ECE1901 – Technical Answers for Real World Problems (TARP)

A Project Report

EEG BASED BRAIN COMPUTER INTERFACE

Submitted by

ELAVARTHI SRUTHI	**(20BEC1028)**
KOLA SAI KISHORE	**(20BEC1224)**
KOPPARAPU GREESHMA LAKSHMI	**(20BEC1228)**
DEVARA HIMABINDU	**(20BEC1353)**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Vandalur – Kelambakkam Road

Chennai – 600127

January 2023

SCHOOL OF ELECTRONICS ENGINEERING

DECLARATION BY THE CANDIDATES

We hereby declare that the Report entitled "**EEG (ELECTROENCEPHALOGRAPHY) BASED BCI (BRAIN COMPUTER INTERFACE)**" submitted by us to VIT Chennai is a record of bonafide work undertaken by me under the supervision of **Dr. Sheena Christabel Pravin**, Professor, SENSE, VIT Chennai.

Sruthi

Elavarthi Sruthi

(20BEC1028)

K. Saikishore

Kola Sai Kishore

(20BEC1224)

Kopparapu Greeshma Lakshmi

(20BEC1228)

Bindu

Devara Himabindu

(20BEC1353)

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our TARP project guide, **Dr**.**M.Jagannath** , School of Electronics Engineering for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias**, Dean of the School of Electronics Engineering (SENSE), VIT University Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of The Department **Dr. Mohanaprasad K**, B.Tech-ECE for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the courses till date.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

BONAFIDE CERTIFICATE

Certified that this project report titled **"EEG (ELECTROENCEPHALOGRAPHY) BASED BCI (BRAIN COMPUTER INTERFACE)"** is the bonafide work of "**Elavarthi Sruthi (20BEC1028), Kola Sai Kishore (20BEC1224), Kopparapu Greeshma Lakshmi (20BEC1228) and Devara Himabindu (20BEC1353)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Sheena Christabel Pravin ,

TARP Supervisor

Assistant Professor,

School of Electronics Engineering,

VIT Chennai.

ABSTRACT

In recent decades, there has been extensive research into the intricate processes involved in brain activity, with the goal of developing a novel communication pathway that utilizes signals detected by EEG electrodes for brain-computer interface purposes.

As neurologically disabled patients got their whole body paralyzed can’t have any body movements and can’t express their views to the outside world. This BCI is a small approach to make their lives much easier. Our project helps the disabled patients to express to the outside world the body part they are trying to move without performing the movement but by analysing their EEG signals. Electroencephalography (EEG) is a non-invasive method used to record electrical activity in the brain.

EEG-based brain-computer interfaces (BCIs) use this technology to translate brain activity into control signals for external devices, without requiring any physical movement. The development of EEG-based BCIs has the potential to revolutionize the way we interact with technology, particularly for individuals with disabilities. Here Brain-Computer Interface (BCI) system that uses four different types of imaginary motor actions to allow individuals to control devices by voluntarily manipulating their brainwaves, specifically their electroencephalogram (EEG). The system is designed to enable individuals to control their EEG through mental imagery, thus supporting device control.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

EEG-based BCI (Brain-Computer Interface) is a technology that allows individuals to communicate or control external devices using their brain activity, as measured by EEG (Electroencephalography) signals. The user performs specific mental tasks or imagines movements, which are associated with specific patterns of brain activity. These patterns are then analyzed by a computer algorithm to classify the user's intention, which can then be used to control a device. However, the technology is still in its early stages and faces challenges such as low accuracy and variability across individuals.

For the classification of user’s intention, we are using an algorithm called ATCNN. This ATCNN model consists of three main blocks:

Convolutional (CV) block encodes low-level spatio-temporal information within the MI-EEG signal into a sequence of high-level temporal representations through three convolutional layers.

Attention (AT) block highlights the most important information in the temporal sequence using a multi-head self-attention (MSA).

Temporal convolutional (TC) block extracts high-level temporal features from the highlighted information using a temporal convolutional layer. ATCNN model also utilizes the convolutional-based sliding window to augment MI data and boost the performance of MI classification efficiently.

1.1 Project Overview

Machine learning is a subfield of artificial intelligence that focuses on the development of algorithms and statistical models that enable computer systems to automatically improve their performance. It involves feeding large amounts of data into an algorithm, allowing it to analyze and identify patterns, and then using those patterns to make predictions or decisions. There are several types of machine learning, each with its own set of techniques and applications. Common applications include image recognition, natural language processing, fraud detection, and recommendation systems.

Deep learning is a subset of machine learning that uses artificial neural networks to model and solve complex problems. It is composed of multiple layers of interconnected nodes that process and analyze data, and can be trained on large datasets. Applications include image and speech recognition, natural language processing, and autonomous vehicles. Deep learning has revolutionized many fields, and is a rapidly growing area of research and development.

Theme: Machine Learning, Deep Learning

1.2 Technology Stack Used

Python IDEs(Integrated Development Environment) provide features such as syntax highlighting, code completion, code navigation, and debugging tools to help code more efficiently and error-free.

TensorFlow is an open-source framework for building and training machine learning models. It provides a variety of tools and APIs for building and training models, such as the TensorFlow Core API, Keras API, TensorFlow Estimator API, and TensorFlow Serving API. To build a machine learning model using TensorFlow, users must define the model architecture, train it on a dataset, evaluate it on a test dataset, and deploy it in production.

Proteus is a simulation software tool used in electronic circuit design and microcontroller simulation. It is widely used to simulate electronic circuits, microcontroller-based systems, and PCB layouts. It includes a schematic capture tool for designing electronic circuits, a virtual testing environment for simulating circuit behavior, and a PCB layout tool for designing printed circuit boards.

