MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

BRAIN COMPUTER INTERFACE DESIGN AND IMPLEMENTATION USING MACHINE LEARNING WITH USER FEEDBACK

EMRE ARI

Ph.D. THESIS
Department of Mechanical Engineering

Thesis Supervisor
Prof. Dr. Ertuğrul TAÇGIN

Thesis Co-Supervisor
Asst. Prof. Dr. Alper ŞİŞMAN

ISTANBUL, 2023

MARMARA UNIVERSITY

INSTITUTE FOR GRADUATE STUDIES

IN PURE AND APPLIED SCIENCES

BRAIN COMPUTER INTERFACE DESIGN AND IMPLEMENTATION USING MACHINE LEARNING WITH USER FEEDBACK

EMRE ARI

(724616007)

Ph.D. THESIS

Department of Mechanical Engineering

Thesis Supervisor

Prof. Dr. Ertuğrul TAÇGIN

Thesis Co-Supervisor

Asst. Prof. Dr. Alper ŞİŞMAN

ISTANBUL, 2023

ACKNOWLEDGEMENT

After years of blood, sweat and tears, this is the end. While this thesis is a scientific study, it was also a challenge to myself. I am proud and happy to be able to complete this work after years of nonstop flights, years of learning and practice in multidisciplinary areas, and periods of deep uncertainty.

It would certainly not be possible for me to complete this journey alone. I would like to express my sincere thanks and respect to my advisor Prof. Dr. Ertuğrul TAÇGIN, from whom I learned a lot, who always supported me with his patience and understanding, never lacked his friendliness and kindness, always focused on solving problems and guided me with his knowledge and foresight. I would also like to thank my co-advisor Asst. Prof. Dr. Alper ŞİŞMAN for encouraging me to start this thesis work.

I would like to express my sincere thanks to Prof. Dr. Vedat ORUÇ, who expressed his trust in me under all circumstances, encouraged me and provided all kinds of convenience for me to complete this thesis. I would like to express my gratitude to Dr. Kasım ŞİMŞEK for his unwavering support in my studies and for supporting me in all circumstances. I would like to thank to my dear friend Muhammed Hilmi KAVAK, who became a lawyer instead of being an engineer, opened his house to me during my travels, and spent sleepless nights questioning life. I would like to thank to my dear friend Abdulkadir YILDIZ, who took my troubles during both my undergraduate and graduate education and gave me all kinds of support in this thesis.

Last but not least, I would like to express my endless thanks and love to my dear mother Hilal ARI and my dear father Abdulgani ARI, who supported me unconditionally and strongly during this thesis period, as they did throughout my entire life, who believed in me in every situation, helped me whenever I needed, and did the best they could. Having parents like you makes me happy and proud. I present my heartfelt thanks and hugs to my dear brothers, middle bee Yunus ARI, little bee Enes ARI and queen bee Sueda ARI, to whom I transfer stress during tiring and difficult times, do their best when I need them, and I am proud to be their elder brother. I would also like to express my love and thanks to my grandmother and aunts, who have never spared their good wishes and support.

July 2023 **Emre ARI**

TABLE OF CONTENTS

ÖZET

KULLANICI GERİBESLEMELİ MAKİNE ÖĞRENMESİ İLE BEYİN BİLGİSAYAR ARAYÜZÜ TASARIMI VE UYGULAMASI

Beyin bilgisayar arayüzleri, son yılların en heyecan verici ve en zorlu araştırma alanlarından biridir. Nörobilim alanındaki ilerlemeler, bilgisayarların işlemci güçlerinin artması ve yeni nesil makine öğrenmesi yöntemlerinin ortaya çıkmasıyla beraber bu alanda umut verici çalışmalar yapılmaya başlanmıştır. Bu çalışmalar sağlıktan eğitime, eğlenceden güvenliğe, çevre etkileşiminden pazarlamaya kadar geniş bir alana yayılmıştır. Beyin bilgisayar arayüzlerinin geliştirilmesinde en önemli noktalardan biri beyin sinyallerinin hızlı, doğru ve robust bir şekilde sınıflandırılmasıdır. Bu sınıflandırma aşamasında elde edilecek başarı, sınıflandırma öncesi aşama olan sinyal toplama ve sınıflandırma sonrası aşama olan dış aygıt kontrolünün performasına doğrudan ve yüksek oranda etki edecektir. Bu tezin temel amacı, motor imagery beyin sinyallerinin bir evrişimli sinirsel ağ modeli geliştirilerek yüksek performanslı bir şekilde sınıflandırılması, bir arayüz yazılımı geliştirilerek sınıflandırıcının bu arayüze entegre edilmesi ve bu arayüz ile dış bir aygıtın beyin sinyalleriyle kontrol edilmesidir. Tezin ilk aşamasında, literatürde en çok kullanılan motor imagery verisetlerinden olan BCI Competition IV – 2A ve BCI Competition IV – 2B verisetleri analiz edilmiş ve bu veriler kullanılarak herhangi bir sinyal önişleme yapmadan ham veriler kullanan ve birçok son teknoloji modelden daha iyi sonuçlar elde eden iki evrişimli sinirsel ağ modeli geliştirilmiştir (NF-EEG, IS-EEG). Tezin ikinci aşamasında bir beyin bilgisayar arayüz yazılımı geliştirilmiştir (NF-BCI). Bu arayüz, üç ana görevi gerçek zamanlı olarak yerine getirebilmektedir. İlk olarak arayüze beyin sinyalleri girmekte ve giren beyin sinyalleri gerçek zamanlı olarak arayüz üzerinde çizdirilmektedir. İkinci olarak arayüze geliştirilen NF-EEG ve IS-EEG modelleri entegre edilmiştir ve bu modeller sisteme giren sinyalleri gerçek zamanlı olarak sınıflandırmaktadırlar. Üçüncü olarak arayüz ile bir mikrokontrolör bağlantısı kurulmuştur ve arayüz bilgisayarı ve bir robotik kolu kontrol edebilmektedir. Tezin üçüncü aşamasında ise eklemeli imalat yöntemiyle bir robot kol üretimi yapılmıştır ve bu robot kol mekanik ve elektronik bileşenleri bir araya getirilerek arayüze bağlanmıştır. Bu sayede hem tekil görevleri yerine getirebilecek hem de gelecek araştırmalarda kullanılabilecek bir beyin bilgisayar arayüzü geliştirilmiştir.

ABSTRACT

BRAIN COMPUTER INTERFACE DESIGN AND IMPLEMENTATION USING MACHINE LEARNING WITH USER FEEDBACK

Brain-computer interfaces are one of the most exciting and challenging areas of research in recent years. With the advances in neuroscience, the increase in the processing power of computers and the emergence of new generation machine learning methods, promising studies have begun to be made in this field. These studies have spread over a wide area from health to education, from entertainment to safety, from environmental interaction to marketing. One of the most important points in the development of brain computer interfaces is the fast, accurate and robust classification of brain signals. Success in this classification stage will directly and highly affect the performance of the signal receiving stage, which is the pre-classification stage, and the performance of the external device control, which is the post-classification stage. The main purpose of this thesis is to classify motor imagery brain signals with high performance by developing a convolutional neural network model, to develop an interface software and to integrate the classifier into this interface and to control an external device with brain signals with this interface. In the first stage of the thesis, BCI Competition IV – 2A and BCI Competition IV – 2A datasets, which are the most used motor imagery datasets in the literature, were analyzed and using these data, two convolutional neural network model has been developed (NF-EEG, IS-EEG) that use raw data without any signal preprocessing and outperformed many state-of-the-art models. In the second phase of the thesis, a brain computer interface software was developed (NF-BCI). This interface can perform three main tasks in real time. First of all, brain signals enter the interface and incoming brain signals are plotted on the interface in real time. Secondly, developed NF-EEG and IS-EEG models are integrated into the interface and these models classify the signals entering the system in real time. Third, a microcontroller connection has been established with the interface, and the interface can control the computer and a robotic arm. In the third stage of the thesis, a robot arm was produced with the additive manufacturing method and this robot arm was connected to the interface by bringing together the mechanical and electronic components. As a result of this thesis, a brain-computer interface has been developed that can both perform individual tasks and be used in future research.

CLAIM FOR ORIGINALITY

BRAIN COMPUTER INTERFACE DESIGN AND IMPLEMENTATION USING MACHINE LEARNING WITH USER FEEDBACK

Brain Computer Interface (BCI) systems have been developed to identify and classify brain signals and integrate them into a control system. Even though many different methods and models have been developed for the brain signals classification, the majority of these studies have emerged as specialized models. In addition, preprocessing and signal preprocessing methods which are largely based on human knowledge and experience have been used extensively for classification models. These methods degrade the performance of real-time BCI systems and require great time and effort to design and implement the right method. The novelties of this research are stated below:

- A new CNN architecture has been developed (NF-EEG) that outperforms many state-of-the-art models without using any signal processing methods. In this way, labor and time required for signal processing methods were saved, and a generalizable and robust model with low standard deviations was obtained.
- The effects of different input shape structures on system performance during the training of CNN models with EEG signals were investigated. For this subject, which has not been specifically researched in the literature before, the new IS-EEG architecture has been developed and this model has been trained by selecting the appropriate input shape structure and outperformed many state-of-the-art models by using raw EEG data.
- A new real-time end-to-end brain computer interface software has been developed (NF-BCI). This interface can read EEG signals and plot them on the interface. It can perform real-time classification by running NF-EEG and IS-EEG classifiers in the background. It can also establish a connection with the computer it is working on and with a microcontroller. By using the classification results it has obtained, it can control the computer it is working on and an external device through the microcontroller with pre-defined control scenarios.

SYMBOLS

$\boldsymbol{\mu}$	: Mean
$\boldsymbol{\sigma}$	: Standard deviation
$\boldsymbol{\sigma}^{\mathbf{2}}$	: Variance
$\mathbf{B}_{\mathbf{c}}$	: Number of biases
C	: Number of channels
K	: Size of kernels
N	: Number of kernels
$\mathbf{P}_{\mathbf{c}}$	: Number of parameters
$\mathbf{W}_{c}$	: Number of weights

ABBREVIATIONS

ANN	: Artificial neural network
AR	: Autoregressive components
BCI	: Brain computer interface
BMI	: Brain machine interface
BOLD	: Blood oxygen level dependent
CNN	: Convolutional neural network
CPU	: Central processing unit
CSP	: Common spatial pattern
c-VEP	: Code-modulated visual evoked potential
CWD	: Choi-Williams distribution
CWT	: Continuous wavelet transform
DBN	: Deep belief network
DFT	: Discrete Fourier transform
DNN	: Deep neural network
ECoG	: Electrocorticography
EEG	: Electroencephalography
EKG	: Electrocardiogram
ELU	: Exponential linear unit
EMG	: Electromyogram
EOG	: Electrooculography
ERD	: Event-related desynchronization
ERP	: Event related potentials
ERS	: Event-related synchronization
FBCSP	: Filter bank common spatial pattern

FBCSPT	: Filter back common spatial pattern time
FFT	: Fast Fourier transform
fMRI	: Functional magnetic resonance imaging
fNIRS	: Functional near infrared spectroscopy
f-VEP	: Frequency-modulated visual evoked potential
GA	: Genetic algorithms
GPU	: Graphics processing unit
HMI	: Human machine interface
Hz	: Hertz
I/O	: Input/output
I2C	: Inter-integrated circuit
IC	: Integrated circuit
ICA	: Independent component analysis
ICSP	: In-circuit serial programming
IS-EEG	: Input Shape EEG
K-NN	: K-nearest neighbors
LDA	: Linear discriminant analysis
MEG	: Magnetoencephalography
MF	: Matched filtering
MHz	: Megahertz
MI	: Motor imagery
MLP	: Multilayer perceptron
moVEP	: Motion onset visual evoked potentials
NF-BCI	: No-Filter BCI
NF-EEG	: No-Filter EEG

NIR	: Near infrared
NIRS	: Near-infrared spectral imaging
NN	: Neural Network
PCA	: Principal component analysis
PET	: Positron emission tomography
PWM	: Pulse width modulation
QDA	: Quadratic discriminant analysis
RAM	: Random access memory
RBF	: Radial basis function
RBM	: Restricted Boltzmann machines
RELU	: Rectified linear units
RGC	: Riemannian geometry-based classifiers
RNN	: Recurrent neural network
ROM	: Read only memory
SAE	: Stacked autoencoders
SBS	: Sequential backward selection
SCPs	: Slow cortical potentials
SFS	: Sequential forward selection
Slap	: Surface Laplacian
SNN	: Spiking neural network
SNR	: Signal-to-noise ratio
SpecCSP	: Spectrally weighted common spatial pattern
SPoC	: Source power co-modulation
SSA	: Stationary subspace analysis
SSVEPs	: Steady-state visual evoked potentials

STD	: Standard deviation
STFT	: Short-time Fourier transform
SVM	: Support vector machines
TPU	: Tensor processing unit
t-VEP	: Time-modulated visual evoked potential
TVEPs	: Transient visual evoked potentials
VEPs	: Visual evoked potentials
WPD	: Wavelet package decomposition
WT	: Wavelet transform

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this section, first of all, brain-computer interfaces are introduced and detailed information about these systems is given. After explaining the basic components and working principles of these systems, the signal acquisition methods used to obtain the brain signals needed in the brain-computer interfaces are explained. Researchers have used various signal preprocessing methods to make the obtained brain signals more understandable and processable. These preprocessing methods are mentioned after the signal acquisition methods. After signal preprocessing, various processing and feature extraction methods are used to extract meaningful information from brain signals. After the preprocessing sub-section, these methods are also mentioned. The next sub-section describes how brain signals are classified and what methods are used for classification. In the following two sub-titles, information is given about the different paradigms used to develop brain-computer interfaces and the usage areas of brain-computer interfaces. In the last part of this section, the main purpose of the thesis, the developed methods and achievements are explained in detail together with the organizational structure of the thesis.

1.1. Brain Computer Interfaces

Brain Computer Interface (BCI) is the name given to systems that allow external devices to be controlled via brain signals using software and hardware communication systems. These systems are also called Brain Machine Interface (BMI) or Human Machine Interface (HMI). The primary purpose of these systems is to enable people with paralysis or neurological disorders to gain communication and movement skills. Thanks to these systems, people can use brain signals as control signals without using their nervous and muscular systems and interact with their environment through these signals. In this way, they can use computers, mobile phones, prostheses, auxiliary equipment, peripheral equipment or any desired device without using their muscular or nervous systems. In addition, BCI systems are used in entertainment areas for healthy individuals and in commercial areas for various industries [1-5].

A traditional BCI development process consists of signal acquisition, signal preprocessing, feature extraction, classification, application interface, and implementation processes. The generalized steps of a brain computer interface are shown in figure 1.1.

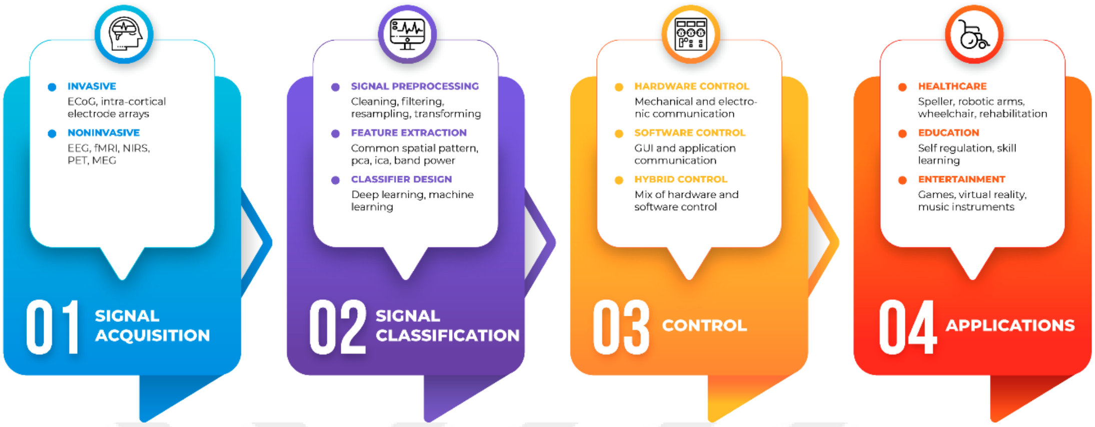

INVASIVE
ECoG, intra-cortical electrode arrays
NONINVASIVE
EEG, fMRI, NIRS, PET, MEG
01 SIGNAL ACQUISITION
SIGNAL PREPROCESSING
Cleaning, filtering, resampling, transforming
FEATURE EXTRACTION
Common spatial pattern, pca, ica, band power
CLASSIFIER DESIGN
Deep learning, machine learning
02 SIGNAL CLASSIFICATION
HARDWARE CONTROL
Mechanical and electronic communication
SOFTWARE CONTROL
GUI and application communication
HYBRID CONTROL
Mix of hardware and software control
03 CONTROL
HEALTHCARE
Speller, robotic arms, wheelchair, rehabilitation
EDUCATION
Self regulation, skill learning
ENTERTAINMENT
Games, virtual reality, music instruments
04 APPLICATIONS

Figure 1.1. Generalized steps of a brain computer interface

1.2. Signal Acquisition Methods in BCI

1.2.1. Electroencephalography (EEG)

The most widely used noninvasive method to collect brain signals is EEG [6,7]. EEG measures voltage fluctuations that occur during electrical current in neurons. To collect EEG signals, electrodes are placed on the skull and the amplitude of the signals is measured with these electrodes. Because the number of electrodes that can be placed on the skull is limited, EEG signals have low spatial resolution. Although EEG signals have lower spatial resolution, they have better temporal resolution [8]. Environmental noises, interferences, skull and cortex-based disorders, as well as mental states, fatigue, and person-to-person characteristics cause EEG signals to have low SNR values. The EEG signal acquisition session is shown in figure 1.2.

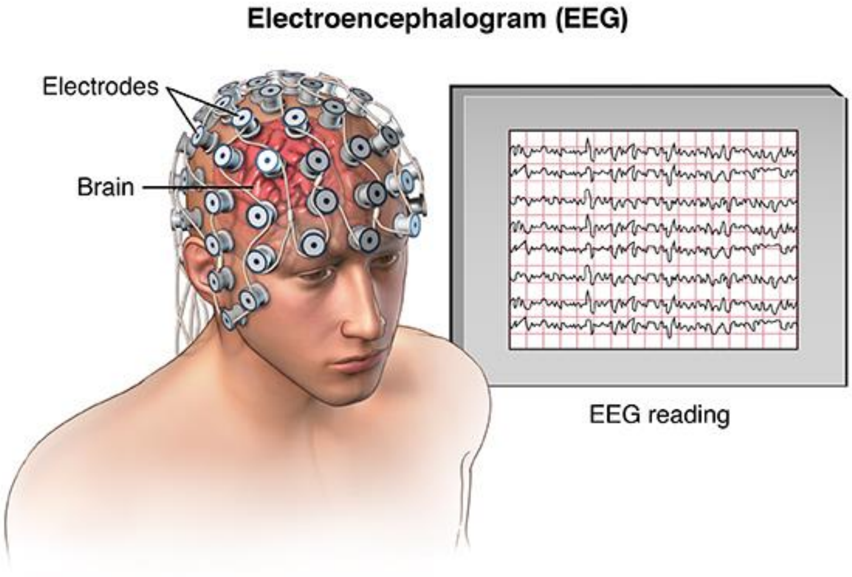


Figure 1.2. Electroencephalography [9]

EEG signals are classified according to different wavelengths. These classes are shown in figure 1.3.

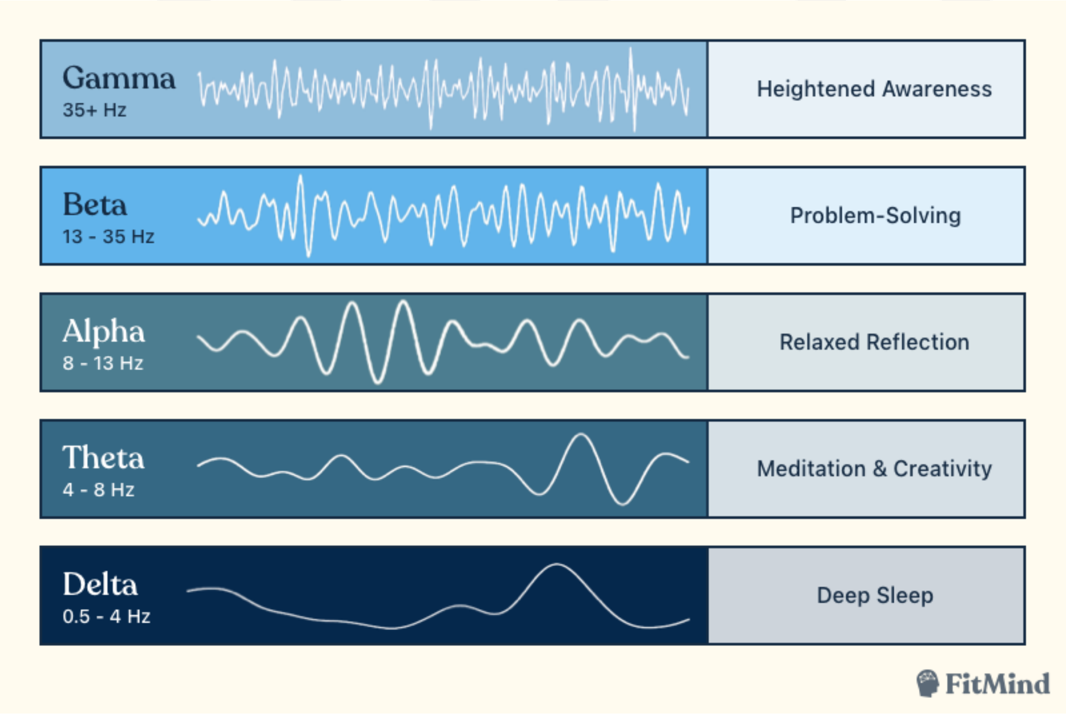


Figure 1.3. Human Brainwaves [10]

1.2.2. Magnetoencephalography (MEG)

With this method, functional neuroimaging is performed by recording the magnetic fields created by electric currents in the brain using very sensitive magnetometers. In order for magnetic fields to be perceived, tens of thousands of neurons must act together and in a certain order. The signal quality in MEG method depends on many factors such as brain region, number of active neurons, neuron distribution, sequence and depth, and therefore it has lower spatial resolution than other methods [11,12]. Despite this disadvantage, MEG method has very high temporal resolution. The reason for this high resolution is that it measures neuron activity directly, not indirectly. MEG devices are not widely used because they are expensive, heavy and not portable. The MEG signal acquisition session is shown in figure 1.4.

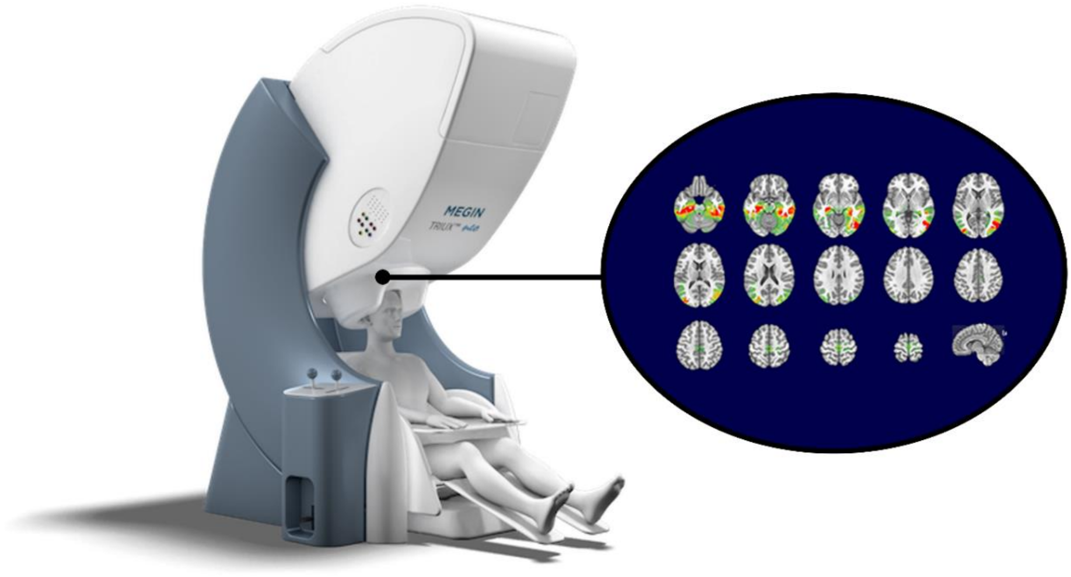


Figure 1.4. Magnetoencephalography [13]

1.2.3. Electrocorticography (ECoG)

ECoG is an invasive electrophysiological method used to collect brain signals. Electrodes are placed under the skull above the epidural to collect brain signals. This method offers high spatial resolution and high SNR, but has a higher surgical risk [14]. ECoG method stands out as a lower risk method compared to intracortical signal acquisition methods. Higher amplitude signals are obtained with ECoG compared to noninvasive signal acquisition methods. The amplitude of EEG signals is less than 20 qV, and ECoG signals are greater than 50 qV [15]. Thanks to this feature, ECoG signals are more resistant to artificial effects and external factors. Also, ECoG signals have higher bandwidth than EEG signals [16]. Thanks to this higher bandwidth, they offer the possibility to obtain a wider range of information from the brain, and this capability can help develop more efficient and high-performance BCI models. In addition to these advantages, the fact that it is an invasive method and surgical risks limit the use of this method in a wider area. The method of obtaining ECoG signal is shown in figure 1.5.

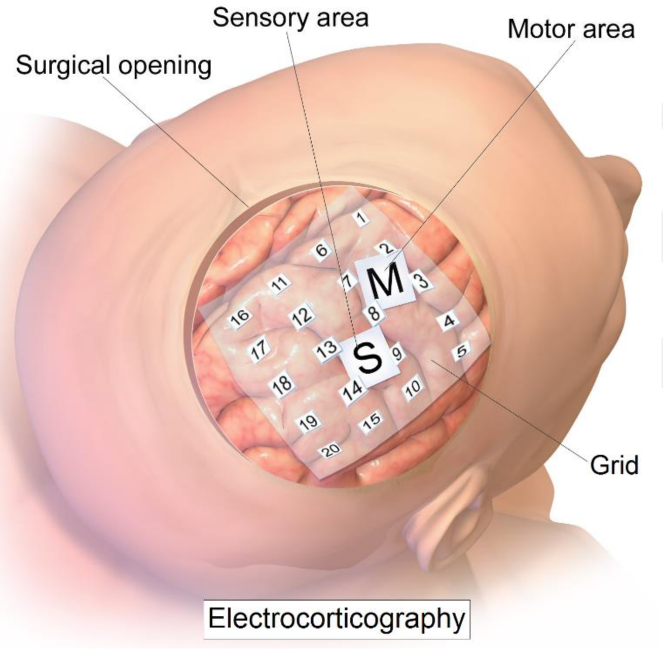


Figure 1.5. Electrocorticography [17]

1.2.4. Intracortical Neuron Recording

In this signal acquisition method, brain signals are collected by placing electrodes in the cerebral cortex of the person. With this method, high resolution brain signals can be obtained. With this method, voltage fluctuations from small groups of neurons to large groups of neurons, spikes and action potentials known as field potentials can be detected. The electrodes used in this method can be a single electrode or arrays of electrodes. The intracortical neuron recording method is shown in figure 1.6.

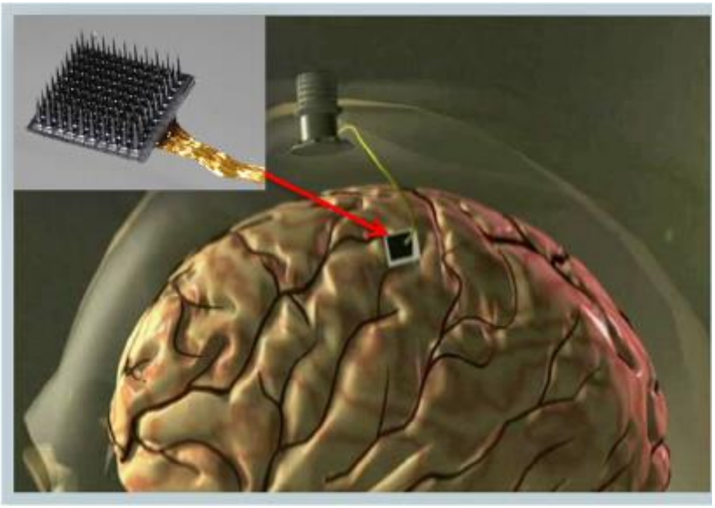


Figure 1.6. Intracortical neuron recording [18]

1.2.5. Functional Magnetic Resonance Imaging (fMRI)

fMRI is a method for monitoring brain activity, allowing to monitor changes in blood flow in the brain [19]. Like the fNIRS method, the fMRI method uses the blood oxygen level dependent (BOLD) response [20]. The fMRI method measures the BOLD response by magnetic rather than optical methods. The penetrating ability of magnetic fields is better than NIR rays, so the activities in the deeper parts of the brain can be monitored and information can be collected with the fMRI method. fMRI and fNIRS methods have similar temporal resolution as they measure blood flow velocity, but the fMRI method has higher spatial resolution than the fNIRS method. Since fMRI devices are heavy, expensive, and low in portability, their use is limited. The fMRI signal acquisition method is shown in figure 1.7.

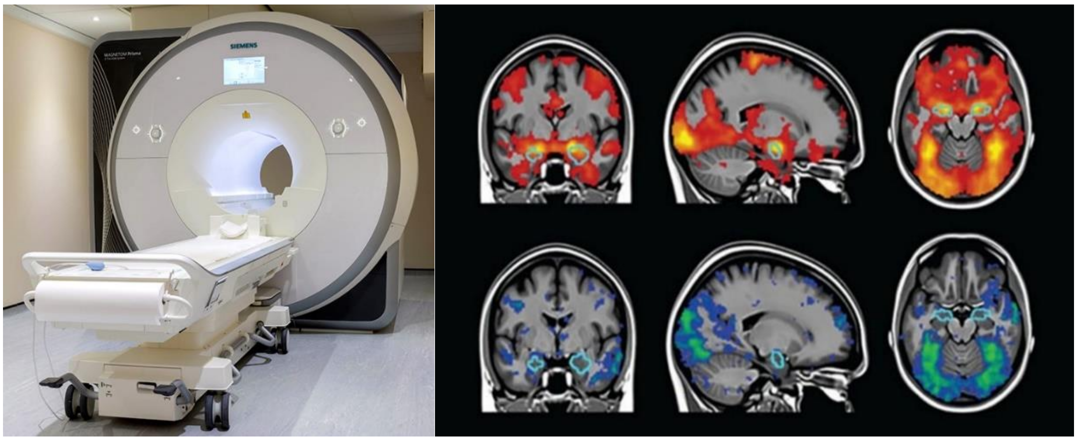


Figure 1.7. Functional magnetic resonance imaging [21]

1.2.6. Functional Near Infrared Spectroscopy (fNIRS)

fNIRS is a non-invasive functional neuroimaging method using near infrared light [22]. This method uses the light absorption capacity of hemoglobin and measures the amount of oxygenated hemoglobin and deoxygenated hemoglobin together with near infrared light. The fNIRS method is a method that creates a functional neuroimage by measuring hemodynamic responses or blood-oxygen-level-dependent (BOLD) responses [23]. The BOLD response detects the level of oxygenated or deoxygenated blood in the brain. Level increases and steady-state level changes other than neural activity indicate activity in neurons and metabolic changes occurring. fNIRS signals have lower temporal resolution compared to electrical and magnetic signals. The spatial resolution of this method depends on the number and distribution of emitter and detector pairs used in the device. Limitations of light emissivity and spatial resolution do not make it possible to measure cortical activity deeper than 4 cm from the brain surface with this method. The fNIRS signal acquisition method is shown in figure 1.8.

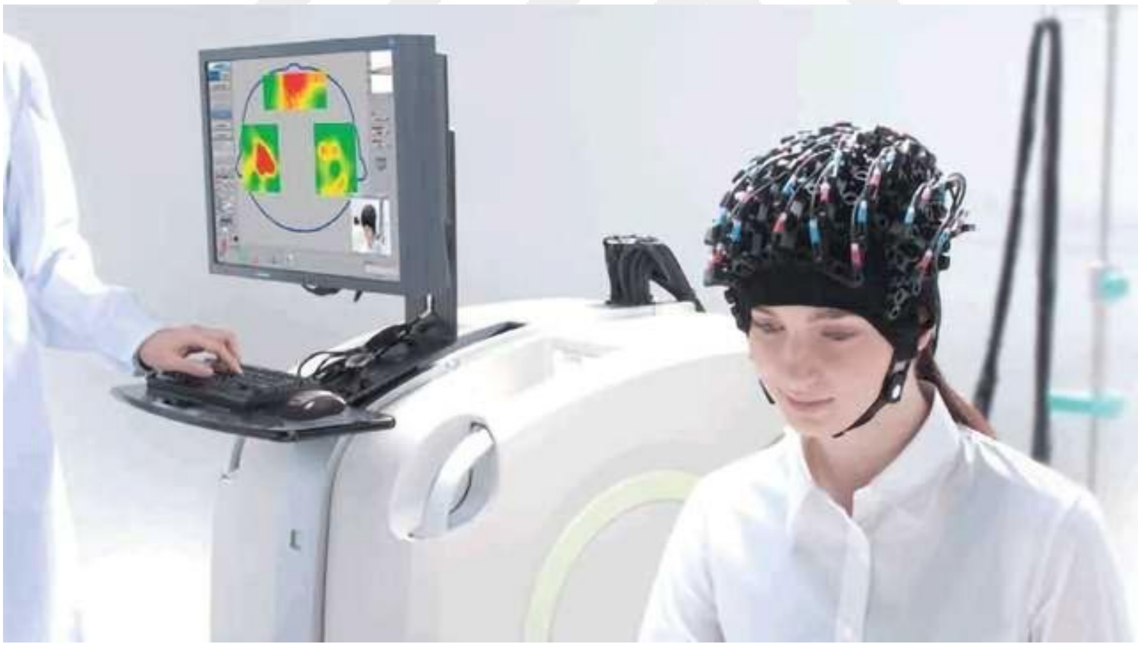


Figure 1.8. Functional Near Infrared Spectroscopy [24]

1.2.7. Electrooculography (EOG)

Unlike the previously described brain signals, EOG signals are derived from voltages that occur as a result of eye movements. These signals can be measured with two electrodes placed on the face and close to the eye at both ends. EOG signals are often used to improve the performance of BCI models in the development of hybrid BCI models. In their studies, the researchers stated that EOG signals have a disruptive effect on motor imagery signals and they detected and reduced this disruptive effect of the EOG signals they measured, and they developed hybrid BCI models [25, 26].

1.3. Signal Preprocessing Methods in BCI

1.3.1. Band Pass Filter

A Band-Pass Filter is a filter that transmits signals within a certain range and attenuates signals outside this range. These filters are designed on the assumption that the brain produces EEG signals in certain frequency ranges and that signals outside this range are artificial or disruptive factors. Generally, signals above 1 Hz and below 36 Hz are used as EEG signals. The band-pass filter is shown in figure 1.9.

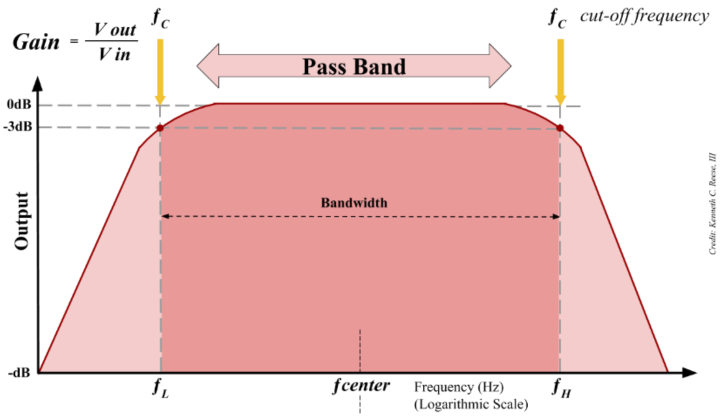


Figure 1.9. Band Pass Filter [27]

1.3.2. Notch Filter

The notch filter is used to filter signals of certain frequency. One of the most common noises when recording EEG signals is power line signal frequencies. It is used to filter out power line noises that usually occur at 50/60 Hz. The notch filter is shown in figure 1.10.

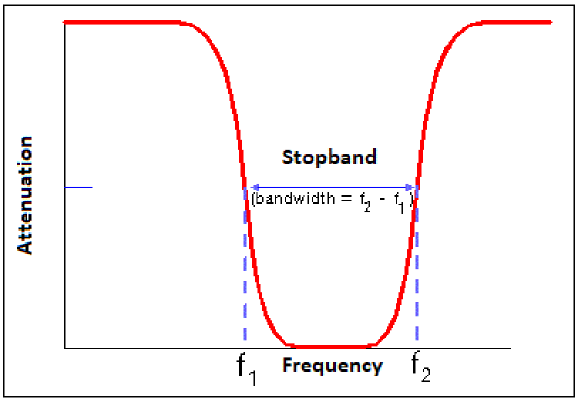


Figure 1.10. Notch Filter [28]

1.3.3. Median Filtering

Median filtering is a nonlinear filtering method used for noise reduction. With this method, the signal is completely scanned to cover each component and each component is replaced by the median value of the neighboring components. Adjacent component patterns are called windows, and this windows slides across the entire signal. In one-dimensional signals, the window represents the first and last few components of the real component, while in two or more dimensional signals it represents a kernel (square, radius, or ellipsoidal) region. The median filter is shown in figure 1.11.

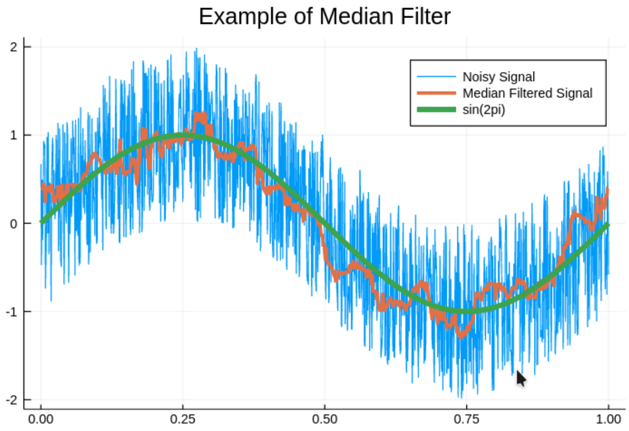


Figure 1.11. Median Filtering [29]

1.3.4. Moving Average Filter

The moving average filter is often used for time series data and provides reduction of random noise while maintaining sharp step responses. With this method, short-term fluctuations are smoothed out and long-term patterns and trends are revealed. By determining a certain number of steps for a series, the values in this step are averaged and replaced with the first digit of the series. This step is then shifted through the series, replacing the other values in the series with the mean values. The number of steps to be averaged can be determined by the type of signal obtained and reveal the desired short or long-term patterns. The moving average filter is shown in figure 1.12.

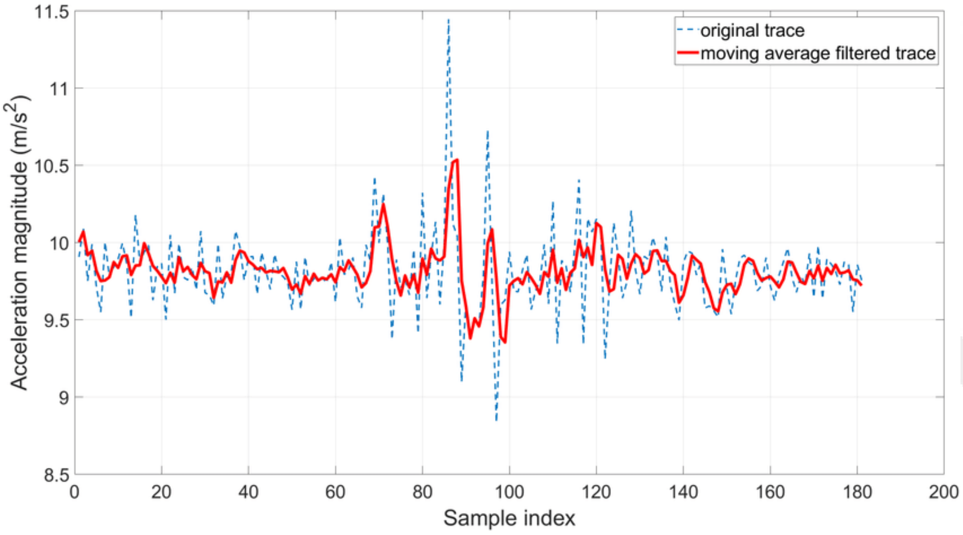


Figure 1.12. Moving Average Filter [30]

1.3.5. Baseline Drift Removal

Baseline drift is defined as a long duration noise and occurs as a result of changing the fundamental position of the signal. Baseline drift occurs as low-frequency slow oscillations. The reasons for these shifts can be many different factors. Muscle and eye movements, brain activity, head and neck movements, fatigue, sweating, movements of electrodes attached to the head and external factors may be the causes of these shifts. Baseline drift removal aims to eliminate these noises and shifts. The baseline drift removal is shown in figure 1.13.

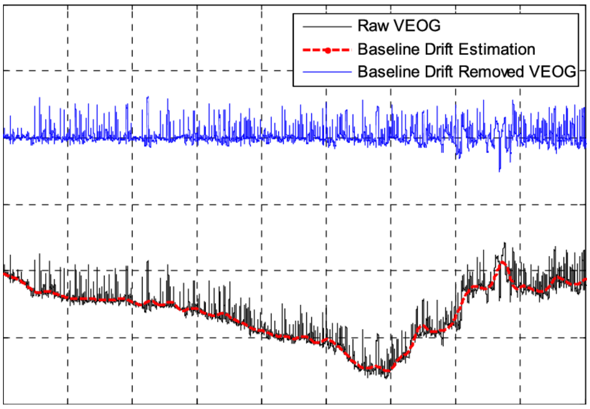


Figure 1.13. Baseline Drift Removal [31]

1.3.6. Wavelet Denoising

The wavelet denoising method localizes the signal features so that important features of the signals can be preserved while denoising the signal. With this method, the signal properties are clarified with high amplitude wavelet coefficients. Wavelet coefficients with small values are generally considered noise and these coefficients are subtracted from the signals to preserve the signal characteristics and obtain clearer signals. Coefficient value limits can be defined by the user and applied to signals according to the desired result or signal type. Wavelet denoising is shown in figure 1.14.

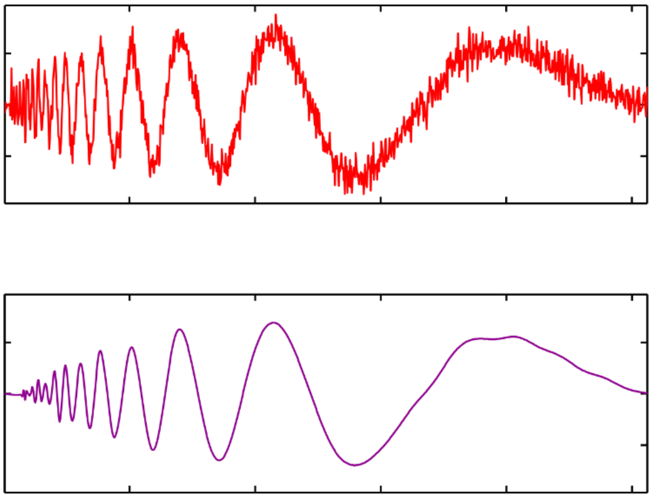


Figure 1.14. Wavelet Denoising [32]

1.4. Signal Processing and Feature Extraction Methods in BCI

1.4.1. Principal Component Analysis (PCA)

Principal component analysis is a size reduction method used to analyze large, complex and multidimensional data. With this method, large multidimensional data sets are divided into smaller variables that will reflect the data without losing their essential properties. It is possible that some of the data applied to PCA will lose their properties, but this method is preferred to make the data more understandable and classifiable. This method can also be applied to redundant data, that is, to some interrelated variables. In this way, different interrelated variables can be converted into a lower-dimensional or single main component, and these components can contain many distinguishing features of other parameters. PCA is applied to analyze EEG data, reduce the size of the data and make the data more understandable and classifiable. Principal component analysis is shown in figure 1.15.

