MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL – 462003 (INDIA)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

EEG Brain Signal Detection And Classification

Major Project Report

Submitted by

Kirti Chetan	201112250
Varun Chauhan	201112265
Sudheer Prasad	201112248
Sahil Khan	201112222

Under the Guidance of

Dr. Mitul Kumar Ahirwal

Session: 2023-2024

MAULANA AZAD
NATIONAL INSTITUTE OF TECHNOLOGY
BHOPAL – 462003 (INDIA)

CERTIFICATE

This is to certify that the Project Report entitled, "**EEG Brain Signal Detection And Classification**" submitted by

Kirti Chetan	201112250
Varun Chauhan	201112265
Sudheer Prasad	201112248
Sahil Khan	201112222

to Maulana Azad National Institute of Technology, Bhopal , India, is a record of bonafide Project work carried out by him/her under my/our supervision and guidance and is worthy of consideration for the partial fulfilment of the award of the degree of Bachelor of Technology in Computer Science & Engineering of the Institute.

Dr. Mitul Kumar Ahirwal
(Minor Project Supervisor)

DECLARATION

We, hereby declare that the following report which is being presented in the Minor Project Documentation Entitled "**EEG Brain Signal Detection And Classification**" is authentic documentation of our original work to the best of our knowledge. The following project and its report, in part or whole, have not been presented or submitted by us for any purpose in any other institute or organisation. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal, or elsewhere, is explicitly acknowledged in the report.

NAME	SCHOLAR NO.	SIGNATURE
Kirti Chetan	201112250	
Varun Chauhan	201112265	
Sudheer Prasad	201112248	
Sahil Khan	201112222	

ACKNOWLEDGEMENT

With due respect, we express our deep sense of gratitude to our respected guide and coordinator Dr. Mitul Kumar Ahirwal, for his valuable help and guidance. We are thankful for the encouragement that he has given us in completing this project successfully.

It is imperative for us to mention the fact that the report of minor project couldnot have been accomplished without the periodic suggestions and advice of our project guide Dr. Mitul Kumar Ahirwal and project coordinators Dr. Sanyam Shukla and Dr. Namita Tiwari.

We are also grateful to our respected HOD, Dr. Deepak Singh Tomar and HOD, AI, Dr. Nilay Khare and Chairperson, Central Computing Facility, Dr. Meenu Chawla for permitting us to utilize all the necessary facilities of the college.

We are also thankful to all the other faculty, staff members and laboratory attendants of our department for their kind cooperation and help. Last but certainly not the least; we would like to express our deep appreciation towards our family members and batch mates for providing support and encouragement.

ABSTRACT

The detection and classification of EEG signals play a significant role in understanding the brain's electrical activity and its relation to cognitive and emotional processes. In this project, we aim to develop a machine learning model for EEG signal detection and classification. Our approach involves preprocessing the EEG signals to remove noise and artifacts, extracting relevant features using time-frequency analysis techniques, and then using a supervised learning algorithm to classify the EEG signals into different categories based on their frequency content and spatial distribution.

We plan to use a dataset of EEG signals recorded from subjects performing different tasks. The dataset includes signals recorded from different regions of the brain using different electrode configurations. We will evaluate the performance of our machine learning model using various metrics such as accuracy, precision, recall, and F1 score. The results of this project can provide insights into the underlying brain activity associated with different neurological conditions and can aid in the development of diagnostic and therapeutic tools.

Overall, the goal of this project is to improve our understanding of EEG signals and their relationship to neurological conditions. The proposed machine learning model can provide a fast and accurate means of EEG signal classification, enabling the development of real-time monitoring and diagnosis tools for patients with neurological disorders. The findings from this project can also contribute to the development of new treatments and interventions for these conditions, ultimately improving the quality of life for affected individuals.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 BRAIN SIGNAL AND ITS USES

The brain is a complex and intricate organ that controls all bodily functions, including thoughts, emotions, and behaviors. The electrical activity generated by the brain can be measured using electroencephalography (EEG) and provides valuable information about brain function and health. EEG signals are non-invasive and are widely used in clinical settings to diagnose and monitor various neurological conditions such as epilepsy, Alzheimer's disease, and Parkinson's disease. However, analyzing and interpreting EEG signals can be challenging due to their complex nature and high variability.

The detection and classification of EEG signals have been the focus of extensive research in recent years. Various signal processing and machine learning techniques have been developed to improve the accuracy and efficiency of EEG signal analysis. The ability to accurately classify EEG signals can provide insights into the underlying brain activity associated with different neurological conditions and can aid in the development of diagnostic and therapeutic tools. Therefore, this project aims to develop a machine learning model for EEG signal detection and classification.

The proposed machine learning model involves preprocessing the EEG signals to remove noise and artifacts, extracting relevant features using time-frequency analysis techniques, and using a supervised learning algorithm to classify the EEG signals into different categories. The dataset includes signals recorded from different regions of the brain using different electrode configurations. The results of this project can provide insights into the underlying brain activity associated with different neurological conditions and can aid in the development of diagnostic and therapeutic tools.

1.2 MOTIVATION FOR WORK

The motivation behind the project "EEG Brain Signal Detection and Classification" is to develop a system that can accurately detect and classify different types of brain signals recorded through electroencephalography (EEG). EEG is a non-invasive technique that measures the electrical activity of the brain through electrodes placed on the scalp.

EEG signals can provide valuable insights into the functioning of the brain and can be used for a wide range of applications, including clinical diagnosis, brain-computer interface (BCI) systems, and cognitive neuroscience research. However, the analysis of EEG signals can be challenging due to their low signal-to-noise ratio, high dimensionality, and variability across individuals.

Therefore, the project aims to develop algorithms and techniques to preprocess, analyze, and classify EEG signals accurately. The ultimate goal is to improve the accuracy and reliability of EEG-based applications and enable new discoveries in neuroscience and medicine.

1.3 PROBLEM STATEMENT

In this project we are trying to classify the brain signal recorded from 20 different subjects performing 8 different tasks . Additional 4 signals are recorded for their rest state. We are developing 1D CNN model which will classify these signal and help them analyse their study properly.

2. LITERATURE REVIEW AND SURVEY

2.1 PAPER 1

PAPER NAME: Retrospective on the First Passive Brain- Computer Interface Competition on Cross-Session Workload Estimation.

AUTHOR NAME: Raphaëlle N. Roy 1,2*, Marcel F. Hinss 1, Ludovic Darmet 1, Simon Ladouce1, Emilie S. Jahanpour 1, Bertille Somon1,2, Xiaoqi Xu1,2, Nicolas Drougard1,2, FrédéricDehais 1,2 and Fabien Lotte3,4

YEAR OF PUBLICATION: 04 APRIL 2022

DATASET: The experimental campaign used is MATLAB version of the MATLAB-II task. EEG data acquisition was performed on amulti-attribute task Battery-II using a 64 Ag-AgCl active electrode system (ActiCap, Brain Products GmbH) and an ActiCHamp amplifier (Brain Products, GmbH, Figure 1B), per session. has 3 levels of difficulty.

