International Journal of Modern Agriculture, Volume 9, No.3, 2020
ISSN: 2305-7246

Control of External Devices with Thoughts by Using BCI (Brain Computer Interface) for IOT-EEG Calibration System

[1]Kavita Goura, [2]Dr. Kiran Kumari Patil, [3]Singirikonda Rohith, [4]Sheela Laxman
[1](AsstProf,CBIT, PhD Scholar(VTU)
[2](Prof. Reva University,VTU Research Guide)
3(IT resident ,Google)
[4](Junior Engineer, Ford Motor Pvt. Ltd.)
[1]gkavita_cse@cbit.ac.in, [2] kiran.kumari@reva.edu.in, 3rohithsingirikonda@gmail.com,
[4]laxman.sheela251@gmail.com

Abstract

A Brain-Computer Interface (BCI) is a device that translates neuronal information into commands capable of controlling external software or hardware such as a computer, a wheelchair, robotic arm. BCI is used to treat neurological disorders and thus helps in the restoration of sensory and motor functions.

BCIs are used with assisted living devices for individuals suffering from motor or sensory impairments. With BCI, a person with all kinds of disabilities but having a functioning brain can do many things that otherwise are impossible to do. With BCI, a person without performing any muscle movement can do various activities using thoughts in the brain. BCI interface is a direct communication pathway between a human or animal brain and an external device.

BCI devices are of two types: invasive and noninvasive. Here, we have used a noninvasive BCI headset device that uses Electroencephalography (EEG) to monitor and measure brain activity. In the market, devices are available that work on Functional Magnetic Resonance Imaging (fMRI) and Magnetoencephalography (MEG). All the existing devices lack an efficient way to train mental commands, maintain different human brain profiles, and a standard way to integrate with IoT devices.

We have done the work to address the above problem. BCI software for EEG headsets is developed, which trains mental commands with calibration and provides easy integration with IoT devices. The BCI software developed offers an interface which can accurately classify the signals of the brain using advanced techniques from machine learning and deep learning domains. The person is able to control external devices with his thoughts. The EEG headset is used to capture the brain waves of the person (signal acquisition phase) and then it is processed using a machine learning model and deep learning model to classify and interpret it. (in signal processing phase).

We have built a brain controlled computer-mouse that integrates with EEG headset. The person will be able to control the mouse clicks on the desktop with human thoughts.The EEG dataset is examined using a deep learning-based approach and machine learning-based approach. The comparison of the accuracy obtained with various algorithms is shown

Index Terms: BCI, Electroencephalography, Neurological disorder, Functional Magnetic Resonance Imaging (fMRI), Magnetoencephalography

Introduction

BCI (Brain-Computer Interface) acts as a channel for the human brain to communicate with a computer system. Users can control external devices with brain signals rather than with muscles. BCI channel is the only way people affected by motor disabilities can communicate their thoughts. It is beneficial to assist patients with impaired motor functions, such as wholly paralyzed patients with amyotrophic lateral sclerosis. BCI aims to interpret brain activity and convert it into digital form which acts as a command for a computer.

BCI's goal is to improve the quality of one's life, and its full potential is yet to be explored. However, many challenges arise in the development of such systems.

The control of real-world apparatus with human thoughts was considered as one of the fantasy, but with the recent advancements in BCI, such technology is no longer absurd and thus giving scope for development of prime applications.

The main objective of the proposed system is to build a EEG classification model with good intuitive user interface, calibration features and easy integration with IoT devices. The system trains a model to identify patterns in EEG and classifies the EEG signals using advanced techniques from the domains of machine learning and deep learning.

The advanced techniques of deep learning and machine learning such as SVM, Decision trees, Gaussian Naive Bayes, Logistic Regression and Convolutional Neural Networks is used for EEG Classification.

The proposed approach is verified by building a desktop application which is connected to the EEG Headset. Real time EEG data thus collected is used to train, test mental commands and map them to external devices. It enables the user to control external devices with thoughts (brain signals exhibiting the mental states). The application developed predicts the mental commands using machine learning and deep learning algorithms on the EEG signal patterns.

Literature Survey

By measuring frontal symmetry one can know how brain responds while doing various tasks and thinking. Increased left-frontal activity may serve as an index of positive feelings, motivation or related emotions like joy and increased right-frontal activity may serve as an index of withdrawal or related emotion like disgust, fear, sadness. To find the mental state of the person, EEG is used to record electrical activity of the brain by keeping the necessary electrodes on the scalp. The graph is plotted as voltage magnitude against time.

There are various EEG headsets available in the market such as Emotiv Insight, Open BCI's Mark IV etc. Raw EEG signals thus collected from the headsets are preprocessed and then the required features are extracted. Normal or abnormal rhythms are determined by frequency (Hz) of the signals. Waveforms can be classified into , alpha, beta, theta, gamma and delta. Brain wave frequency differs corresponding to the different behaviour and mental states of the brain. Alpha (8 - 13Hz) waves aid overall mental coordination, calmness, alertness, mind/body integration and learning. Beta (13- 30Hz) waves are concerned with behaviour and actions. It is related to the sense of what we see, touch, hear, smell and taste. Theta (4 - 8Hz) waves are generated when a person is in a dream, vivid imagery, intuition. Delta (0.5 - 4Hz) brainwaves are generated in deepest meditation and dreamless sleep. Gamma (30-70Hz) waves are associated with perception and consciousness. The extreme frequency ranges play no significant role in the clinical EEG. The general frequency range of interest lies between 0.1Hz and 100Hz for the classification purpose.