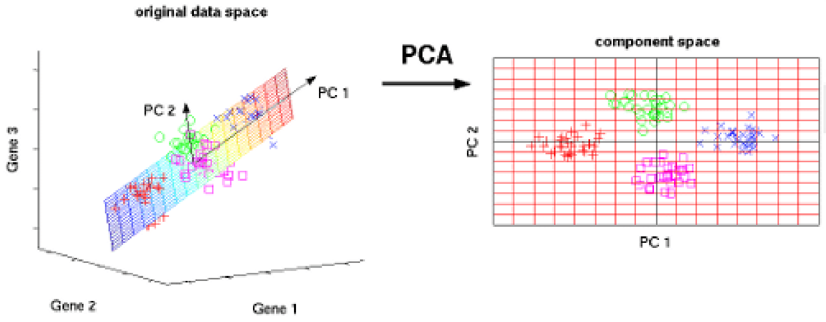


Figure 1.15. Principal component analysis [33]

1.4.2. Independent Component Analysis (ICA)

Independent component analysis is a computational method of blind source separation that divides a multidimensional and multivariate signal into interconnectable subcomponents. When dividing these components, it is assumed that at most one of the components is Gaussian and the other components are statistically independent from each other. ICA is used to detect and distinguish artifacts and disrupting factors in EEG signals, and to solve difficult and complex problems such as source localization processes. Independent component analysis is shown in figure 1.16.

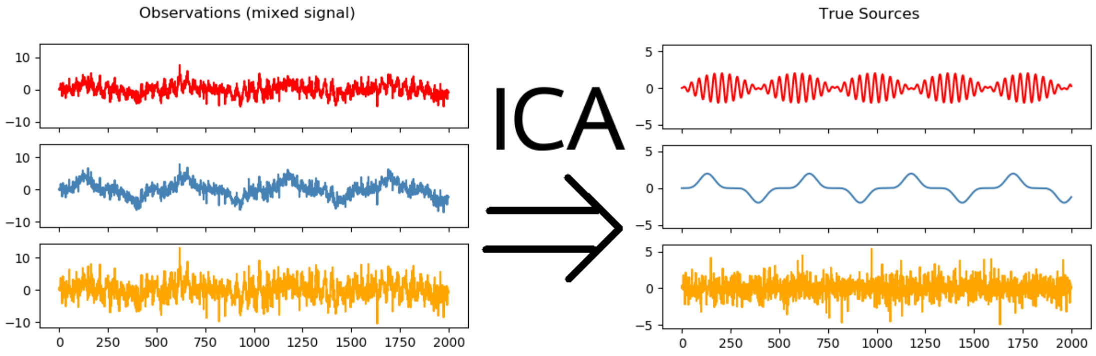


Figure 1.16. Independent component analysis [34]

1.4.3. Common Spatial Pattern (CSP)

The common spatial pattern is a feature extraction method for projecting multi-channel EEG signals into a subspace. With this method, differences between classes are clarified and similarities between classes are minimized. With CSP, spatial filters are designed to make the input data more distinguishable and these filters transform the input data into outputs with optimal variance. CSP models generally perform better with synchronous BCIs that collect signals at predefined time intervals. Due to the non-stationary nature of EEG signals, asynchronous BCIs do not perform as well as synchronous BCIs. The skull placement and position of the electrodes required to collect EEG signals can provide more specific information than other shapes and positions, so the success of CSP models also depends on spatial resolution. For these reasons, various CSP models have been developed so that CSP models can perform better. The common spatial pattern is shown in figure 1.17.

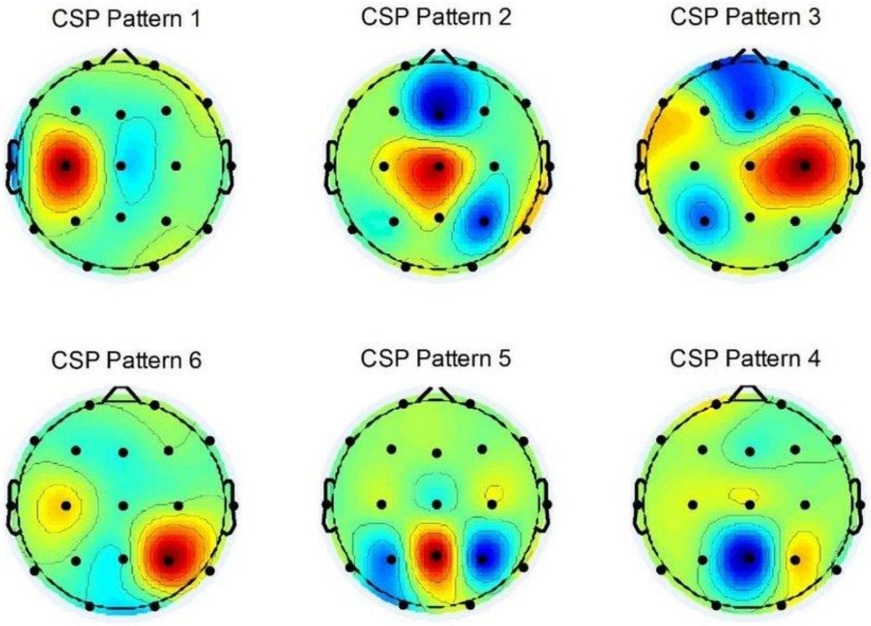


Figure 1.17. Common spatial pattern [35]

1.4.4. AutoRegressive Components (AR)

The autoregressive components method is a spectral estimation method used to model signals. Autoregressive models transform one input into a different output. As input, white noise enters the model and its mean is zero and its variance is q2. In this way, new EEG signals are produced with linear time-invariant filter properties. It is estimated that different thinking activities produce different filter coefficients when generating EEG signals. The purpose of the AR method is to obtain these different filter coefficient values. These coefficients are used as signal properties. AutoRegressive components are shown in figure 1.18.

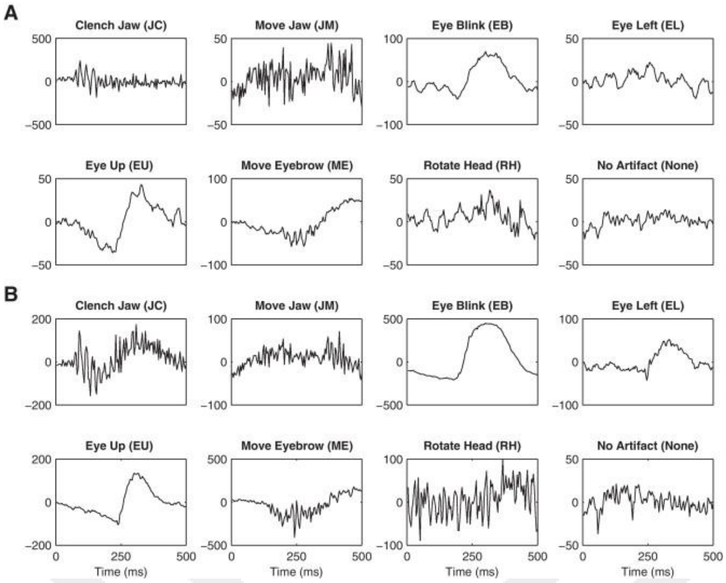


Figure 1.18. AutoRegressive Components [36]

1.4.5. Matched Filtering (MF)

Matched filtering is a feature extraction method that uses predefined and known EEG signals and features. With this method, it is aimed to detect signals that match the predefined EEG signal characteristics. EEG signals produced by individuals are compared with predefined signal sets. Signals with a high correlation are considered more distinctive than other signals and the features of these signals are extracted. Matched filtering is shown in figure 1.19.

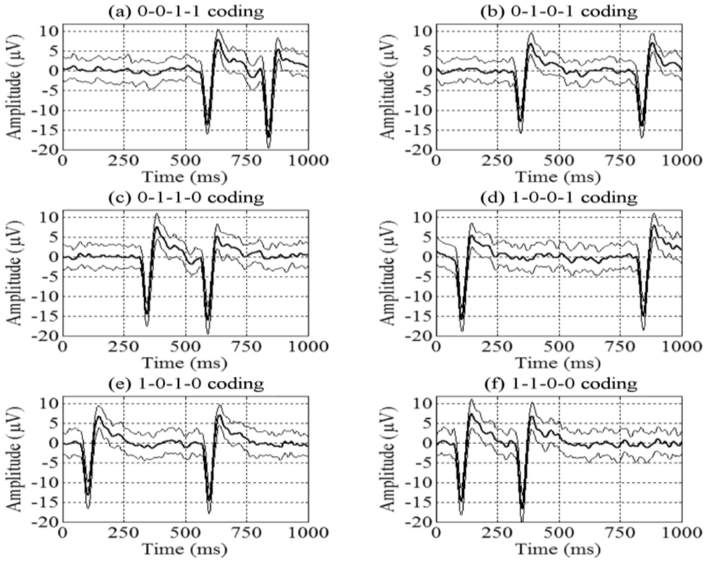


Figure 1.19. Matched Filtering [37]

1.4.6. Wavelet Transform (WT)

Wavelet transform is a feature extraction method used in many different fields and different types of data, from audio to video, from biological signals to images. Since this method can identify and correlate time-frequency properties of signals, it is often used to extract features from non-stationary signals. This method works in the time and frequency domains and produces wavelets as a function of various frequencies. The wavelet transform is shown in figure 1.20.

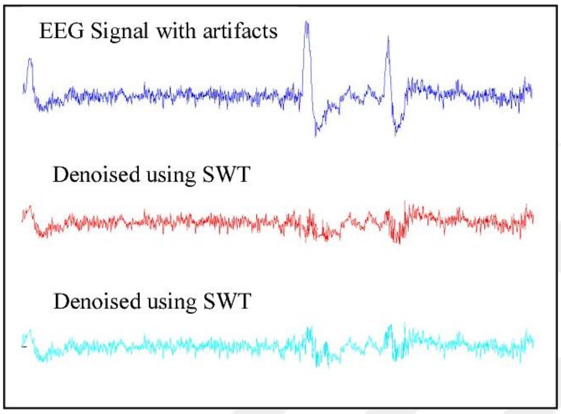


Figure 1.20. Wavelet Transform [38]

1.4.7. Sequential Selection

Sequential selection is a method used to detect optimal EEG signal feature sets. With this method, feature groups are created by adding or subtracting EEG features sequentially. These selections are made by sequential forward selection (SFS) and sequential backward selection (SBS) methods. The sequential selection method is shown in figure 1.21.

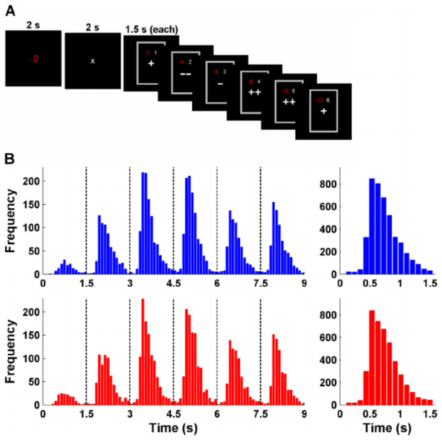


Figure 1.21. Sequential Selection [39]

1.4.8. Genetic Algorithms (GA)

Genetic algorithm is a feature selection and optimization method used to determine which features in EEG signals are more efficient and usable. This method can automatically scan to find correlated EEG features.

1.4.9. Hjorth Parameters

Hjorth parameters are quantitative terms used to describe some general characteristics of EEG signals. These parameters are time dependent but can also be obtained using power spectrum features. This method provides a link between frequency domain definitions and time domain estimation. Although the Hjorth parameters depend on the variance values, complex curves can be obtained by adding these parameters and values together. Hjorth parameters are activity, complexity and mobility parameters used in feature extraction for EEG signals.

1.4.10. Event Related Potentials (ERPs)

Event-related potentials are the brain's response to a particular stimulus or stimuli, and this response manifests itself as very small voltages produced in brain structures [40]. A non-invasive method, the EEG method, is used to observe these events and measure the relevant potential values. Thanks to this method, the response of the brain to different stimuli can be measured non-invasively. During this process, the person connected to the EEG device is given a visual warning on the screen or an auditory warning with an audio device. The person's responses to these stimuli are recorded and used as a distinctive feature in the design of BCI systems. The event-related potentials method is shown in figure 1.22.

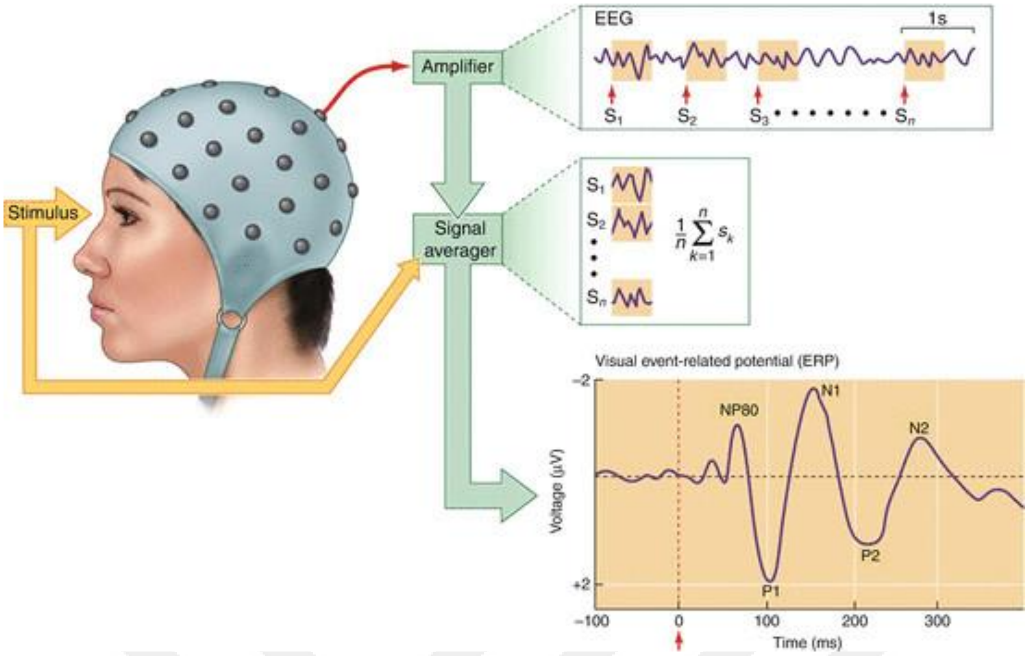


Figure 1.22. Event related potentials [41]

1.4.11. Fast Fourier Transform (FFT)

Fast Fourier Transform is used to convert EEG signals from time domain to frequency domain [42]. In fact, the FFT method is an optimized version of the Discrete Fourier Transform (DFT). FFT is a faster method than DFT as it removes duplicate terms within the method while performing the conversion. With this method, a signal is sampled over a period of time and this signal is broken down into its components. In the time domain, all the features of the signal can be observed together, while in the frequency domain, the signal features can be observed separated into independent frequency components. Thanks to this method, more detailed and distinctive observations can be made in the frequency domain of signals that cannot be clearly observed in the time domain. The fast Fourier transform method is shown in figure 1.23.

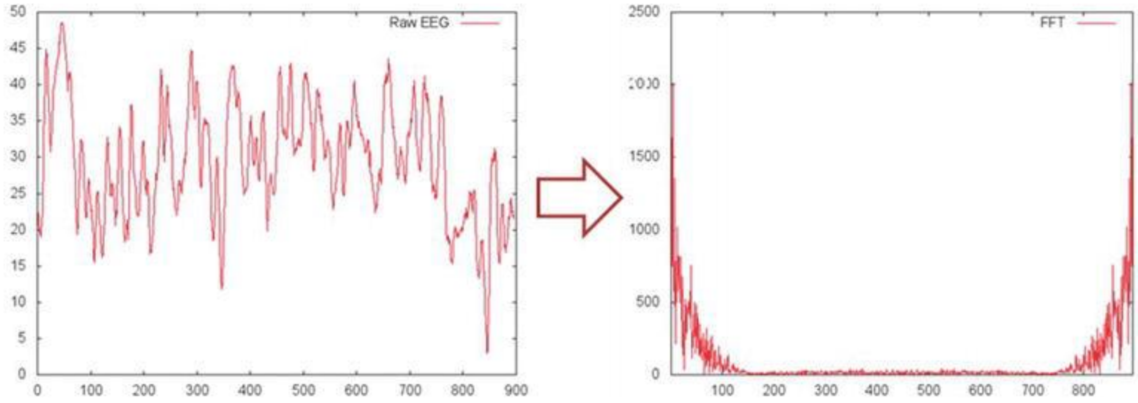


Figure 1.23. Fast Fourier transform [43]

1.5. Classification Methods in BCI

1.5.1. Linear Discriminant Analysis (LDA)

Linear discriminant analysis is a supervised learning method used for classification. This method is used to find linear combinations of features to best distinguish classes in a dataset [44]. LDA applies size reduction and moves data to a lower-dimensional space to better distinguish between classes in the dataset. It uses the distinguishing feature spaces between the classes in the data while doing this reduction. This method assumes that the data has a Gaussian distribution, can be linearly separated, and that the covariance matrices of the different classes are equal [45]. LDA is a simple, lightweight and computationally efficient method that can detect multicollinearity in data. The disadvantages of this method can be explained as it assumes that each data has a Gaussian distribution, assumes that the covariance matrices of different classes are equal, assumes that the data are linearly separable, and does not perform well in high-dimensional feature spaces. Linear discriminant analysis is shown in figure 1.24.

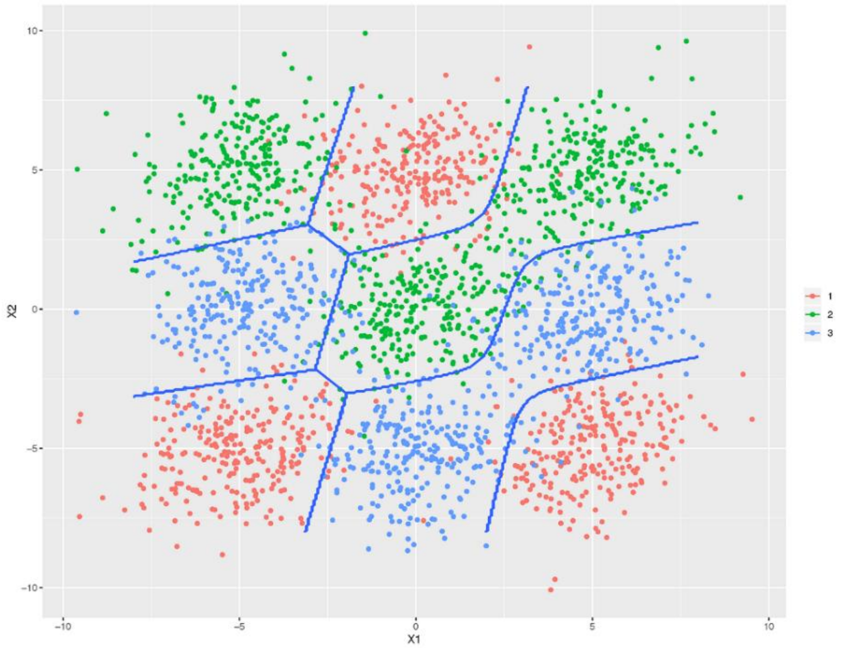


Figure 1.24. Linear Discriminant Analysis [46]

1.5.2. Quadratic Discriminant Analysis (QDA)

Quadratic discriminant analysis is a supervised learning method very similar to linear discriminant analysis. The difference of this method is that the mean and covariance values of the classes in the data set are not considered equal [47]. In the QDA method, the covariance matrix is calculated separately for each class and classification is made using these values. The quadratic discriminant analysis is shown in figure 1.25.

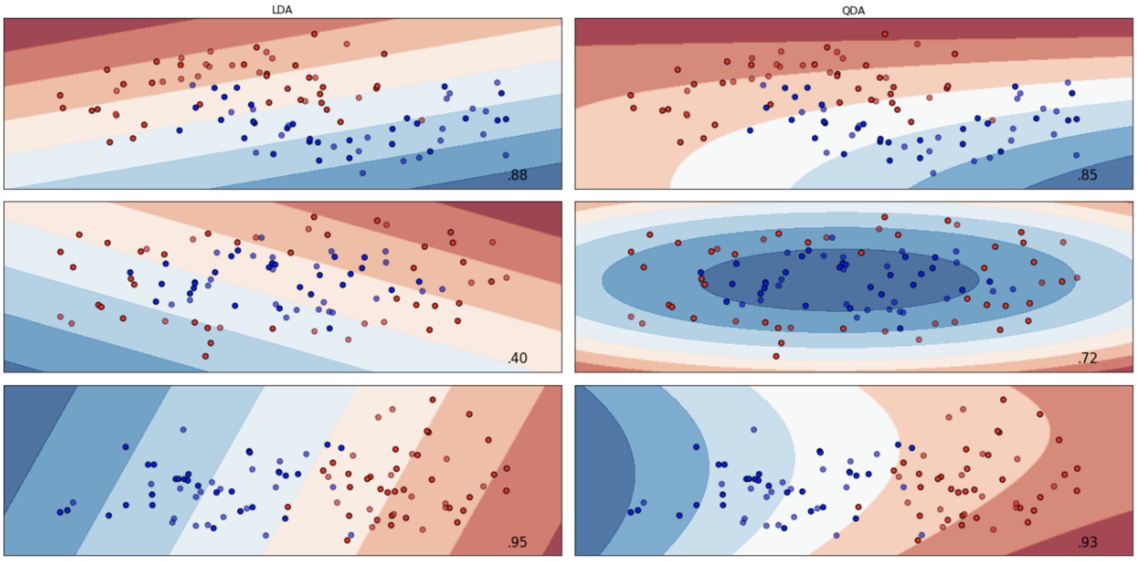


Figure 1.25. Quadratic Discriminant Analysis [48]

1.5.3. Support Vector Machines (SVM)

Support vector machines are a supervised learning method used for regression and classification. This method is a frequently preferred and used method because it requires low computational power and provides relatively high classification successes [49]. Although it is used in regression problems, it is also frequently used in classification problems. In the SVM method, classification is done by dividing the data set into a hyperplane over which it spans. The labeled data used for classification is trained with the SVM model, and a hyperplane is created to best classify the new data that will enter the model. In this hyperplane, the classes in the dataset are divided into different regions. For a two-class dataset, the hyperplane is divided by a line into two planes. For higher dimensional data, the optimal hyperplane is tried to be obtained [50]. The task of the SVM model is to determine the boundaries that best separate the classes from each other. The boundary separating the classes and the data points closest to this boundary from each class are called support vectors. The support vector machine can be used for linear classification operations as well as non-linear classification operations. A method called kernel trick is used for this process. The kernel trick aims to make the data linearly separable by moving the input data to a higher dimensional space. Some of these kernels are kernels such as the polynomial kernel and the radial basis function (RBF) [51]. The SVM method performs well if separation between classes is evident and performs better in high-dimensional spaces. It also performs well on small datasets and exhibits robustness to noise. SVM also has advantages such as good generalization performance, can be used for regression and classification operations, less prone to overfitting, editable and modifiable. In addition to these advantages, there are also disadvantages such as not being very suitable for large datasets, poor performance in close class attributes, being computationally expensive, incorrect kernel selection can greatly affect performance, memory intensive and not very suitable for multi-class. The support vector machines method is shown in figure 1.26.

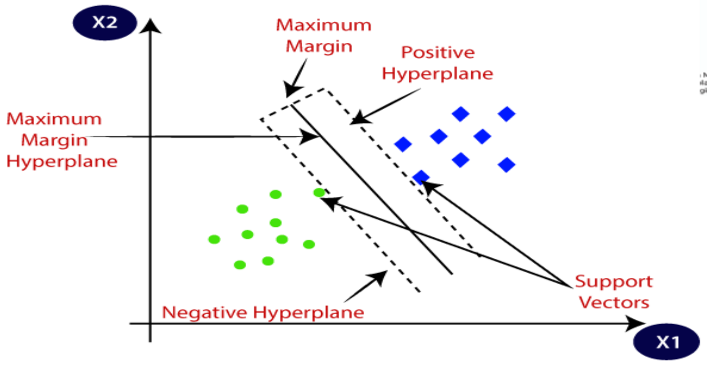


Figure 1.26. Support Vector Machines [52]

1.5.4. K-Nearest Neighbors (k-NN)

K-nearest neighbors is a non-parametric supervised learning method used for regression and classification. In this method, the closeness of the new data to be classified to the previously trained data is calculated [53]. A k number is used while making this calculation and users can determine this k number according to their own problems. The closeness of the new data to be classified to k points is calculated with the k-NN classifier, one of the previously classified data classes. According to this neighborhood proximity, it is decided in which class the new data will be included. Euclidean distance is often used when calculating proximity. In K-NN regression operations, the classifier calculates the mean values of k closest points as output [54]. The closer the K value is to one, the less stable the model tends to be. Conversely, as the K value increases, the model starts to work more stable up to a certain value, but after a value the error increases and the model starts to slow down. This method is simple and easy to implement. The application can be made quickly by changing the parameters of the existing models without installing a new model. It can be used to perform tasks such as classification, regression, and search. However, this method slows down considerably as the number of data and independent variables increases. The k-nearest neighbors method is shown in figure 1.27.

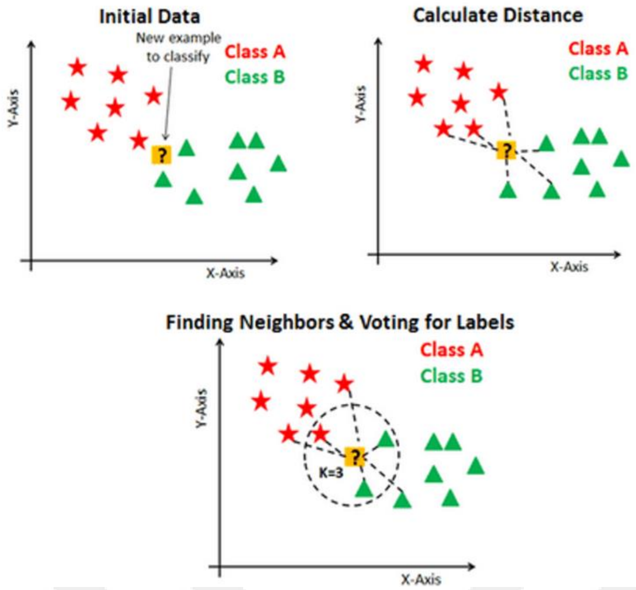


Figure 1.27. K-Nearest Neighbors [55]

1.5.5. Riemannian Geometry-Based Classifiers (RGC)

Instead of detecting and classifying temporal or spatial features, the Riemann geometry classifier method maps data to a geometric space with appropriate metrics [56]. Thanks to such a space, data can be processed in different ways according to the desired purpose. These are operations such as smoothing, interpolation, extrapolation, averaging, and grading. Spreading the EEG data over such an area can enable the computation of covariance matrices of the data. Riemannian geometry studies are done with smooth curved spaces, and these spaces can be approached locally and linearly. Curved spaces are called manifolds. The linear approximations at each point of these manifolds are a tangent space. The tangential space is equipped with inner products that change smoothly from each point to the next [57]. This causes a non-Euclidean distance between two points. This is why external distance is used instead of Euclidean distance. The external distance is an internal distance adapted to the manifold geometry to which the data is mapped. The most common matrix manifolds for BCI applications are Hermitian or symmetric positive definite matrices, Stiefel and Grassmann manifolds [58]. The classifiers method based on Riemann geometry is shown in figure 1.28.

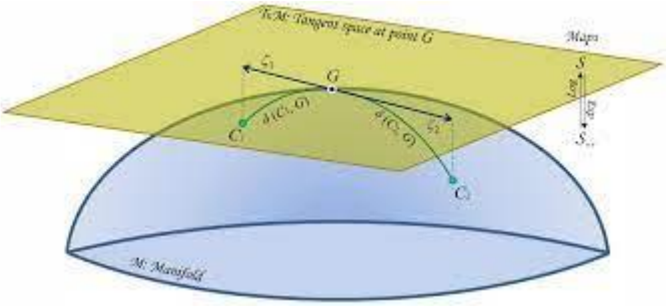


Figure 1.28. Riemannian Geometry-Based Classifiers [57]

1.5.6. Artificial Neural Networks (ANN)

Artificial neural networks are computational systems inspired by the biological neural networks of the human brain [59]. These networks are also called neural networks (NN) or neural nets. ANN models consist of interconnected groups of artificial neurons that mimic neurons in the biological brain. The connections between these interconnected artificial neurons transmit signals like synapses in the brain. Artificial neurons receive the signals transmitted to them, process these signals, and then transmit these processed signals to the neurons connected to them. The signals entering the neurons are a real number, and the signals leaving the neurons are the values calculated by some nonlinear function of the sum of the signals entering it. The structures that connect neurons are called edges. Each neuron and edge have a weight value that is updated as training continues in the neural network [60]. The weight value can increase or decrease, and these increasing and decreasing trends depend on the strength of the signal entering the neuron. Threshold values can be determined to adjust whether a neuron sends the data it receives to other neurons after processing it. In neural network models, neurons are grouped to form groups of neurons called layers. Different layers can be used for different purposes by making different calculations. The first layer of artificial neural networks is called the input layer, the last layer is the output layer, and the layers between these two layers are called hidden layers. Signals enter the model from the input layer, pass the hidden layers, and exit the output layer. During this journey, the signals follow many connections according to the structure of the model and zigzag until the output.

To train neural networks, this data needs a known input and a known label. This is a supervised learning method. During training, the network structure establishes probability-weighted connections between these inputs and outputs and performs classification or regression operations for new data according to these connections. The neural network determines an error value by comparing the output value obtained during training with the actual label of the data. The network then updates the weighted attribution values using this error value. The network continues training until it minimizes the difference between the output value and the actual data label. When the desired classification success is achieved by training with sufficient data, the training is terminated, and the model is ready for use. An artificial neural network is shown in figure 1.29.

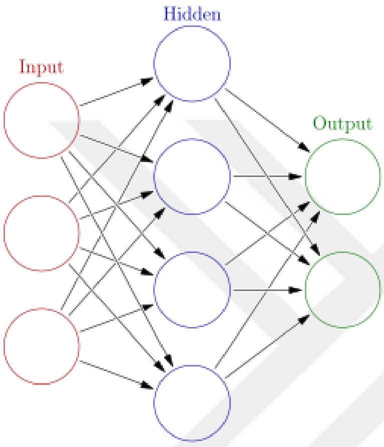


Figure 1.29. Artificial neural network

1.5.7. Deep Learning

Deep learning is a learning method of artificial neural network-based models with some new techniques added and expressing a wider model section. The reason for using the word deep is that there are many layers in the network between the input and output layers [61]. Deep learning methods can be supervised, semi-supervised and unsupervised. Deep learning architectures are methods such as deep neural networks (DNN), deep belief networks (DBN), deep reinforcement learning, recurrent neural networks (RNN), convolutional neural networks (CNN), and transformers [62]. These methods, which have very wide and very different application areas, are used in fields such as computer vision, machine translation, biomedical signal processing, speech recognition, natural language processing, virtual reality, bioinformatics, medical image analysis, weather and climate forecasting, drug design and space research. Deep learning models, with their success, have achieved better results than human knowledge and experience in many areas [63].

Deep learning models have been developed to increase the performance of traditional machine learning methods and to obtain faster and more accurate models. In traditional machine learning methods, feature extraction and feature selection are done manually, and it may be necessary to create new features in addition to these. In deep learning methods, these processes are automated and done by deep learning models. In addition, these models perform much better in training large, complex and nonlinear data sets. The disadvantage of these models is that they require high processing power and large datasets. The deep learning method is shown in figure 1.30.

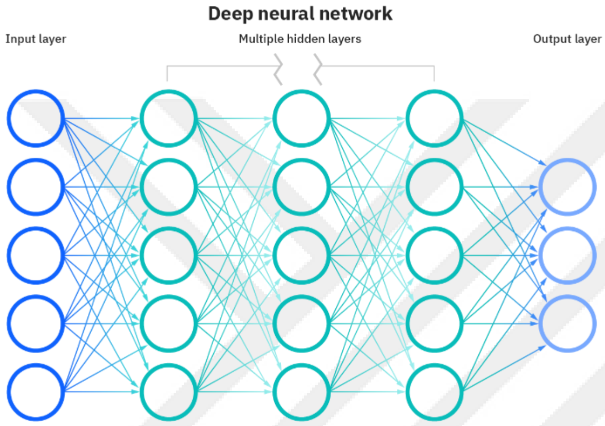


Figure 1.30. Deep Learning [64]

1.5.8. Transfer Learning

Transfer learning method refers to methods developed to use pre-trained machine learning models to solve a new problem [65]. Thanks to this method, it is possible to apply previously produced solutions and models in BCI and similar fields to the new problem to be solved. In this way, it is possible to prevent the repetition of labor, work and costs and to use the information obtained before in solving the current problem. This method may not require the collection of new data, which is very laborious during training of limited EEG data and may improve model performances. The transferred content is also important in transfer learning methods. These transfers can be in the form of features, parameters, or instances, and these methods are called feature-based transfer learning, parameter-based transfer learning, and instance-based transfer learning [66]. The transfer learning method is shown in figure 1.31.

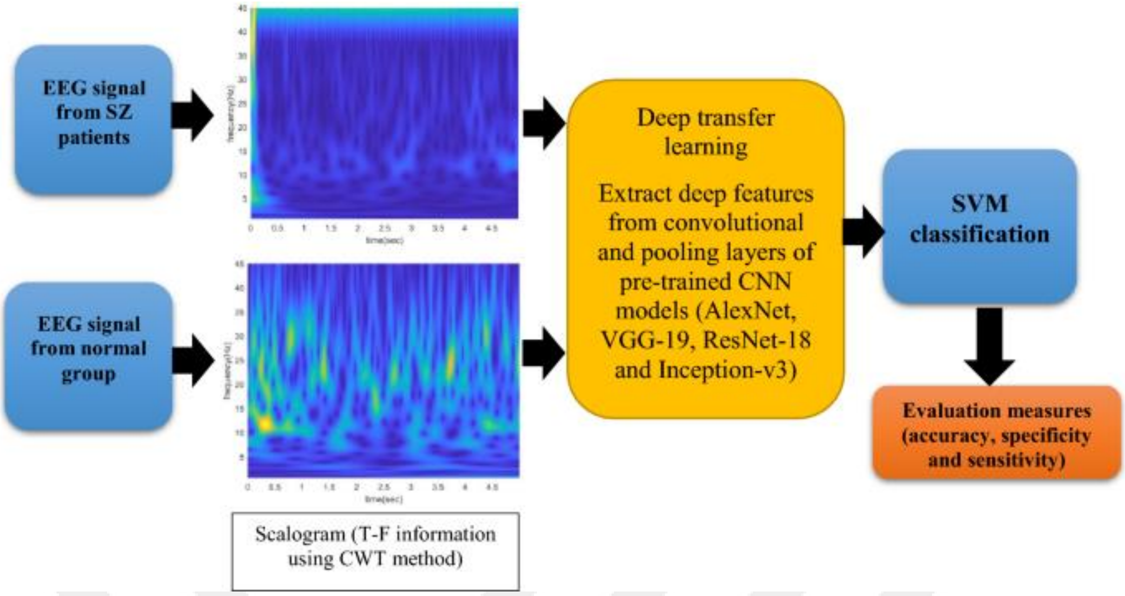

EEG signal from SZ patients
EEG signal from normal group
Scalogram (T-F information using CWT method)
Deep transfer learning
Extract deep features from convolutional and pooling layers of pre-trained CNN models (AlexNet, VGG-19, ResNet-18 and Inception-v3)
SVM classification
Evaluation measures (accuracy, specificity and sensitivity)

Figure 1.31. Transfer Learning [67]

1.6. BCI Paradigms (Control Signals)

1.6.1. Visual Evoked Potentials (VEPs)

Visual evoked potentials are modulations of brain activity that occur in the visual cortex in response to a visual stimulus [68]. These modulations are relatively easy to detect, as the amplitude of the visually evoked potential signals increases drastically as the stimulus center approaches the visual field [69]. Visual evoked potentials can be produced by methods such as flash stimulation or by graphic patterns such as gate, random point map or control panel cage, and it is the morphology of the optical stimulation that causes these signals. Visual evoked potentials can be generated according to the frequencies of visual stimuli. These are responses such as transient VEPs (TVEPs) and steady-state VEPs (SSVEPs). In order for TVEP signals to be produced, the frequency of visual stimuli must be below 6 Hz. Higher frequency stimuli are required for SSVEP responses. BCIs using SSVEP responses can be divided into three categories: code-modulated VEP (c-VEP), frequency-modulated VEP (f-VEP), and time-modulated VEP (t-VEP) BCIs [70]. In addition, VEPs can be produced by field stimulation. These are half-field VEPs, full-field VEPs, and partial-field VEPs, which are defined by the region where the stimulus is located on the screen. The visual evoked potentials method is shown in figure 1.32.

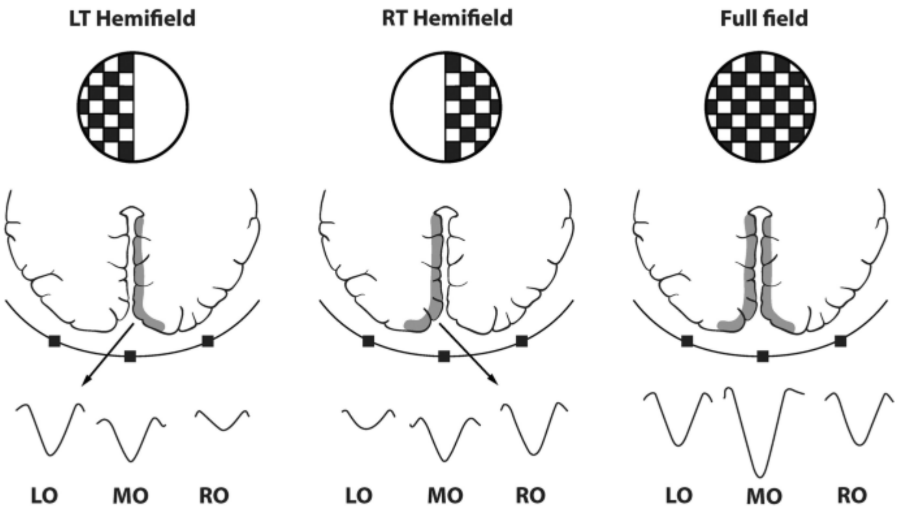


Figure 1.32. Visual Evoked Potentials [71]

1.6.2. Slow Cortical Potentials (SCPs)

Slow cortical potentials are voltage shifts that occur in EEG signals that can last from one second to several seconds. These voltage shifts are found in EEG signals less than 1 Hz [72]. Slow cortical potentials arise as changes in cortical activities occur. There are two types of slow cortical potentials, positive SCPs and negative SCPs. Positive SCPs occur with decreased activity in individual cells, while negative SCPs occur with increased neuronal activity. These signals can be generated by both healthy and paralyzed individuals for use as control signals in BBAs. SCP signals can be used as outputs to move a prosthesis or a digital cursor [73]. Although such signals are used to make basic movements, the information contained in these signals is usually low. Additionally, users must complete months of training and hands-on practice to control SCP-based BCI systems.

1.6.3. P300 Evoked Potentials

P300 evoked potentials are individuals' response to auditory, visual, or somatosensory stimuli. These responses are observed as positive peaks within the EEG signals. P300 evoked potentials often appear approximately 300 ms after external stimuli [74, 75]. Some studies have found that the less likely the stimulus is to occur, the greater the amplitude of the response peak. Users do not need training and experience with BCI systems that use P300 evoked potentials compared to other BCI systems. However, when users use these systems for a long time, it has been observed that the P300 signal amplitudes decrease as they get used to the stimuli [76].

1.6.4. Sensorimotor Rhythms (Mu and Beta Rhythms)

Sensorimotor rhythms consist of rhythms that occur in the mu band between 7-13 Hz and in the beta band between 13-30 Hz. Mu band is also called Rolandic band and beta rhythms are formed as a result of brain activity. Signals in the mu and beta bands are correlated. Some beta rhythms are a kind of harmonic mu rhythms, but some beta rhythms can be independent outside of this relationship. Motor movements can be performed to produce sensory-motor rhythms and cerebral activity can be measured with these movements. The amplitude of these rhythms can vary depending on the type of motor movement performed. However, there is no need to physically perform motor movements to generate sensorimotor rhythms and modulate their amplitude [77,78]. These signals can also be produced by thinking and mentally rehearsing an action without actual motor action to generate the signals [77]. Sensory motor rhythms are widely used in BCI systems because they do not require any physical activity and people can learn to generate these signals [79,80]. Sensorimotor rhythms can be divided into event-related desynchronization (ERD) and event-related synchronization (ERS). ERS signals can be obtained through motor behavior, motor imagery, and sensory stimulation paradigms [79]. ERD refers to the reduction and suppression of sensorimotor rhythms, while ERS refers to the increased amplitude of these rhythms.

One of the interesting areas of sensory-motor rhythms is the motor imagery signals that can be produced by mental activity without any motor action. These signals enable internal BCI designs, and internal BCI systems are more beneficial than external BCI systems. Therefore, sensorimotor rhythms and motor imagery signals are extensively studied in BCI research. BCI systems that use sensorimotor rhythms as control signals can operate synchronously or asynchronously. It has been determined that these movements can be detected without physical movement by using sensorimotor signals. While perceiving these movements, the person does not need to make any physical movement and thinking that he or she is making a movement allows this movement to be perceived [81].

1.7. BCI Applications

Brain computer interfaces allow an external device to be controlled only by brain signals, without the use of peripheral nerves or muscles. In these systems, brain signals are identified, classified and converted into control signals. BCI systems have attracted the attention of researchers due to these capabilities, and by developing these capabilities, it is aimed to develop applications that will be beneficial especially for people with severe motor disabilities. With the increase in research in this area, significant progress has been made especially in EEG-based BCI applications and various applications have been developed for both healthy and paralyzed people. The reasons for using EEG signals in most BCI systems are that EEG is noninvasive, inexpensive, easy to use, portable, and has high temporal resolution. Patients who have completely lost their motor control, who have severe cerebral palsy or who are in the last stage of ALS are the primary target groups of BCI applications. Secondly, there are patients who are almost completely paralyzed but can perform voluntary movements such as blinking, eye movement, lip movement. In the third group, there are patients who are disabled but can speak and can make movements such as hand, arm and eye movements. BCI systems are of great importance to the first two groups, as these two groups include patients with virtually no motor control and whose brain signals may be vital to them. It is possible for the third group to have systems that produce control signals by using their muscles and hand, eye and arm movements more quickly and efficiently than BCI systems using brain signals. In addition, BCI is actively used in the detection of epilepsy, schizophrenia, depression and similar diseases. Today, BCI systems have begun to be used for patients as well as healthy people. It is used in the analysis of situations and products that people like or dislike in the field of neuromarketing, as a joystick in the game industry, in monitoring the emotional states of the players, and in improving the concentration and focusing skills of the students in the field of education.

1.7.1. Healthcare

Brain-computer interfaces allow people with neurological conditions or limb loss to communicate with their environment in a variety of ways. It is a great need for people to communicate and interact with their environment. Researchers are making great efforts and developing solutions to meet these needs.

Many BCI applications designed for communication use an on-screen virtual keyboard. With this virtual keyboard, a person can select letters or symbols on this keyboard using brain signals and eventually create words and sentences. Communication solutions are usually made with this method and the feature that distinguishes the systems from each other is the type of control signal they use. The P300 method is shown in figure 1.33.

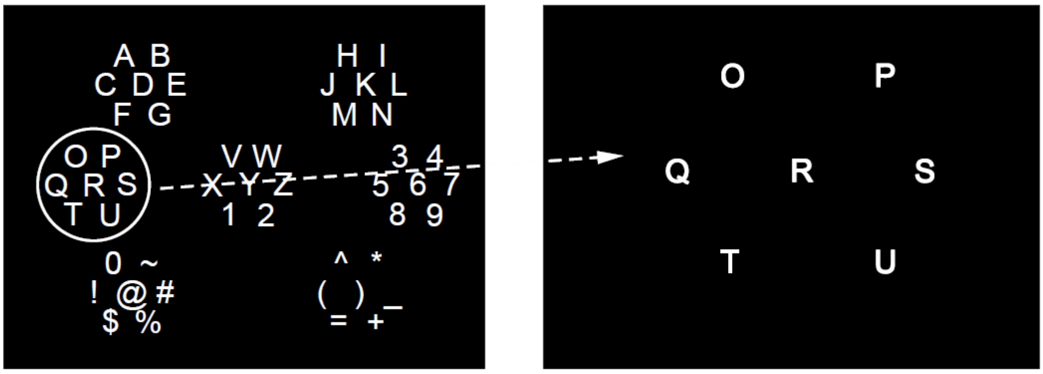


Figure 1.33. P300 [82, adapted from [83]]

Visual event-related potentials (VEP) signals have previously been used in P300 speller applications, but these signals are very difficult to generate for patients who have lost sight or eye movement skills. To solve these problems, researchers turned to auditory stimuli instead of visual stimuli and started to use auditory stimuli instead of visual stimuli [84-87].

