i] != None Then
If P_{best}.layer [i]["typ"] == D.layer[i]["typ"] Then
Add P_{best}.layer[i] to NP
Else
If rand() < α then
Add P_{best}.layer[i] to NP
Else
Add D.layer[i] to NP
End
End
End For
Return NP

4. Results and Discussion

A Dataset Description

1 Neurosky Dataset

From 30 subjects, raw signals from EEG were registered. It takes 5 seconds for data recording in 10 milliseconds duration and three trials for every person. A first-trial person should make a blinking movement, a second should be in focus, a third is to move the eye up, and a fourth movement is an eye down. The set of signals recorded in EEG is of 11 features. To classify the MI tasks following features are availed in our training model Signal Quality, Attention value, Meditation value, delta, theta, low-alpha, high-alpha, low-beta, high-beta, low-gamma, and mid-gamma.

2 MI BCI – IV 2a Dataset

BCI competition IV-2a dataset is one of the most popular motors imagery task data sets. The dataset consists of 4 classes: Right-hand movement, Left-hand movement, Feet, and Tongue movement. There a total of 9 subjects were used to collect the EEG signals. The dataset consists of 72 samples of training and test data in totally.

The algorithm generates the optimal architecture, which provides higher accuracy for the dataset, and it is shown in Table 2. The parameter in architecture is initialized by the value chosen from the range mentioned in Table 1.

Table 1
Parameter Values for the WOAIMCNN Algorithm

Description	Value
WOAIMCNN Algorithm	
No. of Iteration	20
Whale's Population size	15
CNN Architecture	
Min, Max Number of Feature Maps	3,280
Min, Max Number of Kernal size	3x3, 7x7
Min. Max Number of Neurons in FC layer	1,300
Min, Mac Number of layers	5,20
CNN Training	
No. of Epoch for Whale Evaluation	1
No. of Epoch for Global Best	100
Dropout Rate	0.5
Batch Normalization Layer	Yes

B Performance Comparison with the Mutualism Phase

The proposed algorithm with mutualism phase and without is compared based on test accuracy and training accuracy in various runs, as shown in Figures 6-7 which represent Neurosky and BCI IV 2a dataset, respectively. The result provided with the mutualism phase is better in all the cases. The mutualism phase provides the exploitation capability of the algorithm, and thus it increases the accuracy. The run in the proposed algorithm means the number of times the entire algorithm is repeatedly executed. As the number

Figure 6
Boxplot for the test accuracies of the Neurosky dataset obtained using the CNN architecture designed using the proposed WOAIMCNN Algorithm

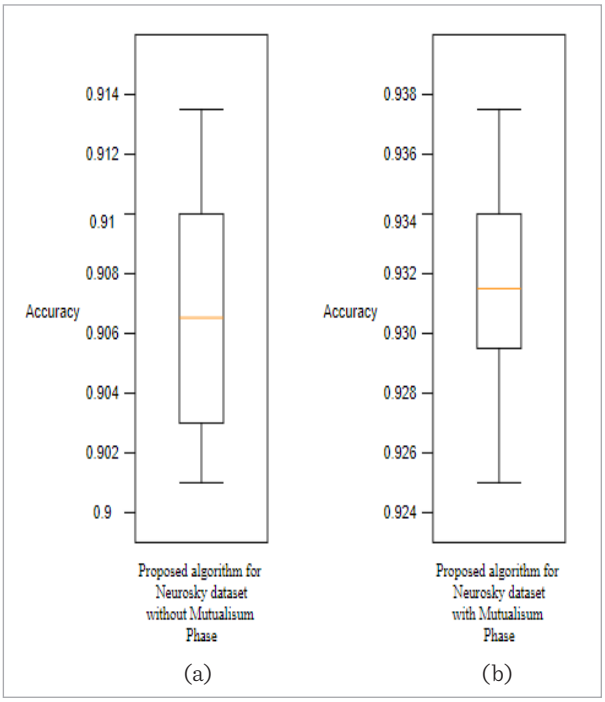
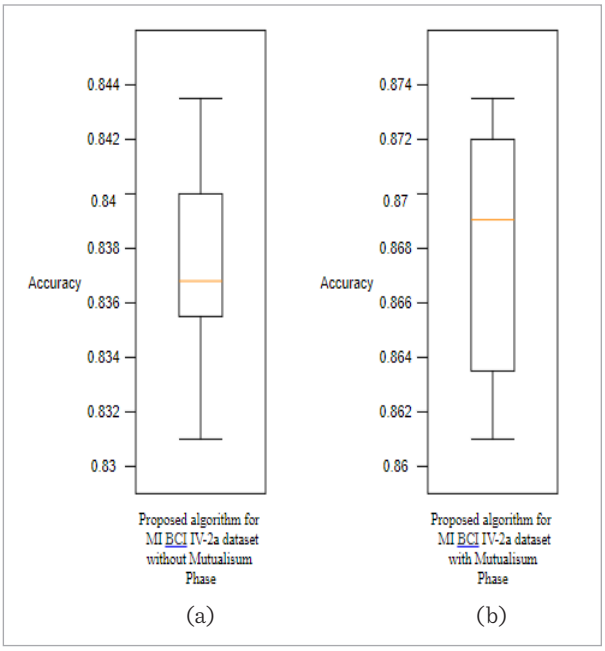