2.1 Existing Solutions

Both invasive BCI and non invasive has its own advantages and disadvantages. Invasive technique has large bandwidth compared to the noninvasive technique but the invasive technique requires surgery to place the electrodes under the skull.

A critical issue for the development of a BCI application is training the users to operate the BCI. Some groups have shown that the subjects can learn to control the brain activity by undergoing training for longer periods so as to generate fixed EEG patterns and that inturn transforms into external actions.

Many of the existing solutions lack efficient training and testing. In the below given implementation, advanced machine learning and deep learning approaches with a good calibration system is used for efficient training , testing and mapping of commands to various devices.

2.2 Related Works

In [1] , the author proposed a novel approach for learning representation that are invariant to inter and intra subject differences as well as to inherent noise associated with EEG data collection from multi-channel EEG time series data and demonstrated its advantages in the context of classification task.The author [2] shows the Classifier Development and performance analysis of K-Nearest Neighbour (K-NN) Classifier on EEG Data. The author [3], shows the EEG signals recorded based on the actual physical movements and Neural network which are used to detect left and right command. [5] primarily focuses on the EEG signal and its characterization with respect to various states of the human body. **[6]** The survey offers valuable information for researchers to find out which signal processing methods have been used for certain ASSC (Automatic Sleep Stage Classification) schemes and discusses their performances and efficiency.

Design Of The Proposed System

3.1 Introduction

The main challenge is correctly recognizing mental states from observed brain activity and constructing a model that is robust to translation and deformation of signal in space, frequency, and time, due to inter-subject and intra-subject differences, as well as signal acquisition protocols.

The system developed gives an opportunity for easy training of mental commands by utilizing the capabilities of deep neural networks for modelling cognitive events from EEG data. The system trains a model to identify patterns in EEG and classifies the EEG signals. The system developed also has a good intuitive User Interface, calibration features and easy integration for IoT devices.

The system acquires the raw EEG time series data from multiple locations and spectral power within three prominent frequency . Then a sequence of topographical maps is combined to form a sequence of 3-channel images which are fed into a recurrent convolutional network for representation learning and classification. During the training phase, EEG data is collected by displaying a moving cube animation to the subject corresponding to a mental command. After training, the system allows to visualize the trained commands by displaying the cube animation. Then the trained commands can be mapped to input of any device.

3.2 Block Diagram

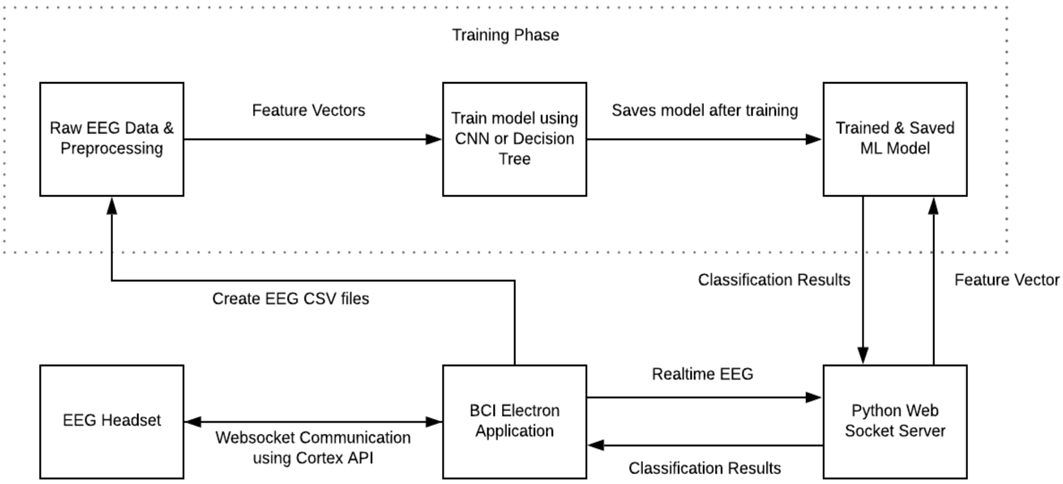
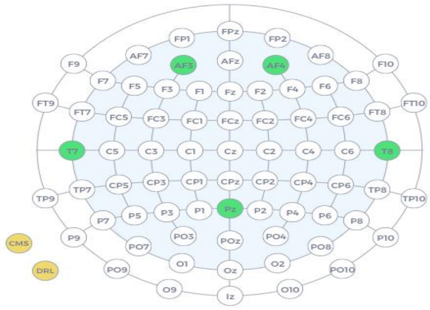


Figure 3.1: Block diagram of the proposed BCI system.

The activities mentioned in the blocks of fig 3.1 are typically carried out in BCI based applications. The following operations are performed by these blocks. : a) EEG Headset: Emotiv Insight - 5 Channel EEG headset (fig 3.2) is used with the channel locations on the International 10-20 system (fig 3.3). b) BCI Electron Application: is a frontend to the user. c) Python web socket server: acts as a EEG processing backend d) Extraction of relevant features is done in Raw EEG data and Pre-processing block e) Training: block performs EEG Classification. As mentioned before, the model is trained by using Deep Learning approach and Machine Learning approach.