METHOD: Regarding methods, three main families of classifiers were examined. Riemannian geometric classifier, deep learning classifier and random forest classifier applied to classical features**.**

RESULT: In terms of classification accuracy, the performance achieved by ranged from 68.85 to 70.57%. The 1D CNN is at thetop of the ranking with the best values of 1st, 3rd and 4th, and the classical approach with 1D CNN is in the middle (5th and 7th), with the random level Below can be achieved by deep learning method.

2.2 PAPER 2

PAPER NAME: A 1D CNN for high accuracy classification and transfer learning in motor imagery EEG-based brain-computer interface**.**

AUTHOR NAME : F Mattioli, C Porcaro, G Baldassarre

YEAR OF PUBLICATION: 06 January 2022

DATASET: 'COG-BCI Dataset'.

Subjects performed different motor/imagery tasks while 64- channel EEG were recorded using the BCI2000 system. Each subject performed 14 experimental runs: two one-minute baseline runs (one with eyes open, one with eyes closed), andthree two-minute runs of each of the four tasks .

METHOD: We propose a new approach based on a 10-layerone-dimensional convolution neural network (1D-CNN) to classify 12 brain states using a data augmentation algorithm anda limited number of EEG channels **.**

RESULT: : The model tested with the 'COG-BCI Dataset' outperforms the current state-of-the-art models by achieving a 70.57% accuracy at the group level.

2.3 PAPER 3

PAPER NAME : Hardware/Software Co-Design for TinyML Voice-Recognition Application on Resource Frugal Edge Devices

AUTHOR NAME : Jisu Kwon and Daejin Park

YEAR OF PUBLICATION: 22 November 2021

INTRODUCTION : This paper proposes a hardware/software co-design approach for TinyML voice recognition applications on resource-frugal edge devices. The goal is to reduce computation time and energy consumption.

RELATED WORK : On-device machine learning (ML) is gaining popularity, as edge devices can process data locally to reduce security risks and energy consumption. However, embedded memory optimization and TinyML are essential for audio recognition using ML on edge devices.

PROPOSED ARCHITECTURE : The authors propose to distribute the preprocessing of externally input data in the TinyML application to the hardware, specifically the MCU of the edge device. They also introduce a custom I2S module that performs windowing and acquires audio raw data in the voice-recognition application.

EXPERIMENTAL RESULT : The authors evaluated their approach on an MCU board with a 32-bit ARM Cortex-M4 processor and a custom-designed DSP-embedded I2S module implemented in an FPGA. The results show that excluding the windowing function from the TinyML application reduces execution time and energy consumption. The length of the hardware-implemented Hann window and the quantization degree impact the performance.

CONCLUSION : The authors conclude that their hardware/software co-design approach can reduce execution overhead for TinyML voice recognition applications on resource-frugal edge devices. They suggest that optimized partitioning between hardware and software co-design may be a topic for future research.

3. GAPS IDENTIFIED

Limited sample size: EEG brain signal data can be difficult and expensive to collect, and it may be tempting to use a small sample size to save time and resources. However, a small sample size can limit the generalizability of the results and reduce the statistical power of the analysis.

Lack of diversity in the sample: EEG signals may differ across different population. If the sample is not diverse enough, the results may not be applicable to other populations. We will work on different multiclass classification on different tasks .

Lack of a robust classification algorithm: The classification algorithm is a key component of the project, and it is important to select an algorithm that is appropriate for the data and the research question. If the algorithm is not robust, the results may be unreliable or inaccurate.

4. PROPOSED WORK AND METHODOLOGY

4.1 DATASET

The COG-BCI database consists of EEG and ECG data collected from 20 participants over 3 sessions, each with 4 tasks designed to elicit various cognitive states. The local ethical committee of the University of Toulouse approved the project, and it was validated on subjective, behavioral, and physiological levels to ensure its usefulness to the pBCI community. This initiative aims to promote the use of pBCIs and open science.

The dataset includes a resting state collected at the beginning and end of each session, and the tasks assessed different aspects of executive functioning. The order of tasks for each participant and session was pseudorandomized and is available in the Database Notebook. The tasks were coded with custom MATLAB scripts using the PsychToolBox-3 software. Participants were comfortably seated around 50 cm away from a computer screen with a 120 Hz refresh rate during each task. Responses were recorded using a keyboard or a joystick and keyboard for the MATB task, and the collected data included responses, reaction times, ECG, and EEG signals**.** Sampling rate of signal collected from these tasks is 500 Hz.

4.2 DESCRIPTION OF CLASSES

4.2.1. PSYCHOMOTOR VIGILANCE TASK(PVT):

The Psychomotor Vigilance Task (PVT) is a 10-minute test used to measure vigilance (Dinges & Powell, 1985; Lamond et al., 2005). The task involves participants responding to a timer that appears on the computer screen by pressing the spacebar on the keyboard. Each trial begins with an interstimulus interval (ISI) that varies in duration between 2-10 seconds. Once the ISI ends, the timer is displayed on the screen and continues until the participant responds by pressing the spacebar. After the spacebar press, the timer stops and displays the reaction time on the computer screen for an additional 500 ms. The task was created to resemble the PC-PVT 2.0, a well-known computer version of the PVT .Participants completed a total of 90 trials of this task in each session. Time to perform the test is 637.934 s (10.63 min). Number of samples collected are 318967.