1.3 Objectives

Generally, we have seen many patients with spinal cord injuries and cerebrally paralyzed often lose their ability to communicate or control their environment using their limbs due to loss of motor function.

Our project mainly focuses on classification of human motion intentions from non-stationary EEG signals which are collected from these paralyzed patients who cannot move their limbs, this project mainly aims to classify and detect the part that they are willing to move by the collected EEG data.

CHAPTER 2

REVIEW OF LITERATURE

[1] Reshmi G, Amal A (2013), Title of the paper is Design of a BCI system for Piloting a Wheelchair Using Five Class ML based EEG. The workflow followed in this paper follows in this way. EEG signals are acquired, analyzed using wavelet transform and band pass filter and classified using SVM. TAT performs classification tasks by constructing hyper planes in multidimensional spaces. Advantages of this paper is the classified output is fed into the microcontroller which provides an appropriate control signal for the rotation of motors in a smooth and controlled manner. Drawbacks of this paper is for fine analysis, it becomes computationally intensive its discretization the discrete wavelet transform, is less efficient and natural.

[2] Antora Dev, Md. Asifur Rahman, Nursadul Mamun (2018), Title of the paper is Design of an EEG-based Brain Controlled Wheelchair for Quadriplegic Patients. The workflow followed in this paper follows in this way. Using Neurosky headset, EEG signals are recoded and the fast fourier transform is used for extraction of information about alpha and beta waves. With the help of Bluetooth module and Arduino Uno board, wheelchair is operated. Advantages of this paper is Wheelchair can be controlled based on the double eye blink of the patient. Patient will be able to move the wheelchair in four directions. Drawbacks of this paper is the battery of the wheelchair has to be recharged after certain period daily with the help of others.

[3] Chung-Kang Huang, Zuo-Wen Wang, Guan Wei Chen, Chia-Yen Yang (2017), Title of the paper is Development of a Smart Wheelchair with Dual Functions: Real-time Control and Automated Guide. The methodology used here is Emotiv EPOC is used to measure EEG signals. There are two modes- Real-time Control mode based on EEG signals featuring and Automated Guided mode using GPS, movement of wheelchair is determined. It has two main advantages. They are flexibility, and other than manually operating the wheelchair and convenience, it saves their time and energy to operate the wheelchair. Key drawback in this paper is Accuracy is not up to the mark. Automated guide mode is dangerous for large distances.

[4] Shenghong He, Rui Zhang, Qihong Wang, Yang Chen, Tingyan Yang, Zhenghui Feng, Yuandong Zhang, Ming Shao, and Yuanqing Li (2017), Title of the paper is A P300-Based Threshold-Free Brain Switch and Its Application in Wheelchair Control. The workflow followed in this paper follows in this way. EEG signals are acquired using NUAmps device. Graphical User Interface has four buttons acts as pseudo keys and one target key using this movement of wheelchair is controlled. Advantages of this paper is as the SCI patients suffer from neuromuscular disorders and required solution that is brain switch to interact with their environments. It has the control to start/ stop. The main drawback is the patients can move from one place to other but they can get a hold on the objects.

[5] Arjon Turnip, M. Agung Suhendra and Mada Sanjaya W. S (2015), Title of this paper is Brain-Controlled Wheelchair based EEG-SSVEP Signals Classified by Nonlinear Adaptive Filter. The methodology used here is a four choice signal paradigm with different frequencies is used to stimulate the subjects. EEG signal is pre-processed and features are extracted using Adaptive filter, used in noise cancellation and NAF technique is used. Advantages of this paper is one way of gaining further insights into EEG signal is by applying NAF technique. Drawbacks of this paper is difficulty to identify the associated signals with respect to the given stimulus. For the implementation of FIR filter complex computational techniques are required to implement.

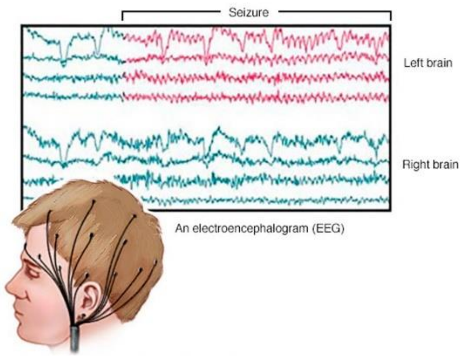
[6] Cesar E. Hernandez-Gonzalez, Juan M Ramırez-Cortes, Pilar Gomez-Gil,Jose Rangel-Magdaleno,Hayde Peregrina-Barreto and Israel Cruz-Vega (2017), Title of the paper is EEG Motor Imagery Signals Classification using Maximum Overlap Wavelet Transform and Support Vector Machine. The workflow followed in this paper follows in this way. EEG signals are extracted and using discrete wavelet transform (DWT) and Maximum overlap discrete wavelet transform (MODWT), alpha, beta, delta and theta waves are acquired. Classification is done by using Support Vector Machine (SVM). Advantages of this paper is the usage of Maximum overlap discrete wavelet transform (MODWT), provides an excellent class separability which helps in feature extraction. Drawbacks of this paper is long training time for large datasets. Difficult to understand and interpret the final model, variable weights and individual impact.

[7] Arjon Turnip, Taufik, Hidayat,, Dwi Esti Kusumandari (2017), Title of the paper is Development of Brain-Controlled Wheelchair Supported by Raspicam Image Processing based Raspberry pi. The methodology used here is Wheelchair is based bio-signal & non-bio-signal approach which translated into movement commands by the Arduino microcontroller is proposed. In bio-signal approach, based on EEG signals movement is controlled. In another approach, ultrasonic sensors and speed controller are attached so that obstacles are identified. Advantages of this paper is beside the use of the brain signals as a control movement direction, the wheelchair also supported by several sensors like ultrasonic sensors for obstacle detection and inertial measurement unit for speed controlling helps to make sure the user safety. Drawbacks of this paper is the accuracy of an obstacle objects is highly affected by the distance and light intensity.