Figure 7
Boxplot for the test accuracies of the MI BCI IV-2a dataset obtained using the CNN architecture designed using the proposed WOAIMCNN Algorithm



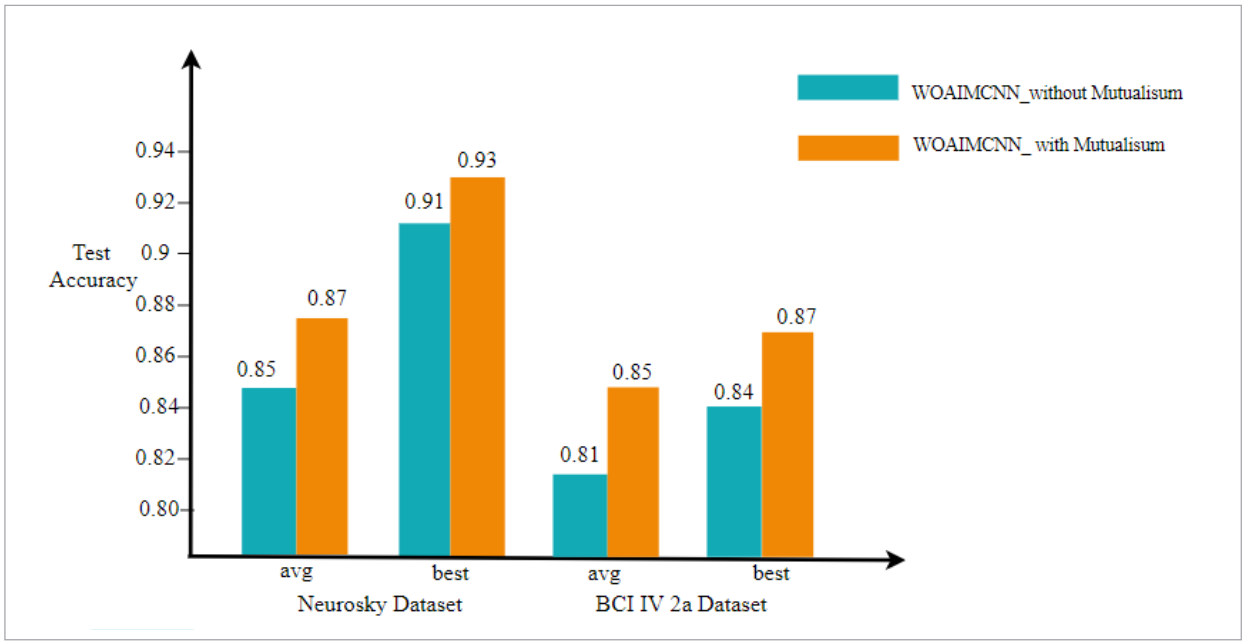
of runs and iterations increases, the probability of finding an optimal architecture increases. The optimal solution is reached at the early stage of iteration. The average and the best accuracy obtained for both dataset is provided in Figure 12.

The layers and their parameter values for each layer of the CNN architecture represented as the population in the proposed algorithm for both datasets is shown in Table 2. The algorithm generates architecture with a shallower network, and an optimal architecture that provides higher accuracy is shown. The proposed algorithm avoids deeper architectures at the initial stage for the population since it will increase the time taken for convergence.