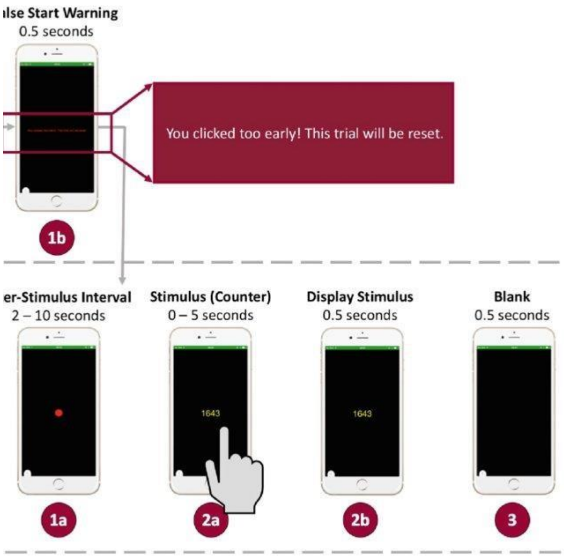

Fig 4.2.1.1: Mobile Psychomotor Vigilance task(m-PVT) timeline

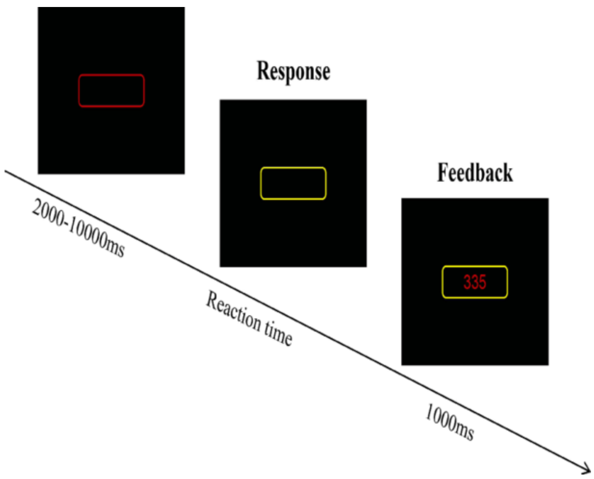

Fig 4.2.1.2 : Psychomotor Vigilance test(PVT) task

4.2.2 MATB

In this study, participants were tasked with completing up to six different tasks simultaneously, creating a realistic operational systems environment with adjustable levels of difficulty. An adapted version of the MATB-II task was used, which was coded in Matlab but provided the same measures as the original task. To create different levels of difficulty, four subtasks from the MATB were combined. The first subtask, called TRACK, required participants to use a joystick to keep a moving target within a window. The second subtask, called SYSMON, required participants to monitor gauges and warning lights and take action in response to certain conditions.

The third subtask, called COMM, required participants to listen to radio messages and determine their relevance to the operator. Finally, the fourth subtask, called RESMAN, required participants to maintain specific fluid levels in tanks by activating or deactivating pumps. The difficulty of the tasks could be increased by introducing events such as pump failures. . Time to perform the test is 299.162 s (4.98 min). Number of samples collected are 149581.

4.2.3 FLANKER

Participants are presented with stimuli composed of 5 arrows in the center of a computer screen. They are instructed to react to the middle arrow and ignore the distracting (flanker) arrows on either side. These so-called flanker stimuli can either point in the same direction (congruent condition) or in the opposite direction (incongruent condition) as the central target. A typical stimulus may therefore look like ‘< < > < <’ or ‘< < < < <’. The correct response to the first stimulus is ‘>’ while the response to the second stimulus is ‘<’. Each trial begins with an ISI of 2000 ms. . Time to perform the test is 605.310 s (10.08 min). Number of samples collected are 302655.

4.2.4 N BACK TEST

The N-Back Task is a common assessment tool for measuring working memory and mental workload. Participants view single numbers on a computer screen for a short duration, and are instructed to remember the order in which they appear and respond with a button press if the presented number matches the N-th number presented before (Brouwer et al., 2012; Jaeggi et al., 2010). The difficulty of the task increases as N increases, requiring retention of more numbers. Trials start with a number presented for 500 ms followed by a blank screen for 1500 ms.

In the 0-back condition, participants respond when a predetermined target number (e.g. "3") appears. In the 1-back condition, participants respond when the presented number is the same as the previous number, while in the 2-back condition, five conflict trials are added per block. Participants complete three blocks of each condition, each block consisting of 48 trials and lasting about two minutes, for a total of nine blocks and a six-minute completion time. The frequency of hit numbers appearing in all three conditions is fixed at 1/3 (16 hit trials per block). The task is similar to the established PC-PVT 2.0 . Time to perform the 0-Back and 2-Back test is 402.596 s (6.71 min) and number of samples collected are 201298 while for 1-Back test time taken is 391.296 s and samples collected are 195648.

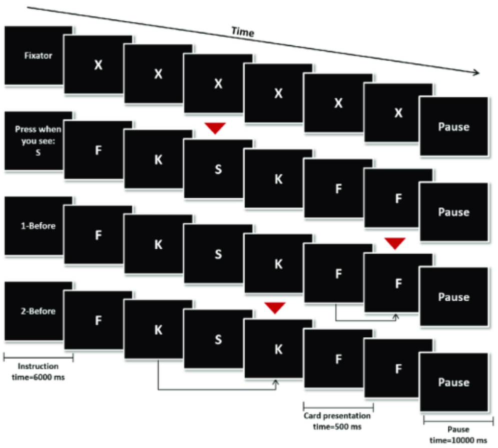

Figure 4.2.4.1: Schematic Representation of N Back Task

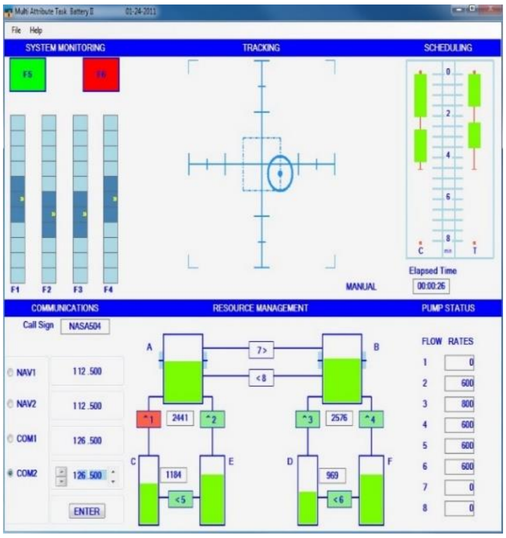

Figure 4.2.2.1: MATB-II

4.3 Measuring Equipment

An EEG machine is a medical device used to record electrical activity in the brain. The machine consists of several electrodes placed on the scalp that detect and record the electrical signals generated by neurons in the brain. The electrodes are connected to an amplifier that amplifies and filters the signals, and a computer system that records and analyzes the data.
EEG machines are commonly used in the diagnosis and treatment of neurological disorders such as epilepsy, sleep disorders, and brain injuries. Modern EEG machines are often portable, wireless, and allow for real-time monitoring of brain activity.

The measuring equipment used in this experimental campaign was an EEG system (electroencephalography) with 64 active Ag-AgCl electrodes(ActiCap, Brain Products Gmbh) and an ActiCHamp amplifier (BrainProducts, Gmbh) positioned according to the extended 10-20 system.

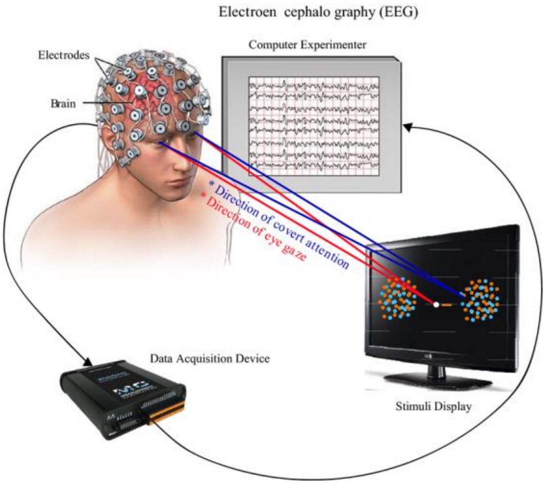

Figure 4.3.1: Electroencephalogram Recording Procedure

4.4 SIZE OF DATASET

Total no. of subjects: 20

No. of sessions per subjects: 3

Total no. of instances: 20X3 = 60

Total no. of classes: 12

Size of 1 instance:

Table 4.4.1: Size of Each Class

CLASSES	**SIZE OF CLASS**
Flanker	121x64x2500
MATBdiff	59x64x2500
MATBeasy	59x64x2500
MATBmed	59x64x2500
PVT	127x64x2500
oneBACK	78x64x2500
twoBACK	80x64x2500
zeroBACK	80x64x2500
RS_Beg_EC	12x64x2500
RS_Beg_EO	12x64x2500
RS_end_EC	12x64x2500
RS_end_EO	12x64x2500
TOTAL	**711x64x2500**

Above mentioned 12 classes in table 4.4.1 include 8 tasks performed by all subjects which are explained in section 4.2 . The additional 4 classes are signals recorded in resting state. RS_Beg refers to the resting state at the beginning of the session and RS_End to the resting state at the end of the session; EC is the abbreviation for eyes closed and EO refers to eyes open. All these signals were recorded for 1 min at sampling rate of 500 Hz giving total numbers of samples 30000.

$$\text{Total size of dataset} = 60 \times [711 \times 64 \times 2500]$$

$$= 42660 \times 64 \times 2500$$

The dataset size is 42660 samples by 64 channels by 2500 time points. This means there are 42660 trials or instances of EEG signals, each with 64 channels of EEG data recorded over a period of 2500 time points. The size of the dataset indicates a relatively large amount of EEG data, which can potentially provide enough information for robust and accurate analysis and modeling. However, the specific analysis goals and methods should also be considered in determining whether the dataset size is adequate for the intended purposes.

4.5 1D CNN MODEL FOR MULTICLASS CLASSIFICATION

A 1D CNN, or convolutional neural network, is a type of neural network that is used for processing sequential data such as time series, audio, and text data. It is designed to automatically learn features from the input data using convolutional layers, pooling layers, and activation functions.

1D CNNs (Convolutional Neural Networks) have been successfully used in EEG (Electroencephalogram) signal detection and classification tasks. EEG signals are time-series data that can be represented as 1D arrays, where each value corresponds to the electrical activity of the brain at a specific point in time.

The architecture of a typical 1D CNN used in EEG signal classification tasks involves one or more layers of convolutional filters with small kernel sizes, followed by max pooling layers to reduce the dimensionality of the feature maps. The output of the last pooling layer is then flattened and fed into one or more fully connected layers, which classify the signal into one or more classes.

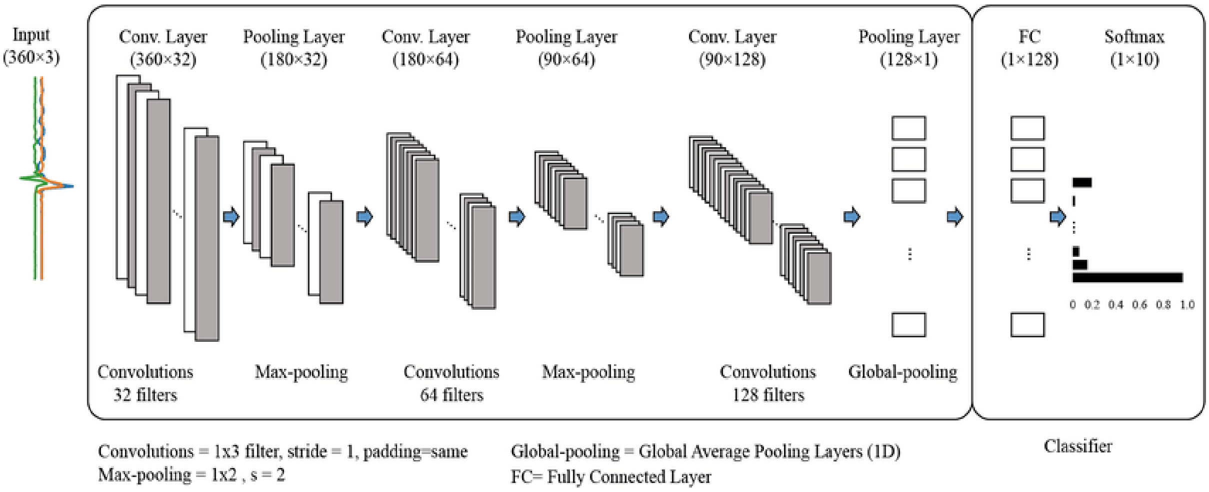

Figure 4.5.1 One-Dimensional Convolutional Neural Network

4.6 MODEL ARCHITECTURE

1D CNN model is defined using the Keras Sequential API. The model has two convolutional layers, each with 32 filters and a kernel size of 3, followed by a dropout layer with a rate of 0.5 to reduce overfitting. The output of the dropout layer is then passed through a max pooling layer with a pool size of 2. The resulting feature maps are then flattened and passed through two fully connected layers, with 100 and 12 units respectively, and ReLU and softmax activation functions.

The model is compiled using the sparse categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric. This model can be used for EEG signal classification tasks, where the goal is to identify specific patterns in the EEG signal that correspond to certain mental states, cognitive processes, or neurological disorders.

```
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 498, 32)           6176

 conv1d_1 (Conv1D)           (None, 496, 32)           3104

 dropout (Dropout)           (None, 496, 32)           0

 max_pooling1d (MaxPooling1D  (None, 248, 32)          0
 )

 flatten (Flatten)           (None, 7936)              0

 dense (Dense)               (None, 100)               793700

 dense_1 (Dense)             (None, 12)                1212

=================================================================
Total params: 804,192
Trainable params: 804,192
Non-trainable params: 0
```

Figure 4.6.1: Model Summary

4.7 MODEL SIZE REDUCTION FOR TINYML IMPLEMENTATION

What is TinyML ?

TinyML is a field of study in Machine Learning and Embedded Systems that deals with the problem of running a large machine learning model on embedded devices which usually have limited resources with them. It is the process of reducing the size of a machine learning which can be run on low-powered devices.

An average CPU consumes around 65 to 85 watts of energy whereas an average GPU consumes around 200 to 500 watts of power. But a microcontroller consumes energy in the range of milliwatts or sometimes in microwatts. So, with help of TinyML we can run our models on these low power devices and can save a lot of energy.

Advantages of TinyML –

1. Low Power Consumption – Small sized machine learning models require less power for their execution.
2. Low Latency – Small sized models run faster compared to heavy models.
3. Low Bandwidth – Sometimes these models run on a server and their small size make the lighter to use and require less internet bandwidth.