CHAPTER 3

POTENTIAL COMPETITORS

Table 3.1 Potential Competitors

Topic	Picture	Drawbacks
Manual Extraction of EEG signals		1. Time consuming 2. Inconsistency 3. Limited Frequency Range 4. Limited Spatial Resolution
Head Set	 	1. Compatibility 2. Limited Warranty 3. Cost 4. Wear and Tear

CHAPTER 4
METHODOLOGY

4.1 Procedure

The Proposed EEG based BCI system works by the following steps:

Step – 1: Signal acquisition is the process of capturing a signal from a physical system and converting it into a digital form. It is essential for many scientific and engineering applications, as errors or distortions can affect the quality and usefulness of the data. In this project the features are collected from BNCI HORIZON 2020 website.

Step – 2: Pre-processing is an essential step in data analysis and modelling, involving techniques such as cleaning, transforming, scaling, handling missing values, and feature selection. It helps to ensure accurate, consistent, and relevant data.

Step – 3: The next step would be training the model. Here the training model used is ATCNN. The ATCNN model requires expertise in deep reinforcement learning, neural network architecture design, data preprocessing and analysis. Data preparation includes historical air traffic data, preprocessing involves converting raw data into a format, model architecture is based on a deep reinforcement learning network, hyperparameter tuning is critical, loss function is designed to minimize the difference between predicted values and actual values, training is done using the prepared data, optimized hyperparameters, and loss function, validation and testing is done to ensure the model is not overfitting to the training data, and generalization is achieved.

Step – 4: Preparing testing data is essential for ATCNN to make predictions. Loading a pre-trained ATCNN model is necessary, converting the testing data into a numerical representation, running the data through the model, and evaluating its performance.

Step – 5: Next, we will visualize the predicted classes using Proteus. The predicted classes from testing is given as input to raspberry pi which controls the four motors through motor drivers L298D.

4.2 Process Overview

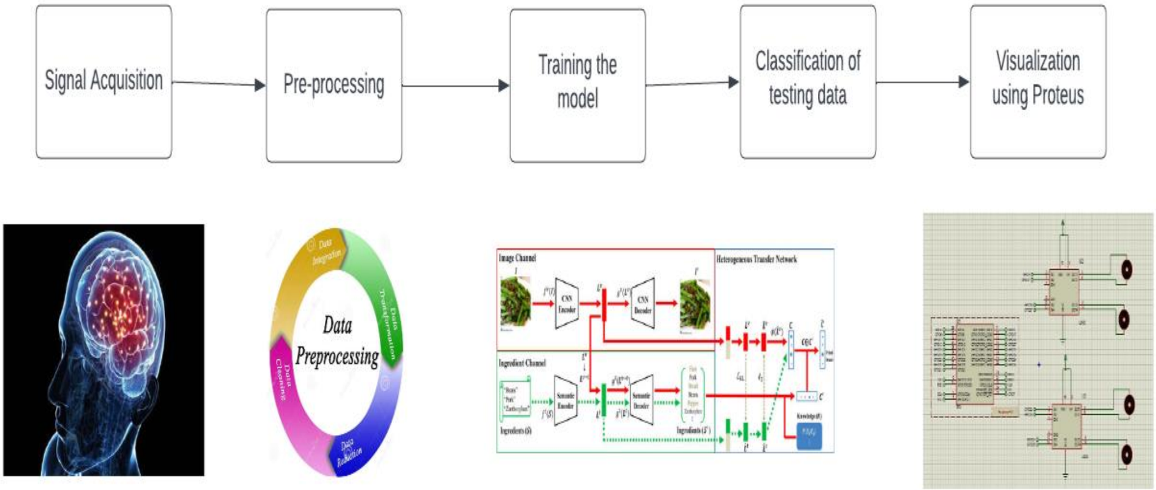


Figure 1: Process Overview

CHAPTER 5

PROPOSED SYSTEM'S ATTRIBUTES

5.1 Data Description

In this project we are using BCI competition iv dataset 2a which has 9 subjects and 4 classes namely right hand, left hand, both foot and tongue. The data is collected at sampling frequency of 250 Hz. They are bandpass filtered between 0.5 Hz to 100 Hz with the 50 Hz notch filter enabled.

The dataset has 25 features in which 22 features are related to EEG signals and 3 features are related to EOG signals which record eye movements.

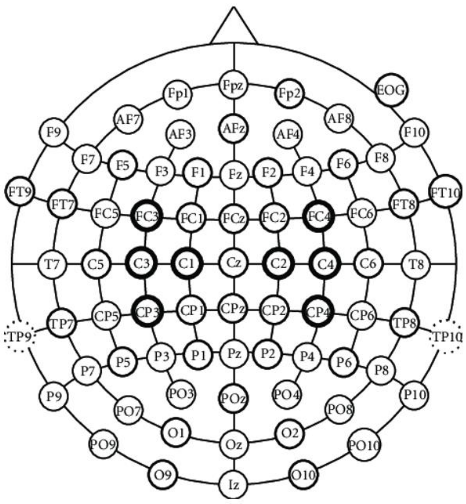


Figure 2: 10-20 electrode system

22 electrodes of this figure are used for recording EEG signals. The specific electrode positions used in this dataset are as follows: Fz, FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, Pz. The electrodes are labeled according to their location on the scalp. "F" refers to frontal, "C" refers to central, "P" refers to parietal, and "z" refers to the midline. Odd numbers (e.g., C3, CP3) refer to positions on the left side of the head, while even numbers (e.g., C4, CP4) refer to positions on the right side of the head. The number indicates the distance from the midline, with lower numbers being closer to the midline and higher numbers being farther away.