Researchers have investigated how smoking and alcohol affect attention and therefore brain signals [88–92]. Based on this information, the researchers stated that the brain signals of people with distraction, focusing problems or motion sickness can be followed, and also EEG signals can be followed to prevent possible distractions and focusing problems that may cause a dangerous situation [93, 94]. Consciousness level was not only measured for drivers, but also studies were conducted to measure and monitor the level of consciousness of healthy individuals [95].

Some neurological diseases, injuries or losses can cause damage or complete loss of motor functions and senses of individuals. While these losses decrease the quality of life of patients, they cause patients to need home care services and auxiliary tools and equipment throughout their lives. Motor restoration applications, which can be obtained through BCI systems, try to fulfill the motor function abilities of individuals, can enable patients to cope with the physiological and psychological problems of the situation they are in, and enable them to integrate with their social environment.

For these restoration operations, research has been done on mobile robots that perform various tasks to carry out the daily activities of locked people [96,97]. In addition to these robots, prosthetic limbs that will perform the functions of various limbs have been designed and produced and integrated with BCI systems to restore the motor skills of the patients [98-100]. These systems are also being investigated for the recovery of motor skills after stroke and for faster control over the nervous systems of individuals during physiotherapy processes [101,102].

Brain-computer interfaces are also used for early diagnosis of some diseases. These are diseases such as epilepsy, dyslexia, tumors, encephalitis and narcolepsy. Researchers have tried to detect these diseases by using brain signals in their studies and have obtained promising results [103-108]. The early diagnosis of these diseases or the prediction of their attacks both enable early diagnosis and treatment for individuals and provide treatment of some diseases before they reach complex dimensions. Especially early diagnosis and treatment of disorders such as dyslexia is very important for the development of the skills of growing children.

1.7.2. Environmental Control

One of the areas of use of brain-computer interfaces is systems where people interact with their environment. These can be smart devices, smart vehicles, smart homes, systems that can be connected with the internet of things, smart working environments, machines and equipment that can perform daily tasks or a specific task. These systems can help adjust the working conditions by monitoring the cognitive status of the operator using any device, protect the cognitive status of people working under intense stress, and prevent problems that may arise by monitoring the stress and cognitive level of a doctor who performs a risky surgery [109-115].

In [116,117], researchers designed a system in which people interact with their environment using brain-computer interfaces, and with this system, they adapted the components to be controlled in the user's environment according to the user's situation by monitoring the cognitive status of the users.

In addition, distraction or fatigue of people using transportation vehicles can be detected in advance and the relevant follow-up units can be warned. Researchers have studied the measures that can be taken to detect these dangerous situations and prevent them from causing an accident [118-121].

A car company developed a system in which the driver's brain connects with the car and called this technology Brain-to-Vehicle [122]

When developing this system, the company started from the knowledge that the brain produces signals before people perform their motor activities and that there is a 0.4-1 second difference between signal production and motor activity. This system aims to receive the sudden brake signals of the drivers and to activate the system at the same time, to gain a braking distance of approximately 27 meters at 100 km/h. In addition, with this system, distraction and fatigue levels of the person can be detected. The brain-to-vehicle system is shown in figure 1.34.

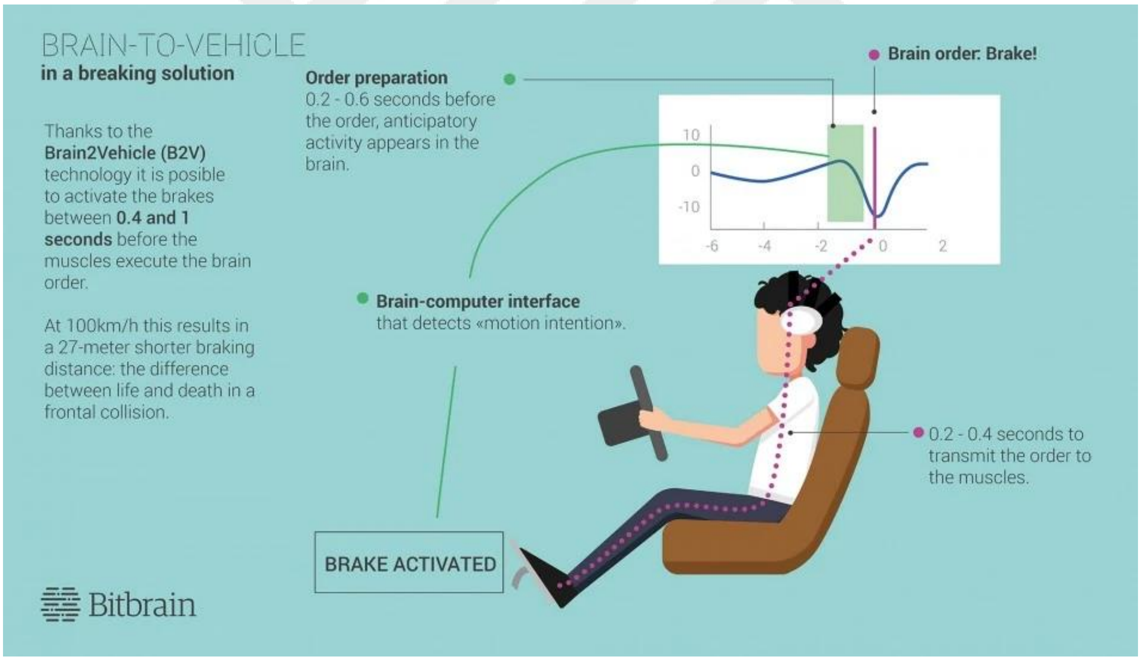

BRAIN-TO-VEHICLE
in a breaking solution
Thanks to the Brain2Vehicle (B2V) technology it is possible to activate the brakes between 0.4 and 1 seconds before the muscles execute the brain order.
At 100km/h this results in a 27-meter shorter braking distance: the difference between life and death in a frontal collision.
Order preparation
0.2 - 0.6 seconds before the order, anticipatory activity appears in the brain.
Brain order: Brake!
10
0
-10
-6
-4
-2
0
2
Brain-computer interface
that detects «motion intention».
0.2 - 0.4 seconds to transmit the order to the muscles.
BRAKE ACTIVATED
Bitbrain

Figure 1.34. Brain to vehicle technology [122]

1.7.3. Education

Most brain-computer interfaces used in education use neurofeedback or information derived from it. Brain signals produced by individuals before, during and after work are very important. Thanks to this information, people's concentration levels, distraction rates, stress levels during learning and post-learning brain activities can be followed. Another feature of these systems is that they can be customized according to the person. Considering the different learning skills of each individual, it would be beneficial to use these systems as support mechanisms for individuals.

At the same time, a solution is being developed through brain-computer interfaces to rehabilitate the learning and attention of children with neurodevelopmental developmental disorders [123]. There are also studies that try to make the learning process faster and more efficient by combining technologies such as BCI technologies and virtual reality (VR) technologies [124].

Thanks to these tracking systems, people have the opportunity to improve their skills such as focusing, learning and preventing distraction. Researchers have determined the mental states of individuals in their studies, offered suggestions to improve them, and suggested customizable systems [125-129].

1.7.4. Marketing

One of the areas where brain-computer interfaces are intensively researched is marketing and advertising. These systems can measure people's reactions to a product, environment or advertisement through brain signals and classify and use these signals according to people's interests, likes and demands. However, they can also be used to gather feedback from the propaganda methods used to influence the determination of political preferences. Researchers have investigated the effects of advertising and marketing techniques, propaganda and similar content on individuals [130-132].

1.7.5. Entertainment

Brain-computer interfaces are being extensively researched and a wide variety of applications are being developed not only for people with physical disabilities or limited mobility, but also for healthy people. BCI's research interests include controlling virtual vehicles in two-dimensional and three-dimensional spaces, using brain signals instead of keyboards or joysticks in a computer game, gamification therapy using brain-computer interfaces for people undergoing stress and rehabilitation therapy and also games designed to measure levels of stress and meditation for users. [133-135].

Users can use pre-classified brain signals to move a digital object on the computer. For example, in a BCI model where right and left motor imagery signals are correctly classified, users can use these signals to move an object. Given that right hand motor imagery signals are defined to move an object to the right, the user can generate right hand motor imagery signals and move that object to the right.

1.7.6. Security and Authentication

Security and authentication-based systems often rely on brain signals produced by different people or produced by people in different environmental conditions. These systems can use brain signals produced by people with different characteristics to identify people. In addition, by detecting different signals produced by the change of environmental conditions, it can be determined whether people are under any security threat. Researchers are working to identify different people and to measure and describe people's responses under different environmental conditions [136–139].

1.8. Aim and Significance

The main purpose of this thesis is to develop a brain computer interface using EEG motor imagery signals. In order to develop this brain-computer interface, research and development was done on three fundamental components and a system was formed by combining these components.

The first of these components is the fast, accurate and robust classification of EEG motor imagery signals. To fulfill this task, Convolutional Neural Networks were preferred and two different CNN architectures were developed. These developed models use raw EEG data without any signal processing. In the second chapter of this thesis, we explained in detail the effects of signal processing on classification processes and the advantages of classification with raw data under the title of literature review. The first of the models we developed is the NF-EEG model, in which input reshaping and data augmentation methods are used and optimum hyperparameters are selected using a large parameter pool. This model outperformed many state-of-the-art models in its field and emerged as a generalizable CNN model with low standard deviation values. The second model we developed is the IS-EEG model, in which we investigated the effect of different input shapes on CNN classification performance. For this model, the existing data set was converted into 2D and 3D structures and the classification success of these structures was examined. This model outperformed many state-of-the-art models in its field and the effects of different input shape structures on system performance were explained in the results section.

The second component required to develop a brain-computer interface is the development of interface software that can run the developed CNN models in real time and control a computer or external device with the classification results. The NF-BCI interface was developed to achieve this purpose. This interface can read EEG motor imagery signals in real time. It allows the user to observe the EEG signals by displaying the reading signals on the interface. The NF-BCI program can run NF-EEG and IS-EEG models in the background and classify the EEG signals it reads in real time. In addition to these, NF-BCI software can also perform various control scenarios. In order to realize these scenarios, it can both connect with the computer it is working with and control an external device by connecting to a microcontroller. These mentioned control scenarios are scenarios such as turning the sound of the computer on and off, controlling a robotic arm. NF-BCI, which can perform all these tasks at the same time, also stands out as a research platform candidate where researchers can work with their own data and classifiers.

The third component needed to complete the brain-computer interface is a robotic arm that will act as an external device. In order to realize this aim, the robotic arm, which is explained in detail in the fourth chapter, has been produced. This robotic arm is in constant communication with the NF-BCI and performs the desired movements with the incoming control signals. These movements are opening and closing the palm, moving certain fingers, turning the arm, etc. These movements are defined on the NF-BCI and according to the classification results, the robotic arm is provided to make these movements.

By combining all these components, a real-time end-to-end NF-BCI model has been developed. Both of the developed classifier models are original CNN models that have emerged as a result of intensive work. Since these models use raw data, it has eliminated the need for long signal processing to classify signals. Although the models eliminate this need, they outperformed many state-of-the-art models and achieved high classification success. With the IS-EEG model, the effects of different input shape structures, which have not been studied specifically in the literature, on classification performance were investigated. As a result of these examinations, it has been revealed that the choice of input type is very important in the relevant classification areas. Comparative results and discussions of these studies are given in chapters 5 and 6. The NF-BCI interface software is a new and original interface developed from the ground up. With this interface, both EEG signals can be displayed in real time, EEG signals can be classified, and an external device can be controlled by connecting to a microcontroller with classification results. In addition, NF-BCI aims to be a software that researchers can benefit from with its flexible structure. Together with the robotic arm, the real-time operation of this entire system has emerged as a physical movement and has been a beacon of hope for the development of this system and its use in real life.

1.9. Organization of Thesis

This thesis consists of 7 main chapters. These sections are;

1. Introduction
2. Literature Review
3. Dataset, Tools and Methods
4. NF-BCI: An End-To-End Brain Computer Interface and Its Components
5. Robotic Arm
6. Experimental Results
7. Discussion and Conclusions

In the introduction, it is aimed to make this thesis more understandable by giving general and comprehensive information about brain-computer interfaces. In addition, the aims and contributions of this thesis are stated and the achievements are explained.

In the literature review section, information such as the development of brain-computer interfaces, the methods used, the approaches developed, the results obtained, the advantages and disadvantages of different methods are given.

In the Dataset, Tools and Methods section, general information about the programming languages used in the development of the EEG motor imagery datasets, CNN models and

NF-BCI software used in this study and the programming languages and libraries used in the programming of the robotic arm are given, and about the stages and for what purpose they are used. explanations have been made.

There are 3 subtitles in the NF-BCI: An End-to-End Brain Computer Interface and Components section. The first part is the part where NF-EEG and IS-EEG architectures and working principles are explained. The second section describes the input shape designs and explains the methods and purposes of these designs. In the third section, the NF-BCI software is introduced and the task of each section on the interface is explained.

In the Robotic Arm section, information such as robotic arm design, mechanical and electronic components of the arm, and motion capabilities of robotic arm are given.

In the Experimental Results section, classification successes of NF-EEG and IS-EEG models are given with graphics and tables, and these results are compared with many current models in the literature. The results of the NF-EEG model with and without data augmentation were also compared among themselves. The results obtained as a result of using different input shape structures for the IS-EEG model are given and comparisons are made.

In the Discussion and Conclusion section, all the above-mentioned results were evaluated, the prominent input shape structures were examined and suggestions for future studies were presented.

2. LITERATURE REVIEW

Brain Computer Interface (BCI) is the name given to systems that allow communication and interaction between a device and the brain. People with neurological disorders, bedridden, paralyzed, and those who cannot perform their motor functions, can control a device through their brains using these systems. In addition, it can be used in the fields of utility tools and entertainment for who do not have any health problems, and it can be used for commercial purposes for different industries [1-5]. These BCI systems collect and classify the brain signals of people and turn them into commands. In this way, people who experience limb loss or cannot control their limbs, can independently control their artificial or existing limbs with these systems. In addition, these systems have recently started to be used for post-stroke rehabilitation processes by moving the limbs via brain signals, not by moving the muscles of the person [140-142].

BCI signals are generally collected from electroencephalography (EEG), magnetoencephalography (MEG), electrocorticograms (ECoG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), local field potentials and action potentials, and near-infrared spectral imaging (NIRS). One of the most used sources from these methods is EEG, because it is non-invasive, low cost, easy to apply and has high temporal resolution [143]. EEG signals are generally defined as a curse of dimensionality problem because of non-stationery signals and they have channel correlation, multi-channel recording paradigm, noise and artifacts and also fast, reliable and promising systems are systems that can accurately and quickly detect hidden variables in these physiological signals [144]. EEG-Based BCI systems are applied in many different areas and there are applications in neuromarketing [4], epileptic seizure detection [145], BCI-based wheelchairs [146, 147], upper or lower limb assistive robots [3, 5], neuro entertainment [148] and some similar systems. BCI applications have started to find a place for themselves in the game and entertainment industry as well. Tetris [149], World of Warcraft [150] and BrainArena [151] are some of them. EEG-based BCI systems have different applications such as P300 evoked potentials, visual evoked potentials (VEP), steady-state visual evoked potentials (SSVEP), motion onset visual evoked potentials (moVEP), motor imagery (MI) and so on [152, 153]. Motor imagery classification is a very popular BCI paradigm using the EEG signals. A person can generate EEG signals in brain by imagining making any movement (such as hand, arm, foot movements) without physically moving, and these EEG signals can be collected through electrodes and classified automatically with this method. There are 5 traditional steps in EEG motor imagery classification: Signal acquisition, signal preprocessing, feature extraction, feature selection and classification as shown in figure 2.1.

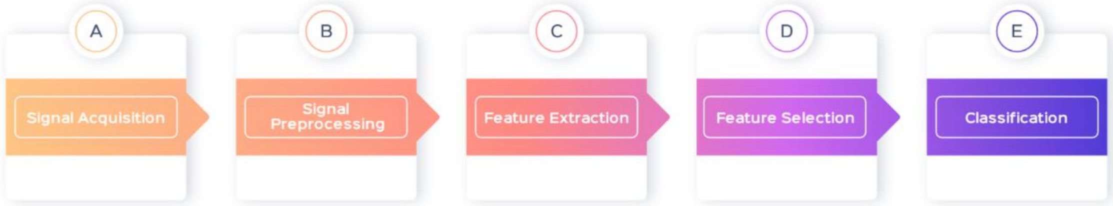


Figure 2.1. Traditional steps of EEG motor imagery classification

Signal preprocessing methods are used extensively before and during feature extraction processes and most of the methods used at this stage are designed manually and are based on human experience and knowledge [154-156] and it is highly likely to cause unwanted losses on the data with signal preprocessing and feature extraction. In the feature extraction of EEG signals, it is aimed to extract the information in the time, frequency and space domains, and with this information, neural networks are trained and classifications are made. In addition, for the classification of EEG signals, the signals were converted into 2D images with various preprocessing methods, and these images were trained and classified with deep learning algorithms. In these studies, mostly a specific time range or features in the specific frequency range are extracted to detect EEG features. In order to convert EEG signals to 2D images, the image of the signal at a fixed time or frequency was extracted, or the average of the images obtained at different times or frequencies was transformed. EEG signals may contain important information at different times or frequencies, however, these time and frequency regions may vary from person to person and these applied methods may cause to missing important information for classification, low classification success and inability to generalize the models. In many studies, wavelets [154,157] or short-time Fourier transform (STFT) [155,158] have been used for time-frequency feature extraction of EEG signals. In [158], researchers created 2D images from EEG signals using short-time Fourier transform and classified these images by training them using CNN and stacked autoencoders (SAE) where CNN was used to extract the features and SAE was used to do the classification. In [159], the authors used dynamic frequency feature selection algorithm and wavelet packet decomposition (WPD) for feature selection. In [160], authors converted EEG signals into 2D images using continuous wavelet transform (CWT) and used CNN to train and classify these images. In [161], the authors made a choice of time period with desynchronization / synchronization (ERD / ERS) and used WPD. In [162], deep learning algorithms used for the classification of signals such as Electroencephalogram (EEG), Electrocardiogram (EKG), Electrooculogram (EOG) and Electromyogram (EMG). Another important feature extraction from EEG signal processing methods is common spatial pattern (CSP) for space domain [1,161,163]. The performance of the CSP operation varies depending on the frequency band to be processed. To solve this problem, authors developed a filter bank common spatial model (FBCSP) [164]. FBCSP creates different frequency bands and applies the CSP process to these frequencies and uses an automatic feature selection algorithm among these bands. This process increases the classification accuracy in general. FBCSP uses a specific time range and ignores the most effective time ranges for various subjects and therefore cannot use time domain information efficiently enough. [160] and [165] indicate the best time choice for FBCSP, but the increase in classification accuracy is not a notable increase. In [166], various EEG paradigms are reviewed and possible DNN algorithms for each paradigm are discussed. In [144], it was stated that the neural network architecture used for the classification process, the input data structure and the frequency range used during the analysis are important in the classification of MI EEG signals with deep learning methods. Limited human knowledge and experience may limit the classification accuracy of these applied methods and systems [167,168]. However, to design a feature manually requires a lot of time-consuming experimentation and observations, and eventually these features need to be tested and their performance have to be measured for many different scenarios. In order to solve this problem and to obtain the needed features, the researchers were able to automatically perform feature extraction and classification steps using neural networks and they achieved remarkable results with these structures [169-174]. Recently, the use of deep learning techniques in the analysis of large data sets has produced promising results and high classification success [175]. Deep learning methods gave much more successful and stable results in extracting valuable features and in-depth information from big data compared to traditional neural networks. Deep learning techniques have also eliminated most of the disadvantages such as feature extraction, preprocessing, model changing according to the subject, and loss of information during training, which rely on human knowledge and experience and cause time and effort [176]. Deep learning methods are very effective for image, video, speech and pattern recognition and very good results have been obtained. Apart from these data types, it has started to be applied to time series data types and very good results have been obtained in this field. Deep learning models, which have also been used in the EEG brain signals classification, have achieved promising results. The most popular one among deep learning methods is the CNN method, due to its great achievements in various fields. In addition to CNN, researchers have also used different neural network structures. The authors developed a CNN by applying techniques such as the exponential linear unit activation function and batch normalization and aimed to achieve higher accuracy than FBCSP [177]. In [178], the authors proposed a multilayer perceptron (MLP) CNN, aiming to achieve a high classification accuracy by separately processing static energy and dynamic energy. In [179], they proposed a CNN with good accuracy rates applicable to various BCI paradigms. In [180], researchers used CNN to extract high-level features from motor imagery signals. In [152], authors applied the Choi-Williams distribution (CWD) transform, which reflects the energy distribution in both time and frequency domains, to obtain 2D images from EEG signals. In [172], they applied fast Fourier transform (FFT) and wavelet package decomposition (WPD) to EEG signals and compared them. Then they trained the information they obtained in the frequency domain using deep belief network (DBF) and restricted Boltzmann machines (RBM). In the classification of EEG motor imagery signals, CNN has achieved better classification success than other methods [169,171,174,181,182], but there are some problems to be solved for these methods. Without using any of this signal preprocessing methods and focusing on neural network modeling, we have proposed a neural network-based motor imagery classification steps as shown in figure 2.2.

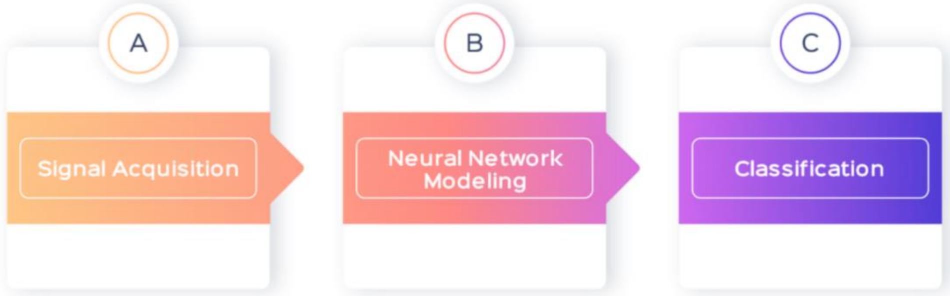


Figure 2.2. Proposed steps of EEG motor imagery classification

In [183], researchers developed a CNN model for training motor imagery signals by transferring VGG-16 network parameters via transfer learning. In [184], authors developed a multi-branch CNN model for the classification of motor imagery signals, which trains the EEG signals in parallel in 3 different CNN models and collects the results from these CNNs. The authors of [185] attempted to classify motor imagery signals by using a deep belief network (DBN). In addition, some studies classified motor imagery signals by using spiking neural network (SNN) [186,187]. While performing the learning process, deep neural networks train millions or even tens of millions of learnable parameters, and they do a lot of matrix multiplication for this training [188-190] causing a very heavy computational cost. GPUs and TPUs are used to perform these calculations more efficiently and faster. Neural network structures have gone beyond human knowledge and experience which make them a limit and have achieved higher classification accuracies. This method also reduces time and effort for feature design because neural networks automatically design the appropriate features needed for successful classification. Existing CNN-based classification methods have been designed and implemented as single and multiple CNN dimensions [182]. Since EEG signals have very low amplitude and high noise, various signal processing methods have been applied to the signals before classification. These signal processing methods can act in a way that reduces the size and complexity of the signal, but also loses the distinctive features within the existing signal. In addition, these classification methods may encounter classification success problems when data to train are insufficient. In the light of all this information, one of the biggest difficulties in MI EEG based BCI systems is the lack of large enough datasets and the inability to train deep learning models deeply enough. Accessible MI EEG datasets contain a limited number of subjects and a limited number of trials.

However, various solutions have been investigated by researchers to deal with this problem. Collecting EEG motor imagery signals is a very difficult and laborious process. The same experimental setup and EEG sensor placements need to be set up in the same way for each subject over and over again, the subjects must not make any physical movements, they must be trained before the experiment, and they must maintain their concentration throughout the experiment. Even blinking one's eyes or a strong heartbeat can distort the EEG signals. For all these reasons, there is very little data collected and accessible. Researchers have used and developed various data augmentation methods to reproduce existing data to overcome this problem. Another problem is the preprocessing of EEG data, which is very sensitive, noisy and difficult to analyze [191,192]. In order to find meaningful properties and in-depth information contained in EEG signals with time series data type, researchers generally used certain frequency bands and time intervals and applied various preprocessing processes to the signals. In addition, many preprocessing methods have been applied to convert the signals to 2D images. It has been stated in various studies that different frequency bands contain important information. However, the preprocessing methods applied to the signals can cause the loss of important information in such sensitive data and create a limit for the classification success. All these mentioned problems prevent the creation of a generalized classifier and cause the creation of customized classifiers that vary from person to person and cause low classifier accuracies. Considering all the above mentioned in this study, NF-EEG, a new generalized CNN classifier model has been developed that increases the accuracy by input reshaping and data augmentation methods without any signal processing.

The distinctive features of EEG signals are obtained by different signal processing techniques. These distinguishing features are classified and integrated into a control system or used as a control signal by connecting to a device and it is provided to fulfill the desired purpose.

Classification processes are done by using signal processing and machine learning models where distinctive features are extracted manually from EEG signals or by using deep learning models where feature extraction is done automatically by the model.

As machine learning and deep learning methods, support vector machine [164], random forest [193], linear discriminant analysis [194], autoencoders [158], artificial neural network, deep belief network, recurrent neural network and convolutional neural network models [134] have been developed and used. In addition, hybrid models consisting of combinations of these methods and models have also been designed. The purpose of these methods and models is to analyze the EEG signals correctly and to extract the necessary distinguishing features. By using these manually or automatically extracted features, it is aimed to classify the brain signals correctly.

With all these methods mentioned, the main goal is to identify the distinctive features of the EEG signals and to classify them with a high classification accuracy. In addition, the designed methods and models are expected to be fast, highly accurate, robust and re-liable systems. These are very important requirements for real-time brain computer inter-face systems to be developed.

Signal preprocessing applied to EEG signals, which are very sensitive and difficult to analyze and understand, are based on a high level of user knowledge and experience [182, 195, 196, 197, 198]. In addition, many different processes are tried and time and energy is spent to choose the right method [196,199]. However, these processes applied to the EEG signals can reveal the distinctive features in the signals, but they can also cause the loss of these features [196,200,201].

In [182], the researchers explained that many feature extraction methods depend on user knowledge and experience, and they said that these methods may limit the success of the model. They also stated that detecting the right features requires extensive experience and observation, which also requires large time and effort. They explained that high classification accuracies can be achieved and time and effort can be saved with feature extraction processes performed automatically by neural networks. In [196], the researchers developed a deep learning network model that performs artifact removing for EEG signals and compared it with different methods. As a result of this comparison, they showed that some traditional models cause data loss and revealed that their model is faster than other models.

Considering all these factors, no preprocessing was applied in this study and training and classification were made with raw data. The way the data is used during training is one of the important factors affecting the success of the model [134]. In our literature review, we observed that researchers use different input shapes in their studies [134, 153, 158, 160, 182, 195, 199, 200, 202]. We could not find a specific study on the effect of input shapes on classification performance. This shortcoming was one of the inspirations for this study. In this study, the effects of different input shapes on system performance were investigated during the training of EEG motor imagery data with CNN models, which has not been specifically examined before.

Classifications using machine learning and deep learning, EEG signals are usually preprocessed, the distinctive features of the signals are extracted, and the developed models are trained and classifications are made. Studies have shown that Convolutional Neural Networks among these models give better results than other models [144].

While developing brain computer interfaces, traditionally, there are 7 stages: signal acquisition, signal preprocessing, feature extraction, feature selection, classification, application interface and application. In this study, a leaner and faster BCI development process is proposed by excluding the signal preprocessing phase using raw EEG signals and feature extraction and feature selection phases with CNN model design. This proposed methodology is shown in figure 2.3.

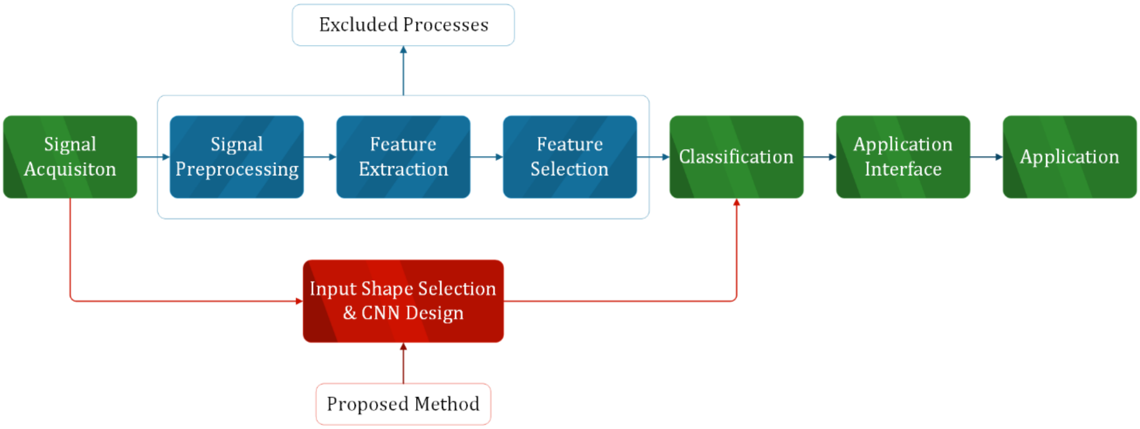


Figure 2.3. Proposed BCI processes.

Researchers have started to train and classify their models with raw data and have achieved remarkable results. In [195], researchers developed a CNN model using raw EEG signals. They stated that using all EEG channels in the dataset to train their model is computationally demanding and includes irrelevant features. To avoid this situation and to create a simpler classifier, combinations with different numbers of EEG channels were tried. They also tried different data augmentation methods to increase the success of the classifier. As a result, they explained their optimal channel configurations and data augmentation methods. In [203], the authors developed a CNN model using separated temporal and spatial filters and used raw EEG signals. With this study, they searched for minimal electrode pairs with which they can achieve high classification success and aimed to accelerate clinical applications by simplifying BCI designs. They stated that pre-processing methods used for EEG signals can increase the signal-to-noise ratios and classification success of nonstationary EEG signals, but this is not necessary [203]. In addition, the researchers used raw EEG data in their studies and achieved good results [174, 204-206]. The promising results obtained from previous studies with raw EEG data [174, 182, 195, 199, 203-206] are one of the reasons that encourage us to create a model using raw data.

Since EEG signals are non-stationary and show variable characteristics depending on time, covariate shifts occur and it is very difficult to classify these signals with high performance. The fact that input data distributions vary from person to person and even in different sessions of the same person makes it very difficult to develop real-time adaptive systems [207]. Various methods have been developed to prevent the covariate shift effect in studies using machine learning, signal preprocessing and feature extraction methods. In [207], authors developed an exponentially weighted moving average model to detect covariate shifts and designed an ensemble classifier. They updated their model by adding new classifiers to the ensemble that take into account the changes in the input data distribution and estimated shifts over time, and then they compared their method with different studies in the literature.

Feature distributions of train and test sets can be analyzed with density ratio estimation approaches [207]. Some of these approaches are least-squares importance fitting, Kullback-Leibler importance estimation procedure [208] and kernel mean matching [209].

In [210], the authors aimed to identify the most robust spatial filtering approach using a calibration dataset and a test dataset. They also examined performance variations by applying Stationary Subspace Analysis (SSA). Among the Common Spatial Pattern (CSP), Filter Bank Common Spatial Pattern (FBCSP), Filter Bank Common Spatial Pattern Time (FBCSPT), Source Power Co-modulation (SPoC), Spectrally Weighted Common Spatial Patterns (SpecCSP), and Surface Laplacian (SLap) methods, they showed that FBCSP and PBCSPT methods are the most robust approaches against feature covariance shift. In addition, after applying the SSA method, they achieved higher accuracy values in both datasets.

In deep learning methods, including our CNN model, learning takes place layer by layer, and the output of each layer becomes the input of another layer. Changes that occur in the input distributions during this type of learning cause a covariate shift in the model and the hidden layers try to adapt to the new distribution. This slows down the training and makes it very difficult to train models with saturating nonlinearity [211]. In [212], the researchers developed the batch normalization layer to overcome this problem. By this layer, each batch is normalized to have zero mean and unit variance. In this way, each batch is centered around the zero value and it is ensured that each feature value remains on the same scale.

In this study, among the deep learning classification methods mentioned above, two different CNN models (IS–EEG) consisting of the same layers and parameters were developed for 2D and 3D data structures. BCI Competition 2A and BCI Competition 2B datasets were used to train and evaluate these CNN models. These used data were transformed into 8 different input shape structures and the models were fed with these structures. Training and classification processes were performed separately for each input shape, and accuracy values for each subject, average training accuracy values and epoch times were obtained. In the results section, these obtained values are given in tables. Confusion matrix values and training and validation graphs are shown for each input shape and finally, comparisons of the obtained results are made. In the discussion section, the pros and cons of these designed input shape structures are discussed and input shape structures that can be used in brain computer interface designs are discussed. One of aims of this thesis is to investigate the effect of input shapes on classification and to propose classification methods with raw data that will save researchers time and effort in classifying noisy EEG signals, which are very difficult to train and classify [196,182].

3. DATASET, TOOLS AND METHODS

In this thesis, it is aimed to compare the proposed methods with other studies and classifiers by using Graz BCI Competition IV-2A and Graz BCI Competition IV-2B datasets [212], which are the most used datasets for EEG motor imagery classification. While developing their methodologies, researchers make great efforts to find the right input shape structures in which they can achieve the best and stable classification accuracy [144]. In this study, a 3D input shape structure is proposed to develop a faster model and achieve a better classification accuracy.

3.1. NF-EEG Datasets

As mentioned above, Graz BCI-IV-2A and Graz BCI-IV-2B datasets were used to train and test the NF-EEG and IS-EEG models. The BCI Competition IV 2A dataset contains data collected from 9 healthy individuals with a sampling rate of 250 Hz. These data consist of four classes: left hand, right hand, both feet and tongue. Motor imagery data were recorded with two different sessions, training and evaluation, for each person. Each session contains 288 trials and the dataset contains 5184 attempts in total. Persons sitting in front of a computer screen were first shown a fixation cross at time t = 0s, and a warning tone was rung at the same time. Two seconds later, at t = 2s, the left, right, upper or lower (left hand, right hand, foot and tongue classes respectively) parts were shown on the screen via an arrow and remained on the screen for 1.25s. In this way, people were asked to produce the desired motor imagery signals until t = 6s and the fixation cross disappeared from the screen. After this process, a short break was taken. These trial stages are shown in figure 3.1.

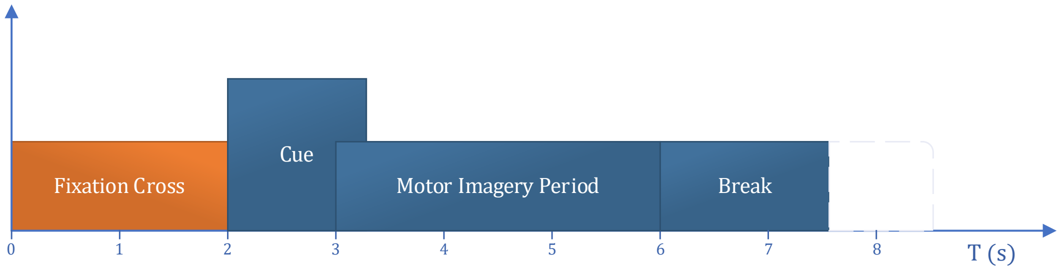


Figure 3.1. BCI Competition IV – 2A signal acquisition session

In the BCI Competition IV 2B dataset, there are data collected from 9 healthy individuals, different from those in the 2A dataset, with a sampling rate of 250 Hz. Data were collected with 5 sessions for each subject, 120 trials for the first two sessions, and 160 trials for the last three sessions, with a total of 6520 attempts. This dataset consists of 2 classes as right-hand and left-hand movements. While collecting data for the people in this dataset, a trial was started with the fixation cross displayed on a computer screen at time t = 0. Then, an arrow pointing to the left or right is displayed, indicating that the right-hand or left-hand movement should be imagined at the time t = 3s, and this arrow remained on the screen for 1.25 seconds. The person started to imagine the relevant hand movement at t = 4s, and this process took approximately four seconds. Finally, there was a break of at least 1.5 seconds. These trial stages are shown in figure 3.2.

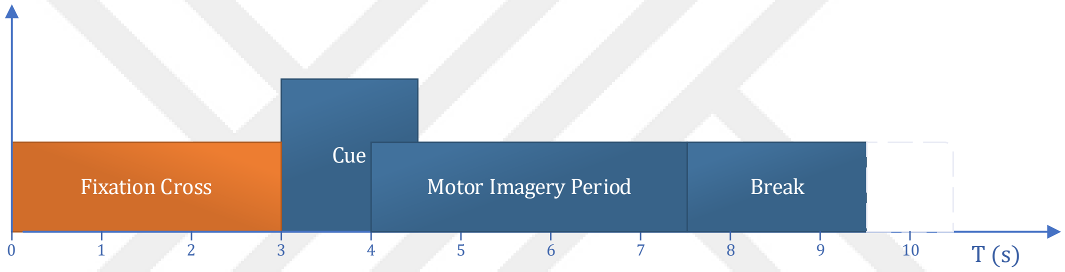


Figure 3.2. BCI Competition IV – 2B signal acquisition session

3.2. IS-EEG Datasets

Graz BCI-IV-2A and BCI-IV-2B datasets [212], which are two of the most widely used datasets for classification of EEG motor imagery signals, were used in this thesis. There are four different classes in the BCI-IV-2A dataset: left hand, right hand, both feet and tongue. These data were collected from 9 different healthy individuals with 22 EEG channels at a sampling rate of 250 Hz and included 288 trials and 5184 attempts. The dataset contains a different set for training and a different set for testing.

In the BCI-IV-2B dataset, there are two classes as left hand and right hand. These data were collected from 9 different healthy individuals with 3 EEG channels at a sampling rate of 250 Hz and included 280 trials and 6520 attempts. The dataset contains a different set for training and a different set for testing. In order to make an accurate comparison of the BCI-IV-2B dataset with the BCI-IV-2A dataset, only the left hand and right hand classes from the 2A dataset were used.

3.3. Convolutional Neural Networks

In this study, convolutional neural networks, which is one of the most used methods to classify EEG signals, were preferred as explained in detail in the literature section. CNN is a type of neural networks designed to reveal meaningful information and relationships in data such as time series and image data. CNN models try to obtain relationships within the data by using matrix calculations intensively.

Convolutional neural network models usually contain three main components. These are the convolutional layer, the pooling layer, and the fully connected layer. Apart from these, there are different customized layers such as normalization, dropout etc. A CNN model starts with a convolutional layer and ends with a fully connected layer. Between these two layers, there may be layers with very different numbers and parameters. The number and types of these layers determine the depth and complexity of the model. The complexity of the model increases as the data entering the CNN model moves towards the fully connected layer. As complexity increases, the CNN model is expected to reveal more complex and deeper relationships and features.

Convolution operations take place within the convolutional layers. These operations are done through structures called kernels that perform matrix operations by scanning a certain region of the data. Kernel scans all data by hovering over the data. During each scan, a dot product multiplication is made between the kernel and the data. After these processes, a feature map is obtained and classification success is determined according to the success of the model in obtaining a correct map. Pooling layers are used to achieve a faster and less complex training process by reducing the number of parameters at a user-specified rate. With the fully connected layer, the classification process is performed using the features extracted by the CNN model. A CNN model example is shown in figure 3.3.

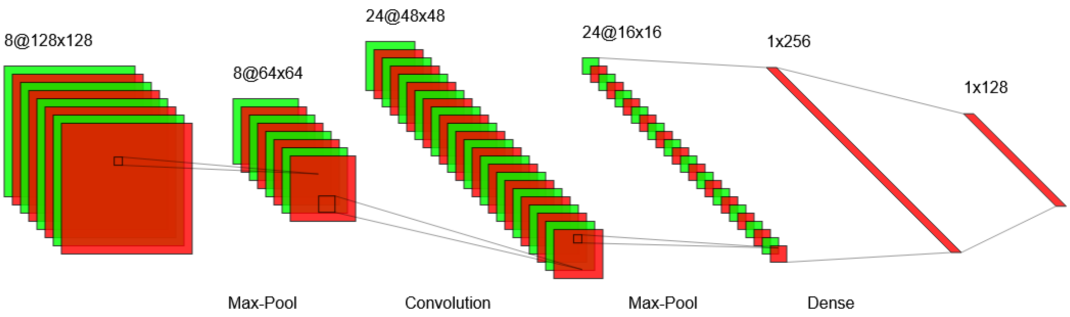
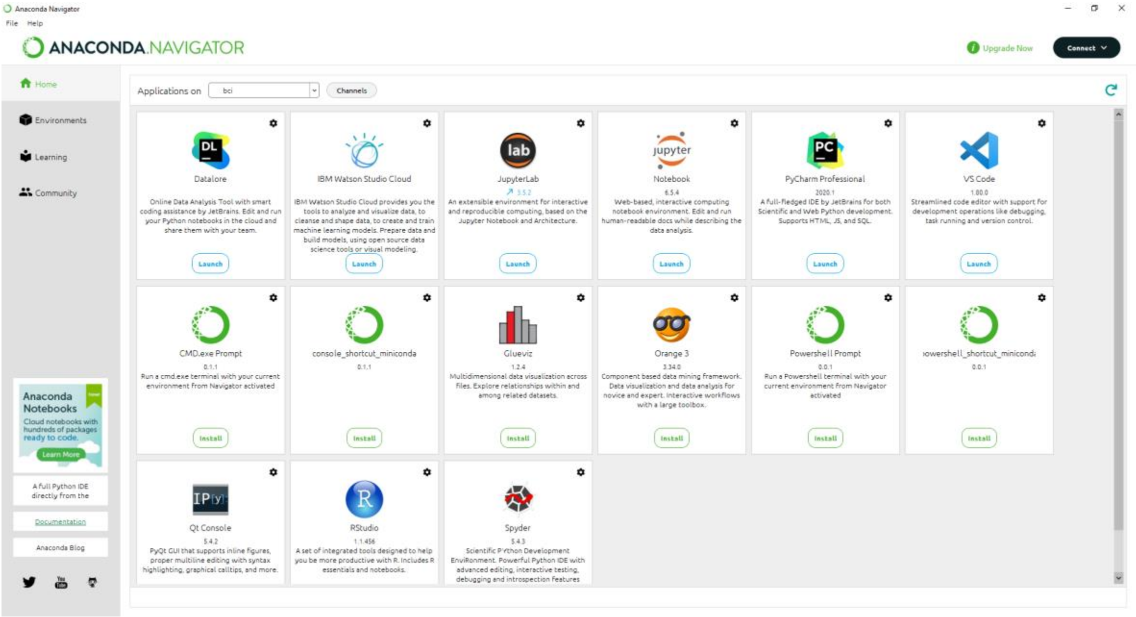


Figure 3.3. A convolutional neural network representation

3.4. Anaconda Navigator / JupyterLab

Anaconda Navigator is a graphical user interface for managing different libraries, environments and packages. Via Navigator, compatibility of different libraries needed in the programming environment is checked and the loading of incompatible library versions that may disrupt each other's operation is prevented. In this way, users' programming environments are prevented from being corrupted. In addition, the libraries needed are installed, deleted and managed via the user interface, not the terminals. The Anaconda Navigator graphical user interface used in this study is shown in figure 3.4.



Cloud notebooks with hundreds of packages ready to code.

A full Python IDE directly from the

Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab

↗ 3.5.2

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook

6.5.4

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyCharm Professional

2020.1

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

VS Code

1.80.0

Streamlined code editor with support for development operations like debugging, task running and version control.

CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

console_shortcut_miniconda

0.1.1

Glueviz

1.2.4

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Orange 3

3.34.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

›owershell_shortcut_miniconda

0.0.1

Qt Console

5.4.2

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Spyder

5.4.3

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Figure 3.4. Anaconda Navigator graphical user interface

JupyterLab is a web-based development environment. In this development environment, certain parts of the code can be run instead of running the entire code each time. Thanks to this feature, it is possible to work very efficiently in the fields of scientific calculations, machine learning, deep learning, data visualization and data science. Instead of running the entire code and searching for specific pieces each time, a section of the code is run for quick results. The JupyterLab development environment used in this study is shown in figure 3.5.

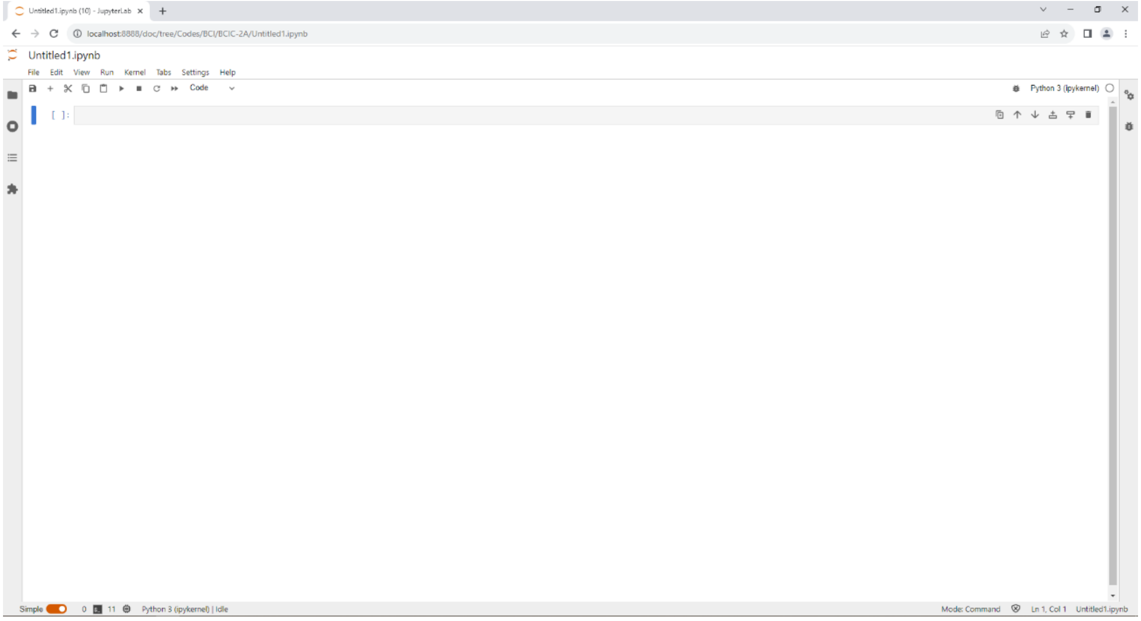


Figure 3.5. JupyterLab development environment

3.5. Python Programming Language

Python is a high-level and object-oriented programming language that has been frequently used in data science, machine learning and scientific computing in recent years. Its syntax and easy-to-read structure helped the python language spread rapidly. In addition to these, the fact that it has a large community and develops efficient and fast libraries for many different fields has made this language widely used in wider areas. In this study, python language was used for data reading, writing, editing, visualization processes and design and training of deep learning models. Libraries such as NumPy, pandas, SciPy, matplotlib, seaborn were also used with Python.

3.6. TensorFlow / Keras Libraries

TensorFlow is an end-to-end machine learning platform library that can be used with python. Thanks to this library, many machine learning and neural network models can be developed quickly and easily. Keras is also a deep learning API developed for TensorFlow. Thanks to Keras, deep learning models can be developed in a simple, flexible and fast way. In this study, GPU supported TensorFlow and Keras libraries were used.

3.7. Visual Studio Code

Visual studio code is a code editor with very rich and powerful development tools. Thanks to this editor, many different programming languages can be used simply and quickly with tools developed specifically for those languages. In addition, it has a large community and this community members develop new tools, allowing to quickly obtain useful solutions for many different problems. The visual studio code development environment used in this study is shown in figure 3.6.

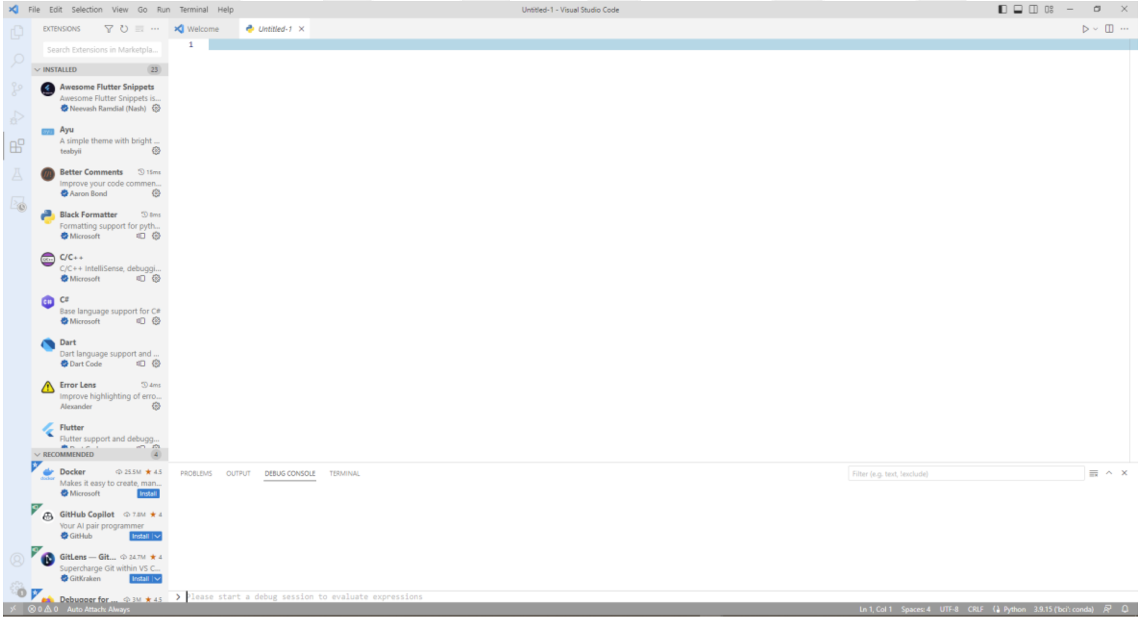

File Edit Selection View Go Run Terminal Help
Untitled-1 - Visual Studio Code
EXTENSIONS
Welcome
Untitled-1
Search Extensions in Marketpla...
INSTALLED 23
Awesome Flutter Snippets
Awesome Flutter Snippets is...
Neevash Ramdial (Nash)
Ayu
A simple theme with bright ...
teabyii
Better Comments 15ms
Improve your code commen...
Aaron Bond
Black Formatter 8ms
Formatting support for pyth...
Microsoft
C/C++
C/C++ IntelliSense, debuggi...
Microsoft
C#
Base language support for C#
Microsoft
Dart
Dart language support and ...
Dart Code
Error Lens 4ms
Improve highlighting of erro...
Alexander
Flutter
Flutter support and debugg...
RECOMMENDED 4
Docker 25.5M ★ 4.5
Makes it easy to create, man...
Microsoft
Install
GitHub Copilot 7.8M ★ 4
Your AI pair programmer
GitHub
Install
GitLens — Git... 24.7M ★ 4
Supercharge Git within VS C...
GitKraken
Install
Debugger for ... 3M ★ 4.5
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Filter (e.g. text, !exclude)
Please start a debug session to evaluate expressions
⊗ 0 ⚠ 0 Auto Attach: Always
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.15 ('bci': conda)

Figure 3.6. Visual Studio Code development environment

3.8. Flutter

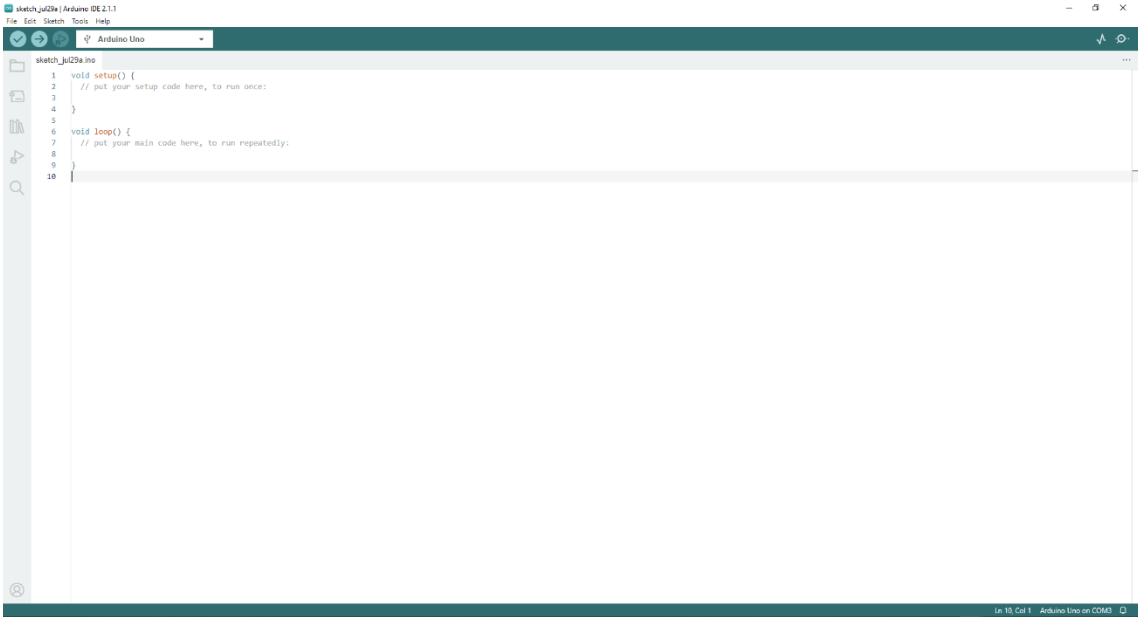
Flutter is a framework for developing multiplatform applications for desktop, mobile and web environments and runs on the Dart programming language. We used the flutter framework to develop the NF-BCI interface software.

3.9. Flet

Flet is a multiplatform framework for developing flutter applications via python. We used the flet framework to make the NF-BCI and NF-EEG and IS-EEG models work together.

3.10. Arduino IDE

Arduino integrated development environment is the user interface for communicating with Arduino boards and uploading programs to the boards. Thanks to this program, coding can be done through the processing language, libraries needed for Arduino cards can be downloaded, sensors connected to the cards can be read, and a program can be uploaded to the microcontroller to control Arduino cards and external devices. We used the Arduino IDE software to communicate with the NF-BCI and control the robotic arm. The Arduino development platform is shown in figure 3.7.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

Figure 3.7. Arduino integrated development environment

3.11. Model Generalization

Generalization is the ability of an artificial neural network to be sensitive to new and previously unseen data from the same distribution after it has been trained with a particular data [213] and also the ability to obtain evaluation results close to the classification success achieved during training. Neural networks are required to be fast, robust and have high classification success during implementation. In order to develop models with these capabilities, generalization methods are applied and it is aimed that the model approaches test successes close to the training success. For this purpose, layers and parameters such as dropout, normalization, regulators, constraints are used and hyperparameter fine tuning is performed. Situations such as overfitting and underfitting may occur that may hinder the generalizability of a model. Overfitting is a situation where a model results from memorizing the data exactly, rather than finding the relationships between the data used for training. In this case, although the model achieves very high training accuracy during training, it achieves very low success in validation and test sets. Underfitting, on the other hand, occurs when a model cannot find the relationships between the data used during training, and in this case, the training accuracy values are very low and the model needs to be developed before starting the testing process.

In this study, kernel constraint and kernel regularizer parameters were used, together with the above-mentioned dropout and normalization layers, in order to find the relationships between multidimensional, complex and noisy EEG data and to obtain a generalized model. These layers and parameters have been optimized with fine tuning and grid search, overfitting of the model has been prevented and generalizability has been increased. Optimization processes for generalization and used layers and parameters are explained in section 4.1.1 and section 5.1.

3.12. Data Augmentation

The ability of neural networks to achieve a good classification success largely depends on the size of the training data. As the size of the dataset increases, the desired features are found in the dataset in many different scenarios and forms, thus making it easier for neural networks to establish relationships within the data. Conversely, if the dataset is small or insufficient, classification success is usually low.

The datasets used for classification of EEG motor imagery signals are generally few in number and these datasets contain a limited number of trials. Strict rules such as the subject's going through a training and preparation stage for the acquisition of EEG signals, the person's limb and muscle movements were prohibited, and person's requirement to maintain a constant state of focusing while imagining a movement are various obstacles to the creation of these datasets in large sizes. In order to overcome this problem, researchers obtained new training data both by making various transformations on the original data and adding various noises to the original data. [183, 184, 188-190]

In this study, in order to improve the classification accuracy, we augmented the dataset by applying jittering and scaling operations which explained in [214]. Details of the jittering and scaling operations and variables are explained below.

3.12.1. Jittering

Jittering or jitter, is a way of simulating additive sensor noise. In order to apply jittering to the signals, Gaussian distribution was applied with two parameters, standard deviation (STD) and mean. Gaussian noise was obtained by using all data in each dataset with different STD values and added to the signals and then these signals added to dataset. STD values of 0.03, 0.07 and 0.1 and mean values of 0 were applied.

The probability density for the Gaussian distribution is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.1}$$

Where σ the standard deviation and μ is the mean. The square of the standard deviation,

σ^2, is called the variance.

3.12.2. Scaling

Scaling or scale is the name given to the process that changes the magnitude of the data by multiplying the data with a random scalar. Random scalar is obtained by using STD and mean through Gaussian distribution in the scaling process and scaled data is obtained by multiplying these values with the data. Random scalar was obtained by using all data in each dataset with different STD values and the values were multiplied with the signals and then these signals added to the dataset. STD values of 0.03, 0.07 and 0.1 and mean value of 1 were applied.

4. NF-BCI: AN END-TO-END BRAIN COMPUTER INTERFACE AND ITS COMPONENTS

In this section, the components of the developed brain computer interface are explained in detail. These components are;

- NF-EEG & IS-EEG: Developed CNN Architectures
- Input Reshaping
- NF-BCI: A Real Time End to End Brain Computer Interface Software

In the first part, the architectures of the CNN models, the types of layers they contain and the number of layers, hyperparameter values, the path and transformations of the data through the models are explained. In the input reshaping section, the designed input shape models are introduced, how these models are created, how the data are distributed and for what purposes these models are created are explained. In the NF-BCI software section, the designed interface is introduced and the functions of the sections on the interface are explained. In addition, the control scenarios and working principles designed for the robot arm to be controlled with this software are explained in detail. In the following Robotic Arm Chapter, the robotic arm, which will act as an external device for the developed brain-computer interface, is introduced. The mechanical and electronic components of the robotic arm are introduced, and information is given about the motion capabilities of the robotic arm, which is ready for use.

4.1. NF-EEG & IS-EEG: Developed CNN Architectures

Within the framework of the methods and problems mentioned in the previous sections, we developed the NF-EEG model, which is a generalized CNN model that increases the classification success without applying any signal preprocessing to the raw data. Input reshaping has been done to make this model faster and more efficient, as explained in the input shape section above. In addition, we applied two different data augmentation methods to generate new data from the original data and enlarge the dataset in order to increase the classification success. The details of the model architecture and input reshaping methods are explained below.

4.1.1. NF-EEG Architecture

NF-EEG was developed as a CNN model that performs automatic feature extraction using raw EEG data. The training process of the model is carried out without developing any filters for the signals or using existing filters. In order to extract the distinctive and in-depth information in the signals, this process is carried out automatically within the model without developing any feature extraction method. CNN structures contain layers such as convolution, pooling and fully connected layers and in this study, depthwise and separable convolution layers are used together with the mentioned layers. Convolution and average pooling layers were used to extract relationships and features, and to perform classification fully connected layers were used. CNN structures contain relatively fewer parameters than other neural networks and are less prone to overfitting. The first priority during model design is to develop a model that extracts data features well, generalizable, robust, has low standard deviation, suitable for real time BCI systems, has high classification success, and has appropriate hyperparameters. Previous studies generally used a filter bank consisting of frequency ranges from 4 to 7 Hz, from 8 to 13 Hz, and from 13 to 32 Hz and also from 8 to 32 to obtain information in beta and mu bands [191, 192, 215] and also theta band helps to identify difference between motor imagery tasks [216, 217]. In many studies, researchers used different model parameters for different subjects and trials and couldn’t generalize the models to work in a wide area according to these parameters, or they trained many models with different kernel sizes and created hybrid models. In [182], authors stated that different kernel size values gave better results in different trials from person to person and even for the same person, and kernel size value could not be generalized, and they trained more than one model with different kernel size values.

High variability among individuals can be observed when recording EEG signals. However, variability can be observed in recordings of EEG signals from a single individual at different times. In order for the designed brain computer interface systems to be generalizable, robust and predictable, long-term stability must be established. In addition, it is important to establish long-term stability in order for the brain computer interfaces to have steady state performance.

NF-EEG aims to create a generalized model for all individuals and trials without using or developing any filter bank or signal preprocessing methods, rather than using signal preprocessing methods, training multiple models, using different parameters for different models and subjects, and focusing on to develop models from person to person or from trial to trial. Besides, for the hyperparameter adjustment of this developed model, it is aimed to increase the success of the model by using grid search in a wide variety of parameters. Detailed structure and hyperparameter lists of the developed NF-EEG model are given in figure 4.1.

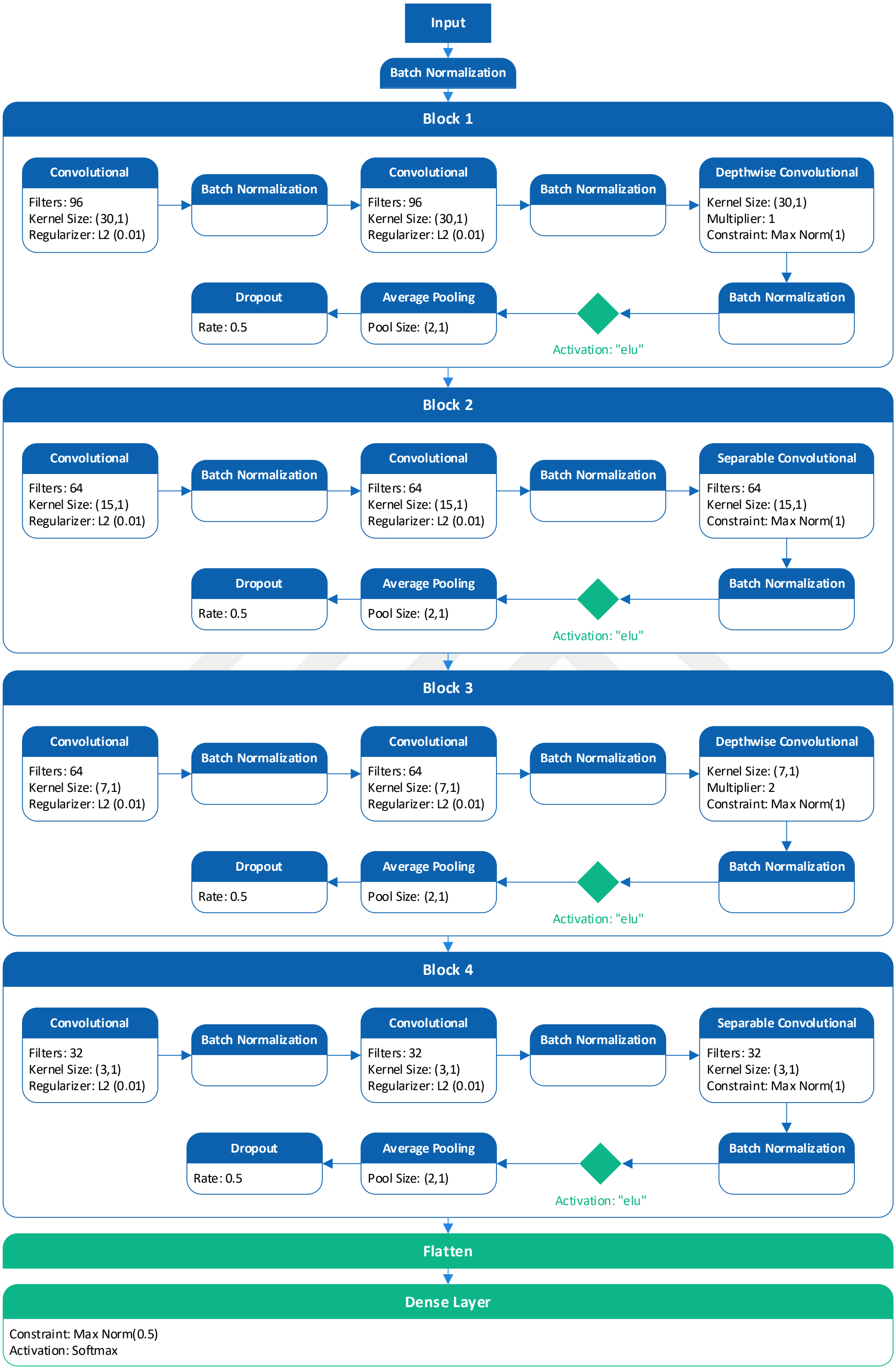


Figure 4.1. NF-EEG Structure

In the proposed model, first the EEG data converted to L x 1 x Nc form is entered into the system. Input shape, where B to represent batch size, is a 4D tensor in the form of B x L x 1 x Nc. Then, the data group as much as the batch size is entered into the batch normalization block to be normalized among themselves as mini batches.

The normalized data first enters Block 1 and passes through the layers in this block to extract the temporal features. While extracting these features, convolutional, normalization, depthwise convolutional layers were used. 1D kernels are used along the vertical axis to extract time, frequency and space features between CNN layers. Then the Elu activation function is used, and then the data is average pooled at a rate of 2x1 and sent to Block 2 with a dropout rate of 0.5. The data coming to Block 2 is again subjected to the Elu activation process by passing through the convolution, separable convolution and normalization layers to extract the deep temporal features, and then it enters the dropout process at a rate of 0.5 by applying 2x1 average pooling and is sent to Block 3.

In Block 3 and Block 4, it was aimed to extract in-depth temporal features within the signals and to establish temporal and spatial relationships better by passing through the same layers in Block 1 and Block 2, respectively, but with different parameters. The data in the form of L x 1 x Nc coming out of Block 4 enters the flatten layer for classification and is converted to a 1D feature vector. The feature vector is sent to the dense layer for classification, where it performs classification. After the classification process, classification success is achieved by testing for each subject and also all data collectively by evaluation.

In order to determine the model parameters, the classification accuracies were examined by testing many different combinations from a wide range of options and selections were made from the model grid search list according to trained model and tested classification results. Model parameters and grid search lists are shown in figure 4.1 and figure 5.1.

4.1.2. IS-EEG Architecture

Two Convolutional Neural Network (CNN) models were created to train and classify the selected datasets. These models have same layers and parameters and have the architecture needed to train 2D and 3D data. Signal preprocessing methods used in the vast majority of studies in this field were not used in this study, training and classification were made with raw data. It is predicted that the filters used for different wavelengths can make the distinguishing features blurrier in very sensitive, complex and difficult to understand EEG signals, and feature extraction is preferred to be done with raw data and CNN models designed with the right parameters. The developed model is shown in figure 4.2.

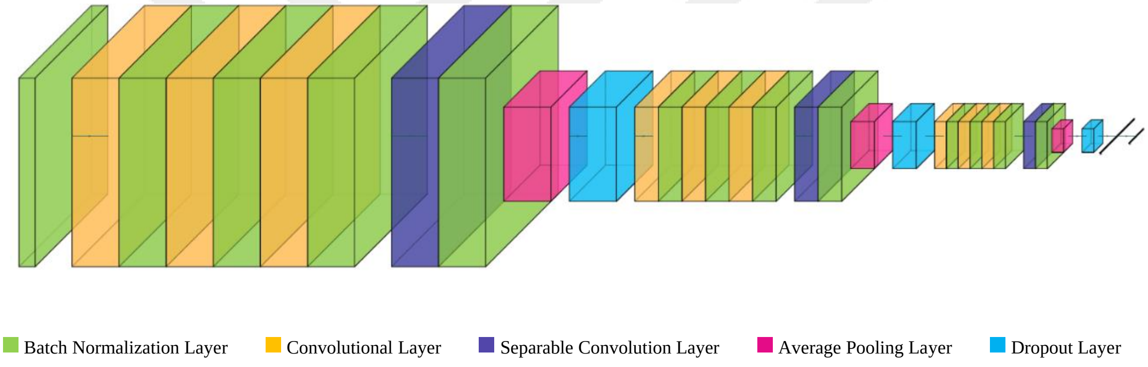


Figure 4.2. IS-EEG Architecture

Proposed model starts the training with the batch normalization layer and then data enters 3 consecutive blocks. Each block consists of 3 x (convolution – batch normalization), separable convolution, batch normalization, Elu activation function, average pooling and dropout layers, respectively. After these three blocks, the data enters the flatten layer, dense layer and then is ready for classification.

Since we will start training with raw data in the model we have developed, the data first enters the batch normalization layer to be normalized. Normalized data enters the convolutional layer to extract temporal, spatial or spatiotemporal features according to the input shape structure. Here, the kernel size for 2D and 3D input shapes is [30x1] and the number of filters is 64. For the batch normalization layers, the momentum is 0.1 and the epsilon value is 1x10-5. After each triple convolution–batch normalization sequence, the data enters the separable convolution layer. In this layer, kernel operations are performed by factorizing the convolutional kernel into two smaller kernels and then data is normalized. For the separable convolutional layer, the kernel size is [15x1] and the number of filters is 64. Then the Elu activation function is applied.

Elu function produce more accurate results by converging cost to zero faster. Where α is a constant between 0 and 1 defined by user and its expression is as follows:

$$R(z) = \begin{cases} z \quad , & z > 0 \\ \alpha.(e^{z} - 1), & z \leq 0 \end{cases}$$

The data enters the average pooling layer to reduce the number of model parameters and perform faster operations by acquiring new features. In this layer, the data is resized by reducing it by ½. Afterwards, 70% of the data is randomly separated with the dropout layer, preventing the overfitting tendency of the model. The data coming out of the dropout layer enters the second block to pass through the same sequence.

In this block, the kernel size for the convolutional layers is [15x1] and the number of filters is 32. For the separable convolutional layer, the kernel size is [7x1] and the number of filters is 32. Batch normalization, activation function, average pooling and dropout parameters are the same as the first block. In the third block, the kernel size for the convolutional layers is [15x1] and the number of filters is 16. For the separable convolutional layer, the kernel size is [7x1] and the number of filters is 16. Batch normalization, activation function, average pooling and dropout parameters are the same as other blocks. After the data is processed in three blocks, it enters the flatten layer and the dense layer, and then softmax activation function is applied.

4.2. Input Reshaping

Input shape structures are one of the most important parameters needed for CNN model designs to achieve high classification accuracies and researchers have used many different input shape structures in their previous studies [29]. In this study, it was investigated whether faster and higher classification success could be achieved by designing different input shape structures, and these input shape structures were compared with each other. The signals in these datasets have been converted into 2D and 3D structures with 8 different input shapes for each dataset. The transformed data for each dataset are explained below.

4.2.1. T x C Structure

With this 2D structure, timestamps data is placed on the horizontal axis and EEG channel data is placed on the vertical axis. T x C input shape structure is shown in figure 4.3.

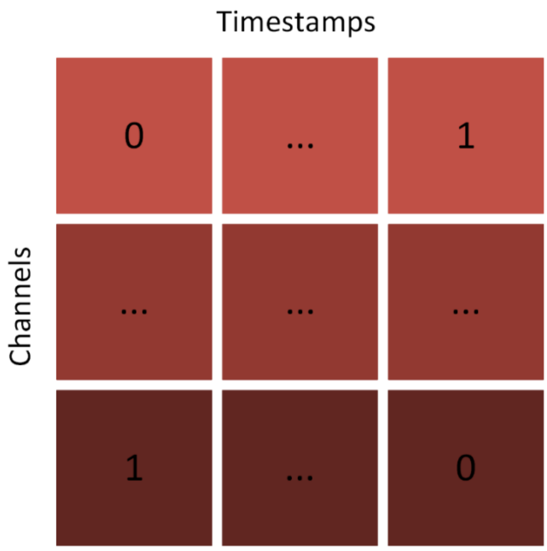


Figure 4.3. T x C input shape structure

4.2.2. C x T Structure

With this 2D structure, EEG channel data is placed on the horizontal axis and timestamps are placed on the vertical axis. C x T input shape structure is shown in figure 4.4.

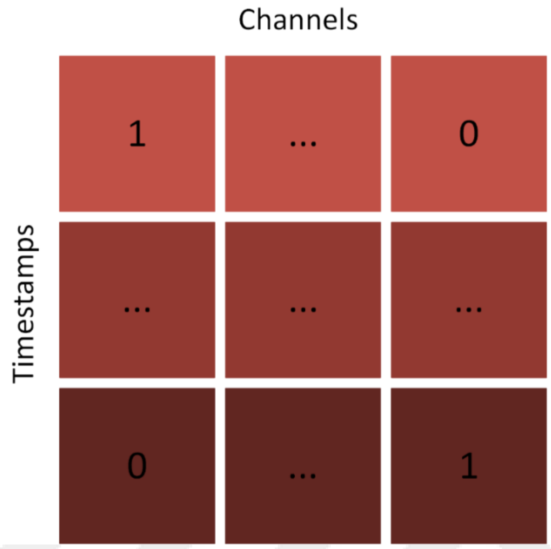


Figure 4.4. C x T input shape structure

4.2.3. T x C x 1 Structure

This 3D structure was created by adding the third axis to the T x C matrix structure, and with this structure, CNN models that use 3D matrices as inputs can be used. T x C x 1 input shape structure is shown in figure 4.5.

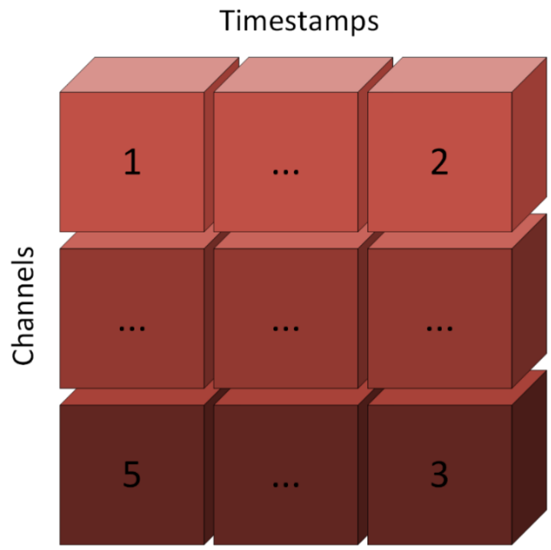


Figure 4.5. T x C x 1 input shape structure

4.2.4. C x T x 1 Structure

This 3D structure was created by adding the third axis to the C x T matrix structure, and with this structure, CNN models that use 3D matrices as inputs can be used. C x T x 1 input shape structure is shown in figure 4.6.

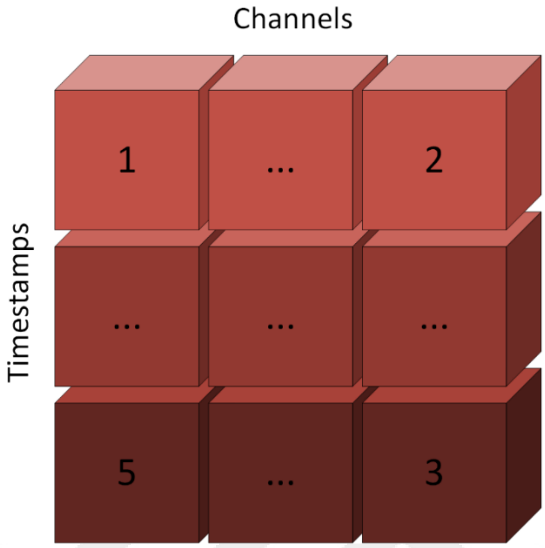


Figure 4.6. C x T x 1 input shape structure

4.2.5. 1 x T x C Structure

With this 3D structure, horizontal axis is arranged to contain a single column and a single EEG channel data is placed in this column. Afterwards, the remaining EEG channels as the third dimension were placed on this axis. 1 x T x C input shape structure is shown in figure 4.7.

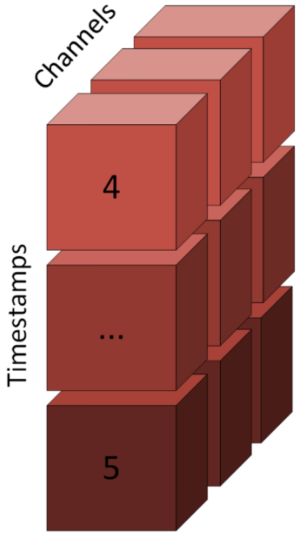


Figure 4.7. 1 x T x C input shape structure

4.2.6. 1 x C x T Structure

With this structure, a single column is placed on the horizontal axis and timestamps data is placed in this column. The remaining timestamps were placed in the third dimension and it was aimed to find the relationships in this way. 1 x C x T input shape structure is shown in figure 4.8.

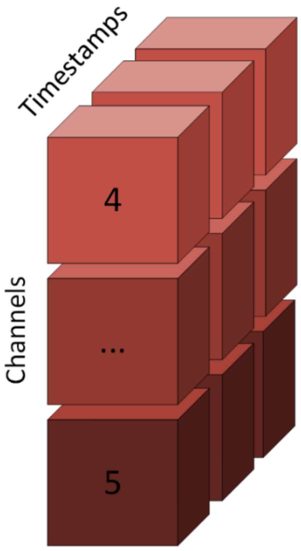


Figure 4.8. 1 x C x T input shape structure

4.2.7. T x 1 x C Structure

With this 3D structure, the vertical axis has been arranged to contain a single row, and the data of all EEG channels has been added to this row. Afterwards, the data of the remaining EEG channels were added as the third dimension. T x 1 x C input shape structure is shown in figure 4.9.

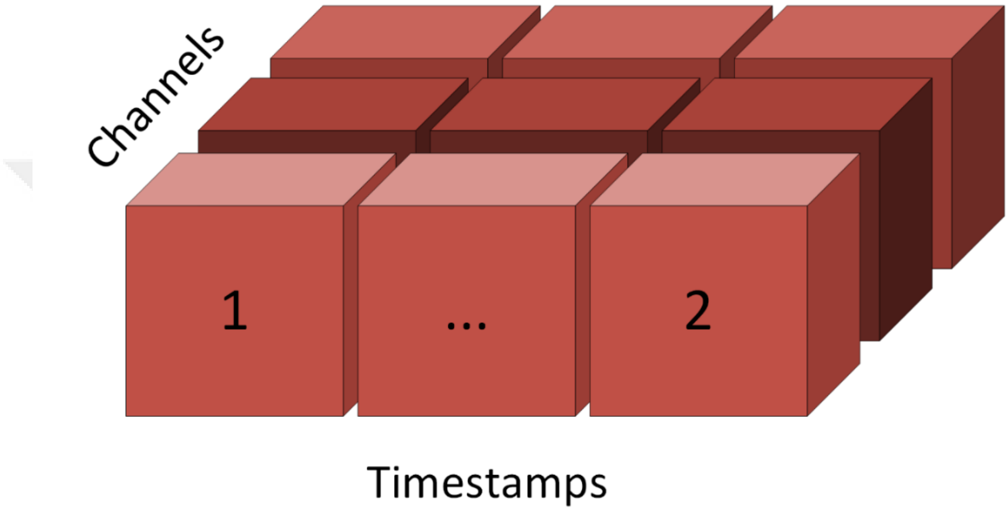


Figure 4.9. T x 1 x C input shape structure

4.2.8. C x 1 x T Structure

With this structure, timestamps data were arranged in a single row on the vertical axis and the remaining timestamps data was added as the third dimension. C x 1 x T input shape structure is shown in figure 4.10.

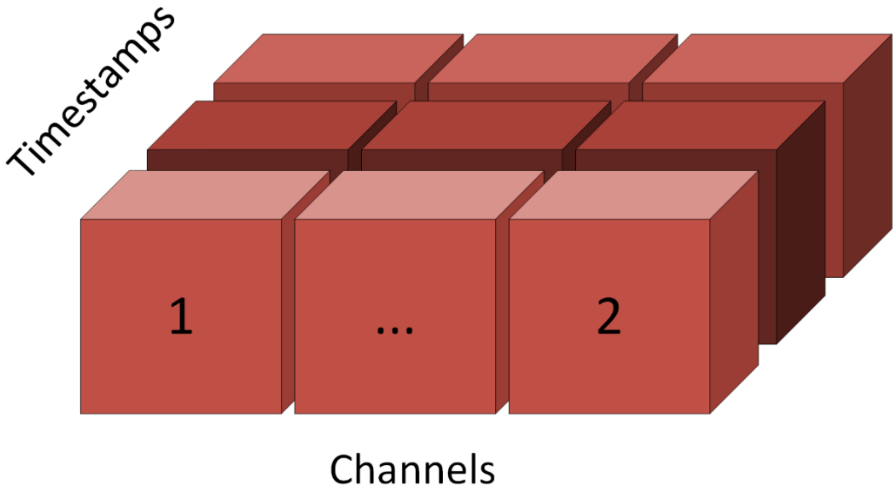


Figure 4.10. C x 1 x T input shape structure

The purpose of these transformations is not only to train the model with 2D data dimensions, but also to train the model with both 2D and 3D data dimensions to find temporal and/or spatial features in EEG signals.

To fulfill this purpose, each trial is structured according to the height, width and channel (H x W x C) parameters of the library used. Height represents the horizontal axis, width represents the vertical axis and channel represents the third axis. These structures are matrices formed as T x C, C x T, T x C x 1, C x T x 1, 1 x T x C, 1 x C x T, T x 1 x C, C x 1 x T, where C represents number of EEG channels, and T represents number of timestamps.

The aim of these different 2D and 3D matrix structures is to examine the effects of shape changes, total number of parameter changes, complexity of calculations on classification speed, robustness and accuracy. Total number of parameters change given in Equations 4.1 – 4.3.

$$W_c = K^2 x\ C\ x\ N \qquad (4.1)$$

$$B_c = N \qquad (4.2)$$

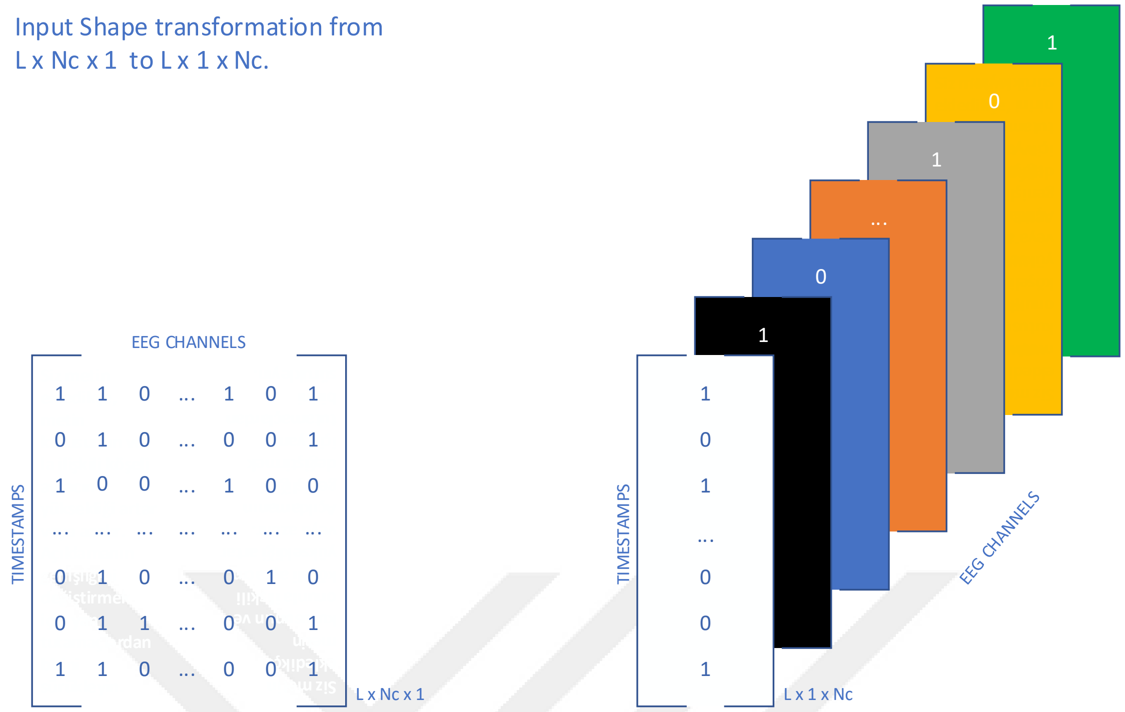
$$P_c = W_c + B_c \qquad (4.3)$$

Where W_c is number of weights of the convolutional layer, K is the size of kernels used in the convolutional layer, C is the number of channels of the input, N is the number of kernels, B_c is the number of biases of the convolutional layer and P_c is the number of parameters of the convolutional layer.

4.2.9. Input Reshaping for NF - EEG

Each trial data was taken with time on the horizontal axis and EEG channel data on the vertical axis. Therefore, for each trial, dataset converted to matrices where row data refers to timestamps and column data refers to data from EEG channels. For each trial, a matrix of L x N_c structure was obtained, where L representing the timestamp and N_c the number of EEG channels. Where N_t representing the number of trials, a matrix of 1 x L x N_c structure was obtained for a trial, and N_t x L x N_c structure was obtained for all trials.

In many studies, 2D matrix structures in the form of L x N_c or N_c x L have been used to extract EEG features. In this study, by adding another dimension to the 2D data for the NF-EEG model, the data was converted into a 3D matrix structure. This process has been done with the aim of reducing the kernel size, the total number of parameters, complexity of calculations rather than using 2D kernels, collecting each sensor data on a single axis, scanning temporal information on a channel basis, and speeding up the classification process. The matrix in the form of L x N_c in the trial example given above was transformed into L x Nc x 1 by adding a third axis, and finally, the EEG channels were transformed into L x 1 x Nc by moving the each EEG channel to the channel section in the deep learning library used. This transformation is shown in figure 4.11.



$$\begin{array}{ccccccc}
1 & 1 & 0 & \dots & 1 & 0 & 1 \\
0 & 1 & 0 & \dots & 0 & 0 & 1 \\
1 & 0 & 0 & \dots & 1 & 0 & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots & \dots \\
0 & 1 & 0 & \dots & 0 & 1 & 0 \\
0 & 1 & 1 & \dots & 0 & 0 & 1 \\
1 & 1 & 0 & \dots & 0 & 0 & 1
\end{array}$$

$$\begin{array}{c}
1 \\ 0 \\ 1 \\ \dots \\ 0 \\ 0 \\ 1
\end{array}$$

Figure 4.11. Input shape transformation for NF-EEG

4.3. NF-BCI: A Real Time End to End Brain Computer Interface Software

NF-BCI (No Filter Brain Computer Interface) is a software that visualizes EEG signals in real time, classifies the EEG signals entering the system with the CNN classifier running in the background, and converts the classification results into control signals and can control a computer, external device or a robotic arm. The NF-BCI software main screen is shown in figure 4.12.