Table 2
CNN Architecture assigned for various populations of WOAIMCNN Algorithm after 5th run

Particle	Architecture for Neurosky dataset	Architecture for BCI IV 2a dataset
Particle 1	conv(30)(5) | max_pool(-1)(2) | conv(249)(5) | avg_pool(-1)(2) | 1) | fc(257)(-1) | fc(5)(-1) | fc(218)(-1) | fc(3)(-1) |	conv(23)(4) | avg_pool(-1)(2) | conv(225)(4) | conv(225)(4) | fc(3)(-1) |
Particle 2	conv(105)(5) | conv(17)(4) | conv(207)(5) | conv(233)(3) | conv(98)(6) | fc(231)(-1) | fc(237)(-1) | fc(143)(-1)	conv(19)(6) | avg_pool(-1)(2) | fc(161)(-1) | fc(3)(-1) |
Particle n	conv(149)(6) | max_pool(-1)(2) | max_pool(-1)(2) | fc(129)(-1) | fc(3)(-1) |	conv(206)(6) | avg_pool(-1)(2) | conv(225)(4) | conv(225)(4) | fc(3)(-1) |
Optimal Architecture	conv(206)(6) | avg_pool(-1)(2) | conv(225)(4) | conv(225)(4) | fc(178)(-1) |	conv(192)(3) | conv(29)(4) | conv(207)(3) | conv(200)(6) | conv(64)(3) | max_pool(-1)(2)| conv(61)(4) | conv(218)(3) | fc(3)(-1) |

Figure 8
Histogram for average and optimal classification accuracy for Neurosky and MI BCI IV-2a Dataset



The accuracy with which identical data or instances are utilized for training and testing is called training accuracy. In contrast, test accuracy refers to the accuracy with which the trained model detects independent data or instances not used in training or training accuracy. Table 3 describes the training and testing accuracy and loss of the Proposed WOAIMCNN-generated architecture for the neurosky dataset. The loss value of a model reveals how well or poorly it performs after each optimization cycle. The algorithm's total performance is evaluated in a standardized manner using an interpretable accuracy metric. Table 4 describes the training and testing accuracy and loss of the Proposed WOAIMCNN-generated architecture for the MI BCI IV 2a dataset.

Table 3
Proposed the Architecture's Training and Testing Accuracy and Loss Values for Neurosky Dataset

Epoch	Training Accuracy	Testing Accuracy	Training Loss	Testing Loss
5	0.47262	0.645	0.6262	0.875
10	0.6174	0.6789	0.7974	0.6389
15	0.8508	0.8264	0.8508	0.8264
20	0.9002	0.8854	0.9002	0.8854
25	0.9103	0.9236	0.9103	0.9236
30	0.9123	0.9236	0.9123	0.9236
35	0.9063	0.941	0.9063	0.941
40	0.9113	0.9258	0.9113	0.8958
45	0.9183	0.9228	0.9183	0.9028
50	0.9104	0.9397	0.9104	0.9097

Table 4
Proposed architecture's training and testing accuracy and loss values for BCI Iva dataset

Epoch	Training Accuracy	Test Accuracy	Training Loss	Test Loss
5	0.7556	0.4889	0.7536	0.8730
10	0.7778	0.8222	0.6051	0.5690
15	0.8444	0.8444	0.4977	0.5520
20	0.8489	0.8411	0.4578	0.4422
25	0.8589	0.8556	0.4330	0.4158
30	0.8778	0.8576	0.3521	0.4415
35	0.8667	0.8556	0.4646	0.3623
40	0.8711	0.8611	0.3851	0.3770
45	0.8733	0.8733	0.3584	0.4069
50	0.8756	0.8771	0.3323	0.3698

Table 5
Comparison of test accuracy with other methods

Method	Methodology	Architecture	BCI IV 2a Dataset
Zhang et al. [41]	CNN (Inception)	5 CONV 2 FC	88.4 ± 7
Liu et al. [21]	CNN (3D) (residual)	10 CONV 3 FC	81.22 ± 6.85
Bang et al. [5]	CNN (3D)	2 CONV 2 FC	87.15
Deng et al. [10]	CNN	3 CONV 1 FC	78.96
Ha et al. [11]	CNN (multilevel pooling)	4 CONV 2 FC	73.19
Xu et al. [38]	CNN	3 CONV 2 FC	84.57
Liao et al. [20]	CNN	3 CONV 1 FC	74.60
Collazos et al. [8]	CNN (multiple input CNN)	4 CONV 2 FC	71.2 ± 7.0
Zhang et al. [3]	Hybrid: CNN/ LSTM	3 CONV 3 LSTM-L	84
Amin et al. [37]	Hybrid: CNN/ MLP	5 CONV 4 FC MLP	75
Proposed	**CNN**	**7 CONV 1 MAXPOOL 1 FC**	**87.71**

C Performance Comparison between the WOAIMCNN Algorithm and the State of Art Algorithms.