	
Figure 3.2: Emotiv Insight - 5 Channel EEG Headset.	**Figure 3.3: Locations of EEG headset sensors on the International 10-20 system.**

3.2.1 Deep Learning approach

The total EEG activity exists in three frequency bands, of theta (4-7Hz), alpha (8-13Hz), and beta (13-20Hz). The sum of squared absolute values within each frequency bands and for each channel are compared and used as a separate measurement.

The standard approach is aggregating spectral measurements for all electrodes to form a feature vector but here we have transformed the measurements into 2-D images to preserve the spatial structure and use multiple colour channels to represent the spectral dimension.

Azimuthal Equidistant Projection (AEP) also known as polar projection is used to transform the spatially distributed activity (electrodes are distributed over the scalp surface) maps as 2-D images. This results in getting three topological activity maps. (one for each frequency band). This 3 channel image is given as input to deep convolutional network.

Recurrent-convolutional neural network is used to handle EEG data. Convolutional Neural Networks are used to deal with variations in space and frequency domains due to their ability to learn good two-dimensional representation of the data.

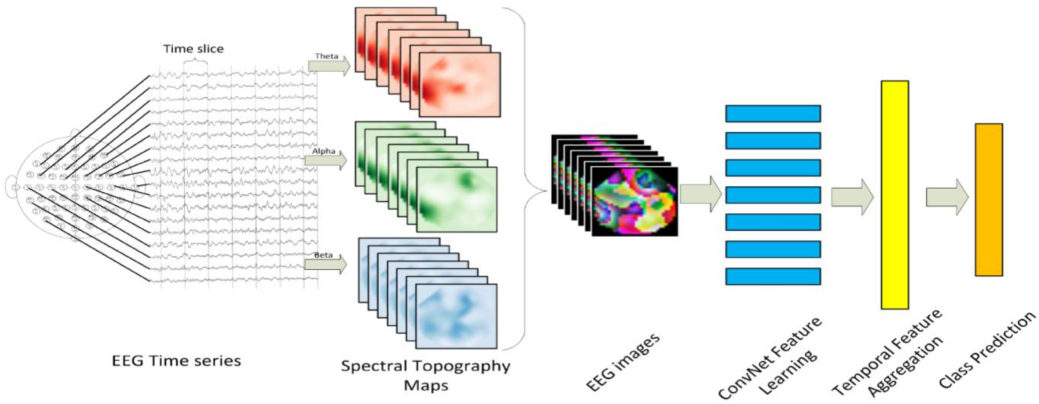


Figure 3.4: Architecture of the deep learning based EEG classification system.

From the Figure 3.4 it can be interpreted that for cognitive state classification, the fast fourier transform can be applied on the raw EEG data collected , to get the spectral power within three frequency bands for each channel. This data is used to construct topographical maps for each band from spectral measurements. Then all the images corresponding to each frequency band will be combined to generate a single image. This process will be repeated for all the time slices. The constructed images can then be used as an input to the recurrent-convolutional network for representation learning and classification. Multi Frame approach is used by dividing each trial into 1 second windows with the overlap of 0.5.

3.2.2 Machine Learning Approach : Aggregation of all the spectral measurements for all electrodes to form a feature vector is proposed. All the five channel data of three frequency bands alpha, beta, theta are transformed to a vector which is given as input to the various classifiers commonly used in this field, like SVM, KNN, Gaussian Naive Bayes, Logistic Regression and Decision Tree.

3.3 User Interface: Users can interact with the application by using user Interface, train and visualize mental commands.

3.4 Theoretical Foundation: A FFT(fast fourier transform) transforms time-domain into the frequency domain representation of signals ie. It changes x-axis of signal from time to frequency, to analyze the different frequencies in a signal. The frequency domain shows the voltage present at varying frequencies. The Fast Fourier is an optimized implementation of a DFT which takes less computational time. It is observed that FFT speeds up the training process in convolutional neural networks.The FFT transform reduces the number of computations needed for a problem of size N from O(N^2) (Discrete Fourier Transform) to O(NlogN).

Implementation Of The Proposed System

BCI application flow chart is shown in the Figure 4.1. Initially the user creates a session in the application where the model, commands and calibration details are stored. The already existing user can resume his session. The new user should go to train mental commands task, to create and train the model. The sequence of tasks or activities are , Raw EEG Data Acquisition, Data Pre-processing, Feature Extraction, Model Building/Training, Testing/Visualization and Calibration, and Mapping of the Commands.

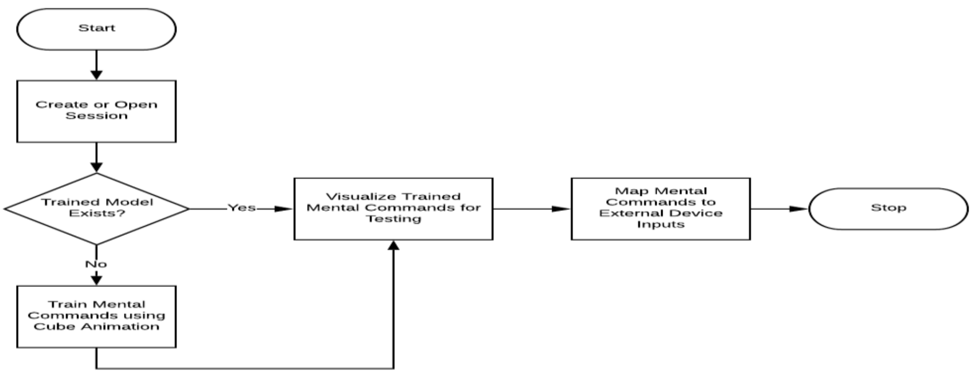


Figure 4.1: Flowchart of the BCI application.