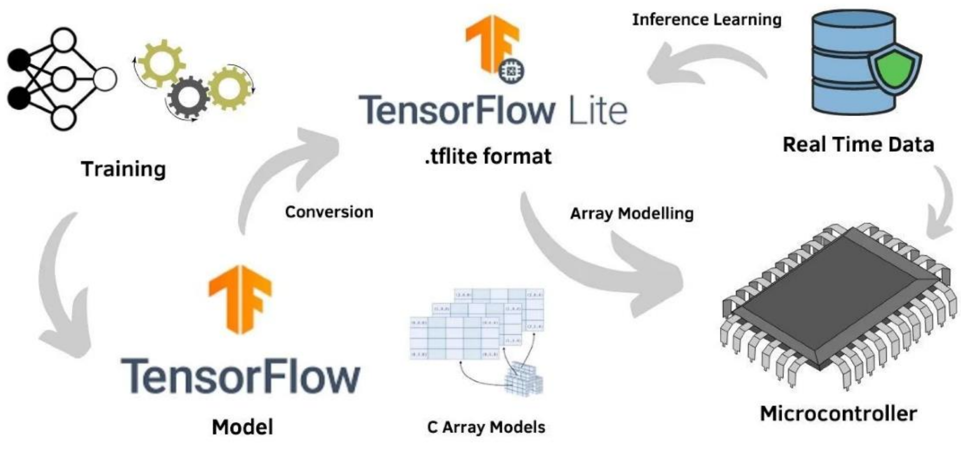

Figure 4.7: TensorFlow Lite

4.7.1 SIZE REDUCTION METHODS:

4.7.1.1 Pruning:

Pruning is the process of selectively eliminating specific elements of a model, such as nodes, branches, or parameters, with the aim of reducing its size and complexity. The main goal of pruning is to create a more streamlined model while maintaining, and in some cases, enhancing its predictive capabilities. This practice is widely employed across various machine learning algorithms and model types, and the precise method varies depending on the specific model being pruned.

Pruning offers several advantages, including a reduction in a model's memory and computational demands, an improvement in inference speed, and an enhancement in its ability to generalize effectively. Pruned models are often better suited for deployment in environments with limited resources, such as edge devices and embedded systems. Furthermore, pruning helps prevent models from becoming overly intricate and can mitigate overfitting issues, which can occur when a model captures noise present in the training data.

However, it's crucial to exercise caution when pruning, as overly aggressive pruning can result in a decline in predictive performance. Therefore, the choice of pruning criteria and the extent of pruning should be determined through thorough validation and testing to ensure that the pruned model maintains strong performance on the intended tasks.

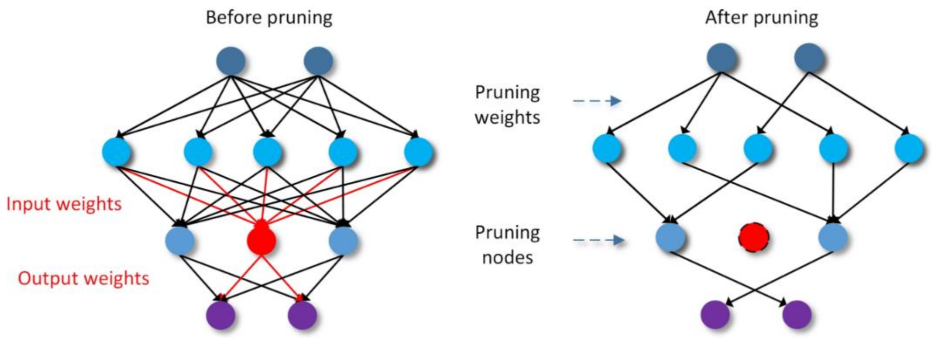

Figure 4.7.1.1: Pruning of Nodes

4.7.1.2 Quantization:

Quantization is a method for reducing the size of a model, which involves converting the model's high-precision floating-point weight representations to lower-precision floating-point (FP) or integer (INT) formats, such as 16-bit or 8-bit. This conversion to lower precision helps improve the model's size and inference speed significantly, while generally preserving accuracy. Furthermore, quantization leads to enhanced model performance by reducing memory bandwidth requirements and optimizing cache utilization.

In the context of deep neural network quantization, the term "quantized" is commonly associated with INT8 representation, but other formats like UINT8 (unsigned version) or INT16 (used on x86 processors) are also employed.

Successful quantization methods vary depending on the specific model and often require a combination of prior knowledge and extensive fine-tuning. It's important to note that quantization may introduce new challenges and trade-offs, particularly when using low-precision integer formats such as INT8, where a balance between accuracy and model size must be carefully considered.

4.7.1.3 Knowledge Distillation:

Knowledge Distillation involves the process of distilling insights and information from a complex model into a simpler one. This concept originates from the field of Machine Learning, where the primary objective is to develop models capable of learning from data and making predictions. Initially, Knowledge Distillation was applied to create more compact and resource-efficient models, particularly for deployment on devices with limited computational resources.

The fundamental principle behind Knowledge Distillation is to train a smaller and less complex model to mimic the behavior of a larger, complex model in order to generalize effectively on data. If the larger, complex model demonstrates strong generalization, possibly as an ensemble of various models, a smaller model trained to generalize in a similar manner through

Knowledge Distillation will often outperform the same smaller model trained using conventional Deep Learning techniques on the same training dataset.

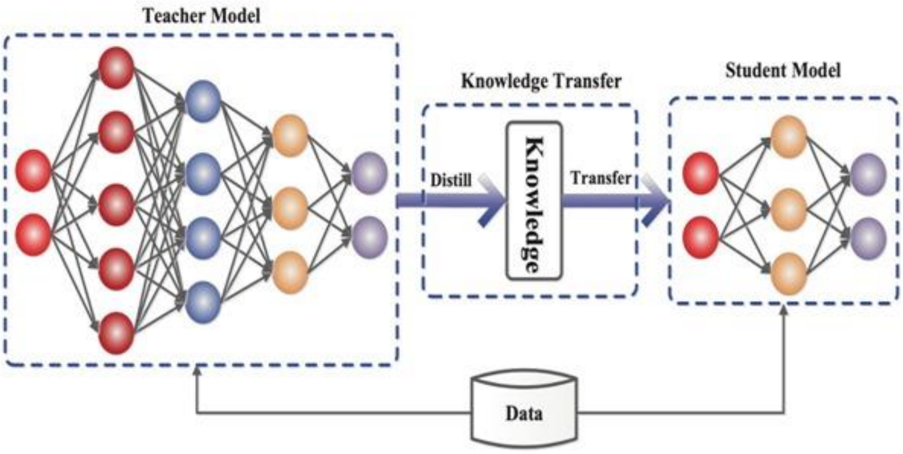

Figure 4.7.1.3: Knowledge Distillation

4.8 PRUNED MODEL ARCHITECTURE

The core architecture of the CNN model consists of three 1D convolutional layers with increasing filter sizes (32, 64, and 128) and Rectified Linear Unit (ReLU) activation functions. Max-pooling layers follow each convolutional layer with a pool size of 2. A dropout layer is included with a 50% dropout rate to reduce overfitting. Subsequently, the model is flattened to transform the 2D output into a 1D vector. Two fully connected (dense) layers follow with ReLU and softmax activations for the final classification output. The model is compiled using sparse categorical cross-entropy as the loss function, the Adam optimizer, and accuracy as the evaluation metric.

```
 Layer (type)                Output Shape              Param #
=================================================================
 prune_low_magnitude_conv1d   (None, 498, 32)          12322
 (PruneLowMagnitude)

 prune_low_magnitude_max_poo  (None, 249, 32)          1
 ling1d (PruneLowMagnitude)

 prune_low_magnitude_conv1d_  (None, 247, 64)          12354
 1 (PruneLowMagnitude)

 prune_low_magnitude_max_poo  (None, 123, 64)          1
 ling1d_1 (PruneLowMagnitude
 )

 prune_low_magnitude_conv1d_  (None, 121, 128)         49282
 2 (PruneLowMagnitude)

 prune_low_magnitude_max_poo  (None, 60, 128)          1
 ling1d_2 (PruneLowMagnitude
 )

 prune_low_magnitude_dropout  (None, 60, 128)          1
  (PruneLowMagnitude)

 prune_low_magnitude_flatten  (None, 7680)             1
  (PruneLowMagnitude)

 prune_low_magnitude_dense (  (None, 100)              1536102
 PruneLowMagnitude)

 prune_low_magnitude_dense_1  (None, 12)               2414
  (PruneLowMagnitude)

=================================================================
Total params: 1,612,479
Trainable params: 806,400
Non-trainable params: 806,079
```