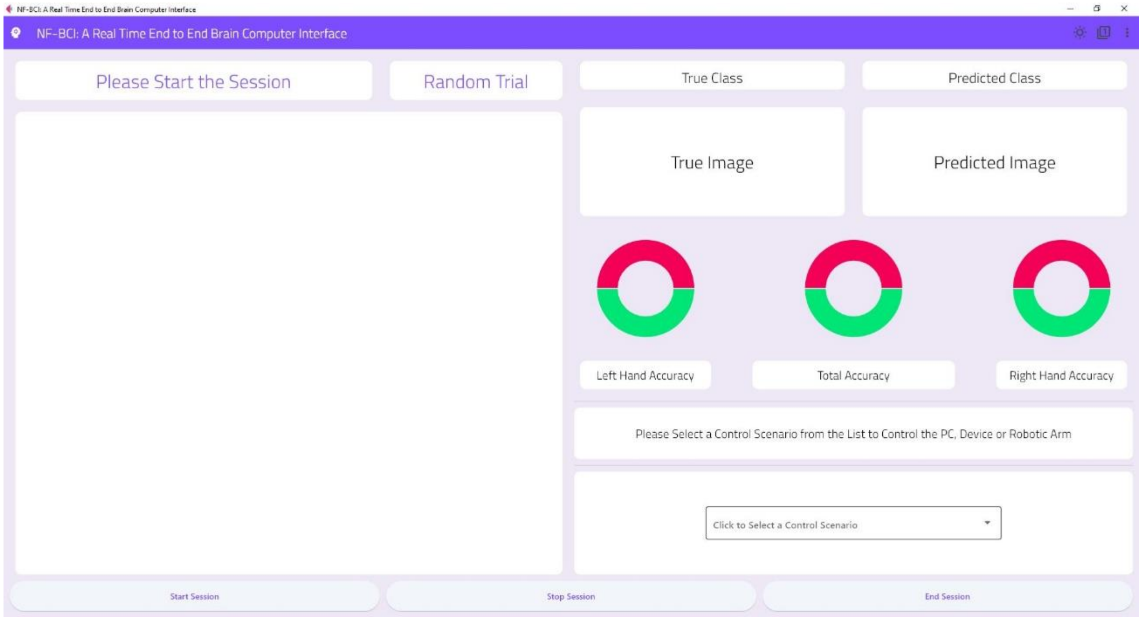

NF-BCI: A Real Time End to End Brain Computer Interface
NF-BCI: A Real Time End to End Brain Computer Interface
Please Start the Session
Random Trial
True Class
Predicted Class
True Image
Predicted Image
Left Hand Accuracy
Total Accuracy
Right Hand Accuracy
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm
Click to Select a Control Scenario
Start Session
Stop Session
End Session

Figure 4.12. NF-BCI Main screen

4.3.1. Trial Selection / Random Trial Generator

In this section, the user can select a specific trial so that the signals in the used data set can be simulated in the system, or can click the random trial button to run the system with a randomly selected trial by the system. If the random trial button is not used, the system will use a specific predefined trial. When the random trial button is used, the system selects a random trial instead of a specific predefined trial. The screens before and after using the random trial button are shown in figures 4.13, 4.14 and 4.15.

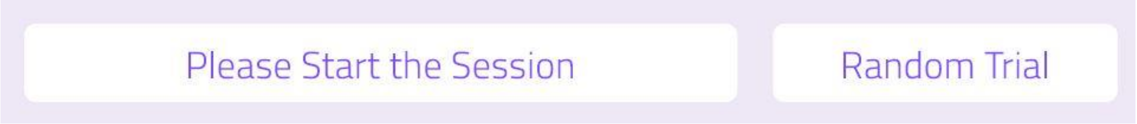
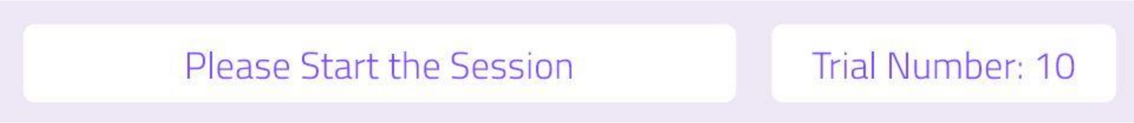
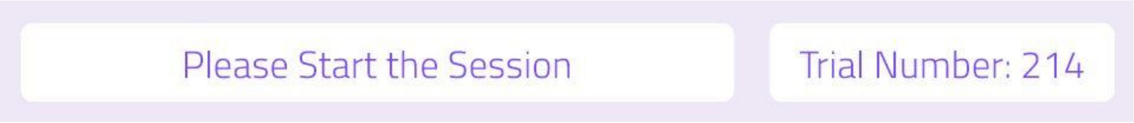


Figure 4.13. Interface displayed when random trial button is not used


Please Start the Session

Trial Number: 10

Figure 4.14. Interface displayed when using random trial button - 1


Please Start the Session

Trial Number: 214

Figure 4.15. Interface displayed when using random trial button - 2

4.3.2. Plot Screen

In this area, the EEG signals entering the system are visualized depending on the number of channels and the EEG signals are displayed to the user based on time. In this way, the user has the opportunity to observe the visuals of each EEG channel. When the program is opened, the plot screen appears as a blank screen. When the user operates the system, EEG signals are plotted and visualized in real time for each channel. On the vertical axis, the signal amplitude is shown in microvolts and the corresponding EEG channel is written in each graph. Figures 4.16 and 4.17 show the images of the plot screen when the program is opened and the system is running.

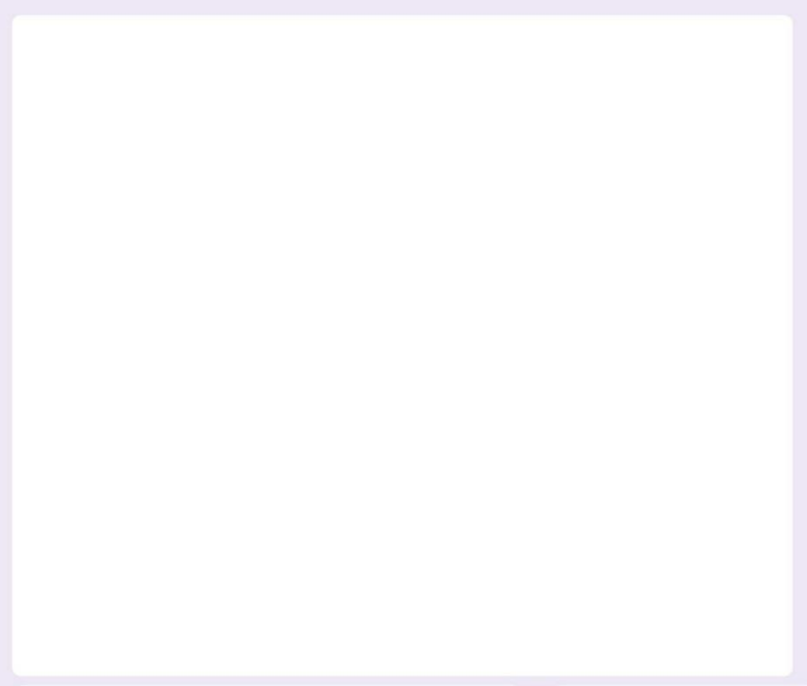


Figure 4.16. Plot screen when the program is opened

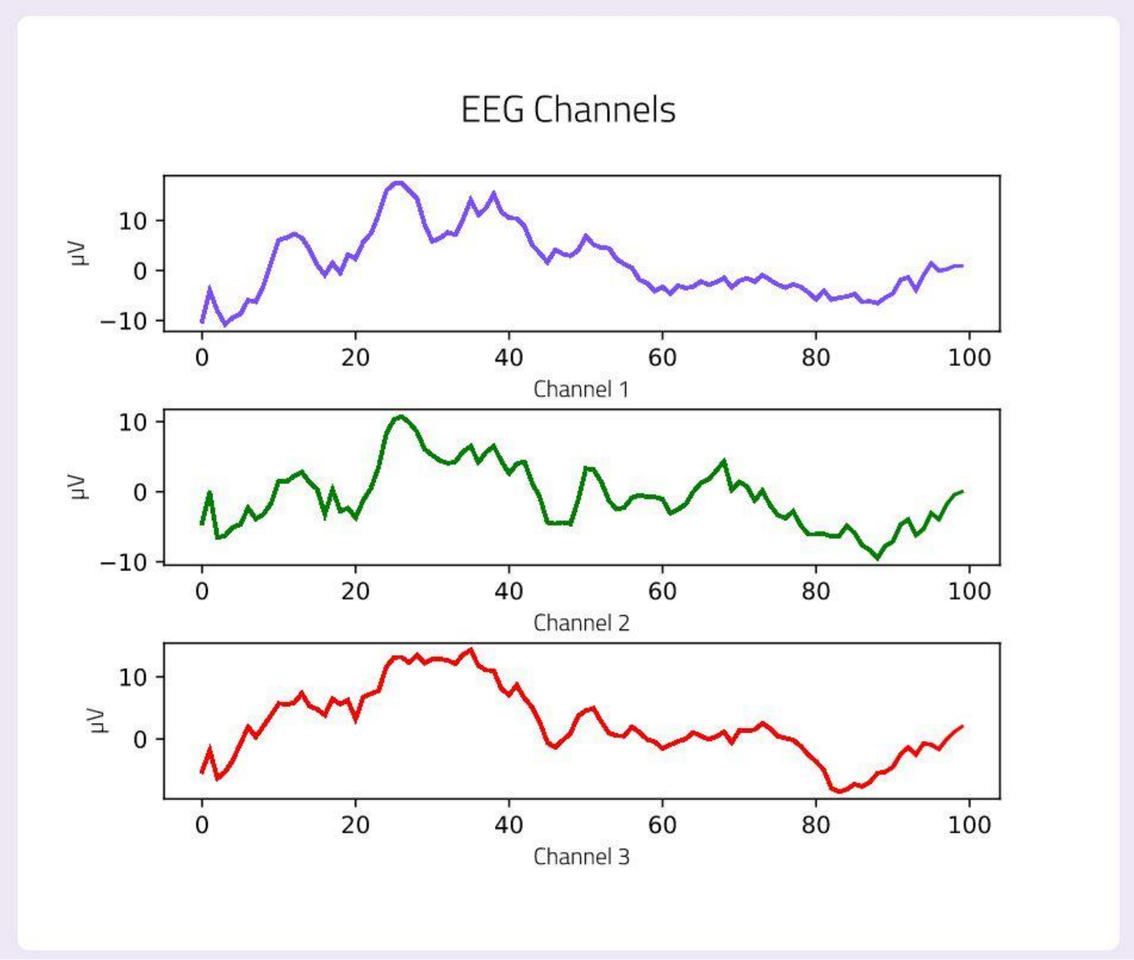


Figure 4.17. Display of the plot screen when the system is run

4.3.3. True Class / Predicted Class

In this field, there are the titles of the screens showing the real class of the signals in the used dataset and the class predicted by our classifier. This part is shown in figure 4.18.

Figure 4.18. True Class and Predicted Class headers

4.3.4. True Image / Predicted Image

This area contains two boxes with text and an image. The left box is used to project the image of the true class to which the EEG signals belong. The box on the right reflects the image of the predicted class our classifier predicts as a result of the classification of EEG signals. This area is shown in figure 4.19.

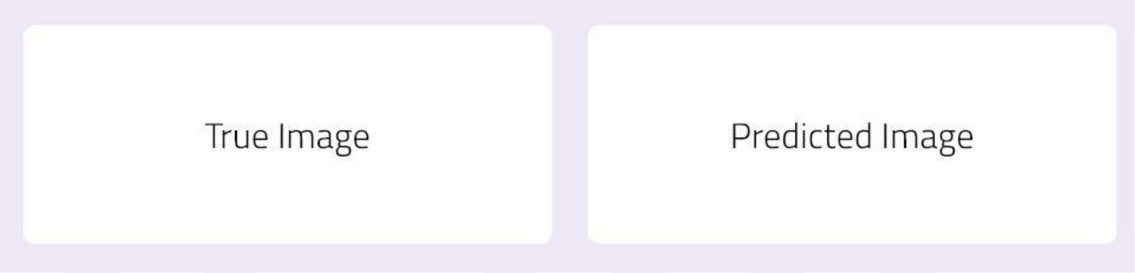


Figure 4.19. True Image and Predicted Image boxes

Two different images are used to display in the True Image and Predicted Image boxes. These are the right-hand and left-hand classes that represent the EEG classes in the dataset. These images are shown in figure 4.20.

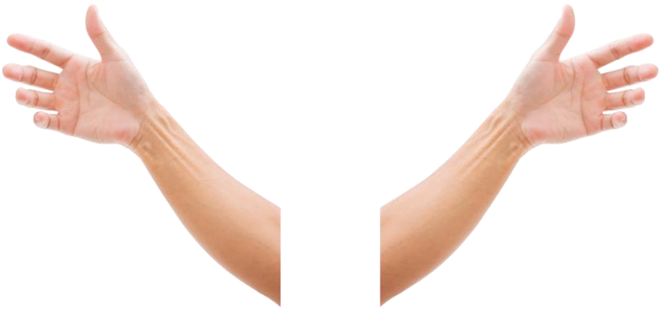


Figure 4.20. Images representing Right-hand and Left-hand classes

When the system is started and the visualization of the EEG signals is finished, the true classes of the EEG signals and the predicted classes of the classifier are indicated in the True Image and Predicted Image boxes with the visuals shown in figure 6.9. If the true class and the predicted class are the same, the background of the boxes is filled with green color, otherwise the true class is represented by a green background, while the incorrectly predicted class is visualized with a red background.

If the true class is right-hand and the classifier also predicted right-hand, a right-hand image is added to both boxes and the background of both boxes is filled with green. This scenario is shown in figure 4.21.

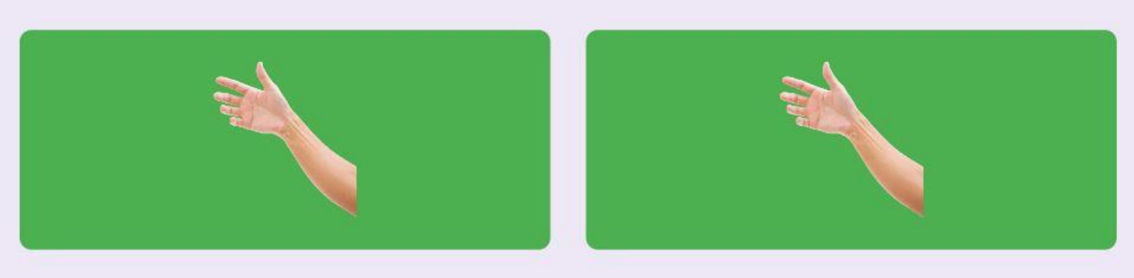


Figure 4.21. Right hand scenario of true class and predicted class

If the true class is left-hand and the classifier predicted left-hand, a left-hand image is added to both boxes and the background of both boxes is filled with green. This scenario is shown in figure 4.22.

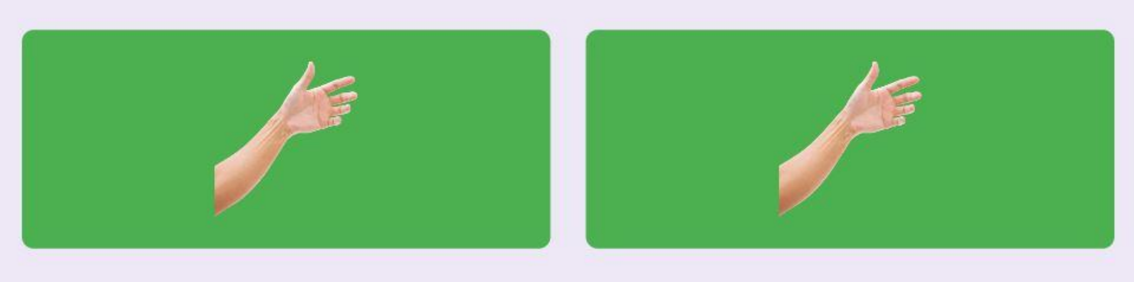


Figure 4.22. Left hand scenario of true class and predicted class

If the true class is left-hand and the classifier predicted right-hand, a left-hand image is added to the True Image box and a right-hand image is added to the Predicted Image box. The background of the True Image box is filled with green, and the background of the Predicted Image box is filled with red. This scenario is shown in figure 4.23.

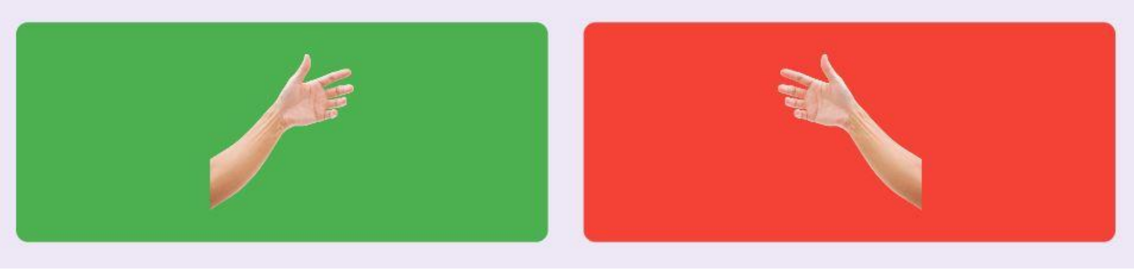


Figure 4.23. The scenario where the true class is left-hand and the predicted class is right-hand

If the true class is right-hand and the classifier predicted left-hand, a right-hand image is added to the True Image box and a left-hand image is added to the Predicted Image box. The background of the True Image box is filled with green, and the background of the Predicted Image box is filled with red. This scenario is shown in figure 4.24.

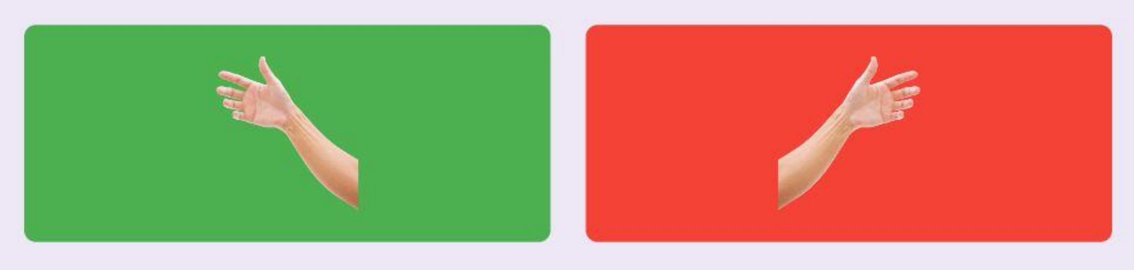


Figure 4.24. The scenario where the true class is right-hand and the predicted class is left-hand

4.3.5. Accuracy Charts

In the accuracies section, there are three different dynamic pie charts and text boxes below these charts. These charts and text boxes are shown in figure 4.25. NF-BCI records the true labels, predicted classes, correctly predicted classes, incorrectly predicted classes and their ratios for each class in the background. On the left are the correct and incorrect prediction percentages of the true left classes of EEG signals entering the system. This chart is updated each time the system completes the classification process and its current rates are displayed on the screen. In the left-hand accuracy box below the left-hand chart, the ratio of the total number of correctly predicted left-hand classes to the actual total left-hand classes is expressed as a fraction.

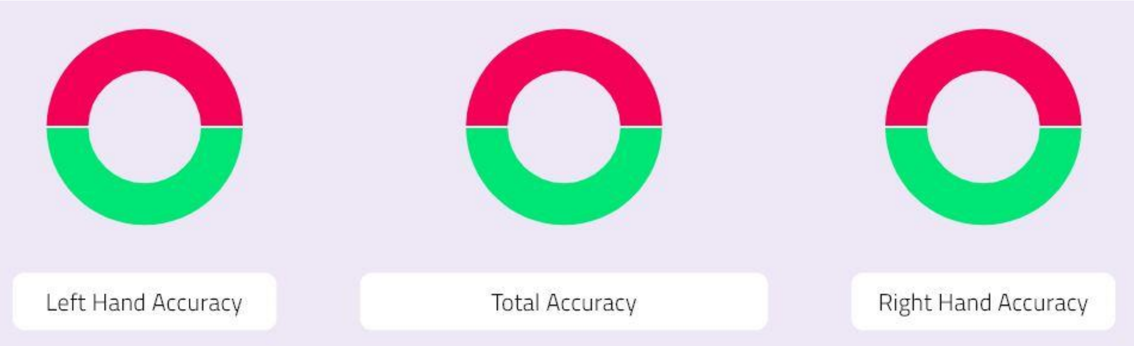
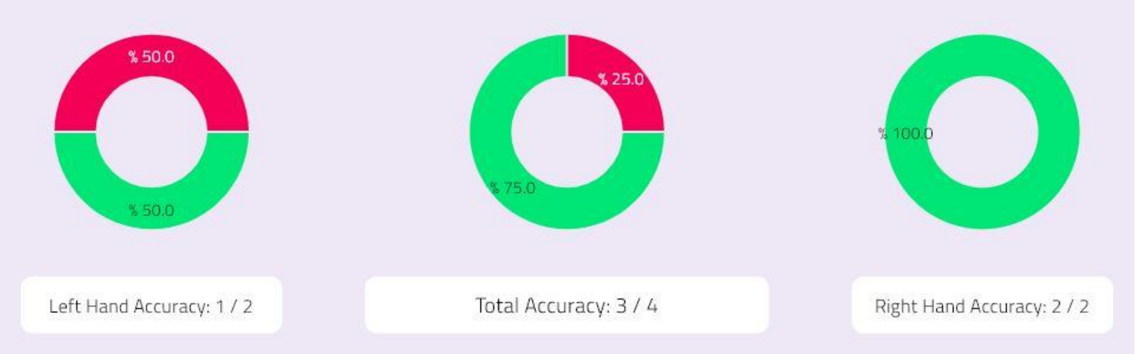


Figure 4.25. Dynamic accuracy pie charts and accuracy boxes

In the middle are the total percentages of correct and incorrect predictions obtained as a result of the classification of the EEG signals entering the system. This chart is updated each time the system completes the classification process and displays the current rates on the screen. In the total accuracy box below the middle chart, the ratio of the total number of correctly predicted left and right hand classes to the actual total left and right hand classes is expressed as a fraction. While the system continues to run, these rates are updated after each trial and these rates and values are displayed on the screen. The graph that occurs while the system is running is shown in figure 4.26.



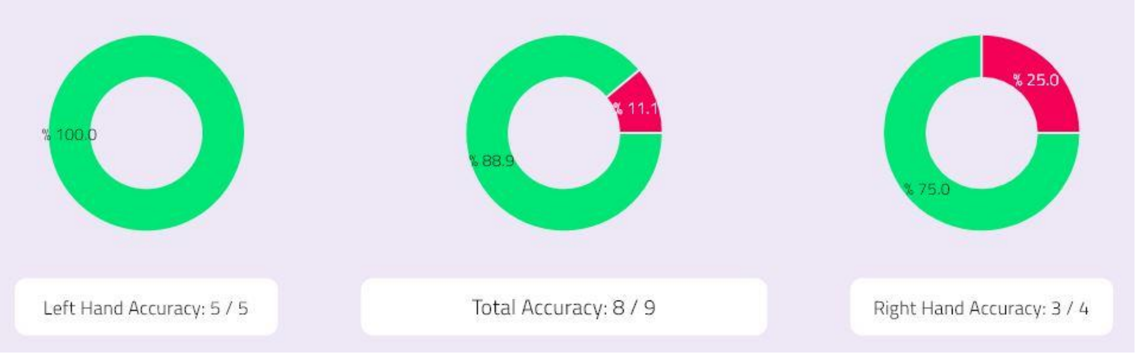
Left Hand Accuracy: 1 / 2

Total Accuracy: 3 / 4

Right Hand Accuracy: 2 / 2

Figure 4.26. Continuous accuracy pie charts and accuracy boxes

On the right are the correct and incorrect prediction percentages of the true right classes of EEG signals entering the system. This chart is updated each time the system completes the classification process and its current rates are displayed on the screen. In the right-hand accuracy box below the right-hand chart, the ratio of the total number of correctly predicted right-hand classes to the actual total right-hand classes is expressed as a fraction. The graph that occurs after the system completes the run is shown in figure 4.27.



Left Hand Accuracy: 5 / 5

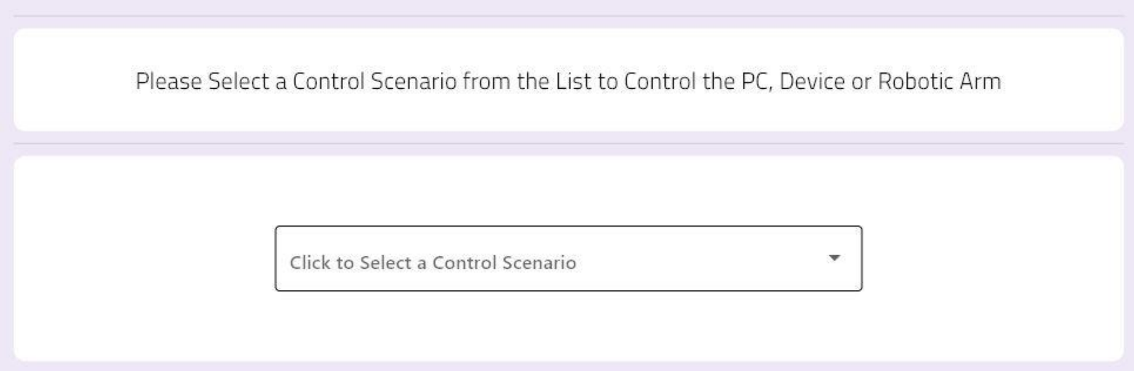
Total Accuracy: 8 / 9

Right Hand Accuracy: 3 / 4

Figure 4.27. Final accuracy pie charts and accuracy boxes

4.3.6. Control Scenarios

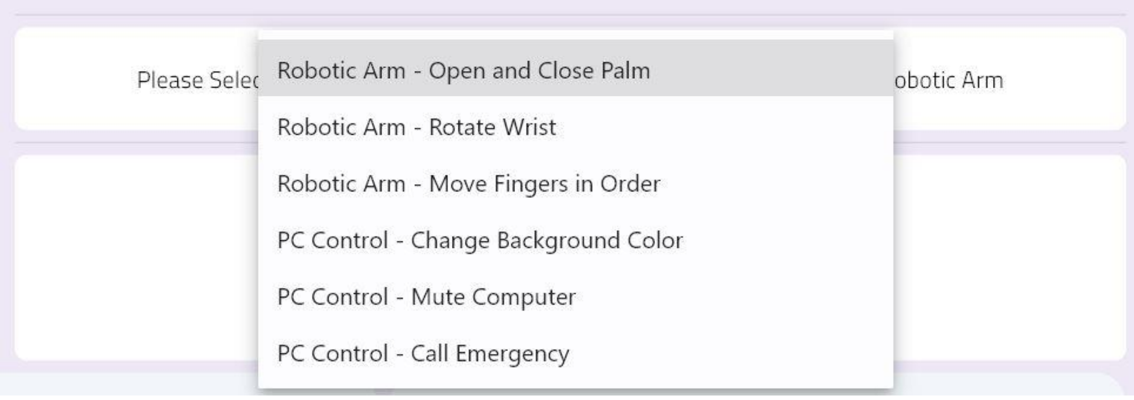
Besides visualizing and classifying EEG signals, the NF-BCI can control a computer, external device or a robotic arm with classification results. The control section is shown in figure 4.28.



Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Figure 4.28. Control scenarios screen

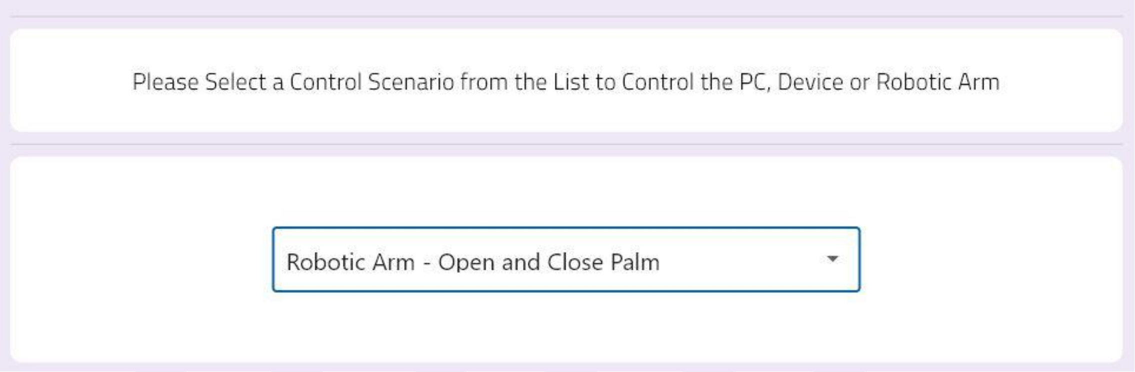
One of the predefined control scenarios can be selected on this control screen. These scenarios include opening and closing all the fingers of the robotic arm at the same time, rotating the wrist of the robotic arm, sequentially opening and closing the fingers of the robotic arm, changing the background color of the computer, muting the computer, and making an emergency call. The selection screen of these scenarios is shown in figure 4.29.



Please Selec
obotic Arm
Robotic Arm - Open and Close Palm
Robotic Arm - Rotate Wrist
Robotic Arm - Move Fingers in Order
PC Control - Change Background Color
PC Control - Mute Computer
PC Control - Call Emergency

Figure 4.29. Predefined control scenarios

Once one of these control scenarios is selected, the NF-BCI establishes a communication link with the computer, external device or robotic arm in the background. After establishing this connection, it generates the necessary control signals and transfers them to the microcontroller and realizes the control scenario in real time. The selected control scenario screen is shown in figure 4.30.



Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Open and Close Palm

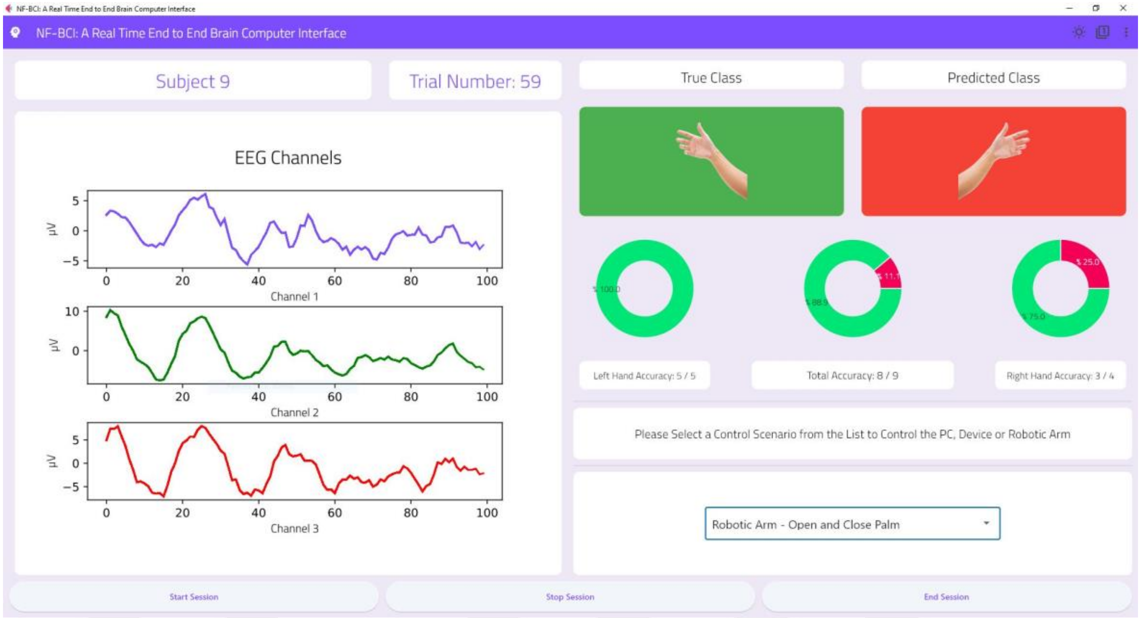
Figure 4.30. Selected control scenario

4.3.7. Start / Stop / End Session Buttons

There are three buttons in this section to start, stop and end the session, these are start session, stop session and end session buttons. The real-time NF-BCI session is started with the start session button. Stopping the session waits for the current session to end, and after the session ends, the general operation is interrupted and then allowed to continue from where it left off. The end session button waits for the current session to end and ends the current session after the current trial ends. The start, stop and end session buttons are shown in figure 4.31.

Figure 4.31. Start / Stop / End Session buttons

NF-BCI plots EEG signals in real time during each trial and updates the subject name box, true images and predicted images boxes, accuracies charts, and accuracies text boxes at the end of the trial. After all sessions are completed, the screen shows the total statistics of previous trials. The screen after the system has finished the trial operations is shown in figures 4.32 and 4.33.



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 9

Trial Number: 59

Left Hand Accuracy: 5 / 5

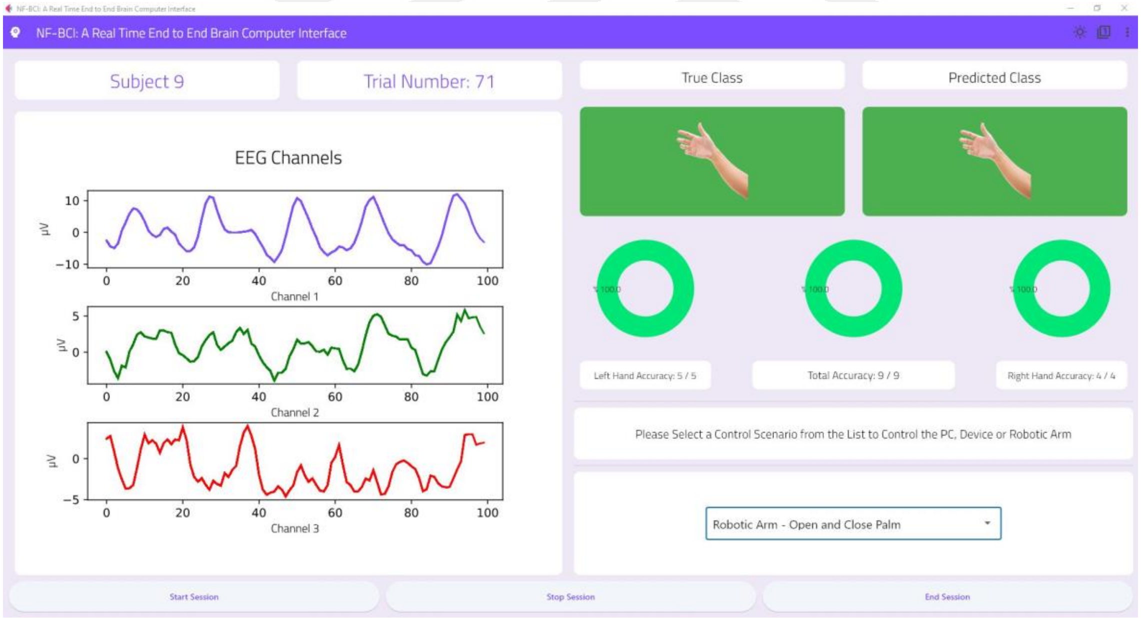
Total Accuracy: 8 / 9

Right Hand Accuracy: 3 / 4

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Open and Close Palm

Figure 4.32. Final screen of sessions - 1



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 9

Trial Number: 71

Left Hand Accuracy: 5 / 5

Total Accuracy: 9 / 9

Right Hand Accuracy: 4 / 4

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Open and Close Palm

Figure 4.33. Final screen of sessions - 2

5. ROBOTIC ARM

One of the aims of this thesis is to use the developed CNN models NF-EEG and IS-EEG and the designed NF-BCI software in real-time control of a robotic arm. NF-EEG and IS-EEG models outperformed many state-of-the-art classifier models in their fields of use. In addition, these models we have developed do not perform any signal processing and use raw data.

NF-BCI software has also been developed in a flexible structure where many different classifiers, especially NF-EEG and IS-EEG, can be integrated. NF-BCI software can run NF-EEG and IS-EEG models in real time in the background. In this section, the robot arm to be controlled with the NF-BCI software is introduced and the developed CNN models are used during these controls. This section also describes the mechanical components, electronic components and features of this robotic arm.

5.1. Mechanical Components

In this study, robotic arm parts of the Humanoid Robotic Torso PROTO1 [235] design were used as a robotic arm. There are eight main parts in this model: fingers, hand, mini servo – thumb connector, wrist connector, arm body, servo motor bearing, arm rotation part and servo horns. The 3D models of these parts are purposefully organized and structured. All parts were then printed using PLA material via a 3D printer. After the printing process is completed, the parts are mounted to each other using elastic string, fishing line, screws and strong adhesive and made ready for application together with electronic component connections.

5.1.1. Fingers

The robotic arm has 5 fingers. The thumb has one knuckle and consists of 2 parts, while the other 4 fingers have 2 knuckles and 3 parts. The fingers are connected to each other with elastic string and fishing line. Each finger has full closing and full opening functions. The fingers can be brought to the desired opening or closing angles by means of the servo motors to be used. Finger models are shown in figure 5.1.

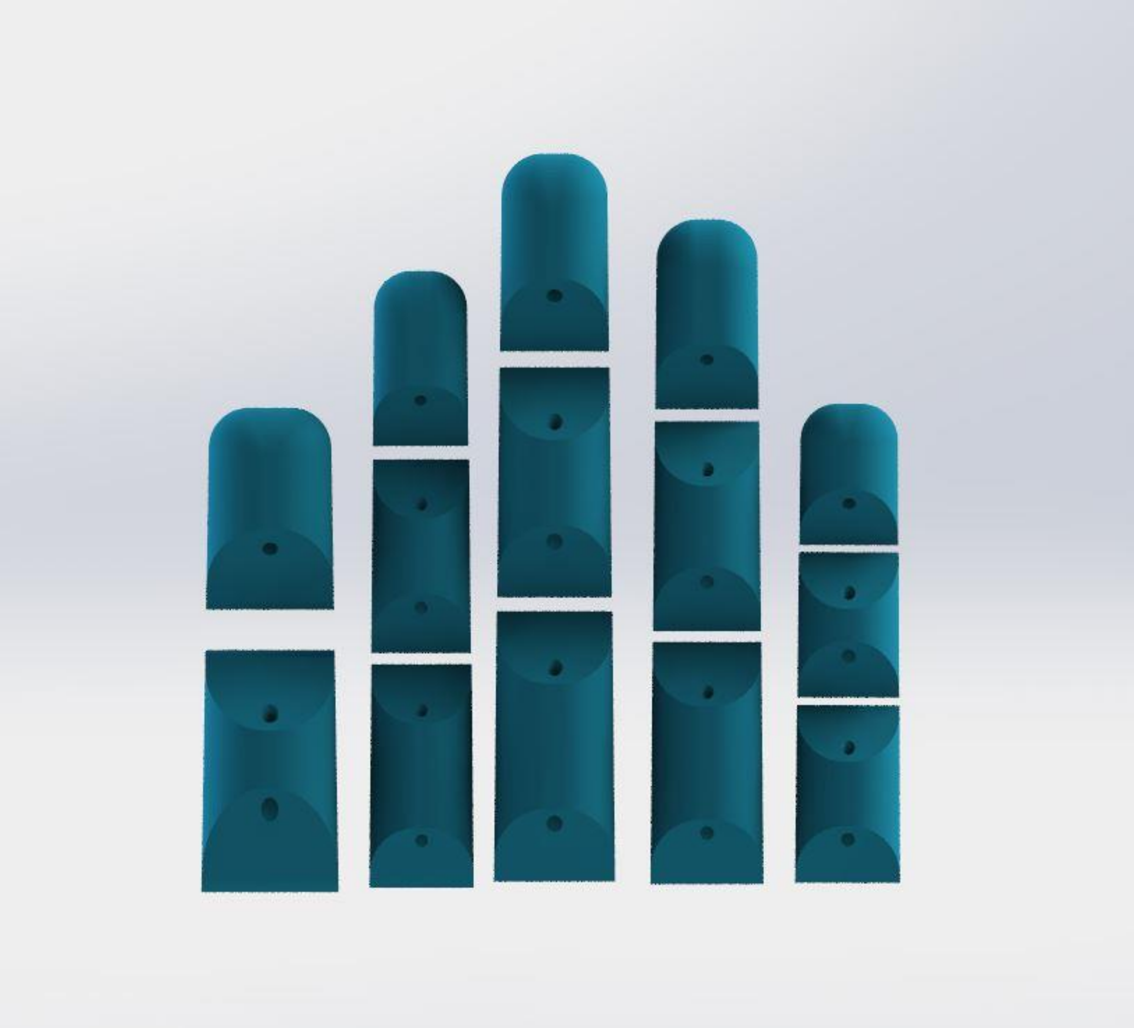


Figure 5.1. Robotic arm fingers

5.1.2. Hand

The hand of the robotic arm is modeled to be joined to the fingers from the upper part and to the wrist from the lower part. The fingers are mounted on the hand with the help of fishing line and elastic rope. Wrist and hand connection is made by shrink fit and adhesive on the wrist. There is also a mini servo slot on the hand, allowing a mini servo to be attached to the thumb. With the servo motor to be placed in this mini servo slot, it is aimed to make the movement that will enable the thumb to come and go towards the palm. The hand model is given in figure 5.2.

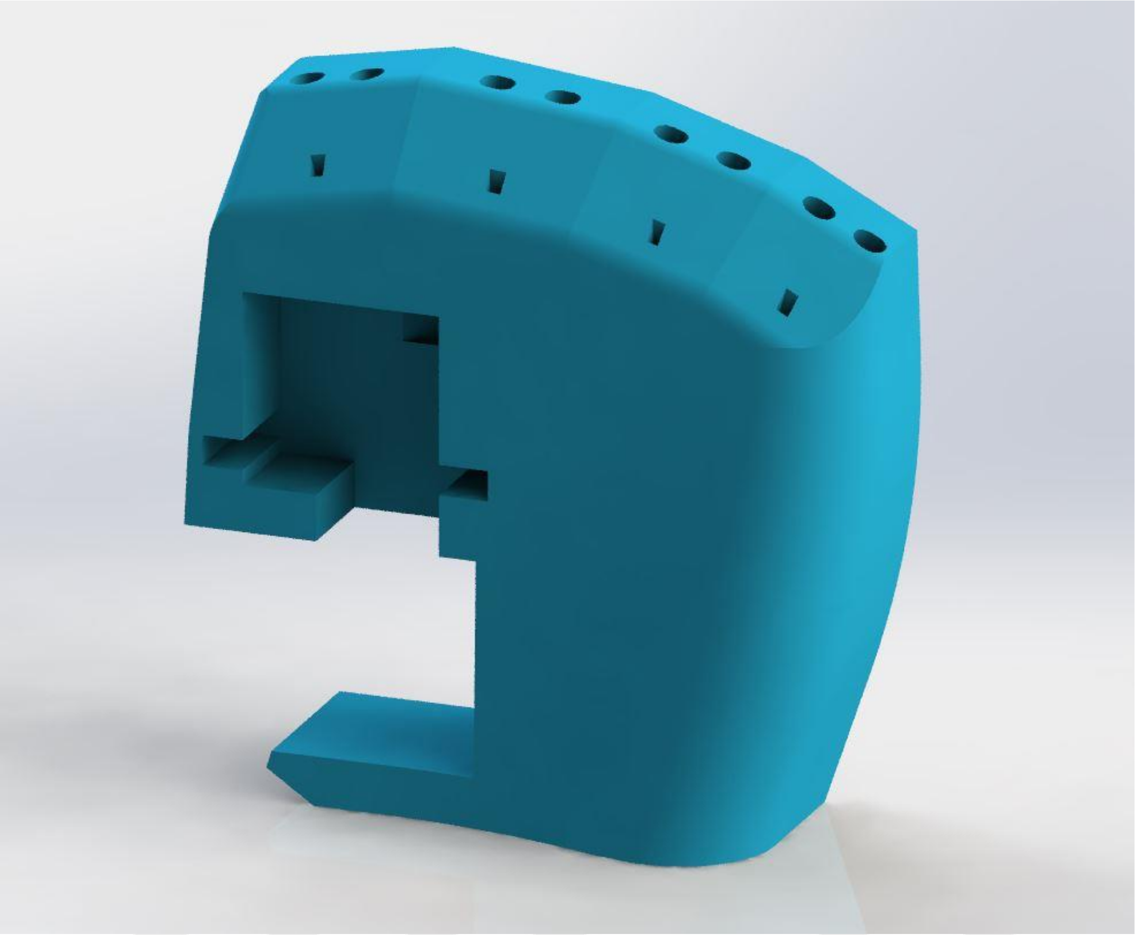


Figure 5.2. Robotic arm hand

5.1.3. Mini Servo – Thumb Connector

This part not only connects the thumb and the mini servo, but also connects the thumb to the hand. It has a mini servo shrink fit slot on the bottom and thumb joints on the front. This piece and thumb are connected to each other by means of fishing line and elastic cord. This part is shown in figure 5.3.