After training, the user navigates to visualization and calibration activity to test the mental commands and adjust the sensitivity to them. The last step is to map the trained mental commands to the external devices that is done in the mapping commands activity.

Here the BCI application is built using Electron, a framework to build desktop applications. It has modules related to data acquisition from EEG headset, training, classification and testing. It connects to the headset via Bluetooth and uses Cortex API for communication. It has command mapping activity, where after commands are visualized and calibrated, they are mapped to external devices whereby the subject can control the external device through thoughts in real time.

Mainly the BCI electron application has the following modules.

4.1 EEG Data Acquisition: Experimental procedure is defined to acquire the data. In this work, a moving cube is displayed to the subject (person) and simultaneously raw EEG signal data is acquired. The person who is considered as subject, is asked to visualize a cube moving to the left or to the right or staying idle for 10 seconds with eyes opened without any facial expression and environmental disturbances. The detailed steps are further explained in the table below.

Action Name	Activity
Left	Subject relaxes his mind(10 sec) and then subject visualizes a cube moving to the left side. The data is recorded and saved. (10sec)
Idle	Subject relaxes his mind(10 sec) and then subject visualizes a cube staying idle without any movements. The EEG data is recorded and saved. (10sec)
Right	Subject relaxes his mind(10sec) and the subject visualizes a cube moving to the right side. The data is recorded and saved (10sec)

Table of experimental actions and their activities.

The data collected consists of voltage amplitudes of 5 channels (AF3, T7, Pz, T8, and AF4), generated from the scalp surface of the human head. It also has a timestamp that indicates the time when the sample was collected.

The raw EEG data from the EEG headset is acquired with cortex API using websocket protocol. The headset we used is of 5 channels and generates 128 samples per second. A sample data acquired from the headset consists of voltage amplitudes of 5 channels (AF3, T7, Pz, T8, and AF4), generated from the scalp surface of the human head. It also has a timestamp that indicates the time when the sample was collected. Spikes were observed in EEG waves when the subject blinks his eyes.

In the training phase, the raw EEG data is acquired in a calm environment ensuring of no external disturbances and also without spikes which may occur due to eye blinks .The acquired EEG data is then scaled using various techniques of scikit-learn such as MinMaxScaler, RobustScaler, StandardScaler and Normalizer. We have found that RobustScalar is good while we have trained the data on various machine learning algorithms.The sample EEG data is as shown in the below figure4.2.

1	title:rohith-b	start timestar	headset ty	headset s	headset f	subject n	channels:
2	Timestamp	EEG.Counter	EEG.AF3	EEG.T7	EEG.Pz	EEG.T8	EEG.AF4
3	1581792126	12	4247.692	4246.154	4128.718	4109.744	4211.282
4	1581792126	13	4243.077	4241.539	4129.231	4101.539	4200.513
5	1581792126	14	4235.897	4259.487	4121.026	4101.026	4190.769
6	1581792126	15	4235.385	4248.205	4110.769	4094.872	4187.18

Fig4.2: Sample Raw EEG dataset

4.2 Data Pre-processing: EEG analysis is done by using the EDF browser. The spikes in EEG waves are observed when the subject blinks his/her eyes. The channels AF3 and AF4 are sensitive to facial expressions and eye blinks. In preprocessing, data with spikes are removed.

4.3 Feature Extraction: After Preprocessing, relevant features that carry significant information are extracted from the EEG data. This data, rich in information is given to the classifier to classify the data into various groups.

The EEG data is divided into 1 second frames as windows with 0.5 second overlap. On each frame, Fast Fourier Transform (FFT) is applied to get frequency bands data. Here make frames procedure is implemented where where 128 samples of data are grouped as snippets with 0.5 overlap and on these snippets, FFT is applied to get alpha, beta and theta values. FFT procedure converts EEG data from time domain to frequency domain. It outputs alpha, beta and theta values by ignoring delta and gamma features as they are only related to sleep, deep meditation, and reduced consciousness.

4.4 Model Building/Training: The frequency bands data of alpha, beta and theta values is used to produce images by generating topographical maps. These images are given as input to the CNN, for classification. CNNs, like any other neural networks, have nodes and are characterized by adjustable weights and biases. The CNN model takes an image as input, processes it and then classifies it. Each image is considered as a multi-dimensional matrix for processing. Each image passes through convolutional layers with filters (kernels), pooling layer and the fully connected layer. Finally, softmax activation is used to classify the image.