The following section compares the result obtained using the proposed architecture with other deep learning techniques. Table 5 presents the test accuracy obtained using the proposed architecture for BCI IV – 2a dataset and compared with different architectures proposed in various papers. The result obtained using the proposed work provides the development equally with the competitors. It gives the best outcome for the Neurosky dataset.

Experiments reveal that the recommended strategy surpasses other competing methods in classification accuracy, making it appropriate for boosting the performance of MI-based BCIs.

5. Conclusion

This research paper aimed to improve classification accuracy in Motor Imagery-based tasks using a novel algorithm called WOAIMCNN. Motor Imagery-based Brain-Computer Interface (MI BCI) is a promising technology that enables people with motor disabilities to interact with their environment using robotic prostheses, wheelchairs, and other devices. EEG signal processing, including feature extraction and classification, is a crucial element of MI BCI technology. In this study, the focus was on Eye blinks and hand and leg movements.

The researchers employed a Convolutional Neural Network (CNN) to classify the MI task using the optimized WOAIMCNN algorithm to achieve their goal. The suggested CNN architecture was compared with various other approaches to training and testing data accuracy. The findings showed that the recommended strategy outperformed other methods regarding classification accuracy. Therefore, it can be concluded that the proposed WOAIMCNN algorithm, coupled with the optimized CNN architecture, can significantly enhance the performance of MI-based BCIs. This improvement in classification accuracy will help make MI-based BCIs more practical and effective for individuals with motor disabilities, improving their quality of life.

References

1. Abid, F., Alam, M., Yasir, M., Li, C. Sentiment Analysis through Recurrent Variants Latterly on Convolutional Neural Network of Twitter. Future Generation Computer System, 2019, 95, 292-308. https://doi.org/10.1016/j.future.2018.12.018
2. Abu Khurma, R., Aljarah, I., Sharieh, A., Abd Elaziz, M., Damaševičius, R., Krilavičius, T. A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem. Mathematics, 2022, 10(3), 464. https://doi.org/10.3390/math10030464
3. Amin, S. U., Alsulaiman, M., Muhammad, G., Mekhtiche, M. A., Hossain, M. S. Deep Learning for EEG Motor Imagery Classification Based on Multi-Layer CNN's Feature Fusion. Future Generation Computer System, 2019, 101, 542-554. https://doi.org/10.1016/j.future.2019.06.027
4. Arshad, H., Khan, M. A., Sharif, M. I., Yasmin, M., Tavares, J. M. R. S., Zhang, Y. D., Satapathy, S. C. A Multilevel Paradigm for Deep Convolutional Neural Network Features Selection with an Application to Human Gait Recognition. Expert Systems, 2022, e12541.
5. Bang, J. S., Lee, M. H., Fazli, S., Guan, C., Lee, S. W. Spatiospectral Feature Representation for Motor Imagery Classification Using Convolutional Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 2021, 1-12.
6. Cervera, M. A., Soekadar, S. R., Ushiba, J., Millán, J. D. R., Liu, M., Birbaumer, N., Garipelli, G. Brain-Computer Interfaces for Post-Stroke Motor Rehabilitation: A Meta-Analysis. Annals of Clinical and Translational Neurology, 2018, 5(5), 651-663. https://doi.org/10.1002/acn3.544
7. Chen, J., Yu, Z., Gu, Z., Li, Y. Deep Temporal-Spatial Feature Learning for Motor Imagery-Based Brain-Computer Interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, 28(11), 2356-2366. doi: 10.1109/TNSRE.2020.3023417. https://doi.org/10.1109/TNSRE.2020.3023417
8. Collazos-Huertas, D. F., Alvarez-Meza, A. M., Acosta-Medina, C. D., Castano-Duque, G. A., Castellanos-Dominguez, G. CNN-based Framework Using Spatial Dropping for Enhanced Interpretation of Neural Activity in Motor Imagery Classification. Brain Informatics, 2020, 7(1), 1-13. https://doi.org/10.1186/s40708-020-00110-4
9. Costa, Y. M. G., Oliveira, L. S., Silla, C. N. An Evaluation of Convolutional Neural Networks for Music Classification using Spectrograms. Applied Soft Computing, 2017, 52, 28-38. https://doi.org/10.1016/j.asoc.2016.12.024
10. Deng, X., Zhang, B., Yu, N., Liu, K., Sun, K. Advanced TSGLEEG Net for Motor Imagery EEG-Based Brain-Computer Interfaces. IEEE Access, 2021, 9, 25118-25130. https://doi.org/10.1109/ACCESS.2021.3056088
11. Ha, K. W., Jeong, J. W. Temporal Pyramid Pooling for Decoding Motor-Imagery EEG Signals. IEEE Access, 2021, 9, 3112-3125. https://doi.org/10.1109/ACCESS.2020.3047678
12. He, K., Zhang, X., Ren, S., Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on