5.2 Working Principle

As discussed earlier, two pivotal components namely the ATCNN and the proteus model contributes as a backbone in the proposed project.

5.2.1 ATCNN Model

ATCNN model consists of three main blocks:

Convolutional (CV) block: The input sequence is first passed through a stack of convolutional layers, which helps in learning local patterns within the input sequence. It encodes low-level spatio-temporal information within the MI-EEG signal into a sequence of high-level temporal representations through three convolutional layers.

Attention (AT) block: The attention mechanism in ATCNNs allows the model to selectively focus on the most informative parts of the input sequence, improving the accuracy of the predictions. It highlights the most important information in the temporal sequence using a multi-head self-attention (MSA).

Temporal convolutional (TC) block: extracts high-level temporal features from the highlighted information using a temporal convolutional layer

ATCNN model also utilizes the convolutional-based sliding window to augment MI data and boost the performance of MI classification efficiently.

5.2.2 Proteus Visualization

In this proteus visualization we are using a Raspberry pi, two motor drivers L298D and four dc motors. After giving the classified input to raspberry pi, the motor detects the intensions of the features taken the motor rotates in clockwise direction and in anti-clockwise direction followed by.

5.3 Features

5.3.1 Features of ATCNN Model

Convolutional Layers: ATCNN uses a series of convolutional layers to extract features from the input data, which allows the model to learn patterns and relationships between different sensor signals.

Recurrent Layers: The model also includes recurrent layers, which allow the network to capture temporal dependencies in the data and identify changes over time.

Attention Mechanism: ATCNN uses an attention mechanism to focus on the most relevant parts of the input data and improve the model accuracy.

5.3.2 Features of Proteus Visualization

3D visualization: Proteus includes a 3D viewer that allows users to see their designs in three dimensions. This can be particularly useful for visualizing complex components or assemblies.

Interactive simulation: Proteus allows users to interact with their simulations in real-time, including the ability to modify input values and observe changes in the output.

Debugging tools: Proteus includes a range of debugging tools, including the ability to set breakpoints, step through code, and view variable values.

5.4. Proposed Algorithm Workflow chart

Data pre-processing is the first step of the ATCNN model, which transforms raw data into a format that can be used by the ATCNN model. Input encoding is then done to encode the pre-processed data into a set of input features. A spatial temporal graph is constructed based on the input features and historical flight data. A Graph Convolutional Network (GCN) is then used to process the graph and learn the relationships between the nodes in the graph. Finally, the ATCNN model outputs a set of predicted trajectories for each aircraft, which can be used by air traffic controllers to manage air traffic flow.

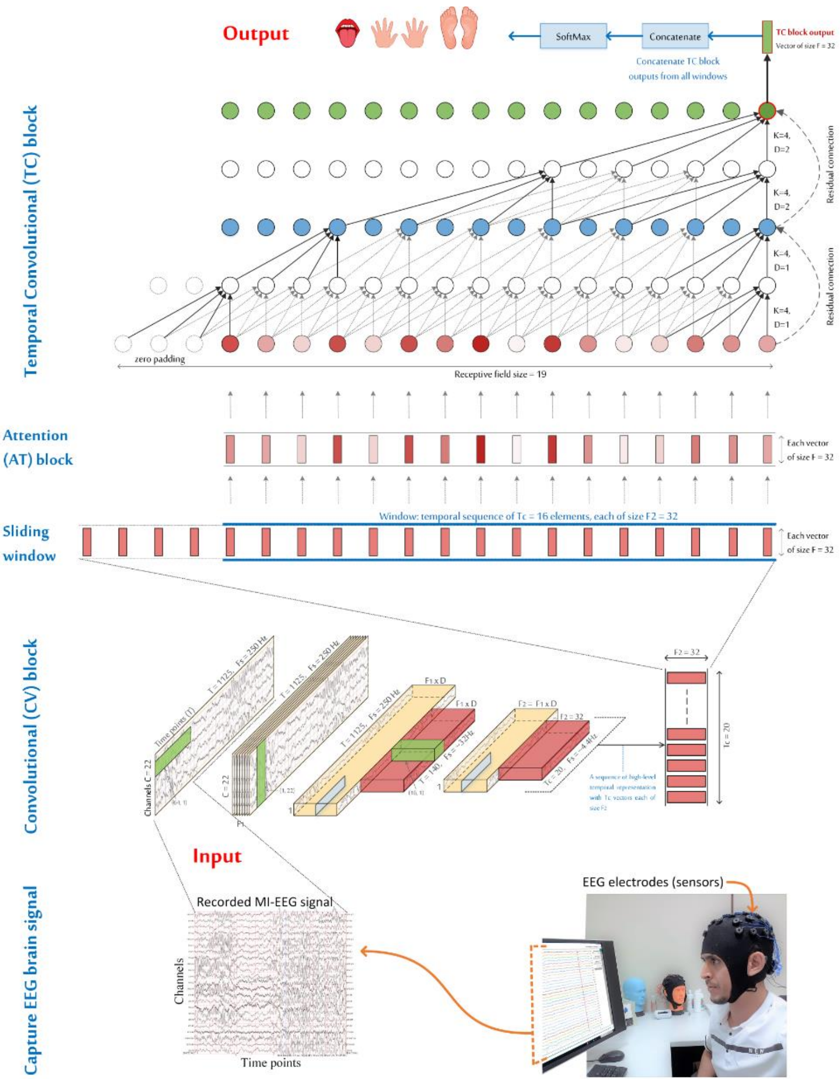


Figure 3 : Proposed Algorithm Workflow Chart

CHAPTER 6
RESULTS AND DISCUSSION

6.1. ATCNN MODEL RESULT

The accuracy of a model trained on EEG (electroencephalogram) data would depend on various factors such as the quality and quantity of data, the preprocessing steps applied to the data, the choice of machine learning algorithm and its hyperparameters, and the evaluation metrics used.