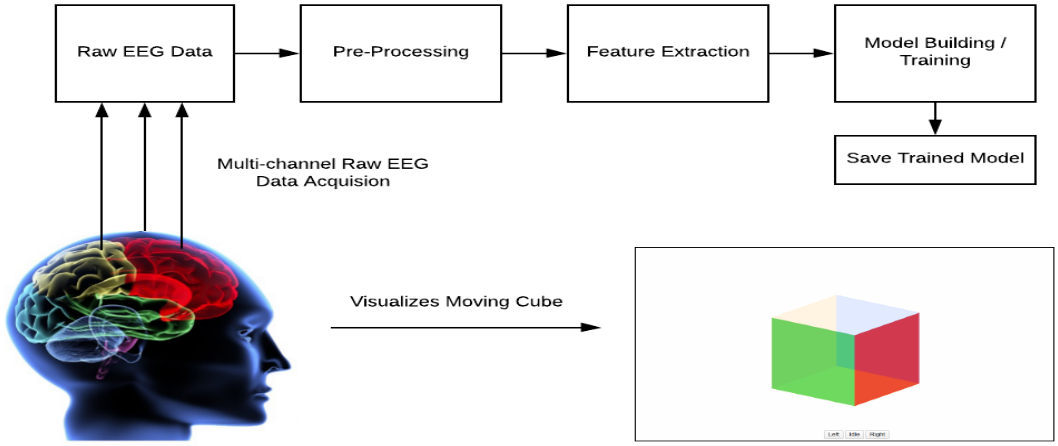


Figure 4.3: EEG Training and Classification Module

The dashboard screen designed, displays the real time statistics from the connected EEG headset. The application has the following webpages (screens) for various activities.1) Dashboard 2) EEG Monitor 3) Train Commands 4) Map Commands 5) Visualize Commands

EEG training and classification module, acquires EEG data from the EEG headset by displaying a moving cube animation. The interface has certain actions like "left", "right" and "idle" which are used to control the cube. The raw data is pre-processed and used for training and classifications as shown in fig 4.3. Visualization and calibration module, enables to test the trained mental commands. The acquired EEG data is sent to Python Web socket server for classification and the results are displayed on the screen with the help of moving cube animation. The module also has a calibration system where the subject can adjust the sensitivity to certain mental commands. In Mental command mapping module, the available mental commands are visualized and calibrated, and are mapped to external devices whereby the subject can control the external device through the thoughts in real time.

Generating images from frequency bands data: After grouping of FFT amplitudes into theta (4-8Hz), alpha (8-12Hz), and beta (12-40Hz) ranges, 2D Azimuthal Projection is carried out where 3 values are projected onto a 2D map of the head.

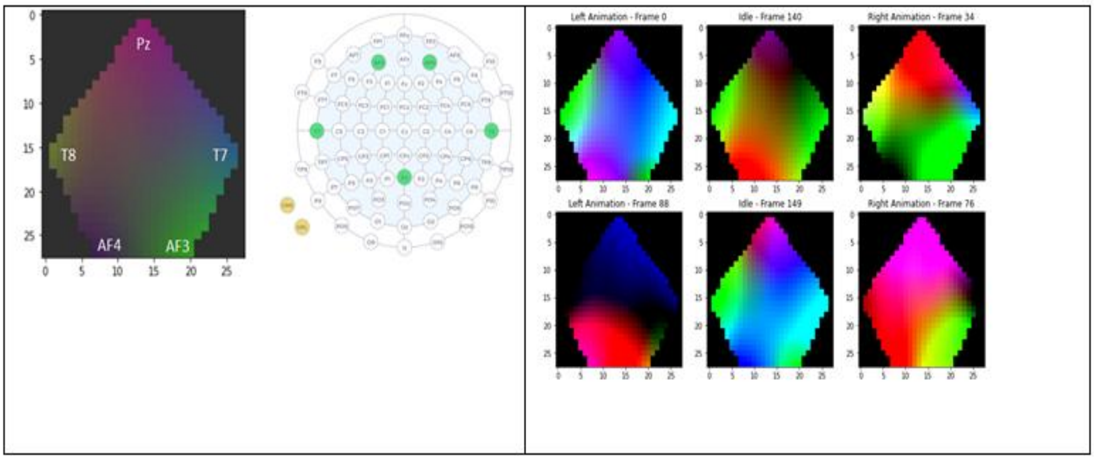


Fig 4.4: Generated Image frame, and location of 5 channels and frames generated from frequency band sets.

Azimuthal Equidistant Projection: The azimuthal projections are formed onto a plane which is usually tangent to the globe at either pole, the Equator, or any intermediate point. In azimuthal equidistant projection, distances from the center of projection to any other point are preserved. Similarly, in our case the shape of the cap worn on a human's head can be approximated by a sphere and the same method could be used to compute the projection of electrode locations on a 2D surface that is tangent to the top point of the head. 3D images are generated corresponding to theta, alpha and beta frequency bands for each frame and 3 images are combined to one image as shown in the figure 4.4.

4.4.3 CNN Architecture

Layers in Typical Convolutional Neural Network are a) Convolutional Layer b) Pooling Layer c) Fully Connected Layer d) ReLU Activation e) Softmax Activation f) RMS Prop Optimizer g) Categorical Cross entropy

We have adopted a generic Convolutional Neural Network architecture on the frames which have 0.5 seconds of overlap between them. In order to reduce the number of parameters in the network, all ConvNets share parameters across frames. We kept the number of neurons in the fully connected layer relatively low to control the total number of parameters in the network.