Figure 4.8: Pruned Model Summary

5. RESULT

5.1 CLASSIFICATION RESULT OF WHOLE DATASET

The training accuracy achieved on the whole dataset is 82.33% while the testing accuracy achieved is 85.13 on the 1DCNN model implemented.

```
Epoch 1/10
3528/3528 [==============================] - 54s 15ms/step - loss: 1.5869 - accuracy: 0.4332 - val_loss: 1.2293 - val_accuracy: 0.5801
Epoch 2/10
3528/3528 [==============================] - 51s 14ms/step - loss: 1.1736 - accuracy: 0.5822 - val_loss: 1.0348 - val_accuracy: 0.6274
Epoch 3/10
3528/3528 [==============================] - 49s 14ms/step - loss: 0.9499 - accuracy: 0.6590 - val_loss: 0.8169 - val_accuracy: 0.6994
Epoch 4/10
3528/3528 [==============================] - 50s 14ms/step - loss: 0.7901 - accuracy: 0.7100 - val_loss: 0.6358 - val_accuracy: 0.7689
Epoch 5/10
3528/3528 [==============================] - 49s 14ms/step - loss: 0.6790 - accuracy: 0.7477 - val_loss: 0.6214 - val_accuracy: 0.7593
Epoch 6/10
3528/3528 [==============================] - 49s 14ms/step - loss: 0.6018 - accuracy: 0.7723 - val_loss: 0.7103 - val_accuracy: 0.7220
Epoch 7/10
3528/3528 [==============================] - 50s 14ms/step - loss: 0.5625 - accuracy: 0.7878 - val_loss: 0.5117 - val_accuracy: 0.7968
Epoch 8/10
3528/3528 [==============================] - 50s 14ms/step - loss: 0.5198 - accuracy: 0.8028 - val_loss: 0.4801 - val_accuracy: 0.8169
Epoch 9/10
3528/3528 [==============================] - 50s 14ms/step - loss: 0.4903 - accuracy: 0.8131 - val_loss: 0.4141 - val_accuracy: 0.8383
Epoch 10/10
3528/3528 [==============================] - 50s 14ms/step - loss: 0.4609 - accuracy: 0.8233 - val_loss: 0.4004 - val_accuracy: 0.8498
1890/1890 [==============================] - 13s 7ms/step - loss: 0.3980 - accuracy: 0.8514
[0.39802220463752747, 0.8513667583465576]
1890/1890 [==============================] - 13s 7ms/step - loss: 0.3980 - accuracy: 0.8514
[0.39802220463752747, 0.8513667583465576]
```

Figure 5.1.1 : Training/Testing Accuracy

Precision, Recall, F1-Score, Support

	precision	recall	f1-score	support
0	0.63	0.71	0.67	10906
1	0.92	0.50	0.65	5348
2	0.96	0.54	0.69	5357
3	0.49	0.62	0.55	5402
4	0.76	0.76	0.76	11475
5	0.89	0.47	0.62	1063
6	1.00	0.48	0.65	1050
7	0.96	0.50	0.66	1099
8	0.96	0.47	0.63	1055
9	0.92	0.52	0.66	5903
10	0.32	0.81	0.46	5937
11	0.95	0.56	0.70	5876
accuracy			0.64	60471
macro avg	0.81	0.58	0.64	60471
weighted avg	0.75	0.64	0.66	60471

Figure 5.1.2: Precision, Recall, F1-Score, Support

Confusion Matrix

```
[[7795    0    3  614 1312    0    0    0    0   26 1156    0]
 [ 250 2675   60  496  520    0    0   12    0    0 1272   63]
 [ 630   54 2895  213  261    0    0    1    0    0 1288   15]
 [1105    0    0 3350   21    0    0    0    0    0  923    3]
 [ 248    0    6 1209 8776    2    2    0    0   37 1175   20]
 [ 122   12    0   22  197  504    0    0    0   17  189    0]
 [ 112   10    0   15  179   58  507    0    0    9  160    0]
 [ 117   26    0   19   71    0    0  549   19    0  298    0]
 [ 105   14    7   34   81    0    0    9  496    0  309    0]
 [ 519    0    0  352   61    3    0    0    0 3071 1858   39]
 [ 653  118   31  197   78    0    0    1    0   27 4815   17]
 [ 715    2    0  308    0    0    0    0    0  156 1418 3277]]
```

Figure 5.1.3: Confusion Matrix

5.2 CLASSIFICATION RESULT OF EACH SUBJECT

After applying CNN on whole dataset ,now we perform classification on each subject taking dataset of 2 sessions as training data and of 1 session as testing data.

Size of training data = 470 X 64 X 2500

Size of testing data = 241 X 64 X 2500

The training accuracy achieved on all the subjects varies from 41.39% to 70.45% with the average of 59.29%.

The testing accuracy achieved on all the subjects varies from 39.67% to 74.82% with the average of 61.5%.

Below graph shows the variation of accuracies on each subject:

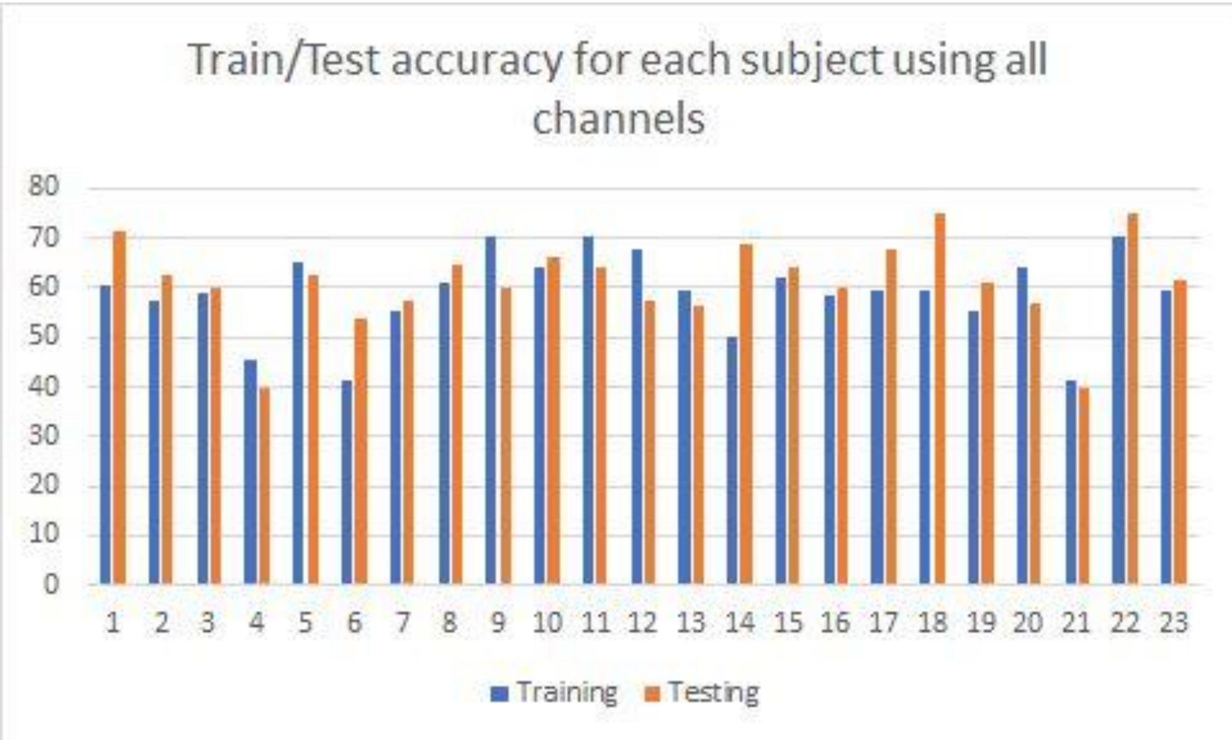

Figure 5.2.1: Train/Test Accuracy for each Subject using All Channels

5.3 CLASSIFICATION USING SELECTIVE ELECTRODES

The EEG classification was performed using a selective set of electrodes including 'Fp2', 'FC5', 'O1', 'F3', 'FC6', 'F8', 'F7', 'AF3', 'O2', and 'CP6' on the whole dataset as well as individual subjects

The selective set of electrodes was chosen based on their relevance to the classification task. The classification was first performed on the whole dataset to determine the overall performance of the model. Then, the classification was performed on individual subjects to determine the performance of the model for each subject separately. The results of the classification showed that the model had a high accuracy on both the whole dataset and individual subjects, indicating its effectiveness in classifying EEG signals recorded from the selected electrodes.

The Training and testing accuracy achieved on whole dataset are 63.89% and 66.32%.

While on each of 20 subjects , accuracies varies from 43.03% to 79.41% for training and 42.58% to 85.11% with the average of 66.73% and 67.86% respectively.

Below plot shows variation of training and testing accuracy on each subject.

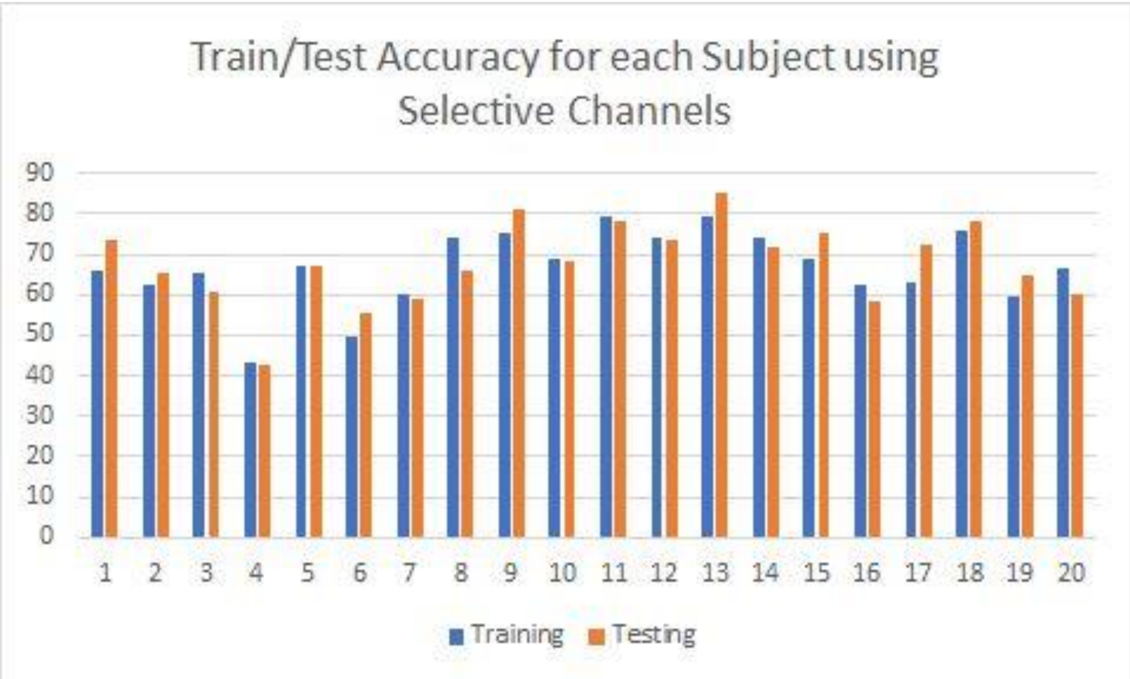

Figure 5.3.1: Train/Test Accuracy for each Subject using Selective Channels

5.4 CLASSIFICATION RESULT OF PRUNED MODEL

The training accuracy achieved on the whole dataset is 83.08% while the testing accuracy achieved is 82.67 on the 1DCNN model implemented.

```
Epoch 1/10
4410/4410 [==============================] - 59s 13ms/step - loss: 1.6998 - accuracy: 0.4010 - val_loss: 1.3222 - val_accuracy: 0.5593
Epoch 2/10
4410/4410 [==============================] - 40s 9ms/step - loss: 1.0813 - accuracy: 0.6146 - val_loss: 1.0971 - val_accuracy: 0.6120
Epoch 3/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.8586 - accuracy: 0.6860 - val_loss: 0.8139 - val_accuracy: 0.7194
Epoch 4/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.7204 - accuracy: 0.7315 - val_loss: 0.7251 - val_accuracy: 0.7454
Epoch 5/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.6260 - accuracy: 0.7629 - val_loss: 0.6564 - val_accuracy: 0.7697
Epoch 6/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.5652 - accuracy: 0.7860 - val_loss: 0.5647 - val_accuracy: 0.7972
Epoch 7/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.5219 - accuracy: 0.8013 - val_loss: 0.5212 - val_accuracy: 0.8149
Epoch 8/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.4833 - accuracy: 0.8160 - val_loss: 0.4844 - val_accuracy: 0.8300
Epoch 9/10
4410/4410 [==============================] - 38s 9ms/step - loss: 0.4634 - accuracy: 0.8242 - val_loss: 0.4508 - val_accuracy: 0.8395
Epoch 10/10
4410/4410 [==============================] - 37s 8ms/step - loss: 0.4452 - accuracy: 0.8308 - val_loss: 0.4850 - val_accuracy: 0.8267
1890/1890 [==============================] - 6s 3ms/step - loss: 0.4850 - accuracy: 0.8267
[0.48499634861946106, 0.8267268538475037]
```