Imagenet Classification. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, 1026-1034. https://doi.org/10.1109/ICCV.2015.123

13. Ioffe, S., Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. International Conference on Machine Learning, 2015, 448-456.
14. Junior, F. E. F., Yen, G. G. Particle Swarm Optimization of Deep Neural Networks Architectures for Image Classification. Swarm and Evolutionary Computation, 2019, 49, 62-74. https://doi.org/10.1016/j.swevo.2019.05.010
15. Katona, J., Ujbanyi, T., Sziladi, G., Kovari, A. Speed Control of Festo Robotino Mobile Robot Using NeuroSky MindWave EEG Headset-based Brain-Computer Interface. In 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, 000251-000256. https://doi.org/10.1109/CogInfoCom.2016.7804557
16. Khan, M. A., Akram, T., Zhang, Y. D., Sharif, M. Attributes Based Skin Lesion Detection and Recognition: A Mask RCNN and Transfer Learning-Based Deep Learning Framework. Pattern Recognition Letters, 2021, 143, 58-66. https://doi.org/10.1016/j.patrec.2020.12.015
17. Khan, M. A., Zhang, Y. D., Khan, S. A., Attique, M., Rehman, A., Seo, S. A Resource Conscious Human Action Recognition Framework Using 26-Layered Deep Convolutional Neural Network. Multimedia Tools Application, 2020. https://doi.org/10.1007/s11042-020-09408-1
18. Lakshmi, M. R., Prasad, T. V., Prakash, D. V. C. Survey on EEG Signal Processing Methods. International Journal of Advanced Research in Computer Science and Software Engineering, 2014, 4(1).
19. Lazarou, I., Nikolopoulos, S., Petrantonakis, P. C., Kompatsiaris, I., Tsolaki, M. EEG-Based Brain-Computer Interfaces for Communication and Rehabilitation of People with Motor Impairment: A Novel Approach of the 21st Century. Frontiers in Human Neuroscience, 2018, 12, 18. https://doi.org/10.3389/fnhum.2018.00014
20. Liao, J. J., Luo, J. J., Yang, T., So, R. Q. Y., Chua, M. C. H. Effects of Local and Global Spatial Patterns in EEG Motor-Imagery Classification Using Convolutional Neural Network. Brain-Computer Interfaces, 2020, 7(3-4), 47-56. https://doi.org/10.1080/2326263X.2020.1801112
21. Liu, T., Yang, D. A Densely Connected Multi-Branch 3D Convolutional Neural Network for Motor Imagery EEG Decoding. Brain Science, 2021, 11(2), 197. https://doi.org/10.3390/brainsci11020197
22. Mudgal, S. K., Sharma, S. K., Chaturvedi, J., Sharma, A. Brain-Computer Interface Advancement in Neurosciences: Applications and Issues. Interdisciplinary Neurosurgery, 2020, 20, 8. https://doi.org/10.1016/j.inat.2020.100694
23. Nakayama, K., Inagaki, K. A Brain-Computer Interface based on a Neural Network with Efficient Preprocessing. In 2006 International Symposium on Intelligent Signal Processing and Communications, 2006, 673-676. https://doi.org/10.1109/ISPACS.2006.364745
24. Rashid, M., Sulaiman, N., Majeed, A., Musa, R. M., Ab Nasir, A. F., Bari, B. S., Khatun, S. Current Status, Challenges, and Possible Solutions of EEG-based Brain-Computer Interface: A Comprehensive Review. Frontiers in Neurorobotics, 2020, 14, 35. https://doi.org/10.3389/fnbot.2020.00025
25. Rashid M., Khan M. A., Alhaisoni M., Wang S. H., Naqvi S. R., Rehman A., Saba T. A Sustainable Deep Learning Framework for Object Recognition Using Multi-Layers Deep Features Fusion and Selection Sustainability, 2020, 12. https://doi.org/10.3390/su12125037
26. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., Le, Q. V., Kurakin, A. Large-Scale Evolution of Image Classifiers. In Proceedings of the 34th International Conference on Machine Learning, 2017, 70, 2902-2911.
27. Rehman, Z. U., Khan, M. A., Ahmed, F., Damaševičius, R., Naqvi, S. R., Nisar, W., Javed, K. Recognizing Apple Leaf Diseases Using a Novel Parallel Real-time Processing Framework Based on MASK RCNN and Transfer Learning: An Application for Smart Agriculture. IET Image Processing, 15(10), 2021, 2157-2168. https://doi.org/10.1049/ipr2.12183
28. Sadiq, M. T., Yu, X., Yuan, Z., Fan, Z., Rehman, A. U., Li, G., and Xiao, G., Motor Imagery EEG Signals Decoding by Multivariate Empirical Wavelet Transform-Based Framework for Robust Brain-Computer Interfaces. IEEE Access, 2019, 7, 171431-171451. https://doi.org/10.1109/ACCESS.2019.2956018
29. Sharif, M. I., Khan, M. A., Alhussein, M., Aurangzeb, K., Raza, M. A Decision Support System for Multimodal Brain Tumour Classification using Deep Learning. Complex Intelligent System, 2021. https://doi.org/10.1007/s40747-021-00321-0
30. Singh, P., Chaudhury, S., Panigrahi, B. K. Hybrid MPSO-CNN: Multilevel Particle Swarm Optimized Hyper Parameters of Convolutional Neural Network. Swarm and Evolutionary Computation, 2021, 63, 100863. https://doi.org/10.1016/j.swevo.2021.100863