The neural network that we have used has 2 convolutional layers with relu activation (each layer generating 32 feature maps), 2 dense layers (each layer having 10 neurons and 3 neurons respectively) and uses max pooling prior to passing the output from convolution layers to the dense layers. We have used RMSProp as the optimizer and categorical cross entropy as the loss function to compile our model. The Figure 4.5 depicts the architecture of our convolutional neural network. The implementation of our convolutional neural network is done by using keras framework.

```
print(model.summary())

Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_5 (Conv2D)            (None, 28, 28, 32)        896       
_________________________________________________________________
activation_9 (Activation)    (None, 28, 28, 32)        0         
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 26, 26, 32)        9248      
_________________________________________________________________
activation_10 (Activation)   (None, 26, 26, 32)        0         
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 13, 13, 32)        0         
_________________________________________________________________
flatten_3 (Flatten)          (None, 5408)              0         
_________________________________________________________________
dense_5 (Dense)              (None, 10)                54090     
_________________________________________________________________
activation_11 (Activation)   (None, 10)                0         
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 33        
_________________________________________________________________
activation_12 (Activation)   (None, 3)                 0         
=================================================================
Total params: 64,267
Trainable params: 64,267
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 4.5: Summary of the CNN Model.

4.5 Testing Commands by Visualization: The "Visualize Trained Commands" activity of the BCI application enables the subject to test the trained mental commands in realtime. A cube is displayed to the subject which moves to the left or right upon the user's thought of action. This activity gets the EEG data in real time when the subject, with his mind, recalls the visualization of a cube moving to the left or right of the training phase. The real time EEG is then sent to the python websocket server for classification. The received classification results are displayed on the screen with the help of the moving cube animation.

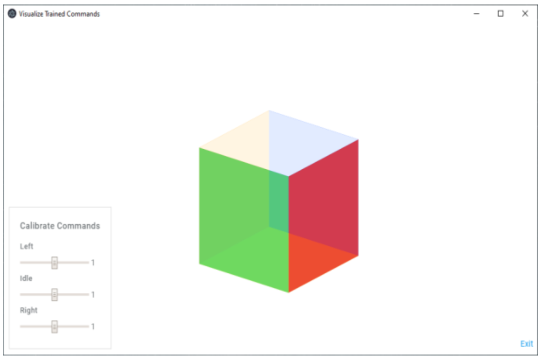


Figure 4.5 Visualize Trained Commands Screen of BCI Application.

Calibration System: The "Visualize Trained Commands" activity has calibration controls which can be used to adjust the sensitivity of trained mental commands. They are useful to overcome the error in action prediction to certain extent. It is implemented by multiplying the adjusted sensitivity (given as input from the subject using range controls) to the model prediction values. After calibration, the adjusted values are saved to the subject profile.

4.6 Mapping of mental commands: After the available mental commands are visualized and calibrated, they are mapped to external devices whereby the subject can control the external device through thoughts in real time. The user can enter shell scripts for each action that will be executed upon the mental command. The scripts are saved to the user's profile. Using this activity, the user can control external IoT devices such as opening a door in a smart home, controlling a smart LED light or moving a robotic arm.

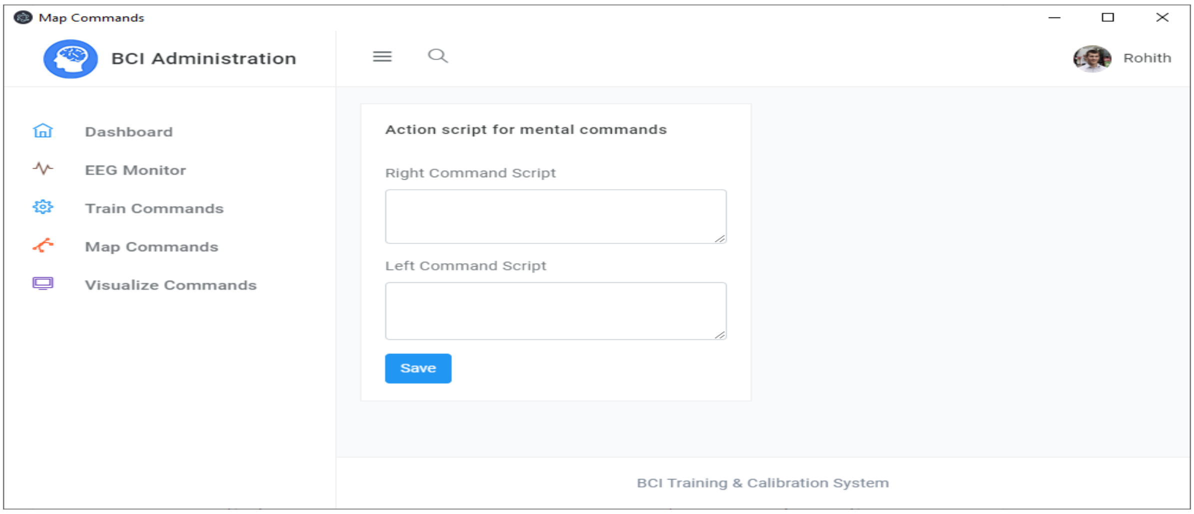


Figure 4.6: Map Commands Screen of BCI Application.

4.7 Websocket Server

The python websocket server acts as a EEG processing backend to the BCI electron desktop application. It receives EEG data through websocket protocol , which after pre-processing is sent to the model for classification. After the classification, the server sends a response back with the classification results. Pandas dataframe is created from the raw EEG data and unnecessary columns are dropped. After preprocessing the generated images are given as the input to the convolutional neural network. The prediction results from the model are sent back as a response.