Figure 5.4: Training/Testing Accuracy of Pruned Model

Table 5.4: Before and After Pruning Comparison

Model	Size of model before quantization	Size of model after quantization	Drop in size of model	Accuracy before pruning andquantization	Accuracy after pruningand quantization	Drop in accuracy
EEG brain signal classification	24.38 MB	9.26 MB	15.12 MB	85.13%	82.67%	0.02%

5.5 MULTI MODEL COMPARISON

Model 1

The Convolutional Neural Network (CNN) model, referred to as "model1," is designed for a classification task. It starts with two 1D convolutional layers with 64 filters each and a kernel size of 3, followed by ReLU activation functions. A dropout layer with a rate of 0.5 is added to reduce overfitting, and a max-pooling layer with a pool size of 2 is used for down-sampling. The output from these layers is then flattened, and two fully connected layers follow, with 100 and 64 units, respectively, using ReLU and softmax activation functions.

```
Layer (type)                 Output Shape              Param #
================================================================
conv1d (Conv1D)              (None, 498, 64)           12352

conv1d_1 (Conv1D)            (None, 496, 64)           12352

dropout (Dropout)            (None, 496, 64)           0

max_pooling1d (MaxPooling1D   (None, 248, 64)          0
)

flatten (Flatten)            (None, 15872)             0

dense (Dense)                (None, 100)               1587300

dense_1 (Dense)              (None, 64)                6464

================================================================
Total params: 1,618,468
Trainable params: 1,618,468
Non-trainable params: 0
```

Figure 5.5.1: Model1 Summary

Model 2

The CNN model labeled as "model2" is designed for a classification task. It consists of a single 1D convolutional layer with 32 filters and a kernel size of 3, employing ReLU activation. To prevent overfitting, a dropout layer with a rate of 0.5 is applied, followed by max-pooling with a pool size of 2. The data is then flattened, and two fully connected layers with 100 and 32 units, respectively, utilizing ReLU and softmax activation functions, are added. The model is compiled with the sparse categorical cross-entropy loss function, the Adam optimizer, and accuracy as the performance metric. This architecture is suitable for classification tasks and may require further fine-tuning based on the specific problem at hand.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 498, 32)           6176

dropout (Dropout)            (None, 498, 32)           0

max_pooling1d (MaxPooling1D  (None, 249, 32)           0
)

flatten (Flatten)            (None, 7968)              0

dense (Dense)                (None, 100)               796900

dense_1 (Dense)              (None, 32)                3232

=================================================================
Total params: 806,308
Trainable params: 806,308
Non-trainable params: 0
```

Figure 5.5.2: Model2 Summary

Model 3

The model has two convolutional layers, each with 32 filters and a kernel size of 3, followed by a dropout layer with a rate of 0.5 to reduce overfitting. The output of the dropout layer is then passed through a max pooling layer with a pool size of 2. The resulting feature maps are then flattened and passed through two fully connected layers, with 100 and 12 units respectively, and ReLU and softmax activation functions.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 498, 32)           6176

conv1d_1 (Conv1D)            (None, 496, 32)           3104

dropout (Dropout)            (None, 496, 32)           0

max_pooling1d (MaxPooling1D  (None, 248, 32)           0
)

flatten (Flatten)            (None, 7936)              0

dense (Dense)                (None, 100)               793700

dense_1 (Dense)              (None, 12)                1212

=================================================================
Total params: 804,192
Trainable params: 804,192
Non-trainable params: 0
```

Figure 5.5.3: Model3 Summary

Table 5.5: Size vs Accuracy Comparison

MODEL NAME	Training accuracy	Testing Accuracy	SIZE
MODEL 1	76.55	74.97	12.67 MB
MODEL 2	83.08	82.67	19.02 MB
MODEL 3	86.30	85.48	24.38 MB

6. CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

The machine learning model developed in this project has achieved a training accuracy of 83.08% and a testing accuracy of 82.67%. This indicates that the model is able to accurately detect and classify different types of EEG signals with a high degree of accuracy.

When tested on each subject taking all the electrodes average training and testing accuracy 59.29% and 61.5% while taking 10 electrodes training and testing accuracies increased to 66.73% and 66.32 respectively. This suggests on decreasing the dataset of each subject accuracies increased.

These results suggest that the model can be used for various applications, such as medical diagnosis or brain-computer interface technology. However, it is important to note that there may be limitations to the model's accuracy, especially in cases where the input data is noisy or contains other confounding factors.

6.2 FUTURE WORK

Future work could include improving the model's accuracy by incorporating additional features or using more sophisticated classification algorithms. It may also be beneficial to test the model on a larger and more diverse dataset to ensure its generalizability to different populations and conditions.

7. REFERENCES

1. M. F. Hinss, B. Somon, F. Dehais and R. N. Roy, "Open EEG Datasets for Passive Brain-Computer Interface Applications: Lacks and Perspectives," 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER), Italy, 2021, pp. 686-689, doi: 10.1109/NER49283.2021.9441214.

2. Ying J, Wei Q, Zhou X. Riemannian geometry-based transfer learning for reducing training time in c-VEP BCIs. Sci Rep. 2022 Jun 14;12(1):9818. doi: 10.1038/s41598-022-14026-y. PMID: 35701505; PMCID: PMC9197830.

3. Raphaëlle N. Roy 1,2*, Marcel F. Hinss 1, Ludovic Darmet 1, Simon Ladouce1,Emilie S. Jahanpour 1, Bertille Somon1,2, Xiaoqi Xu1,2, Nicolas Drougard1,2,Frédéri Dehais 1,2 and Fabien Lotte “Brain-Computer Interface Competition on Cross-Session Workload Estimation’’ 04 April 2022

4. Sara Sedlar, Johann Benerradi, Côme Le Breton, Rachid Deriche, Théodore Papadopoulo, et al.. Rank-1 CNN for mental workload classification from EEG. Neuroergonomics conference, Sep 2021, Munich (virtual event), Germany. ⟨hal-03357020⟩

5. Lan T, Erdogmus D, Adami A, Mathan S, Pavel M. Channel selection and feature projection for cognitive load estimation using ambulatory EEG. Comput Intell Neurosci. 2007;2007:74895. doi: 10.1155/2007/74895. PMID: 18364990; PMCID: PMC2267884.

6. Alexandre Barachant, Stéphane Bonnet, Marco Congedo, Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. Neurocomputing, 2013, 112, pp.172-178. ⟨10.1016/j.neucom.2012.12.039⟩. ⟨hal-00820475⟩