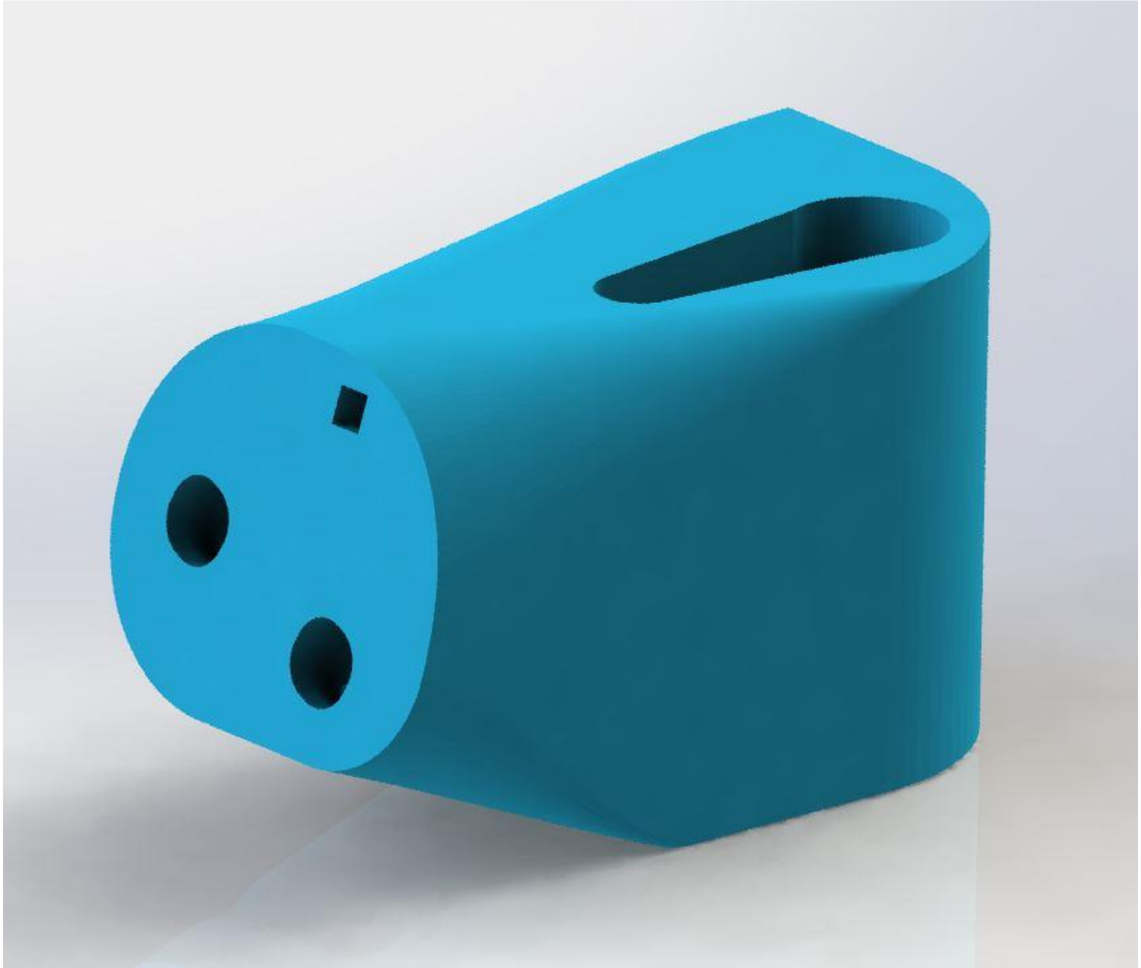


Figure 5.3. Mini servo – thumb connector

5.1.4. Wrist Connector

With the wrist connector, the hand and arm of the robotic arm are connected to each other. On the upper part of the connection piece, there are threads that will fit into the shrink fit sockets under the hand. In addition, there are slots for the fishing line threads that will enable the servo motors to move the fingers and for the connecting cables of the mini servo motor. The wrist connector is shown in figure 5.4.

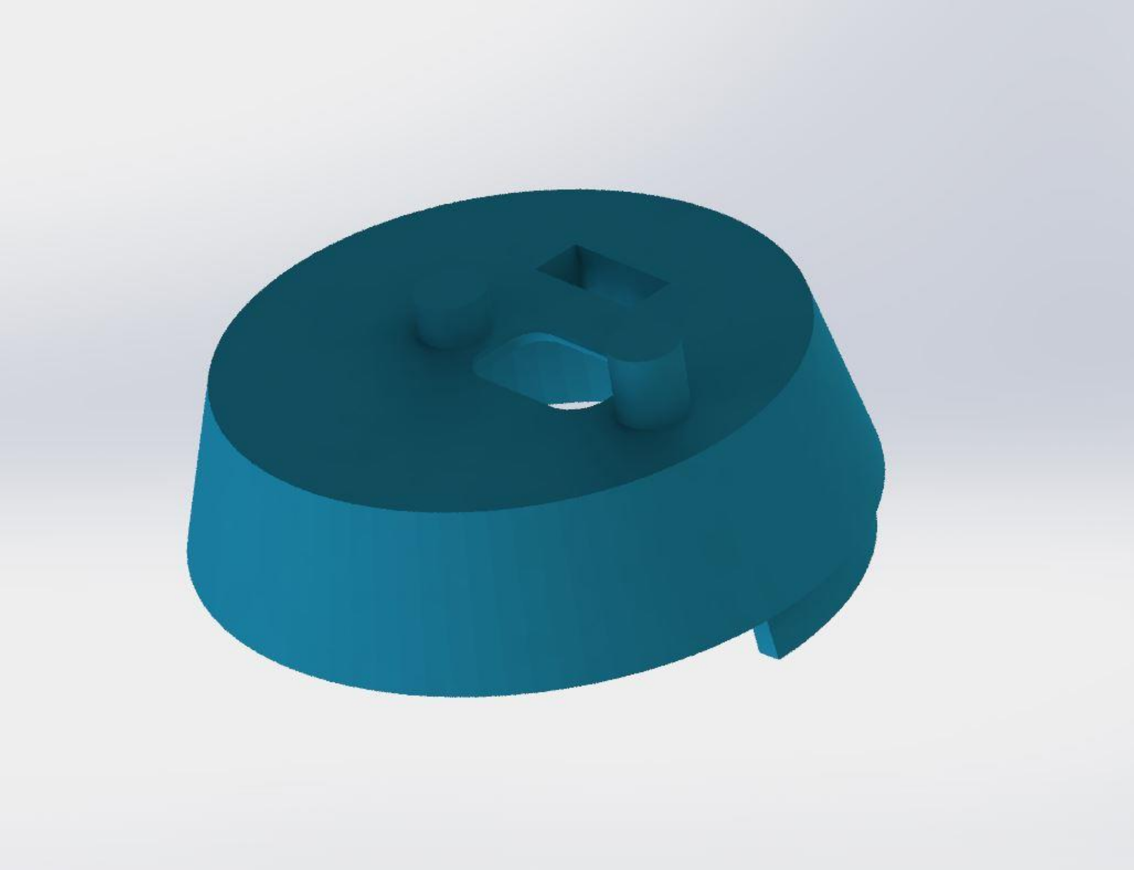


Figure 5.4. Wrist connector

5.1.5. Robotic Arm Body

This part forms the main body of the robotic arm. It connects to the wrist connector part from the upper part and to the arm rotation slot from the lower part. There is a placement area and fixing slots for the servo motor bed to be used to place the servo motors that perform the movements of the fingers on the body. Thanks to this fixation area, the fixation and storage of the servo motors is provided. There is also a cover that can be attached and removed from the main body. In this way, changes to servo motors or servo motor - finger connections are facilitated. The robotic arm body is shown in figure 5.5.

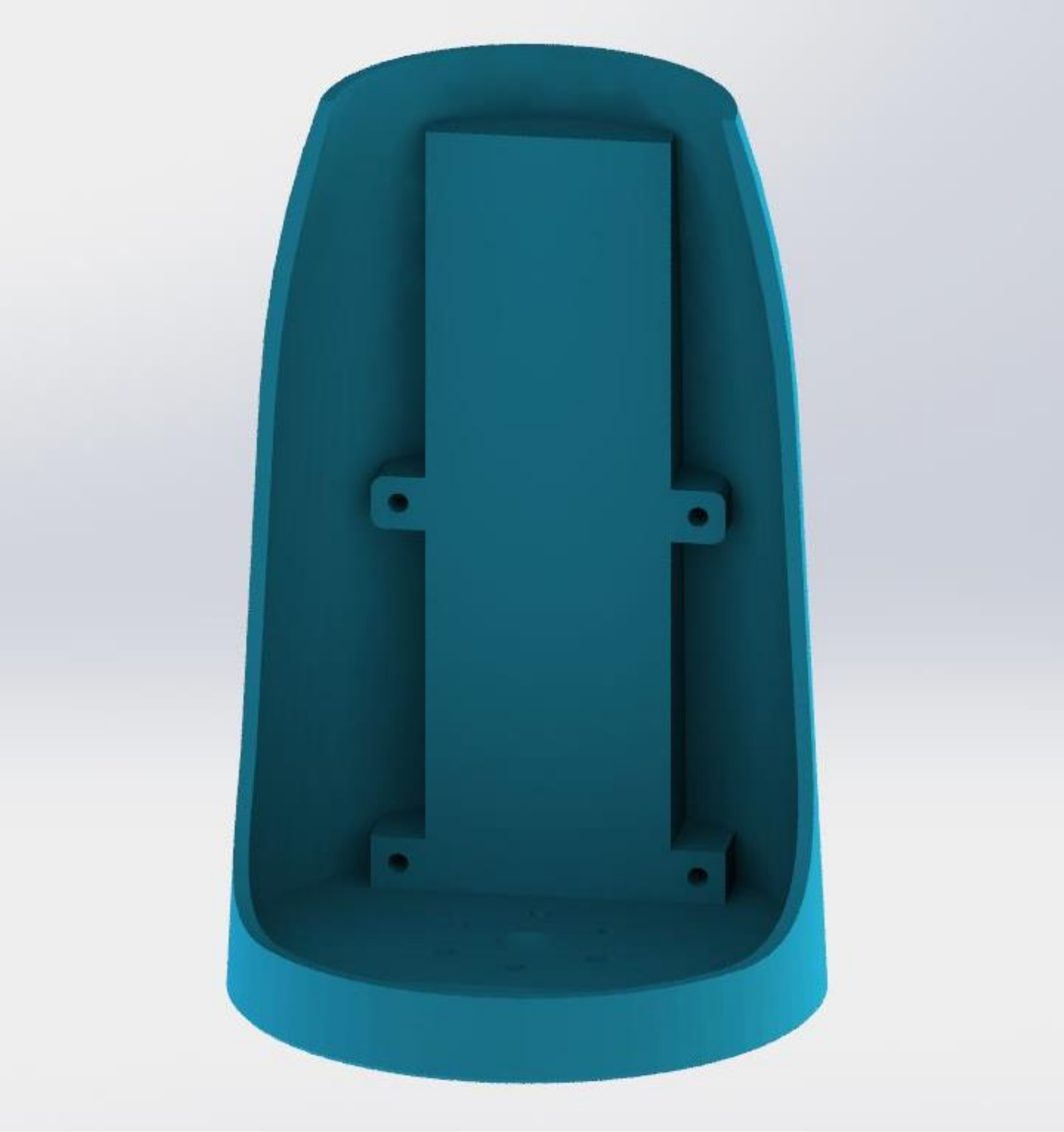


Figure 5.5. Robotic arm body

5.1.6. Servo Motor Bearing

Servo motor bearing is a part on which four servo motors can be placed and fixed to the arm body. Thanks to this part, the servo motors that will be used to move the fingers are fixed and more stable and accurate operation is ensured. Servo motors are placed to form two columns and two servo motors on top of each other. Four servos to be placed on this part control a total of five fingers. A servo motor is used to move the thumb, a servo motor is used to move the index finger, a servo motor is used to move the middle finger, and a servo is used to move the ring and pinky fingers. Fishing line is used for servo motors to move the fingers. Fishing lines for thumb, index finger and middle finger are connected to separate servo motors. The fishing line used for the ring finger and pinky fingers is connected to a single servo motor. During the movement of the servos, there are holes for the fishing line at the end of the servo motor bed so that the fishing lines are in an order and not mixed with each other. Thanks to these holes, the fishing line used for each finger is passed through these holes, preventing them from mixing with each other and preventing each other's movements. Servo motor bearing is shown in figure 5.6.

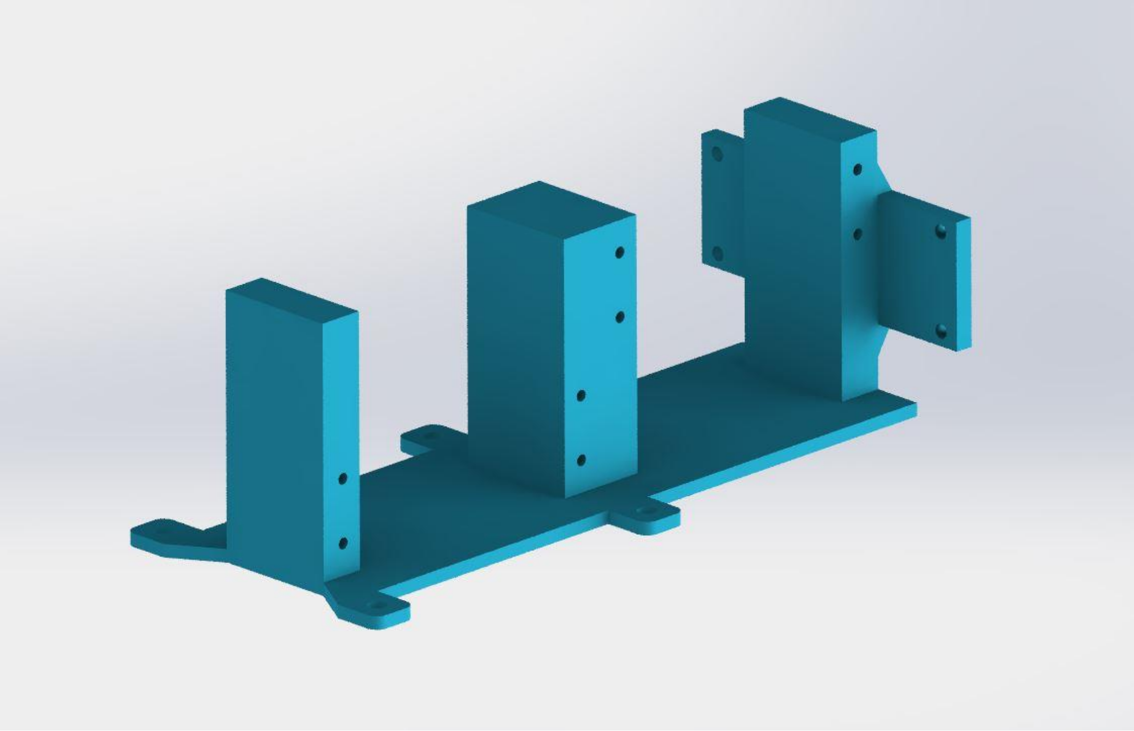


Figure 5.6. Servo motor bearing

5.1.7. Arm Rotation Part

The arm rotation piece is used to rotate the arm main body about the vertical axis. This part is connected to the main body of the arm from the upper part. This part is designed to be fitted with a servo motor. The servo motor placed in the part is fixed and it is provided to rotate the main body of the arm. Thanks to the servo motor, the robotic arm can rotate 360 degrees on the vertical axis. In addition, there are holes where the part connects with the arm main body and on the side for the servo motor cables to pass. Thanks to these holes on the arm rotation part, the servo motor cables are carried to the microcontrollers. The arm rotation part is shown in figure 5.7.

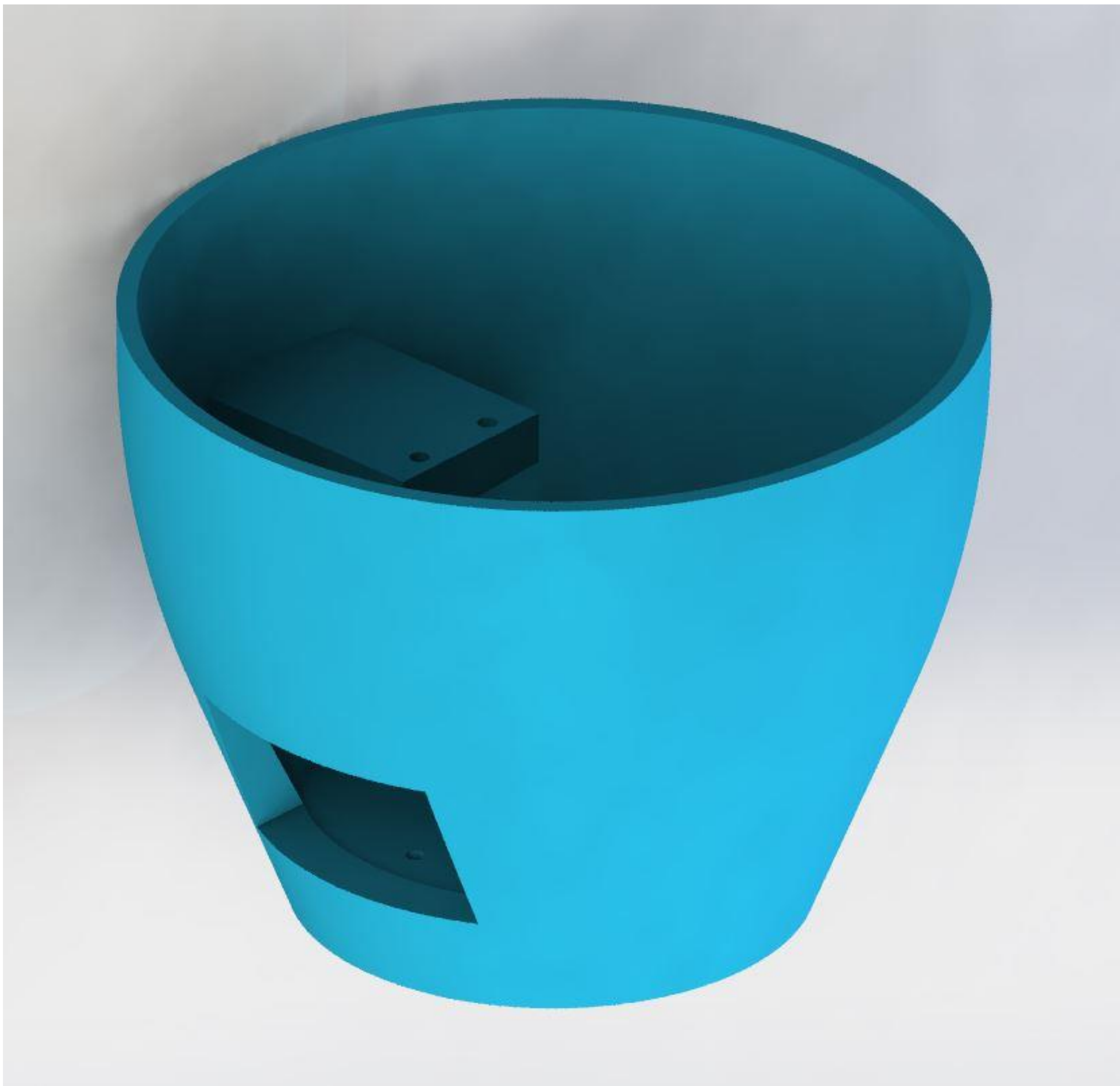


Figure 5.7. Arm rotation part

5.1.8. Servo Horns

Servo horns are connecting parts that are fixed on the gears of servo motors, helping to convert rotational motion into linear motion. They use the rotational movements of servo motors to achieve the movement that allows the fingers to open and close. Servo horns are used for the thumb, index finger and common finger. Fishing line from the fingers is placed in the gaps on these parts and knotted. One servo horn is used for the ring finger and little finger, and the fishing line from both fingers is fixed on a single servo horn. The servo horns fixed to the servo motors are shown in figure 5.8 in order for the fishing line to gain movement.

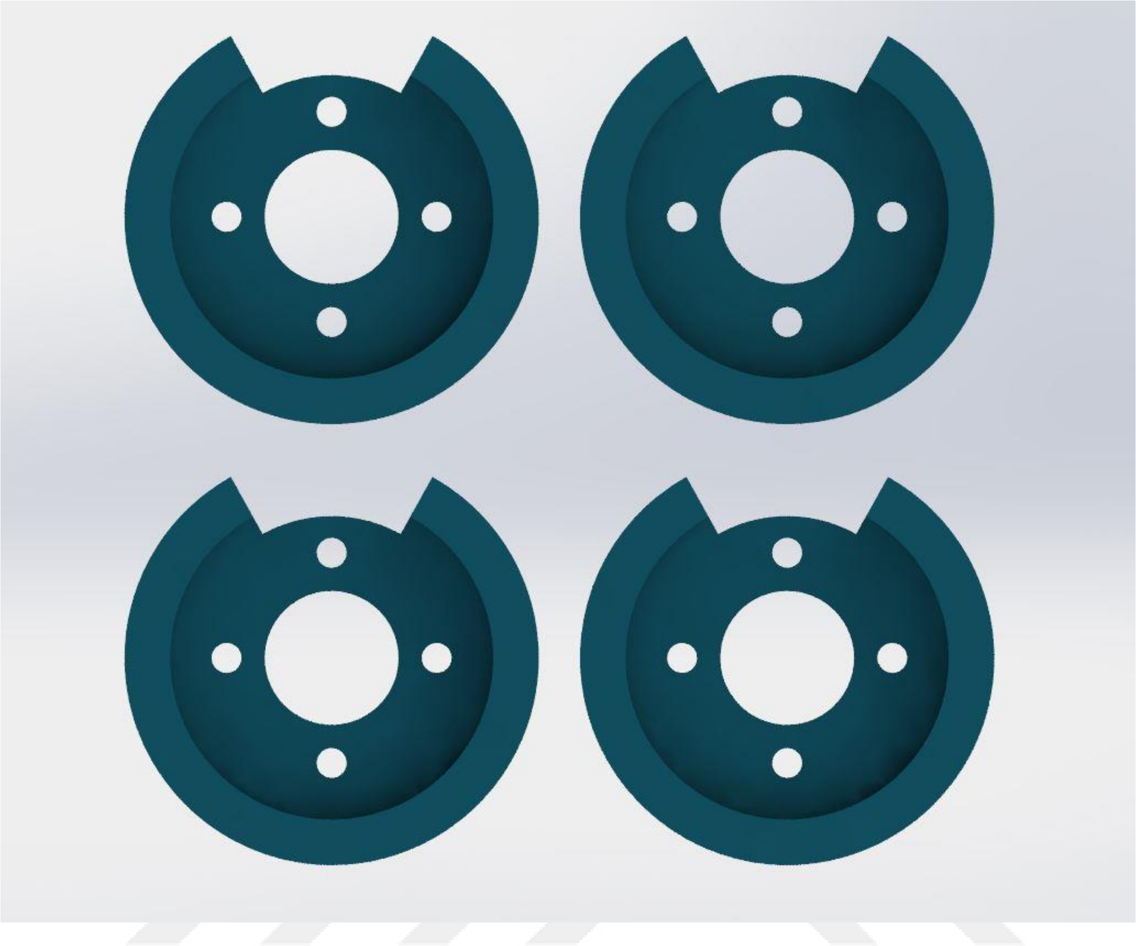


Figure 5.8. Servo horns

5.1.9. Elastic String

Elastic string is used to hold the fingers together and to give flexibility to the fingers. These elastic strings are connected to each other by passing through the parts forming the fingers. In this way, both the unusual movements of the finger are prevented, and flexibility is provided to the fingers so that they can work at different angles.

5.1.10. Fishing Line

Fishing lines are used to make the opening and closing movements of the fingers. The fishing line threads are passed through the fingers and connected to the servo motor horns. In this way, depending on the movements of the servo motors, the fingers are given the ability to move. The flexibility, thickness and durability of the strings used are very important for the fingers to move at the desired speed and angle.

5.2. Electronical Components

In order for the robotic arm to move physically, servo motors are needed, and these motors need to be powered and digitally controlled. Control scenarios and control commands of servo motors are generated by NF-BCI. In order for these control commands produced by NF-BCI to move the servo motors, a connection line must be established between the servo motors and these control commands. This connection line can be set up with microcontrollers and servo driver shields. In this study, microcontrollers and servo motor drivers were used to establish this connection line. These parts used are explained in detail below.

5.2.1. Microcontroller

Microcontrollers are an external device built on an integrated circuit (IC), with CPU (Central Processing Unit), RAM (Random Access Memory), ROM (Ready Only Memory), I/O ports, serial and parallel ports, counters and converters. They are computers that store data in its memory, compile it and get output as a result. Microcontrollers can successfully implement real-time applications. Because of these features, microcontrollers can be used in mechatronics, electromechanics, etc. They act as brains in systems.

In this study, the control signals generated after the control scenario selected on the NF-BCI are transmitted to the microcontroller. In order to control the robotic arm, the control cables of the servo motors are connected to the pins of the microcontroller. In this way, a connection is established between NF-BCI and servo motors.

In this study, Arduino Uno R3 microcontroller was chosen as the microcontroller. Arduino is a microcontroller platform that can be programmed via computers. Thanks to the processors and electronic connections on the Arduino microcontrollers, many different hardware can be controlled and many different applications can be realized. Arduino IDE program is used externally to program Arduino microcontrollers. Arduino microcontrollers can also be programmed in different programming languages through various libraries. The Arduino microcontroller we used in our study is controlled via NF-BCI. Arduino Uno R3 microcontroller is shown in figure 5.9.

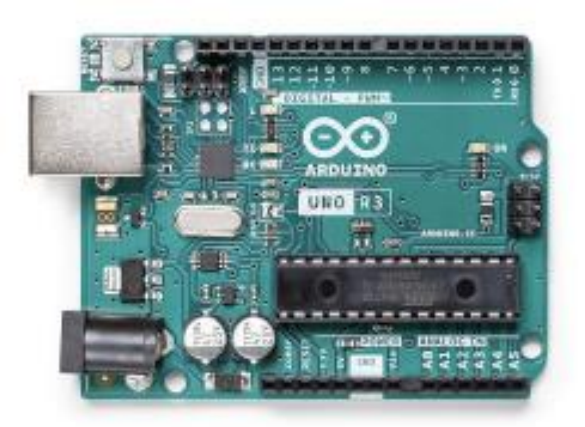


Figure 5.9. Arduino Uno R3 Microcontroller

Arduino Uno has ATmega328P processor. This microcontroller has fourteen digital I/O (input-output) pins, six of which can be used as PWM (Pulse Width Modulation) outputs. In addition, there are six analog inputs, 16 MHz ceramic resonator, power jack, ICSP reader, USB connection and reset button. Technical specifications of Arduino Uno microcontroller are given in table 5.1.

Table 5.1. Arduino Uno R3 Technical specifications

Board	Name	Arduino UNO R3
	SKU	A000066
Microcontroller	ATmega328P	
USB connector	Built-in LED Pin	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	UART	Yes
	I2C	Yes

	SPI	Yes
Power	I/O Voltage	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Power Supply Connector	Barrel Plug
Clock speed	Main Processor	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	Weight	25 g
	Width	53.4 mm
	Length	68.6 mm

However, the pinouts of the Arduino Uno board are given in figure 5.10, table 5.2 and table 5.3.

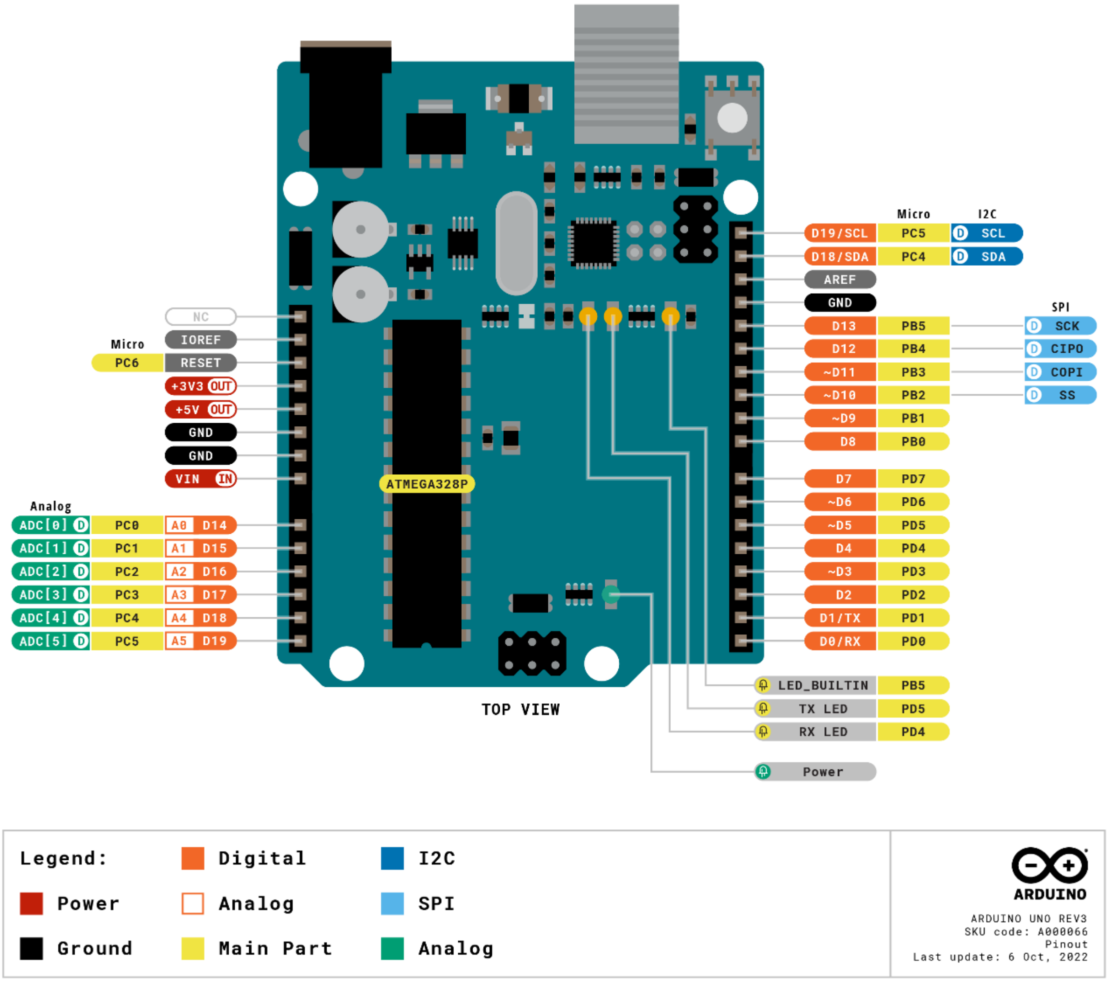


Figure 5.10. Arduino Uno R3 Pinouts

Table 5.2. Analog pinouts

JANALOG			
Pin	**Function**	**Type**	**Description**
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset

4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Analog input 0 /GPIO
9	A0	Analog/GPIO	Analog input 1 /GPIO
10	A1	Analog/GPIO	Analog input 2 /GPIO
11	A2	Analog/GPIO	Analog input 3 /GPIO
12	A3	Analog/GPIO	Analog input 4 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

Table 5.3. Digital pinouts

JDIGITAL			
Pin	**Function**	**Type**	**Description**
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	Main Out Secondary In
13	MISO	Digital	Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground

16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

5.2.2. Servo Motor Shield

There are electronic boards developed for microcontrollers that are designed and produced to perform many different tasks easier and faster. These cards are usually developed to fulfill a specific purpose. They are also used to add additional features that the microcontroller does not have. In this study, a customized electronic card that can control servo motors is used in order to control servo motors more easily and efficiently. This electronic board is the PCA9685 16 Channel I2C PWM/Servo Driver Shield card, which can control 16 different servos and has an external power input. The servo driver board used is shown in figure 5.11.

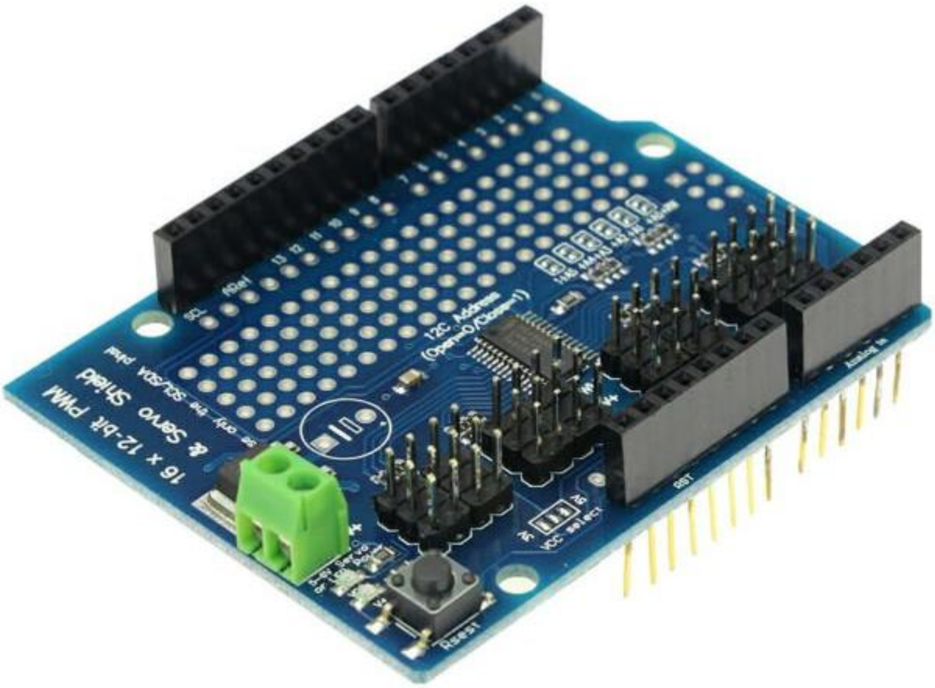


Figure 5.11. CA9685 16 Channel I2C PWM/Servo Driver Shield

One of the purposes of using this motor driver is that there are insufficient pins on the Arduino Uno and there will not be enough pins for other tasks. Thanks to this board, both the pins of the Arduino board are not occupied, and the servo motor connections can be made faster and more effectively. Another reason why this card is preferred is that it has an external power input. The Arduino board cannot provide the power needed by the robotic arm internally. Thanks to this card, the power needed can be added to the card externally, and thus the total power needed by the servo motors can be easily met. The servo motor driver board is connected to the inputs on the Arduino board thanks to its pins. An I2C connection is established between the servo motor driver board and the Arduino board, and thus 16 PWM outputs are obtained using only 2 pins (SDA and SCL).

5.2.3. Servos

In this study, a total of six servo motors were used to move the fingers of the robotic arm, to ensure that the thumb moves towards the palm and to rotate the arm around the vertical axis. Four of these were used to open and close the fingers, one to rotate the arm, and one to move the thumb towards the palm. A total of six servo motors are five of the same type and one of them is a different type of servo motor. Five servo motors are used to open and close the fingers and rotate the arm on the vertical axis. These servo motors are MG996R 13 kg Servo Motors. This servo motor is shown in figure 5.12.

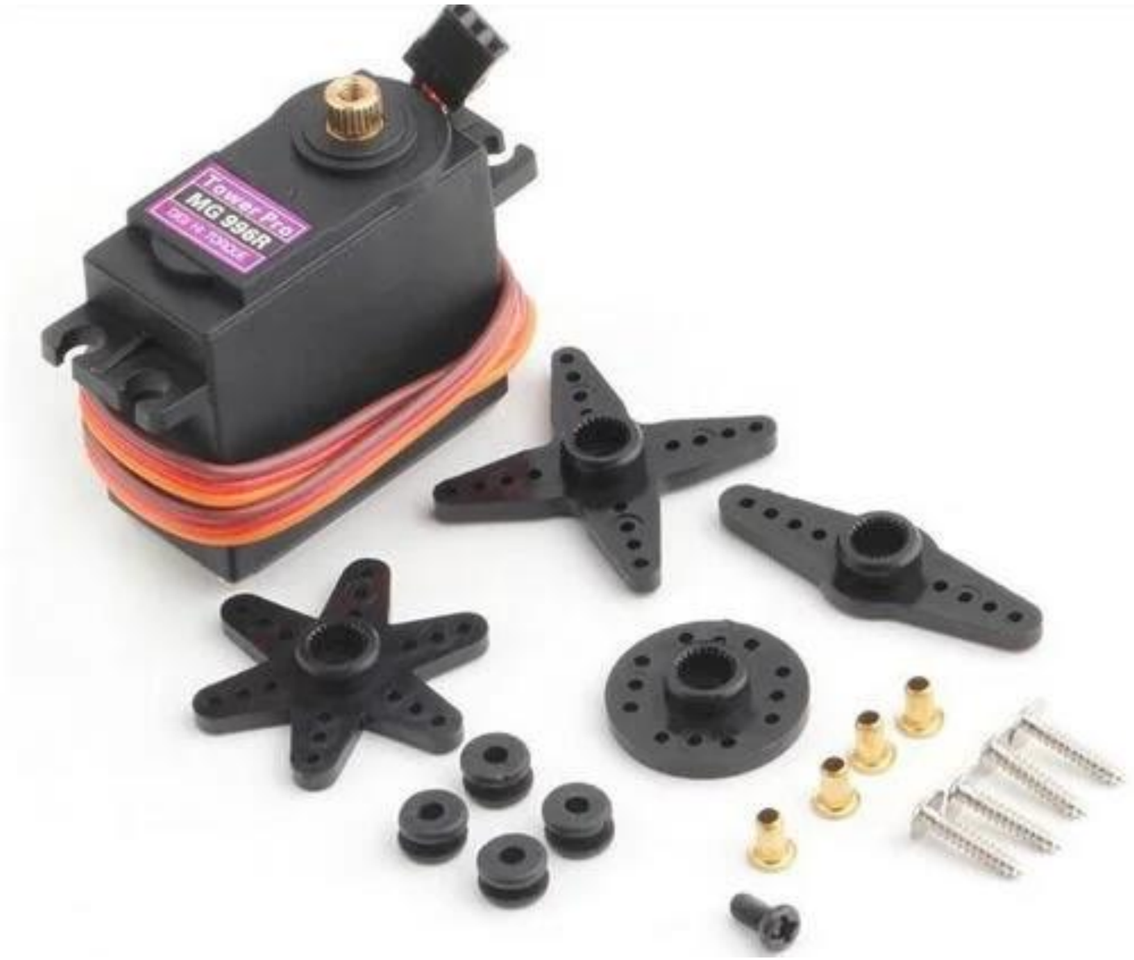


Figure 5.12. MG996R 13 kg servo motor

Technical specifications of MG996R 13 kg servo motor are given in table 5.4.

Table 5.4. Technical specifications of MG996R 13 kg servo motor

Property	**Description**
Brown Wire	Ground wire connected to the ground of system
Red Wire	Powers the motor typically +5V is used
Orange Wire	PWM signal is given in through this wire to drive the motor
Operating Voltage	+5V
Current	2.5A (6V)
Stall Torque	9.4 kg/cm (at 4.8V)
Maximum Stall Torque	11 kg/cm (6V)
Operating speed	0.17 s/60°
Gear Type	Metal
Rotation	0°-180°
Weight	55g

SG90RC Mini Servo Motor is used to enable the thumb to close and open towards the palm. This servo motor is shown in figure 5.13.

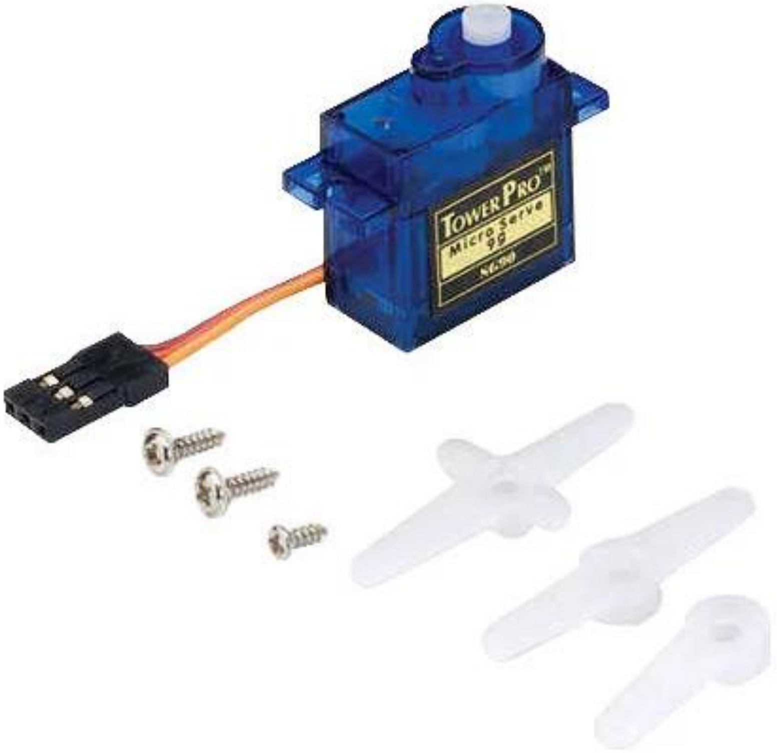


Figure 5.13. SG90RC Mini Servo Motor

Technical specifications of SG90RC mini servo motor are given in table 5.5.

Table 5.5. Technical specifications of SG90RC mini servo motor

Property	**Description**
Brown Wire	Ground wire connected to the ground of system
Red Wire	Powers the motor typically +5V is used
Orange Wire	PWM signal is given in through this wire to drive the motor
Operating Voltage	4.8-6.0V

Current	2.5A (6V)
Stall Torque	0-5 kg/cm
Maximum Stall Torque	@6V: 1.8 kg.cm
Operating speed	0.06 - 0.12 sn/60°
Gear Type	Plastic
Rotation	0°-180°
Weight	9 g
Dimensions	23.1 x 12.2 x 29 mm

5.3. Connections and Assembly

Different cables are used to establish communication between computer, microcontroller, servo motor shield and servo motors. USB connection has been established between NF-BCI (computer) and microcontroller. The connection between the microcontroller and the servo motor driver is established by interlocking the input pins on the microcontroller and the output pins on the servo motor driver. Cable connections have been established between the servo motor driver and the servo motors via PWM outputs. While the system is running, all these connections are protected, and they all work simultaneously.

All these mechanical, electronical and connection parts mentioned above are brought together and the system is brought together to work in real time. While the system is running, all the mentioned parts are actively used, and all connections are protected. The assembled and operational state of the system is shown in figure 5.14.

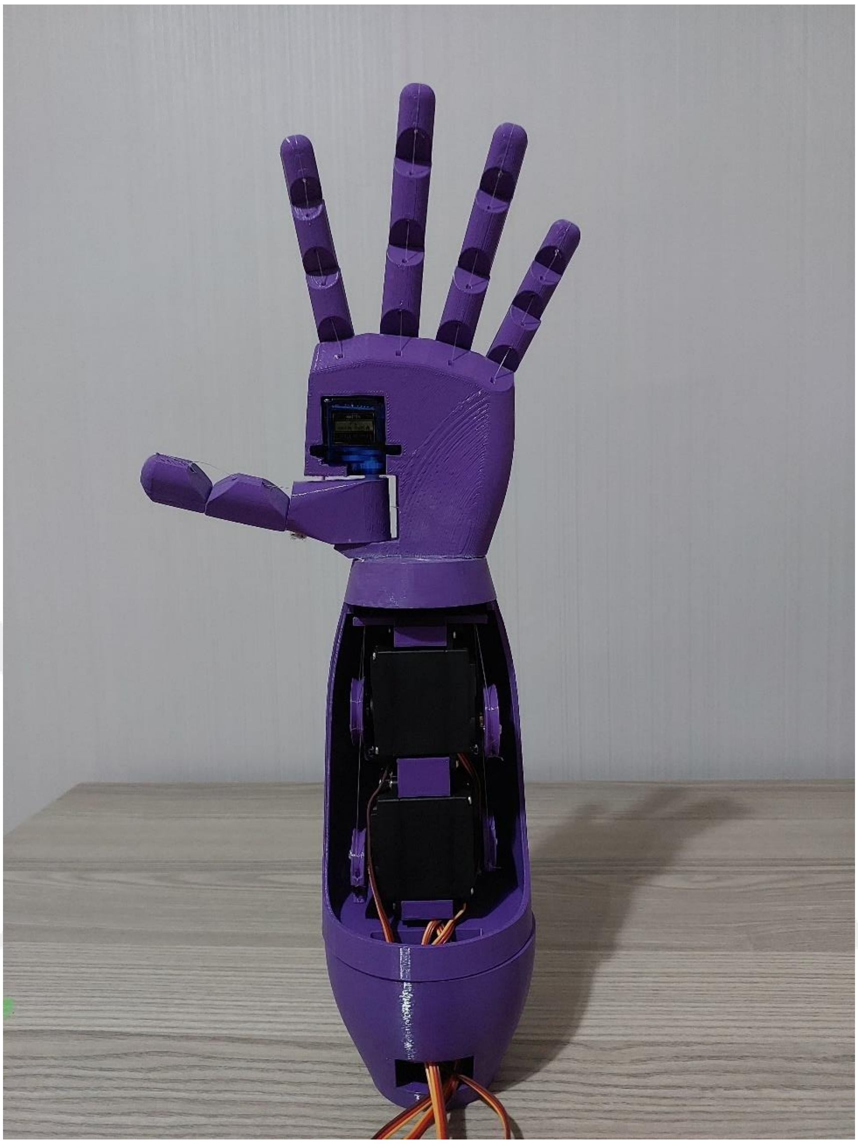


Figure 5.14. Assembly of the robotic arm

5.4. Motion Scenarios

The movements that the robotic arm can do, such as opening and closing the palm, sequentially opening and closing the fingers, moving the thumb and rotating the robot arm, are shown in figures 5.15 – 5.18.