4.8 Comparison drawn between different Learning Models

We compared our approach against various classifiers used in the field of machine learning such as BCI k-NN, SVM, DecisionTree and Gaussian Naive Bayes. After grouping FFT amplitudes into theta (4-8Hz), alpha (8-12Hz), and beta (12-40Hz) ranges, the data is given as input to our various ML models by splitting 70% as training data and 30% as test data. The results section shows the accuracies achieved.

KNN Cross validation technique

Iteration is used to calculate best value of k in models and average accuracy is returned by using cross validation technique. A cv = 5 folds is used where ,in each iteration , it considers one fold as testing and other four folds as training and iteratively on the next iteration it considers other folds as testing and remaining folds as training set.

Results

Collected EEG datasets from headsets were examined by using two approaches. The first approach is by using deep learning-convolutional neural network to classify the EEG data. The second approach is by using machine learning techniques, K-NN, Decision Tree, SVM and Gaussian Naive Bayes. Comparison drawn between the results thus obtained are shown below. The results given below are for various techniques based on two classes datasets and three classes datasets.

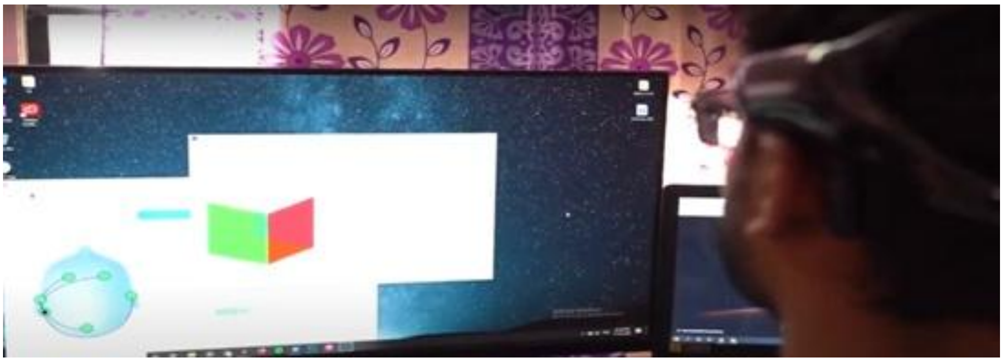


Figure 5.1: Subject controlling the movements of the cube with his mind.

Of all the approaches experimented, the CNN approach gave good results. The CNN model is integrated into the BCI application.

The person is able to control the motion of the cube in the desired direction with thoughts using the BCI application and EEG headset.

Deep Learning Results

● 84% validation accuracy is obtained when the CNN is trained on two classes data corresponding to left cube animation and right cube animation.

● 80% validation accuracy is obtained when the CNN is trained on three classes data corresponding to left cube animation, right cube animation and idle state.

Machine Learning Results

Figure 5.2 shows the accuracy of the various machine learning models:

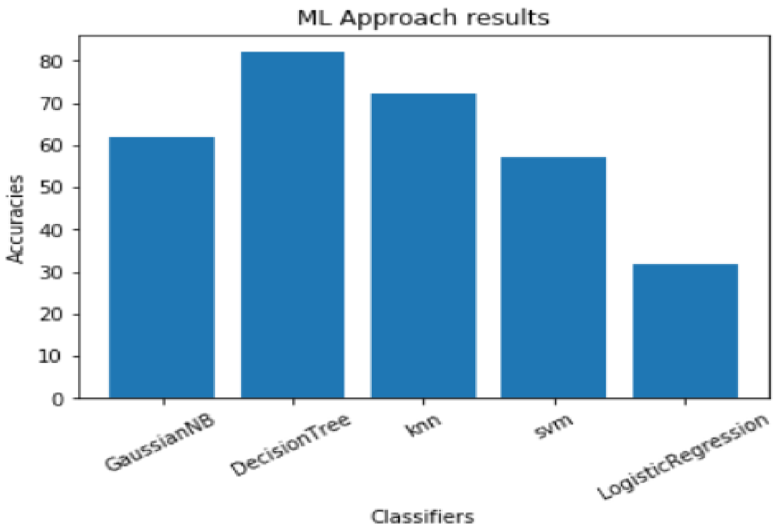


Figure 5.2: Machine Learning results

KNN Cross validation technique result

To know the best value of K, we iterated from the range of 1 to 31, the best value we got for k is 8 with accuracy of 62 using 5 K Folds.

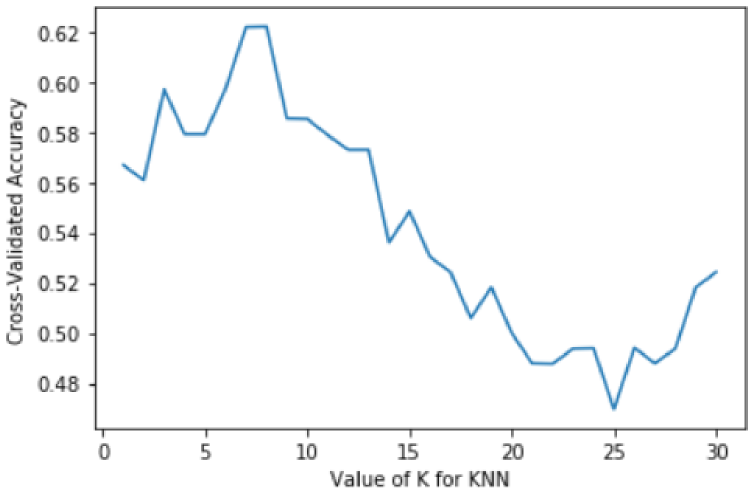


Figure 5.3: Value of K with respect to cross validated accuracy

- 78% accuracy is obtained with k-NN trained on two classes data corresponding to left cube animation and right cube animation.

- 72% accuracy is obtained with k-NN trained on three classes data corresponding to left cube animation, right cube animation and idle state.

Decision Tree Results

- 92% accuracy is obtained with Decision Tree trained on two classes data corresponding to left cube animation and right cube animation.

- 80% accuracy is obtained with Decision Tree trained on three classes data corresponding to left cube animation, right cube animation and idle state.

SVM Results

- 86% is obtained with SVM trained on two classes data corresponding to left cube animation and right cube animation.

- 58% is obtained with SVM trained on three classes data corresponding to left cube animation, right cube animation and idle state.

Gaussian Naive Bayes Results

- 73% is obtained with Gaussian Naive Bayes trained on two classes data corresponding to left cube animation and right cube animation.

- 62% is obtained with Gaussian Naive Bayes trained on three classes data corresponding to left cube animation, right cube animation and idle state.

Conclusion:

The proposed system demonstrates significant improvements in the accuracy of EEG classification by using machine learning and deep learning algorithms. Proposed methodology learns patterns from multi-channel EEG time-series, and demonstrates its advantages in the context of implementing brain computer interfaces.