31. Stanley, K. O., D'Ambrosio, D. B. Gauci, J. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks. Artificial Life, 2009, 15, 185-212. https://doi.org/10.1162/artl.2009.15.2.15202

32. Stanley, K. O., Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. Evolutionary Computing, 10, 2002, 99-127. https://doi.org/10.1162/106365602320169811

33. Sun, Y., Xue, B., Zhang, M., Yen, G. G. Evolving Deep Convolutional Neural Networks for Image Classification. IEEE Transactions on Evolutionary Computing, 2020, 24, 394-407. https://doi.org/10.1109/TEVC.2019.2916183

34. Varsehi, H., Mohammad, S., Firoozabadi, P. An EEG Channel Selection Method for Motor Imagery based Brain-Computer Interface and Neuro Feedback Using Granger Causality. Neural Networks, 2021, 133, 193-206. https://doi.org/10.1016/j.neunet.2020.11.002

35. Wang, B., Sun, Y., Xue, B., Zhang, M. Evolving Deep Convolutional Neural Networks by Variable-Length Particle Swarm Optimization for Image Classification. In 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, 1-8. https://doi.org/10.1109/CEC.2018.8477735

36. Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., Vaughan, T. M. Brain-Computer Interfaces for Communication and Control. Clinical Neurophysiology, 2002, 113, 767-791. https://doi.org/10.1016/S1388-2457(02)00057-3

37. Wu, H. A Parallel Multiscale Filter Bank Convolutional Neural Network for Motor Imagery EEG Classification. Frontiers in Neuroscience, 2019, 13, 1275. https://doi.org/10.3389/fnins.2019.01275

38. Xu, M. Learning EEG Topographical Representation for Classification via Convolutional Neural Network. Pattern Recognition, 2020, 105,107390. https://doi.org/10.1016/j.patcog.2020.107390

39. Yan, X., Fan, Y., Chen, K., Yu, X., Zeng, X. QNet: An Adaptive Quantization Table Generator Based on Convolutional Neural Network. IEEE Transactions on Image Processing, 2020, 29, 9654-9664. https://doi.org/10.1109/TIP.2020.3030126

40. Yang, L., Zhang, J., Wang, X., Li, Z., Li, Z., He, Y. An Improved ELM-Based and Data Preprocessing Integrated Approach for Phishing Detection Considering Comprehensive Features. Expert Systems with Applications, 2021, 165, 113863. https://doi.org/10.1016/j.eswa.2020.113863

41. Zhang, C., Kim, Y, K., Eskandarian, A. EEG-Inception: An Accurate and Robust End-To-End Neural Network for EEG-Based Motor Imagery Classification. Journal of Neural Engineering, 2021, 18(4), 46014. https://doi.org/10.1088/1741-2552/abed81