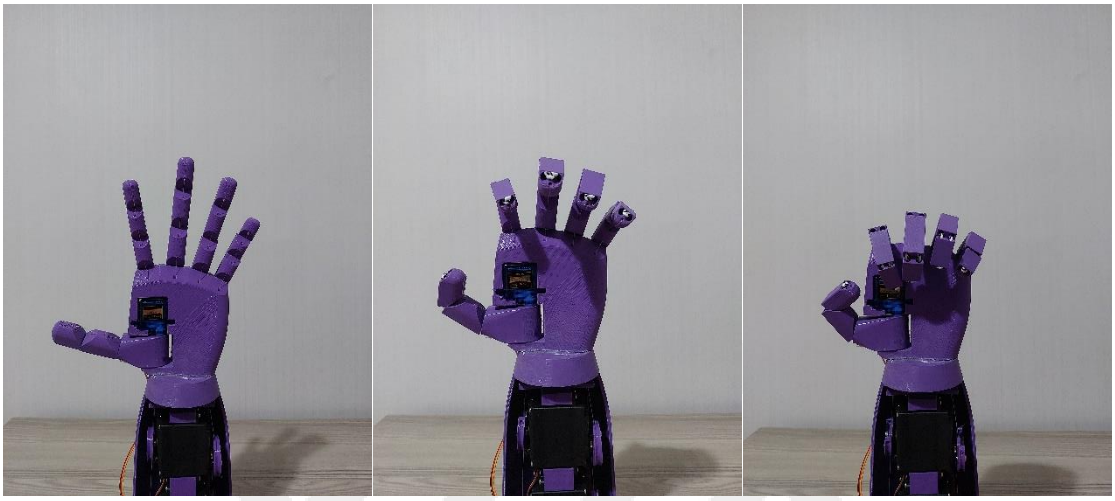


Figure 5.15. Opening and closing the palm

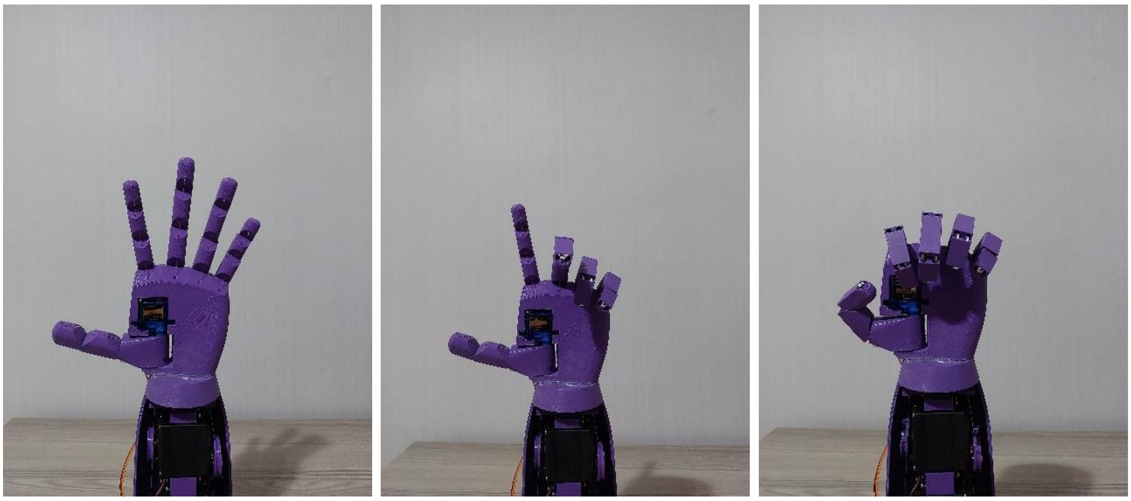


Figure 5.16. Sequentially opening and closing the fingers

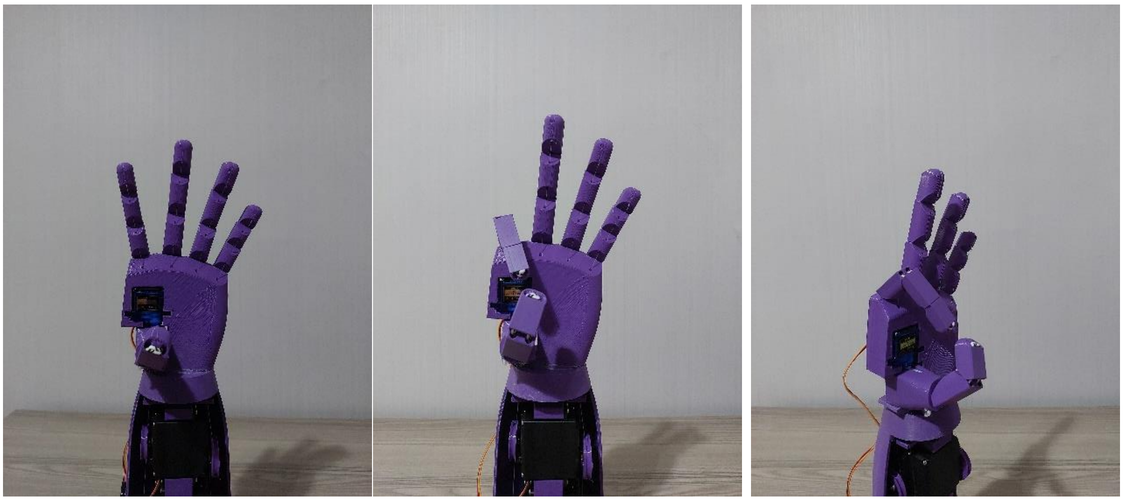


Figure 5.17. Moving the thumb

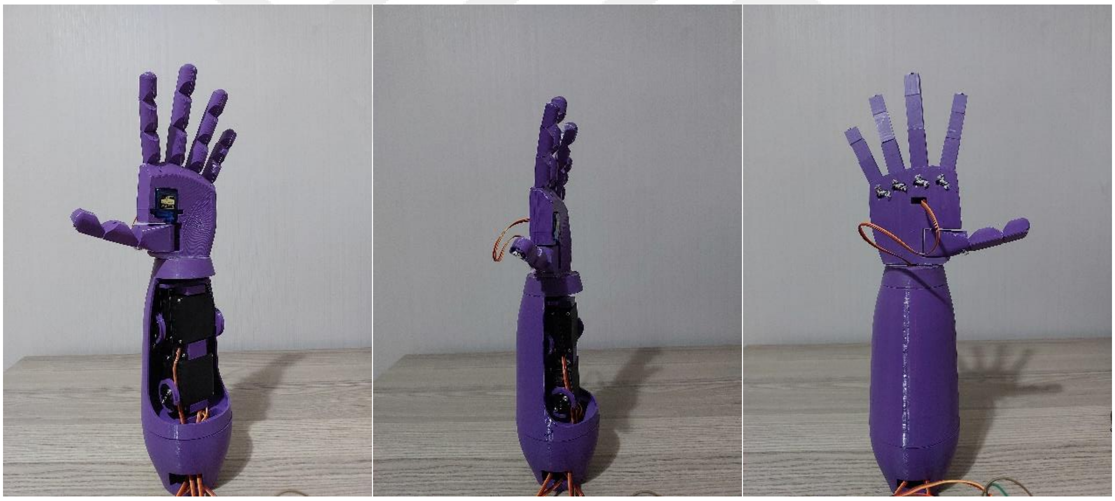


Figure 5.18. Rotating the robotic arm

6. EXPERIMENTAL RESULTS

6.1. NF-EEG Experimental Setup

In this study, python was used as development language and to develop deep learning algorithms TensorFlow and Keras libraries were used. 12 core 3.80 GHz processor, 32 GB ram as memory and GeForce RTX 2080 Super as GPU were used. As mentioned before, there are separate training and test sets in the datasets. All the training processes were done only by using training sets, and then the trained model was tested with the unseen test sets. Training sets were separated as 80% training and 20% validation sets, and the model was trained by performing 5-fold cross validation. Total number of samples for training, validation and test sets are shown in table 6.1.

In order to prevent overfitting, the validation loss value was used as a variable and early stopping was applied and sparse categorical cross entropy was used as loss function. If the validation loss value did not decrease during the repetitive 100 epochs, the model with the best validation loss value was selected among the last 100 epochs, and evaluation was made on the test set with this model. Evaluation process was implemented separately for each subject's data to measure each subject's accuracy. Many different model combinations including different layer numbers and different layer types were examined to determine the model architecture, and many different values were tested for many hyperparameters by applying grid search to determine the model hyperparameters, and the hyperparameters that gave the best classification success were selected from this hyperparameter pool. Grid search lists and selected hyperparameters are shown in figure 6.1.

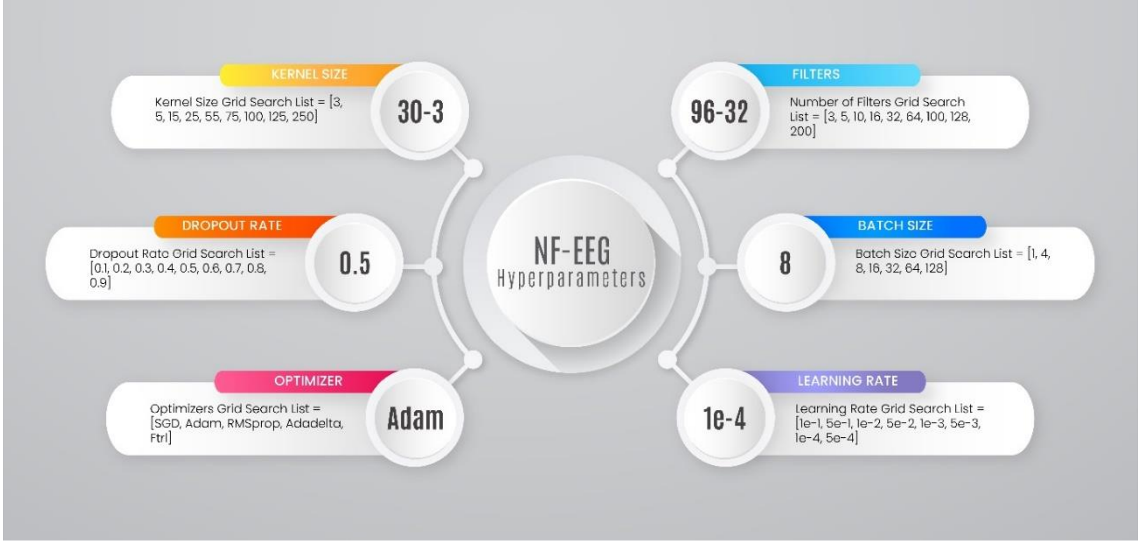

KERNEL SIZE
Kernel Size Grid Search List = [3, 5, 15, 25, 55, 75, 100, 125, 250]
30-3
FILTERS
96-32
Number of Filters Grid Search List = [3, 5, 10, 16, 32, 64, 100, 128, 200]
DROPOUT RATE
Dropout Rate Grid Search List = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
0.5
NF-EEG
Hyperparameters
BATCH SIZE
8
Batch Size Grid Search List = [1, 4, 8, 16, 32, 64, 128]
OPTIMIZER
Optimizers Grid Search List = [SGD, Adam, RMSprop, Adadelta, Ftrl]
Adam
LEARNING RATE
1e-4
Learning Rate Grid Search List = [1e-1, 5e-1, 1e-2, 5e-2, 1e-3, 5e-3, 1e-4, 5e-4]

Figure 6.1. NF-EEG hyperparameters and grid search lists

In this section, total of three main data groups as two-class (left hand, right hand) BCI Competition IV 2A data, two-class (left hand, right hand) BCI Competition IV 2B data and four-class (left hand, right hand, both feet and tongue) BCI Competition IV 2A data were trained and classified separately. The developed NF-EEG model and state-of-the-art CNN-based models and other deep learning and machine learning methods were compared. In addition, the results before data augmentation were compared with the results after data augmentation, and the results before the input reshaping with the results after the input reshaping were also compared. Also, different data augmentation methods were compared with each other. The classification results of the compared methods were taken as the values stated in the researchers' own papers for the models they trained and classified on the same datasets. The highly cited and recommended EEG-Net [169], ShallowConvNet [174] and DeepConvNet [174] EEG classification models were also adapted to existing datasets and retrained with reference to the model structures and parameters specified in their papers for a fair comparison and these classification results were used for comparison. Considering the information mentioned in the previous sections, NF-EEG, which was developed using raw data without any signal preprocessing, achieved the best classification accuracy comparing to all state-of-the-art models. Classification accuracy comparison tables and graphs are explained in detail in the following sections.

Table 6.1. Number of samples for training and test sets

		Number of Samples		
Dataset	**Classes**	Training Set (%80)	Validation Set (%20)	Test Set (Unseen Data)
2 Class BCI-IV-2A	Left Hand, Right Hand	930	233	1183
2 Class BCI-IV-2A w/ Augmentation	Left Hand, Right Hand	6513	1628	1183
2 Class BCI-IV-2B	Left Hand, Right Hand	2421	605	2241
2 Class BCI-IV-2B w/ Augmentation	Left Hand, Right Hand	16946	4236	2241
4 Class BCI-IV-2A	Left Hand, Right Hand, Foot, Tongue	1862	466	2368
4 Class BCI-IV-2A w/ Augmentation	Left Hand, Right Hand, Foot, Tongue	13037	3259	2368

6.2. Performance of The Input Reshaping for NF-EEG

Input reshaping operations have provided great and significant improvements in classification accuracy. Among the first three subjects with the highest results and highest increase in BCI-IV-2A dataset, the classification accuracy increase rate was 62.83% for subject 2, 54.64% for subject 8, 51.78% for subject 1 and 44.67% on average. BCI-IV-2A input reshaping results are shown in figure 6.2. For the BCI-IV-2B dataset, the classification accuracy rate increased by 36.97% for subject 2, 35.08% for subject 3, 23.95% for subject 5 and an average of 17.01%. BCI-IV-2B input reshaping results are shown in figure 6.3.

Also, as mentioned earlier, this input reshaping process has reduced the number of complex computations for the layers to wander along the 2D data axes of the kernels and considerably shortened the time required for each epoch. A shortening of 2.57 times for the 3-channel 2B dataset and 10.75 times for the 22-channel 2A dataset has occurred. As expected, as the number of channels increases, the gain in the 22-channel dataset was higher than in the 3-channel dataset, as the kernels would have to scan a larger area. With this striking change, it is clearly seen that the time to be gained during the training of models that has multi-channel and multiple epochs is quite high. The times taken for an epoch are shown in table 5.2.

Table 6.2. Epoch time comparison of models before and after input reshaping

Dataset	BCI-IV-2A		BCI-IV-2B	
Input Shape	L x Nc x 1	L x 1 x Nc	L x Nc x 1	L x 1 x Nc
Epoch Time	43 seconds	4 seconds	18 seconds	7 seconds

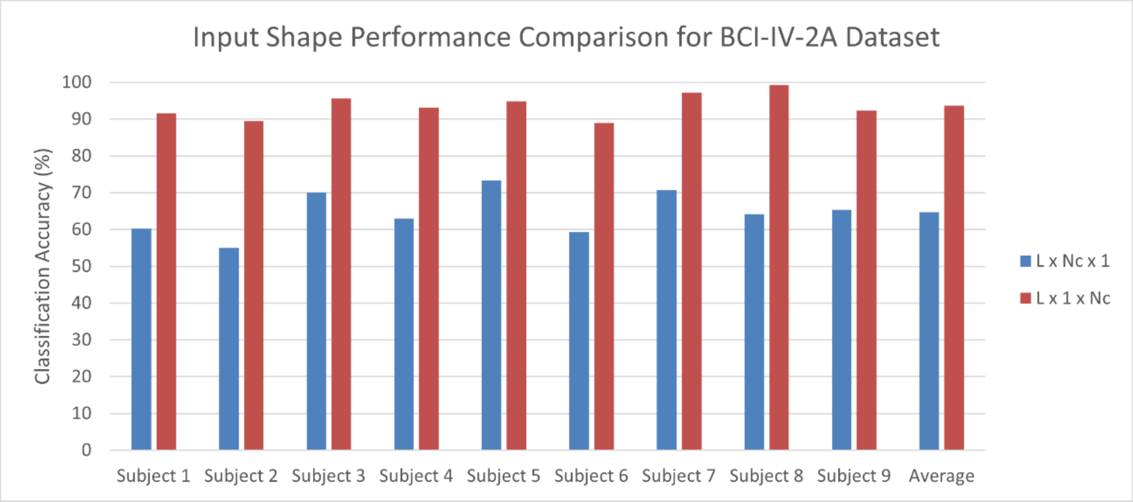


Figure 6.2. Classification performance of different input shapes for BCI-IV-2A

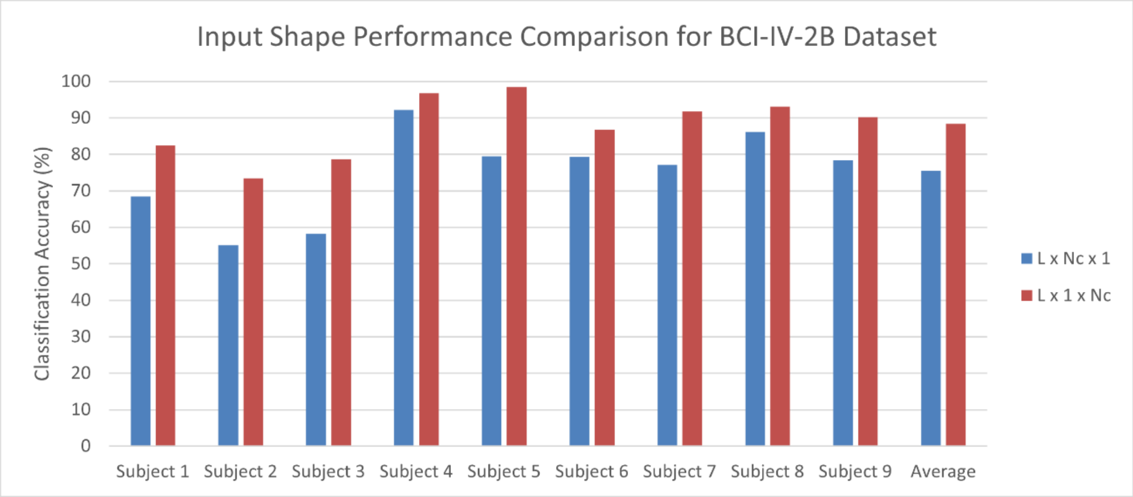


Figure 6.3. Classification performance of different input shapes for BCI-IV-2B

6.3. Performance of The Data Augmentation Methods for NF-EEG

As a result of the jittering and scaling processes applied to the datasets, among the first three subjects with the highest results and highest increase in BCI-IV-2A dataset, the classification accuracy increase rate was 15.39% for subject 9, 7.86% for subject 6, 7.63% for subject 2, and 4.42% on average. BCI-IV-2A data augmentation results are shown in figure 6.4.

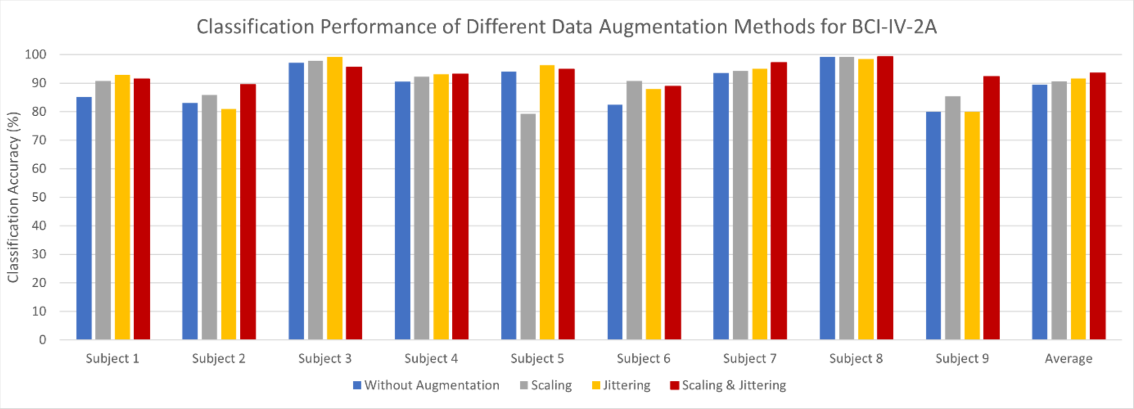


Figure 6.4. Classification performance of different data augmentation methods for BCI-IV-2A

For the BCI-IV-2B dataset, the classification accuracy rate increased by 12.43% for subject 3, 5.29% for subject 9, 4.30% for subject 6 and an average of 2.59%. BCI-IV-2B data augmentation results are shown in figure 6.5.

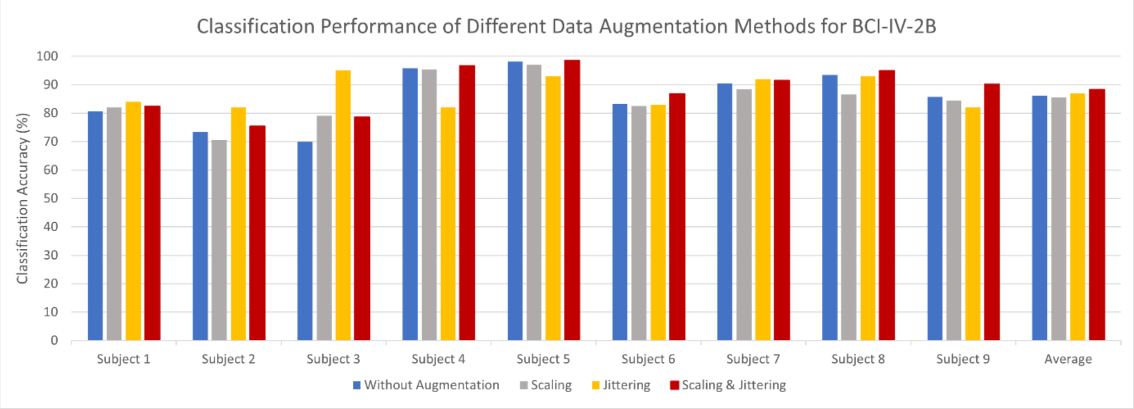


Figure 6.5. Classification performance of different data augmentation methods for BCI-IV-2B

Training and validation accuracies and losses were recorded during training of datasets. These records include information according to number of epochs. These records include information before and after data augmentation and their graphs are shown in figure 6.6, figure 6.7, figure 6.8 and figure 6.9 respectively.

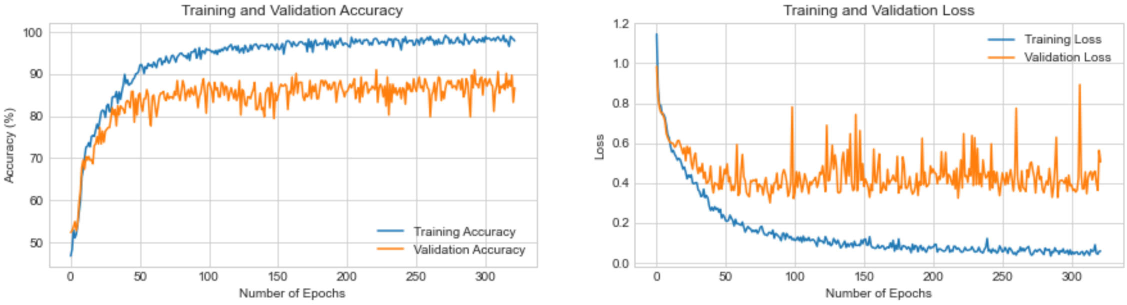


Figure 6.6. Training and validation accuracies and losses for 2 Class BCI-IV-2A

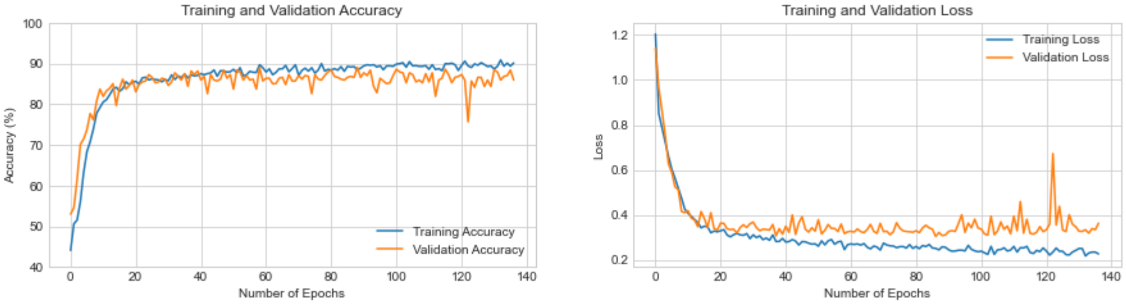


Figure 6.7. Training and validation accuracies and losses for 2 Class BCI-IV-2B

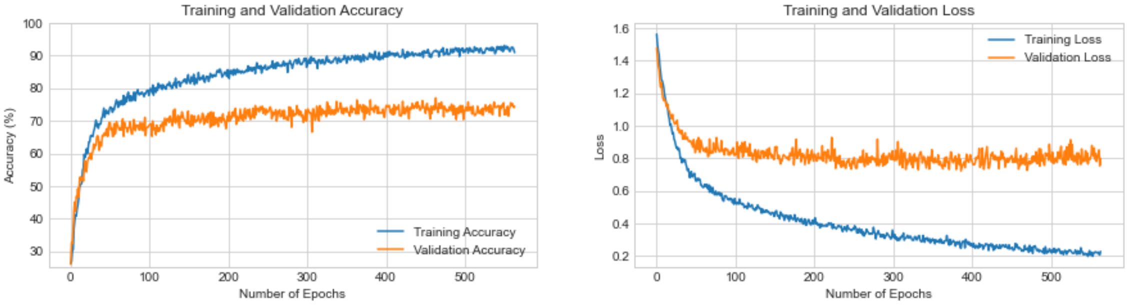
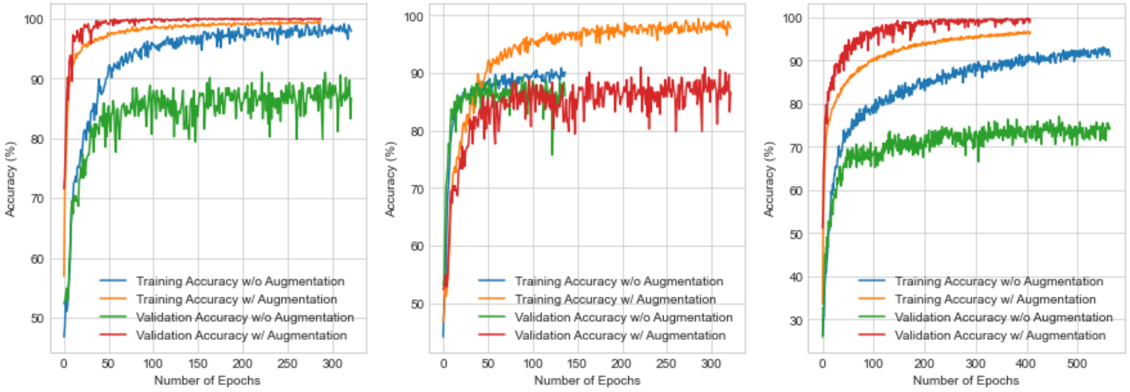


Figure 6.8. Training and validation accuracies and losses for 4 Class BCI-IV-2A



(a) 2 Class BCI-IV-2A (b) 2 Class BCI-IV-2B (c) 4 Class BCI-IV-2A

Figure 6.9. Training and validation accuracies with and without data augmentation

6.4. BCI Competition IV 2A – 2 Class Performance Comparison for NF-EEG

In this section, the proposed method is compared with the methods developed with the 2-class BCI-IV-2A dataset. NF-EEG achieved the best average classification accuracy results among all models compared. Among the first three subjects with the highest results and highest increase in 2-class BCI-IV-2A dataset, the classification accuracy increase rate was 11.41% for subject 2, 5.43% for subject 7, 3.80% for subject 4 and 2.20% on average. This model has a lowest standard deviation as 3.46 among all models compared. BCI-IV-2A 2-class performance comparison results are shown in table 6.3 and in figure 6.10.

Table 6.3. Classification accuracy comparison of state-of-the-art methods for 2-class BCI-IV-2A dataset

Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	AVG
DFFS [159]	63.69	61.97	91.09	61.72	63.41	66.11	59.57	62.84	84.46	68.32
WTRCSP [218]	88.89	54.86	96.53	70.14	65.97	61.81	81.25	95.83	90.97	79.40
ALCSD [219]	90.28	54.17	93.75	64.58	57.64	65.28	65.20	90.97	85.42	73.84
TLCSD [219]	90.28	57.64	95.14	65.97	61.11	65.28	61.11	91.67	86.11	74.92
EMD-MI [220]	66.71	63.90	77.80	63.20	72.20	70.10	64.60	76.40	77.1	70.20
SS-MEMDBF [221]	91.49	60.56	94.16	76.72	58.52	68.52	78.57	97.01	93.85	79.93
EEG-Net [169]	72.92	69.90	74.99	78.72	93.33	82.63	91.29	72.90	72.77	78.75
DeepNet [174]	68.67	68.20	71.34	73.55	93.33	72.44	79.57	67.16	77.38	75.12
ShallowNet [174]	69.38	69.61	77.18	80.45	91.11	76.22	90.29	76.13	78.92	78.84
HS-CNN [182]	90.07	80.28	97.08	89.66	97.04	87.04	92.14	98.51	92.31	91.57
NF-EEG	91.49	89.44	95.62	93.10	94.81	88.89	97.14	99.25	92.31	**93.56**[a]

[a] indicates $p < 0.05$ in Wilcoxon signed-rank test.

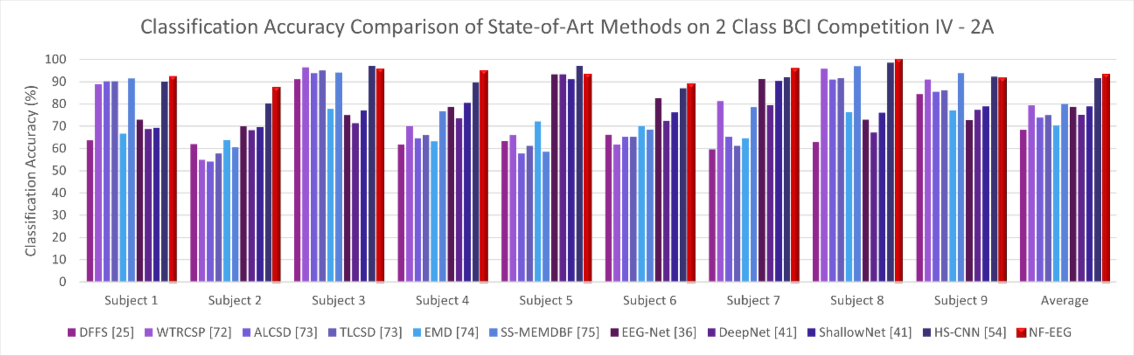


Figure 6.10. Classification accuracy comparison of state-of-the-art methods for 2-class BCI-IV-2A dataset

6.5. BCI Competition IV 2B – 2 Class Performance Comparison for NF-EEG

In this section, the proposed method is compared with the methods developed with the 2-class BCI-IV-2B dataset. NF-EEG achieved the best average classification accuracy results among all models compared. Among the first three subjects with the highest results and highest increase in 2-class BCI-IV-2B dataset, the classification accuracy increase rate was 6.90% for subject 2, 3.25% for subject 9, 2.43% for subject 1 and 1.00% on average. This model has a standard deviation as 8.13. BCI-IV-2B 2-class performance comparison results are shown in table 6.4 and in figure 6.11.

Table 6.4. Classification accuracy comparison of state-of-the-art methods for 2-class BCI-IV-2B dataset

Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	AVG
BSP BCI [222]	77.00	65.00	60.5	96.00	82.00	84.50	75.00	90.50	87.50	80.00
FBCSP [164]	70.00	60.00	61.00	97.50	92.80	81.00	77.50	92.50	87.20	80.00
HCRF [163]	80.00	66.00	53.00	98.50	93.50	89.00	81.50	94.00	90.50	83.00
DFFS [159]	73.20	67.50	63.00	97.40	95.50	86.70	84.70	95.90	92.60	84.10
WPD+SE-IM [161]	84.58	66.25	62.92	95.83	89.17	97.92	82.08	86.25	97.08	84.68
FDBN [172]	81.00	65.00	66.00	98.00	93.00	88.00	82.00	94.00	91.00	84.00
CNN-SAE [158]	76.00	65.80	75.30	95.30	83.00	79.50	74.50	75.30	73.30	77.60
RSMM [223]	72.50	56.43	55.63	97.19	88.44	78.75	77.50	91.88	83.44	77.97
ALCSD [219]	71.90	50.00	52.80	93.40	54.40	73.10	62.20	77.80	75.00	67.90
TLCSD [219]	70.30	50.60	52.80	93.80	63.80	74.10	61.90	83.10	77.20	69.70
EMD-MI [220]	62.80	67.10	98.70	88.40	96.30	75.30	72.20	87.80	85.30	81.60
EEG-Net [169]	71.49	61.22	76.96	93.49	97.07	72.11	92.67	83.04	73.47	80.72
DeepNet [174]	64.47	60.00	73.04	74.59	89.38	71.31	83.62	90.87	59.18	74.16
ShallowNet [174]	66.67	65.31	73.91	80.13	95.24	67.33	90.95	89.13	61.22	76.89
HS-CNN [182]	80.50	70.60	85.60	94.60	98.30	86.60	89.60	95.60	87.40	87.60
NF-EEG	82.46	75.47	78.70	96.74	98.53	86.85	91.61	95.04	90.24	**88.40**[a]

[a] indicates *$p < 0.05$* in Wilcoxon signed-rank test.

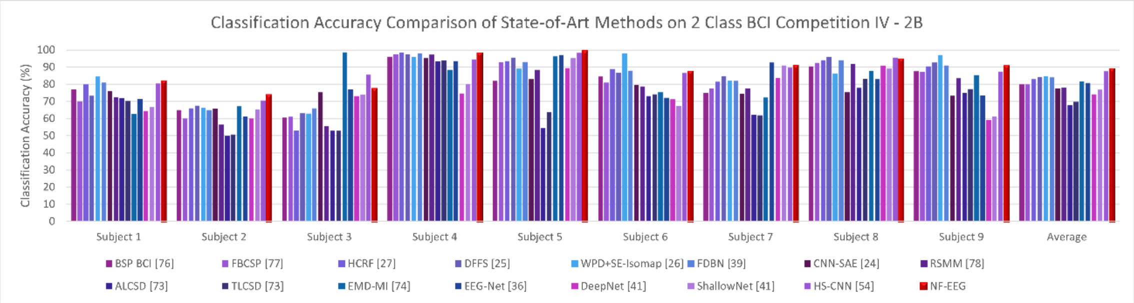


Figure 6.11. Classification accuracy comparison of state-of-the-art methods for 2-class BCI-IV-2B dataset

6.6. BCI Competition IV 2A – 4 Class Performance Comparison for NF-EEG

In this section, the proposed method is compared with the methods developed with the 4-class BCI-IV-2A dataset. NF-EEG achieved the best average classification accuracy results among all models compared. Among the first three subjects with the highest results and highest increase in 4-class BCI-IV-2A dataset, the classification accuracy increase rate was 12.61% for subject 4, 5.55% for subject 2, 3.38% for subject 6 and 2.56% on average. This model has a standard deviation as 7.35. BCI-IV-2A 4-class performance comparison results are shown in table 6.5 and in figure 6.12.

Table 6.5. Classification accuracy comparison of state-of-the-art methods for 4-class BCI-IV-2A dataset

Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	AVG
DFFS [159]	63.69	61.97	91.09	61.72	63.41	66.11	59.57	62.84	84.46	68.32
EMD-MI [220]	66.70	63.90	77.80	63.20	72.20	70.10	64.60	76.40	77.10	70.20
EA-CSP-LDA [194]	87.50	56.25	98.61	73.61	50.00	64.58	68.75	89.58	72.93	73.53
EEG-Net [169]	77.73	63.07	86.52	73.61	68.58	61.60	79.17	67.94	56.03	70.69
DeepNet [174]	74.75	62.24	82.29	72.98	70.94	61.74	82.98	66.25	51.59	69.74
ShallowNet [174]	75.82	60.12	85.59	69.04	68.41	54.30	80.09	68.10	59.17	69.36
WTRCSP [218]	88.89	51.39	96.53	70.14	54.86	71.53	81.25	93.75	93.75	78.01
TLCSD [219]	90.28	54.17	93.75	64.58	57.64	65.28	65.20	90.97	85.42	73.84
C2CM [224]	87.50	65.28	90.28	66.67	62.50	45.49	89.58	83.33	79.51	74.46
FBCSP-SVM [164]	85.31	64.51	90.00	64.02	73.66	52.72	92.10	88.62	72.10	75.89
FBCNet [225]	85.76	61.07	94.51	68.84	82.54	58.71	93.08	86.21	80.54	79.03
NF-EEG	83.63	68.90	90.11	82.89	75.72	73.95	89.53	86.72	78.03	**81.05**[a]

[a] indicates $p < 0.05$ in Wilcoxon signed-rank test.

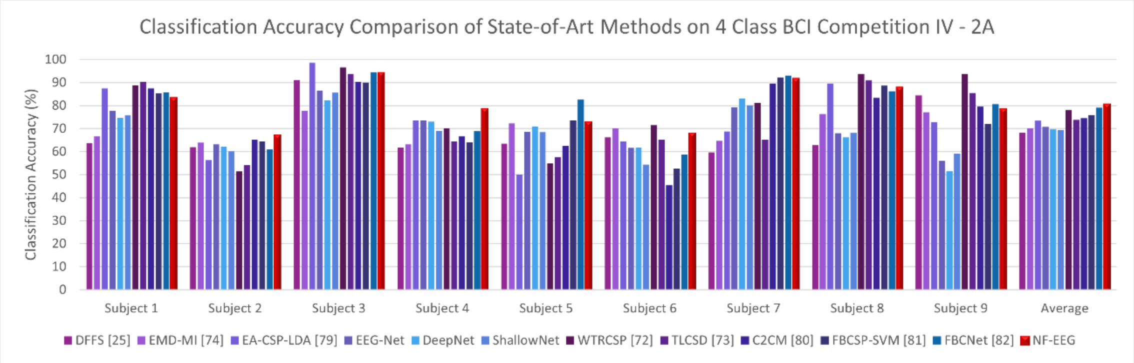


Figure 6.12. Classification accuracy comparison of state-of-the-art methods for 4-class BCI-IV-2A dataset

6.7. Statistical Information of NF-EEG Models

In this section, statistical data of the developed NF-EEG model were calculated and these data were compared with previous studies. In this context, the average accuracy values of the compared models for each dataset were calculated according to the subjects. These values were compared with the accuracy values of the proposed method and the average accuracy increase rates (IR) were calculated. Although it has different ranges in each dataset, it has subject-based accuracy increase rates ranging from 0.43% to 39.51% for all datasets. These values are shown in table 6.6.

Table 6.6. Subject-Based average accuracy increase rates of NF-EEG according to compared models.

	2 Class BCI-IV-2A			2 Class BCI IV-2B			4 Class BCI-IV-2A		
	Average	NF-EEG	IR*	Average	NF-EEG	IR*	Average	NF-EEG	IR*
Subject 1	79.24	91.49	**15.46**	73.49	82.46	**12.20**	80.36	83.63	**4.07**
Subject 2	64.11	89.44	**39.51**	62.45	75.47	**20.84**	60.36	68.90	**14.15**
Subject 3	86.91	95.62	**10.03**	67.41	78.70	**16.75**	89.72	90.11	**0.43**
Subject 4	72.47	93.10	**28.47**	92.94	96.74	**4.09**	68.04	82.89	**21.83**
Subject 5	75.37	94.81	**25.80**	87.46	98.53	**12.66**	65.89	75.72	**14.93**
Subject 6	71.54	88.89	**24.25**	80.35	86.85	**8.09**	61.11	73.95	**21.02**
Subject 7	76.36	97.14	**27.21**	79.19	91.61	**15.68**	77.85	89.53	**15.00**
Subject 8	82.94	99.25	**19.66**	88.51	95.04	**7.38**	79.45	86.72	**9.15**
Subject 9	83.93	92.31	**9.99**	81.43	90.24	**10.82**	73.87	78.03	**5.63**
Average	77.09	93.56	**21.37**	79.33	88.40	**11.44**	73.01	81.05	**11.02**

* Increase Rate

Along with all these information, the true positive, false positive, false negative, and true negative rates and amounts of the NF-EEG model for each dataset and each class are shown with confusion matrices in figure 6.13, 6.14 and 6.15. In addition, the average precision, recall, F1 score, kappa and standard deviation values of the proposed model are given in table 6.7.

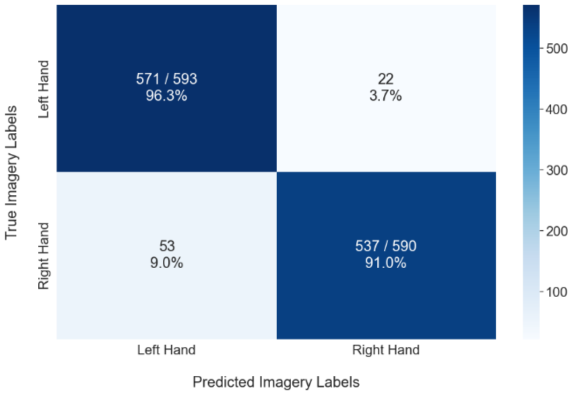


Figure 6.13. Confusion matrix of 2 Class BCI-IV-2A

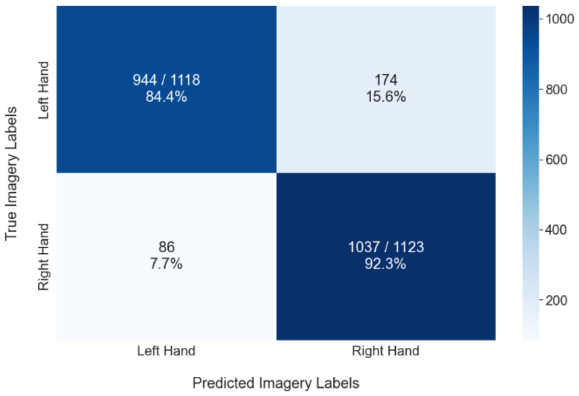


Figure 6.14. Confusion matrix of 2 Class BCI-IV-2B

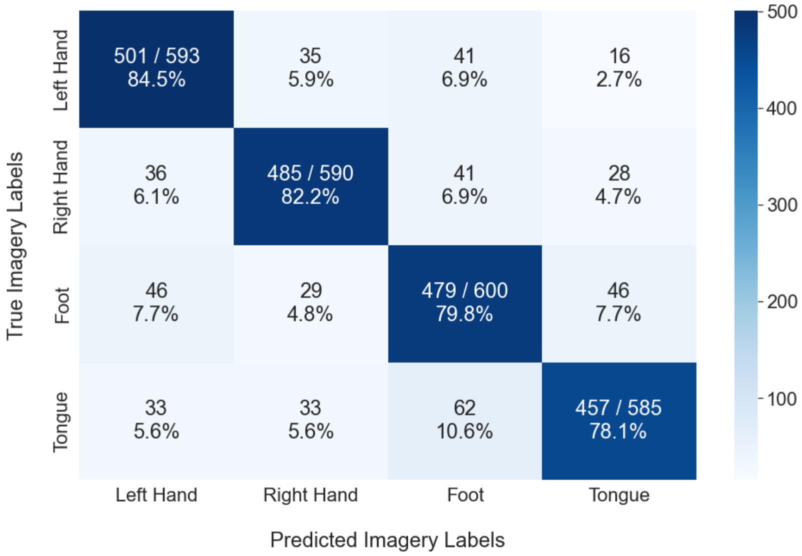


Figure 6.15. Confusion matrix of 4 Class BCI-IV-2A

Table 6.7. Statistical values of NF-EEG models.