The EEG signals are subjective, non-gaussian, non-correlated, random in nature and are considered as chaotic signals. Different signal processing techniques like , linear, nonlinear, time domain, frequency, time-frequency and space-time frequency techniques have been experimented. Experimentation and analyzation was done by using EEG Emotiv headset with 5 channels. The training was carried out by using a low-end CPU. High performance with excellent accuracy can be achieved by using high channel (14,32 etc.) EEG headset and by performing extensive training with high-end GPUs. With this, more number of mental states classification can also be achieved.

The research in the field of BCI with electroencephalography is still in the early stages, many challenging and interesting questions in BCI research are thus still waiting to be explored. The models can be improved to identify patterns invariant to inter-subject and intra-subject differences and to inherent noise associated with EEG data collection. The proposed system makes it a good start in identifying patterns in EEG data for BCI applications.

BCI with EEG has these limitations. a) A person needs to have a calm state of mind and a calm environment. The results obtained are inaccurate when there are disturbances in the environment b) A person should not give facial expressions while collecting EEG data, as it causes the EEG headset to move which leads to acquiring incorrect EEG data.

Accuracy of the system can further be enhanced by including REST API services.

References

1. Pouya Bashivan, Irina Rish, Mohammed Yeasin, Noel Codella, "Learning Representations From EEG with Deep Recurrent Convolutional Neural Network", arXiv.org, arXiv:1511.06448, 29 Feb 2016.
2. Isa, Nurul & Amir, Amiza & Ilyas, M. & Razalli, Mohd, "The Performance Analysis of K-Nearest Neighbors (K-NN) Algorithm for Motor Imagery Classification Based on EEG Signal", MATEC Web of Conferences, Volume 140, 2017.
3. N. Hamzah, N.A.M Syukur, N. Zaini, FHK. Zaman, "EEG Signal Classification to detect Left and Right Command using artificial neural networks", AJOL, Volume 9, No. 4S, 2017.
4. Yuanqing Li, Chuanchu Wang, Haihong Zhang and Cuntai Guan, "An EEG-based BCI system for 2D cursor control", 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008.
5. SatheeshKumar, J. & Bhuvaneswari, P, "Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study Procedia Engineering", Volume 38, 2012.
6. Aboalayon, Khald & Faezipour, Miad & Almuhammadi, Wafaa & Moslehpour, Saeid, "Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation", Entropy, Volume 18, 2016.
7. Parth Agrawal, Parul Khanna, Beaulah Soundarabai P, Niju P Joseph, "Electroencephalogram based Brain Computer Interface System Analysis", IJITEE, ISSN: 2278-3075, Volume-9 Issue-3S, January 2020.
8. Choppalli Sruthi, prof. P.Mallikarjuna Rao, "Feature Extraction And Classification Of EEG Signals Using Neural Network", IJERA, ISSN : 2248-9622, Vol. 8, Issue 11, November 2018.
9. P. M. Shende and V. S. Jabade, "Literature review of brain computer interface (BCI) using Electroencephalogram signal", 2015 International Conference on Pervasive Computing (ICPC), 2015, pp. 1-5.
10. V. P. Oikonomou, K. Georgiadis, G. Liaros, S. Nikolopoulos and I. Kompatsiaris, "A Comparison Study on EEG Signal Processing Techniques Using Motor Imagery EEG Data", 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), Thessaloniki, 2017, pp. 781-786.
11. Zhang, Wenchang, Chuanqi Tan, Fuchun Sun, Hang Wu, and Bo Zhang, "A Review of EEG-Based Brain-Computer Interface Systems Design", Brain Science Advances, 4(2), 2019, pp. 156–167.
12. Vaid, Swati & Singh, Preeti & Kaur, Chamandeep, "EEG Signal Analysis for BCI Interface: A Review", Fifth International Conference on Advanced Computing & Communication Technologies, 2015.
13. Hoffmann, Ulrich & Vesin, Jean-Marc & Ebrahimi, Touradj, “Recent Advances in Brain-Computer Interfaces”, Ecole Polytechnique F´ed´erale de Lausanne (EPFL), 2007.
14. K. Saranya, S. Jayanthy, "BCI based EEG Signals for Emotion Classification", International Journal of Recent Technology and Engineering (IJRTE), Volume 7, January 2018.
15. Sharma, Geeta & Sharma, Neha & Singh, Tanya & Agrawal, Rashmi," A Detailed Study of EEG based Brain-Computer Interface." Annals of Computer Science and Information Systems, Volume 14, 2018