It is also worth noting that EEG data is highly complex and noisy, which can make it challenging to achieve high accuracy. Nonetheless, with careful data preprocessing, feature extraction, and model selection, it is possible to achieve reasonable accuracy.

However, it is essential to assess the performance of the model using appropriate evaluation metrics such as sensitivity, specificity, precision, recall, F1-score, or ROC-AUC, as accuracy alone may not provide a complete picture of the model's performance.

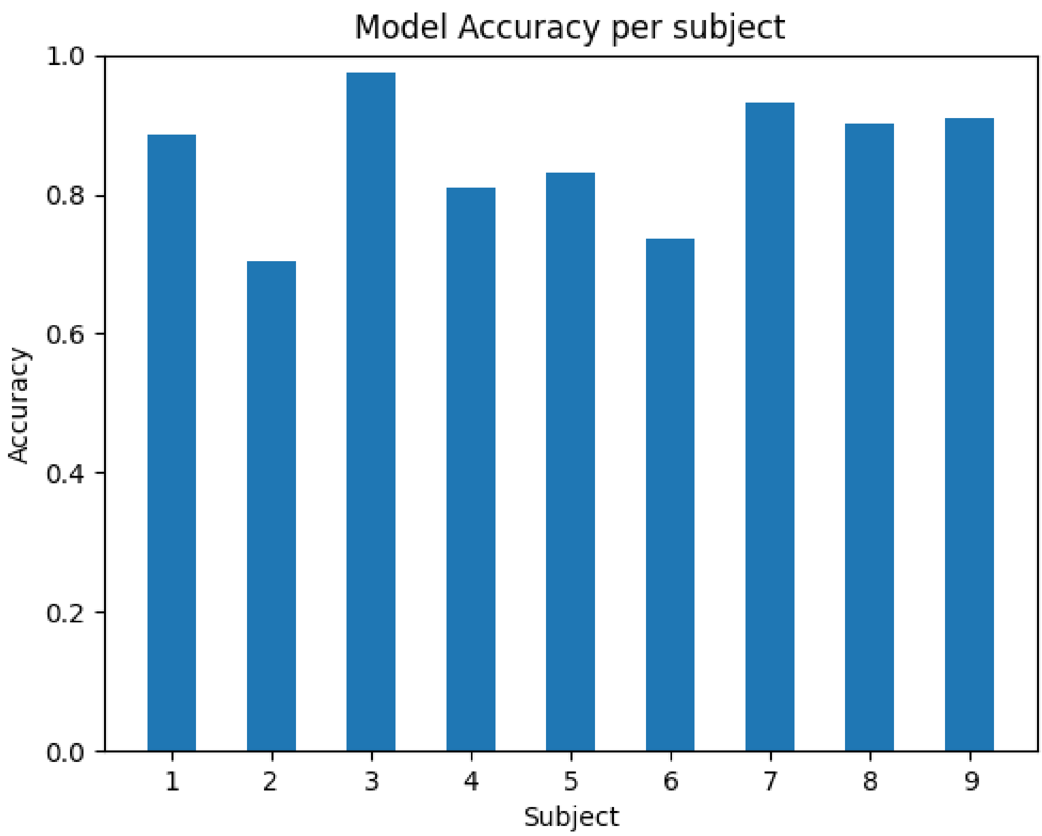


Figure 4 : Model Accuracy per subject

Accuracy is a performance metric used in machine learning to evaluate the performance of a classification model. It is defined as the ratio of correctly predicted instances to the total number of instances in the dataset. Accuracy can be calculated using the following formula:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

where TP (true positive) is the number of instances that are correctly predicted as positive, TN (true negative) is the number of instances that are correctly predicted as negative, FP (false positive) is the number of instances that are incorrectly predicted as positive, and FN (false negative) is the number of instances that are incorrectly predicted as negative.

The above figure shows the accuracy of each subject. The best accuracy among all the subjects is 85.38% and the average accuracy of all the subjects is 81.98%.

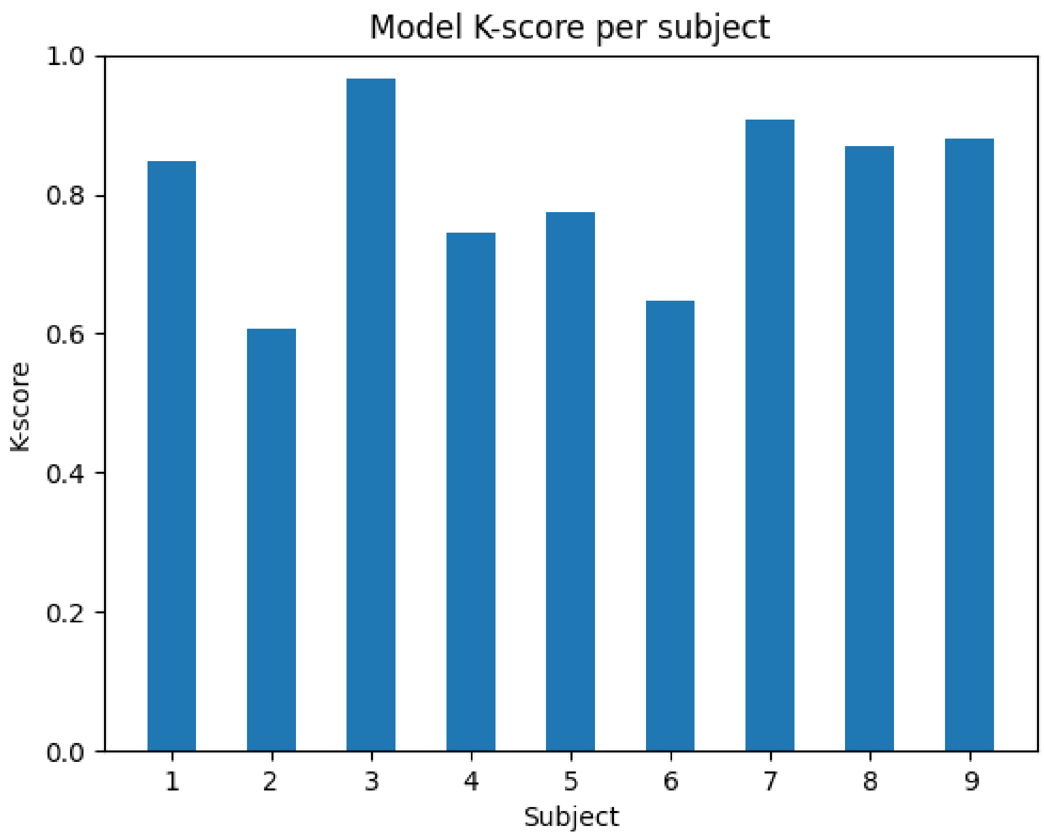


Figure 5 : Model K - score per subject

The kappa score is also a performance metric which compares the agreement between the predictions made by the model and the actual labels of the data, while also considering the probability of chance agreement. The score ranges from -1 to 1, with 1 indicating perfect agreement between the model predictions and the actual labels, and 0 indicating agreement that is no better than chance. A negative score indicates that the model is performing worse than chance. The kappa score can be calculated using the following formula:

kappa = (observed accuracy - expected accuracy) / (1 - expected accuracy)

where observed accuracy is the proportion of instances where the model prediction matches the true label, and expected accuracy is the proportion of instances that would be correctly classified by chance alone.

The above figure shows the K-score of each subject. The best K-score among all the subjects is 0.8050 and the average K-score of all the subjects is 0.7598.

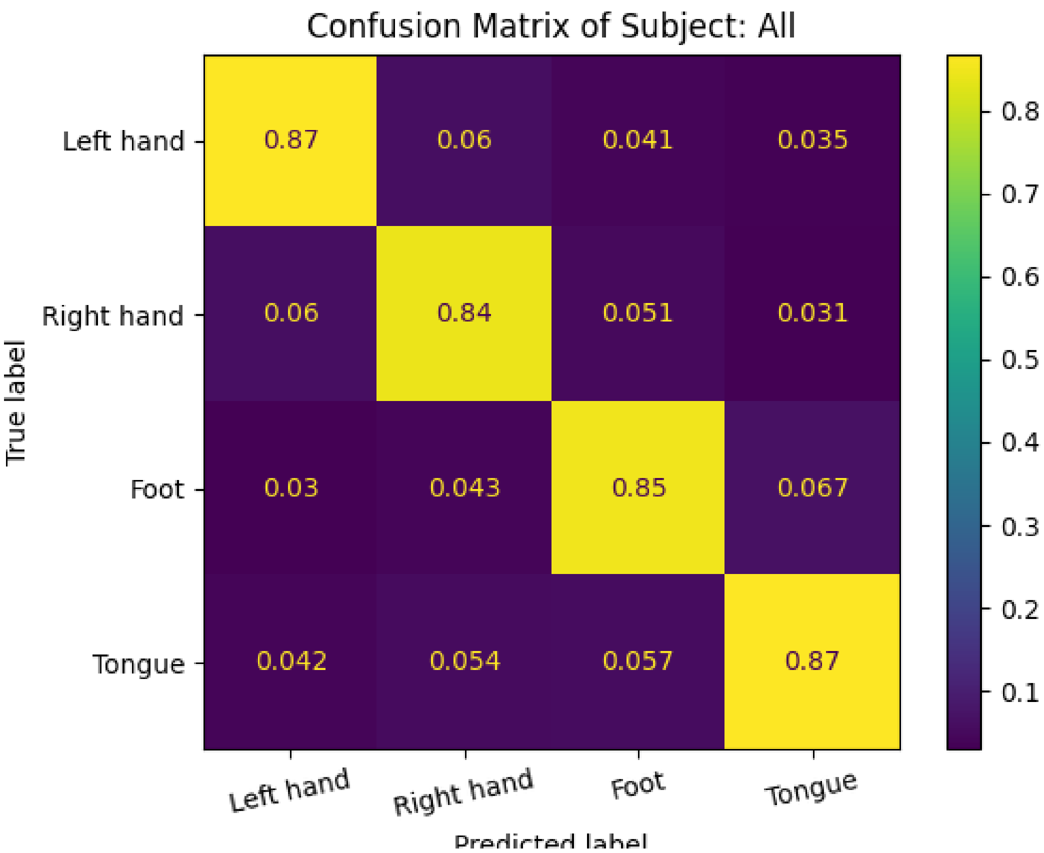


Figure 6 : Confusion Matrix of Subjects: All

A confusion matrix is a performance metric used in machine learning to evaluate the performance of a classification model. It is a table that summarizes the classification results of a model by comparing its predictions to the true labels of the data.

The above figure shows the confusion matrix of all the subjects.

6.2. PROTEUS VISUALIZATION RESULTS

After detecting the motor intensions from the subjects taken, we are implementing the proteus visualization of the willing body part of the subject.

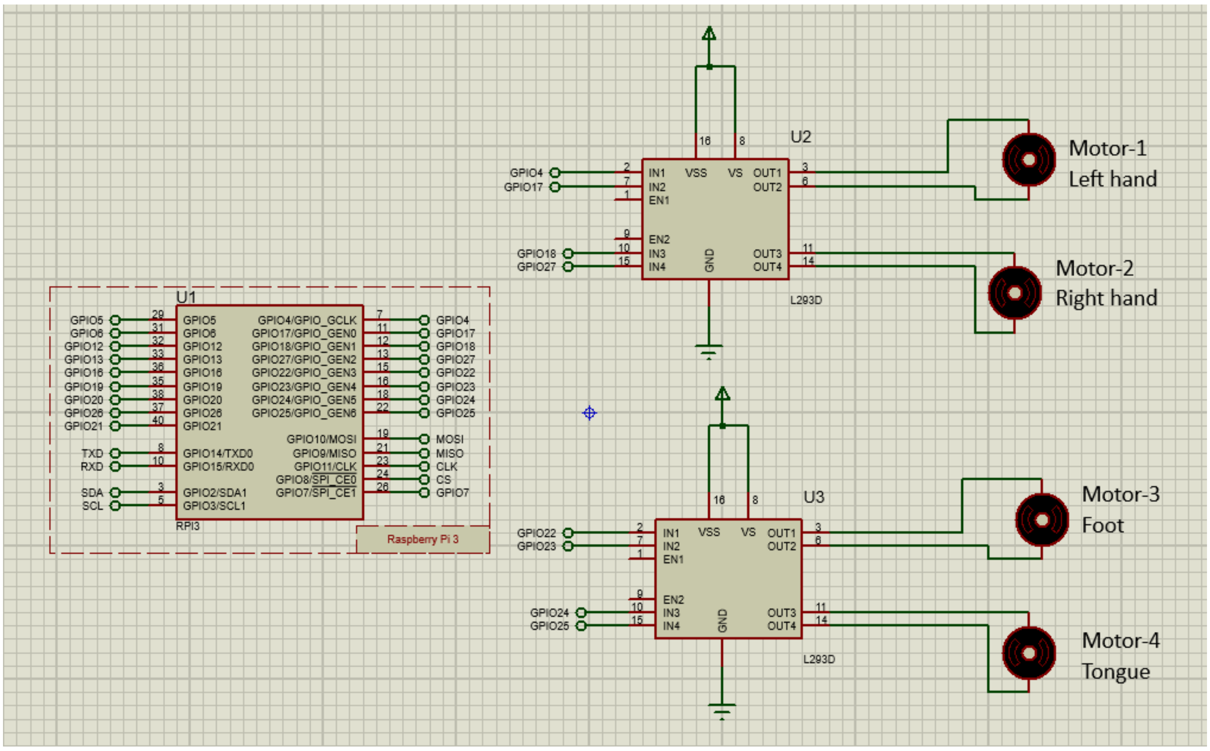


Figure 7 : Proteus Circuit

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion

We have performed the training using ATCNN and tested the model. The predicted classes from testing are visualized using Proteus. From the output of predicted classes, we are giving input to Raspberry pi and rotating the motors with the help of motor drivers.

Through this project, we are trying to visualize the body part the patient wanted to move without even actually performing the movement.

7.2 Future Scope

Advances in signal processing techniques such as machine learning and deep learning could improve the accuracy of EEG-based BCI systems.

We can make our project real-time which continuously transfers the output from machine learning processing to Raspberry pi. This type of BCI's can be used for particular body part and concentrate on specific action so that it can be used in robotic arms and wheelchairs.

There is potential for more applications, wearable EEG devices, improved user experience, and improved performance.

Involuntary blinks are not easy to detect because of the limitation and time lag produced by the single channel device. Overall, the future of EEG-based BCIs looks promising and we can expect to see significant developments in this area in the coming years.

REFERENCES

1. P. Lahane, S. P. Adavadkar, S. V. Tendulkar, B. V. Shah and S. Singhal, "Innovative Approach to Control Wheelchair for Disabled People Using BCI," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529473.

2. A. Turnip, T. Hidayat and D. E. Kusumandari, "Development of brain-controlled wheelchair supported by raspicam image processing based Raspberry pi," 2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro--Mechanical System, and Information Technology (ICACOMIT), Jakarta, Indonesia, 2017, pp. 7-11, doi: 10.1109/ICACOMIT.2017.8253377.

3. C. E. Hernández-González, J. M. Ramírez-Cortés, P. Gómez-Gil, J. Rangel-Magdaleno, H. Peregrina-Barreto and I. Cruz-Vega, "EEG motor imagery signals classification using maximum overlap wavelet transform and support vector machine," 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, 2017, pp. 1-5, doi: 10.1109/ROPEC.2017.8261667.

4. G. Reshmi and A. Amal, "Design of a BCI System for Piloting a Wheelchair Using Five Class MI Based EEG," 2013 Third International Conference on Advances in Computing and Communications, Cochin, India, 2013, pp. 25-28, doi: 10.1109/ICACC.2013.12.

5. C. -K. Huang, Z. -W. Wang, G. -W. Chen and C. -Y. Yang, "Development of a smart wheelchair with dual functions: Real-time control and automated guide," 2017 2nd International Conference on Control and Robotics Engineering (ICCRE), Bangkok, Thailand, 2017, pp. 73-76, doi: 10.1109/ICCRE.2017.7935045.

6. A. Dev, M. A. Rahman and N. Mamun, "Design of an EEG-Based Brain Controlled Wheelchair for Quadriplegic Patients," 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018, pp. 1-5, doi: 10.1109/I2CT.2018.8529751.

7. K. Tanaka, K. Matsunaga and H. O. Wang, "Electroencephalogram-based control of an electric wheelchair," in IEEE Transactions on Robotics, vol. 21, no. 4, pp. 762-766, Aug. 2005, doi: 10.1109/TRO.2004.842350.

8. G. Mezzina and D. De Venuto, "Four-Wheel Vehicle Driving by using a Spatio-Temporal Characterization of the P300 Brain Potential," 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), Turin, Italy, 2020, pp. 1-6, doi: 10.23919/AEITAUTOMOTIVE50086.2020.9307405.

9. S. He et al., "A P300-Based Threshold-Free Brain Switch and Its Application in Wheelchair Control," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 6, pp. 715-725, June 2017, doi: 10.1109/TNSRE.2016.2591012.

10. M. R. N. Kousarrizi, A. A. Ghanbari, M. Teshnehlab, M. A. Shorehdeli and A. Gharaviri, "Feature Extraction and Classification of EEG Signals Using Wavelet Transform, SVM and Artificial Neural Networks for Brain Computer Interfaces," 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, China, 2009, pp. 352-355, doi: 10.1109/IJCBS.2009.100.

APPENDIX

Python Code for training and testing the dataset

https://drive.google.com/drive/folders/1K-GWJE7VIk2tj8GYxoAeYa_rqfbJ6O7P

Proteus Code

```
import RPi.GPIO as GPIO

from time import sleep

GPIO.setmode(GPIO.BOARD)

motor1a=7

motor1b=11

motor2a=12

motor2b=13

motor3a=15

motor3b=16

motor4a=18

motor4b=22

GPIO.setup(motor1a,GPIO.OUT)

GPIO.setup(motor1b,GPIO.OUT)

GPIO.setup(motor2a,GPIO.OUT)

GPIO.setup(motor2b,GPIO.OUT)

GPIO.setup(motor3a,GPIO.OUT)

GPIO.setup(motor3b,GPIO.OUT)

GPIO.setup(motor4a,GPIO.OUT)

GPIO.setup(motor4b,GPIO.OUT)
```

```
x=[2, 1, 1, 0, 0, 0, 1, 3, 1, 3, 0, 2, 1, 3, 3, 3, 3, 3, 3, 3, 2, 1, 0, 0, 2, 3, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0,
1,2, 1, 2, 2, 3, 2, 2, 3, 3, 3, 2, 2, 2, 1, 0, 0, 1, 2, 3, 1, 2, 0, 0, 0, 3, 1, 1, 0, 0, 2, 3, 2, 3, 3, 3, 0, 3,

2, 1, 3, 3, 1, 0, 1, 2, 2, 2, 3, 2, 0, 3, 1, 2, 1, 2, 3, 1, 3, 0, 0, 0, 3, 1, 0, 2, 0, 2, 1, 3, 0, 2, 2, 0, 2,1,
3, 3, 3, 2, 0, 2, 1, 3, 1, 0, 2, 1, 0, 2, 2, 0, 2, 3, 3, 1, 0, 1, 3, 1, 2, 2, 1, 1, 1, 2, 2, 1, 1, 3, 0, 3,3, 3,
0, 0, 3, 1, 2, 3, 3, 0, 1, 2, 0, 3, 0, 3, 2, 1, 3, 2, 0, 1, 0, 2, 3, 1, 0, 0, 3, 1, 0, 2, 1, 1, 0, 0, 3,2, 0, 2,
2, 0, 1, 0, 1, 0, 0, 2, 2, 1, 2, 3, 0, 3, 0, 0, 1, 3, 2, 1, 3, 2, 3, 3, 3, 1, 1, 3, 0, 1, 0, 1, 2, 3,0, 3, 0, 2,
0, 3, 0, 2, 0, 1, 3, 2, 3, 0, 1, 3, 1, 2, 2, 0, 3, 1, 3, 1, 0, 2, 2, 1, 3, 1, 1, 0, 1, 3, 3, 1, 1,1, 1, 3, 3, 3,
3, 1, 1, 2, 1, 0, 3, 0, 3, 0, 0, 0, 0, 2, 2, 3, 1, 2, 2, 2, 3, 2, 0, 2]


n=len(x)

for i in range(n):

  if x[i]==0:

    GPIO.output(motor1a,GPIO.HIGH)

    GPIO.output(motor1b,GPIO.LOW)

    sleep(2)

    GPIO.output(motor1a,GPIO.LOW)

    GPIO.output(motor1b,GPIO.HIGH)

    sleep(2)

    GPIO.output(motor1b,GPIO.LOW)

    sleep(1)

  elif x[i]==1:

    GPIO.output(motor2a,GPIO.HIGH)

    GPIO.output(motor2b,GPIO.LOW)

    sleep(2)

    GPIO.output(motor2a,GPIO.LOW)

    GPIO.output(motor2b,GPIO.HIGH)
```

```
      sleep(2)
GPIO.output(motor2b,GPIO.LOW)
      sleep(1)
   elif x[i]==2:
      GPIO.output(motor3a,GPIO.HIGH)
      GPIO.output(motor3b,GPIO.LOW)
      sleep(2)
      GPIO.output(motor3a,GPIO.LOW)
      GPIO.output(motor3b,GPIO.HIGH)
      sleep(2)
      GPIO.output(motor3b,GPIO.LOW)
      sleep(1)
   elif x[i]==3:
      GPIO.output(motor4a,GPIO.HIGH)
      GPIO.output(motor4b,GPIO.LOW)
      sleep(2)
      GPIO.output(motor4a,GPIO.LOW)
      GPIO.output(motor4b,GPIO.HIGH)
      sleep(2)
      GPIO.output(motor4b,GPIO.LOW)
      sleep(1)
```