Dataset	Precision	Recall	F1 Score	Kappa	STD
2 Class BCI-IV-2A	0.938	0.937	0.937	0.873	3.461
2 Class BCI-IV-2B	0.886	0.884	0.884	0.768	8.131
4 Class BCI-IV-2A	0.813	0.812	0.812	0.749	7.346

6.8. IS-EEG Experimental Setup

Python programming language, TensorFlow and Keras libraries were used to design the training and classification processes covering the datasets, input shapes and CNN models defined in the previous sections. GeForce RTX 2080 Super GPU card, 32 GB ram and 12 core 3.80 Ghz processor were used as hardware. During the training of the datasets, only the training sets were used, and the test sets were not included in the training in any way. Training sets are divided into two parts, 80% training and 20% validation. Determining the validation set let to investigate the tendency of the model to be overfitting or underfitting during training. Early stopping method was used to prevent overfitting. In this method, the validation loss value was chosen as a variable and the loss value was recorded after each epoch. If the validation loss value does not decrease during the following 20 epochs, the training session is finished and the model with the best value is taken. With this method, the performance of the model was measured with an unseen test set. Accuracy values were measured for each subject according to each input shape, and their average accuracy values were calculated according to the input shapes.

In this section, training and classification of BCI-IV-2A and BCI-IV-2B datasets are done by using 8 different input shapes and 2 CNN models that we have designed. Training and validation accuracy graphs, training and loss values graphs and epoch time values were obtained, and after training, accuracy values for each subject, their average values and confusion matrices were obtained and statistical values of models were given.

6.9. Training and Validation Graphs for IS-EEG

While both data sets were trained with the developed CNN models and 8 different input shapes, the training - validation accuracy changes and the graphs of the training - validation loss changes were obtained so that the training process could be observed and analyzed better. These graphs are shown in figures 6.16 – 6.31 for the BCI-IV-2A dataset and BCI-IV-2B dataset.

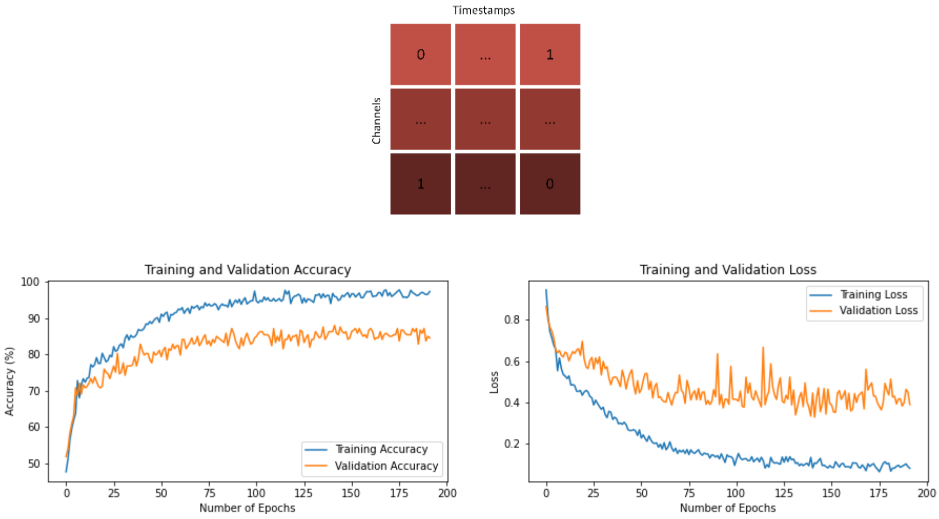


Figure 6.16. Training-Validation accuracies and losses of T x C input shape for BCI-IV-2A

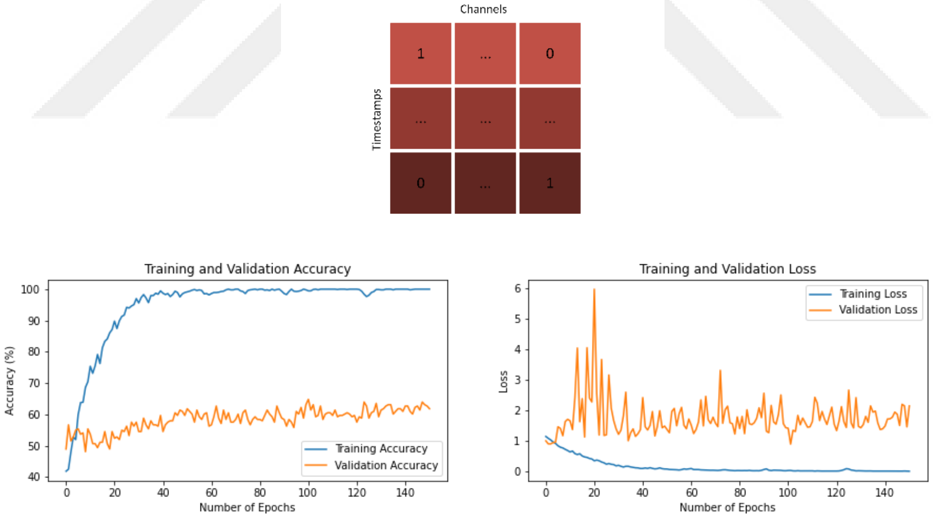


Figure 6.17. Training-Validation accuracies and losses of C x T input shape for BCI-IV-2A

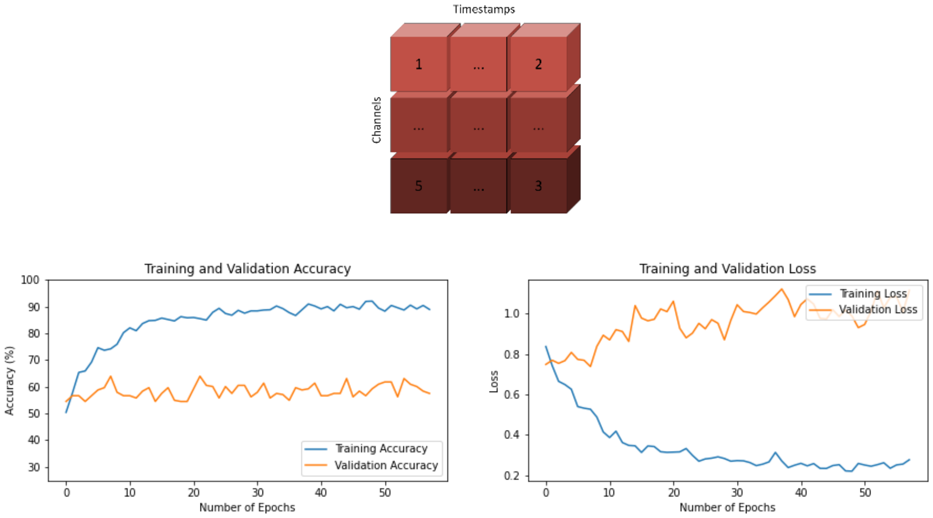


Figure 6.18. Training-Validation accuracies and losses of T x C x 1 input shape for BCI-IV-2A

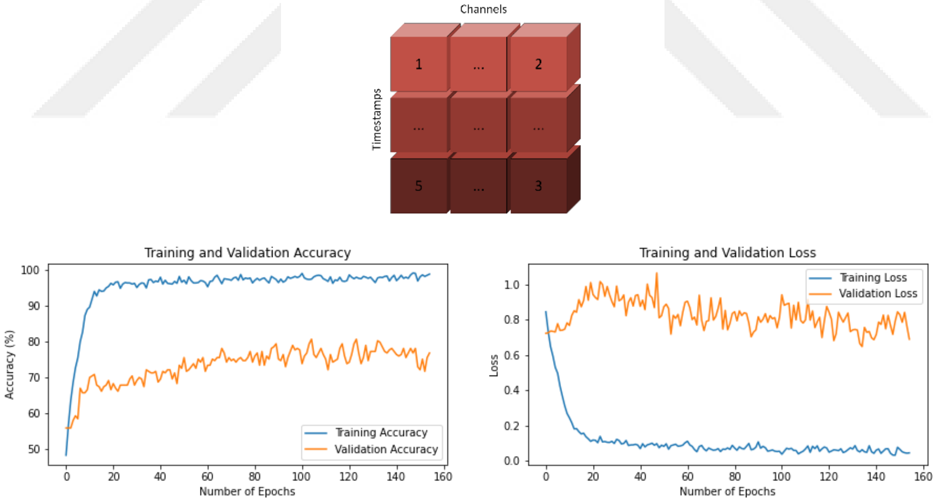


Figure 6.19. Training-Validation accuracies and losses of C x T x 1 input shape for BCI-IV-2A

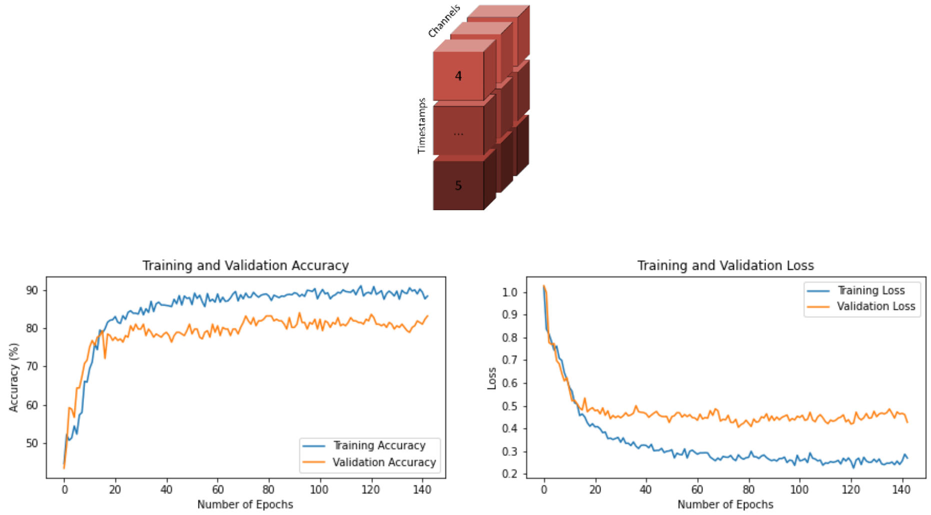


Figure 6.20. Training-Validation accuracies and losses of 1 x T x C input shape for BCI-IV-2A

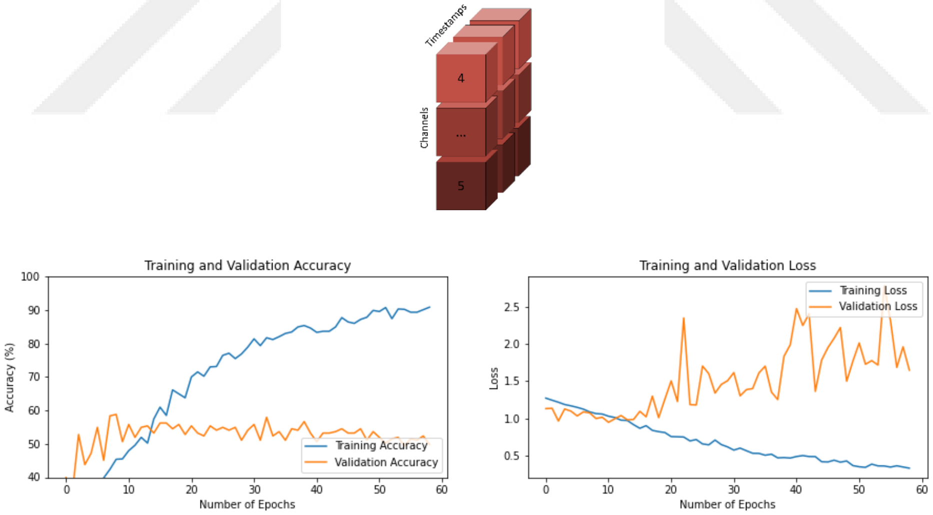


Figure 6.21. Training-Validation accuracies and losses of 1 x C x T input shape for BCI-IV-2A

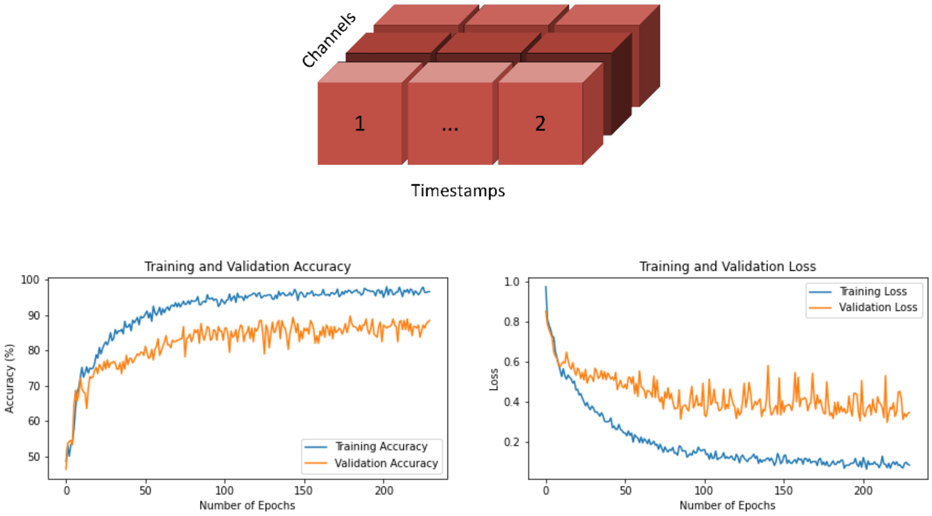


Figure 6.22. Training-Validation accuracies and losses of T x 1 x C input shape for BCI-IV-2A

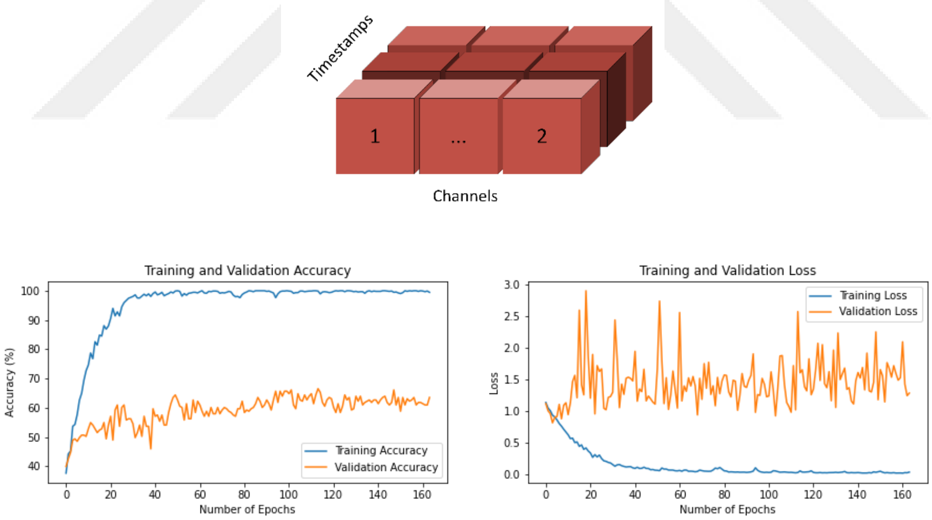


Figure 6.23. Training-Validation accuracies and losses of C x 1 x T input shape for BCI-IV-2A

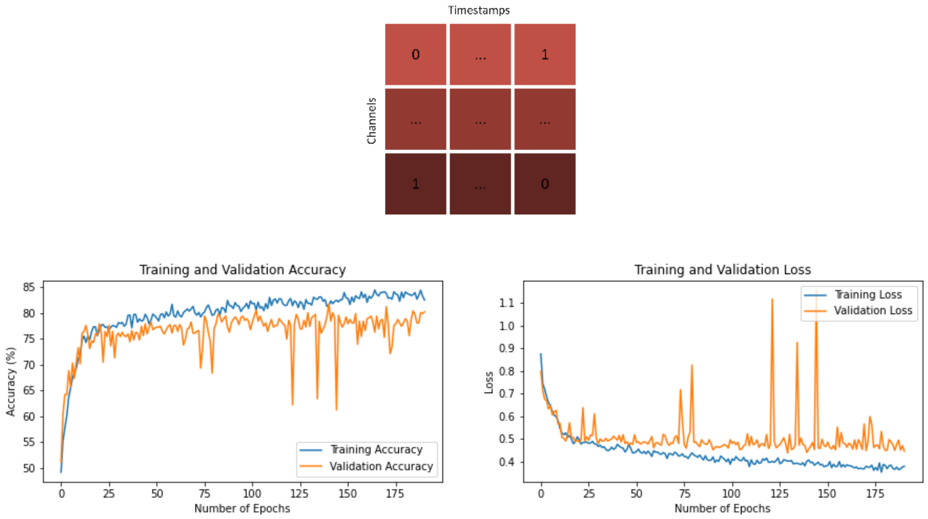


Figure 6.24. Training-Validation accuracies and losses of T x C input shape for BCI-IV-2B

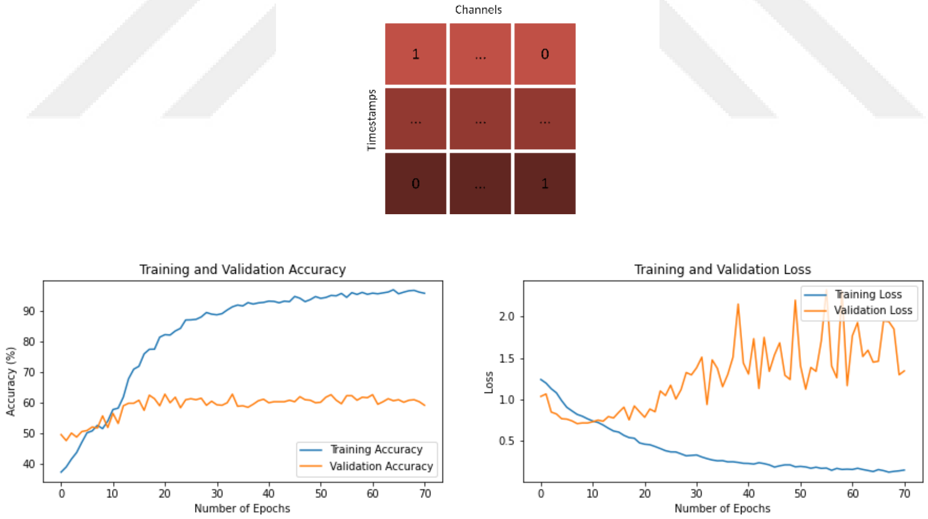


Figure 6.25. Training-Validation accuracies and losses of C x T input shape for BCI-IV-2B

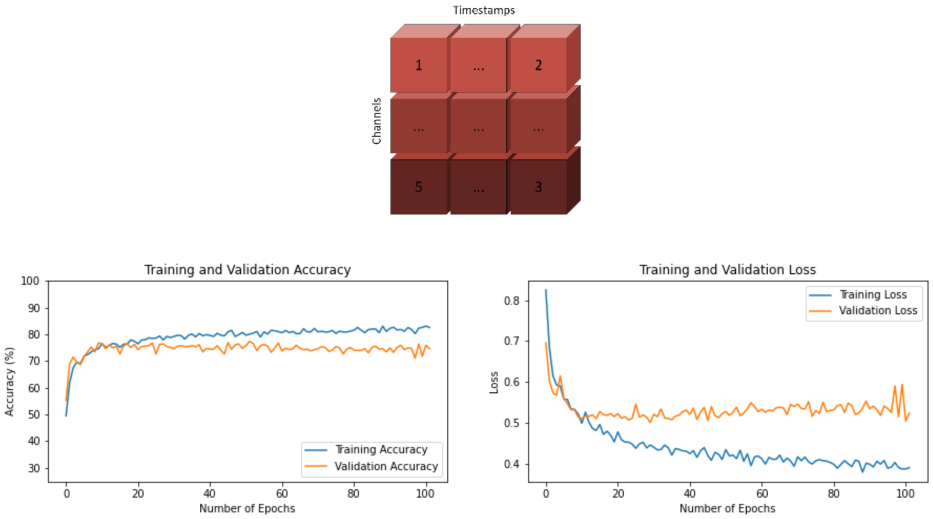


Figure 6.26. Training-Validation accuracies and losses of T x C x 1 input shape for BCI-IV-2B

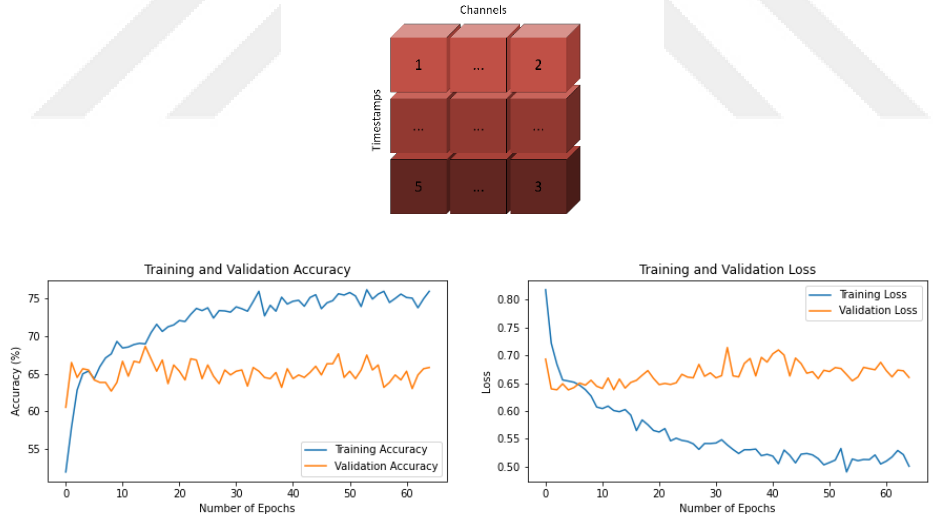


Figure 6.27. Training-Validation accuracies and losses of C x T x 1 input shape for BCI-IV-2B

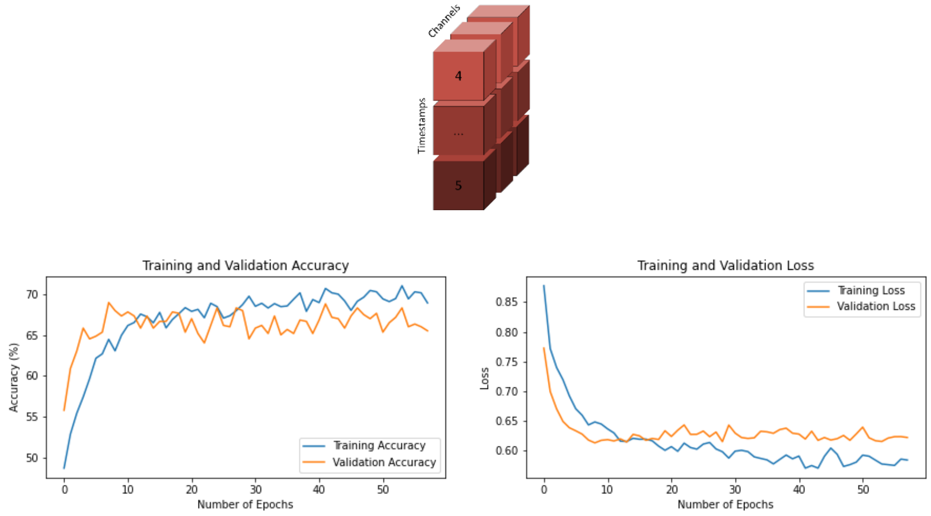


Figure 6.28. Training-Validation accuracies and losses of 1 x T x C input shape for BCI-IV-2B

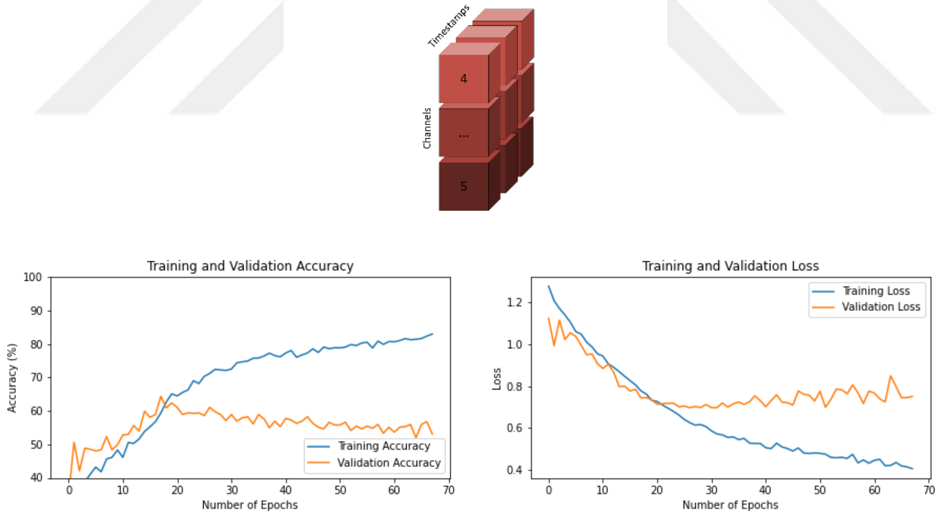


Figure 6.29. Training-Validation accuracies and losses of 1 x C x T input shape for BCI-IV-2

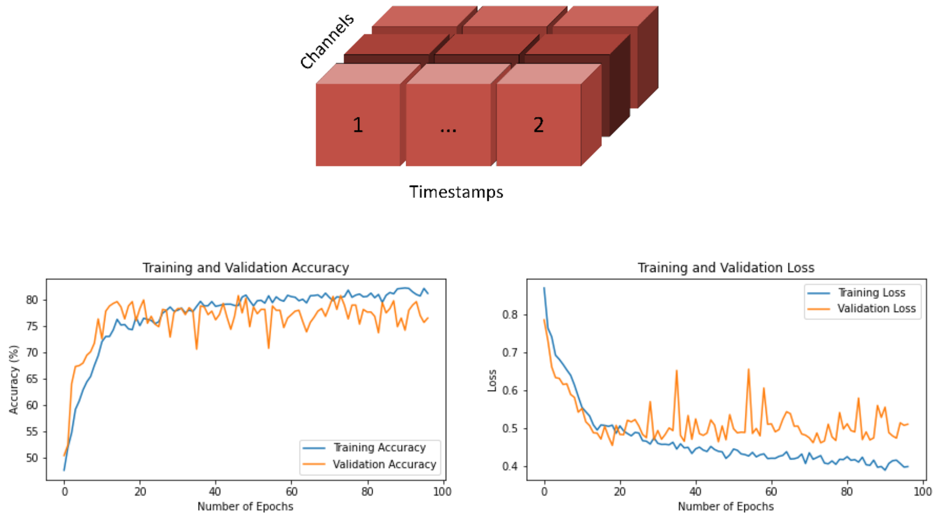


Figure 6.30. Training-Validation accuracies and losses of T x 1 x C input shape for BCI-IV-2B

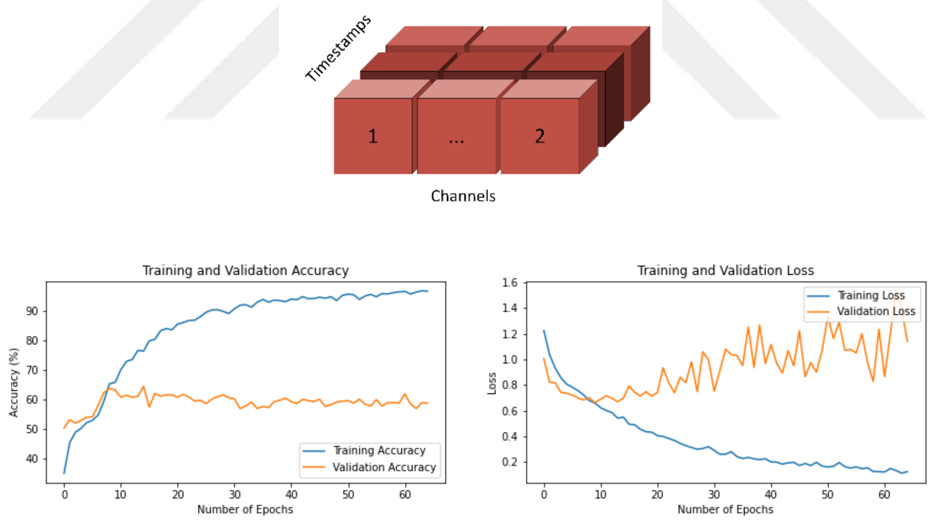


Figure 6.31. Training-Validation accuracies and losses of C x 1 x T input shape for BCI-IV-2B

6.10. Accuracy Values for IS-EEG

In order to measure the effect of the designed input shape structures on the success of the classifier, the subject-based and average accuracy values of each input shape were calculated. While calculating these values, the test set, which is a separate group in the dataset, was not used in the training phase and not seen by the model before was used. The obtained accuracy values are given in table 6.8 for BCI-IV-2A dataset and in table 6.9 for BCI-IV-2B dataset.

Table 6.8. Subject-Based and average accuracy values of BCI-IV-2A.

	T x C	C x T	T x C x 1	C x T x 1	1 x T x C	1 x C x T	T x 1 x C	C x 1 x T
S1	84,40	60,28	54,61	63,12	68,09	47,52	81,56	60,28
S2	78,87	54,23	51,41	72,54	80,28	51,41	81,69	58,45
S3	96,35	54,74	48,18	71,53	75,18	51,82	88,32	59,12
S4	88,79	61,21	60,34	79,31	81,90	51,72	84,48	64,66
S5	92,59	72,59	68,15	87,41	94,07	57,04	96,30	74,81
S6	84,26	59,26	56,48	74,07	80,56	50,00	83,33	62,04
S7	92,14	72,14	65,00	91,43	85,71	56,43	93,57	74,29
S8	97,76	55,97	55,97	79,10	84,33	55,22	94,03	59,70
S9	88,46	54,62	53,85	81,54	82,31	56,15	88,46	60,00
Average	89,29	60,56	57,11	77,78	81,38	53,03	87,97	63,71
STD	5,47	6,41	5,70	7,70	6,38	2,96	5,02	5,74

Table 6.9. Subject-Based and average accuracy values of BCI-IV-2B.

	T x C	C x T	T x C x 1	C x T x 1	1 x T x C	1 x C x T	T x 1 x C	C x 1 x T
S1	78,95	62,72	70,61	64,91	63,16	59,65	78,95	61,40
S2	70,20	56,73	63,67	71,43	64,49	66,53	64,49	61,63
S3	73,48	60,87	67,39	68,70	76,09	67,83	66,96	58,70
S4	96,74	57,33	94,14	64,17	61,56	58,96	94,79	58,31
S5	96,70	93,04	92,31	95,24	97,07	93,77	98,53	89,01
S6	83,67	60,96	76,10	69,32	68,53	64,14	78,09	61,35
S7	90,09	80,60	87,93	81,47	89,66	82,76	93,10	76,72
S8	91,74	64,35	88,26	81,30	78,26	62,61	89,57	62,17
S9	82,86	51,84	82,45	56,33	55,92	51,43	79,59	54,29
Average	84,94	65,38	80,32	72,54	72,75	67,52	82,67	64,84
STD	8,61	11,70	10,08	10,44	12,27	11,63	10,85	9,79

6.11. Epoch Times for IS-EEG

As a result of the training for each input shape, the training epoch times were measured for the datasets used. Epoch times are seconds measured for a single epoch. The measured epoch times for the BCI-IV-2A and BCI-IV-2B datasets are given in table 6.10.

Table 6.10. Epoch times for each input shape for BCI-IV-2A and BCI-IV-2B.

Input Shape	Epoch Time (second / epoch)	
	BCI – IV – 2A	BCI – IV – 2B
T x C	4	10
C x T	3	9
T x C x 1	18	10
C x T x 1	23	10
1 x T x C	2	6
1 x C x T	2	4
T x 1 x C	2	6
C x 1 x T	2	5

6.12. Confusion Matrices for IS-EEG

Confusion matrix graphs were created to measure the ability of the trained models to distinguish the classes in the dataset and to determine which classes can be better distinguished with which input shape. With these graphs, true positive, true negative, false positive and false negative numbers and ratios of the models were obtained. These obtained values are given in figure 6.32 – 6.47 and in table 6.11 for BCI-IV-2A and BCI-IV-2B datasets.

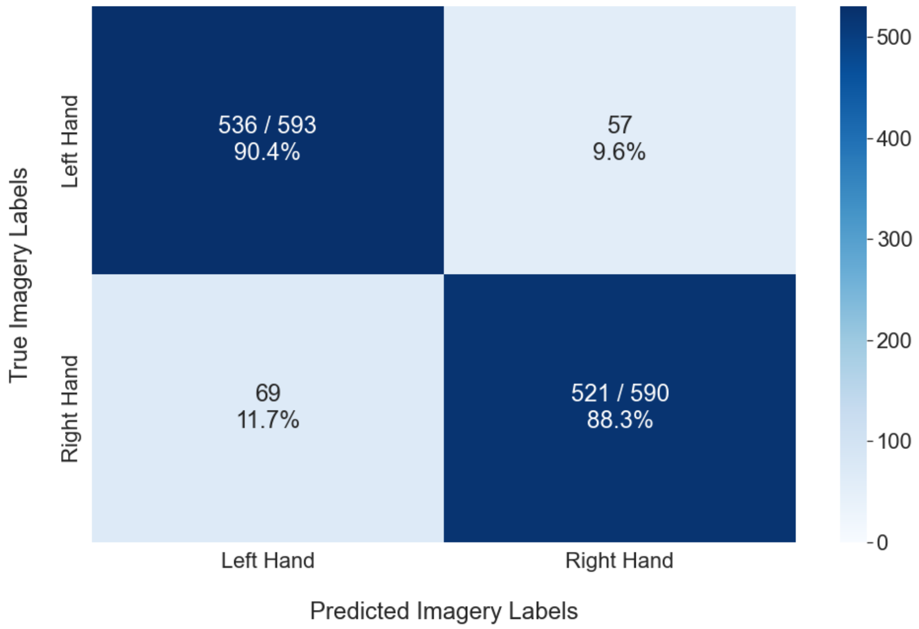


Figure 6.32. Confusion matrix of T x C input shape for BCI-IV-2A

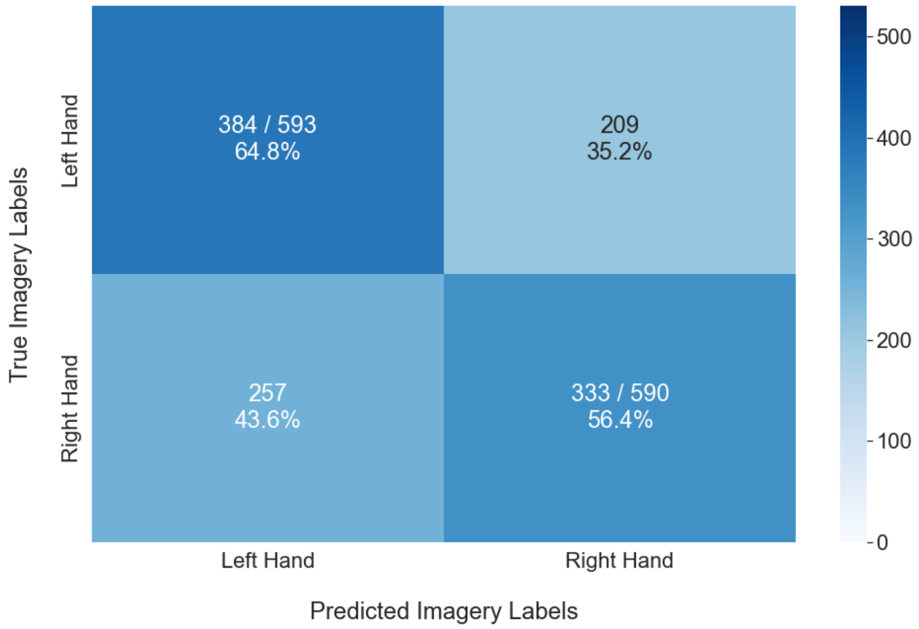


Figure 6.33. Confusion matrix of C x T input shape for BCI-IV-2A

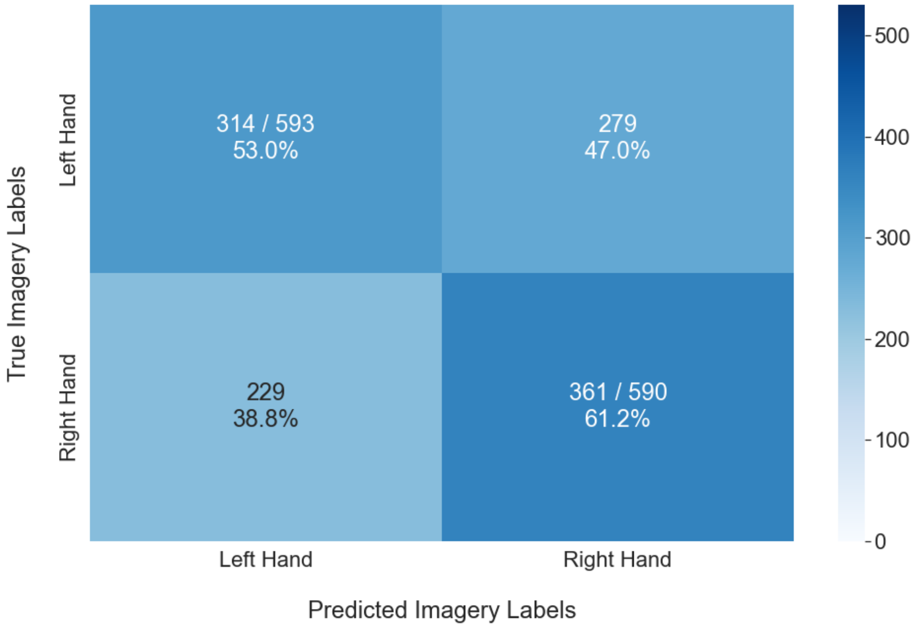


Figure 6.34. Confusion matrix of T x C x 1 input shape for BCI-IV-2A

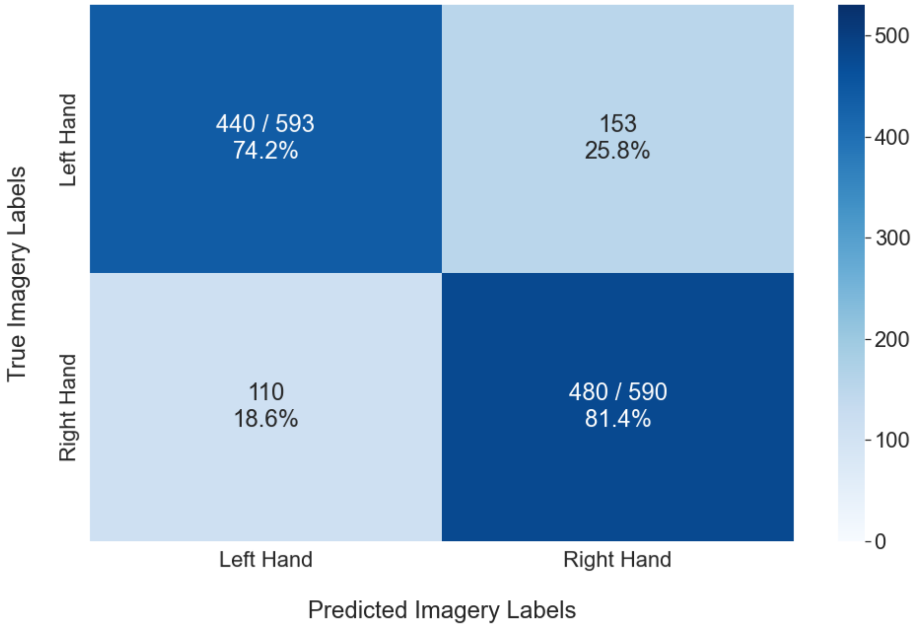


Figure 6.35. Confusion matrix of C x T x 1 input shape for BCI-IV-2A

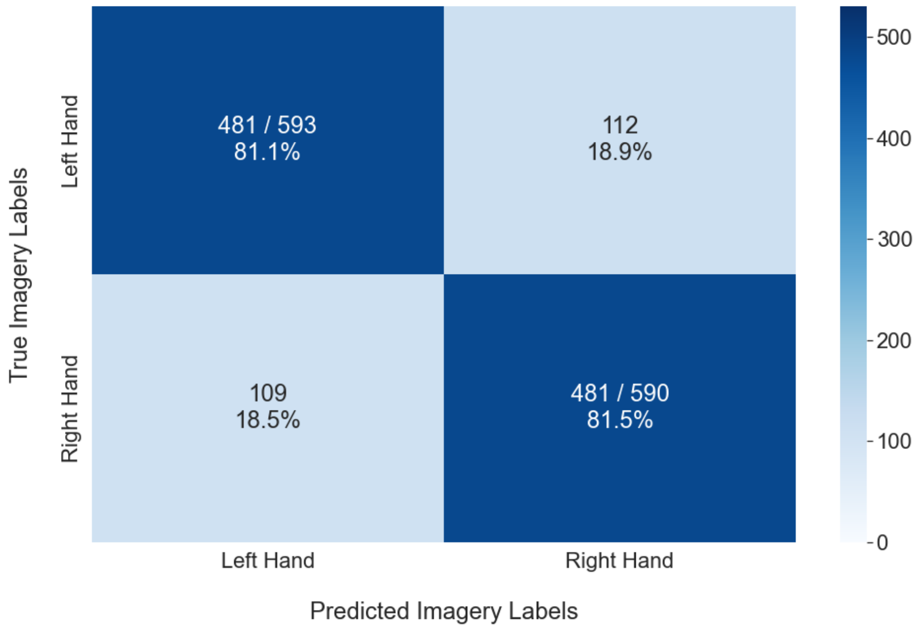


Figure 6.36. Confusion matrix of 1 x T x C input shape for BCI-IV-2A

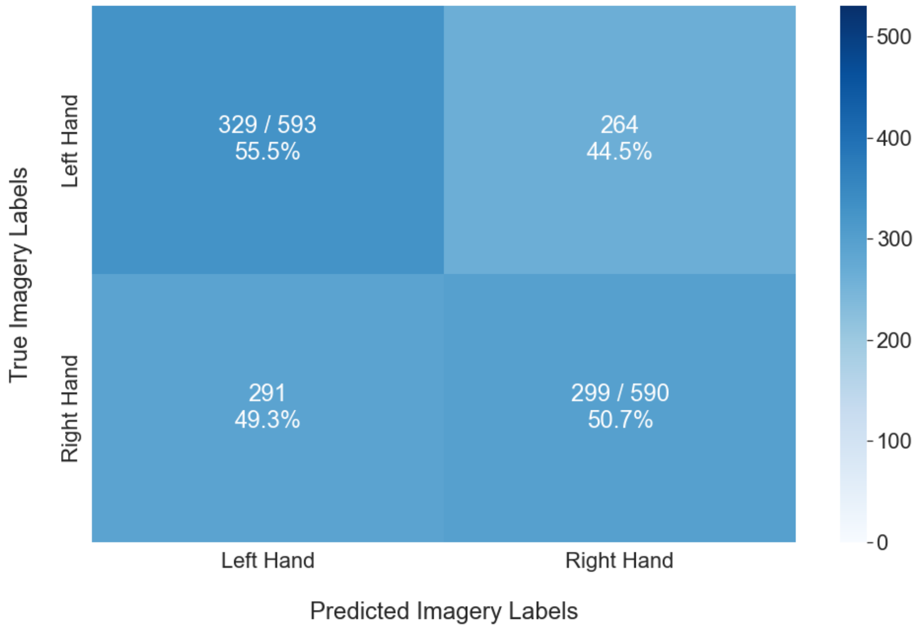


Figure 6.37. Confusion matrix of 1 x C x T input shape for BCI-IV-2A

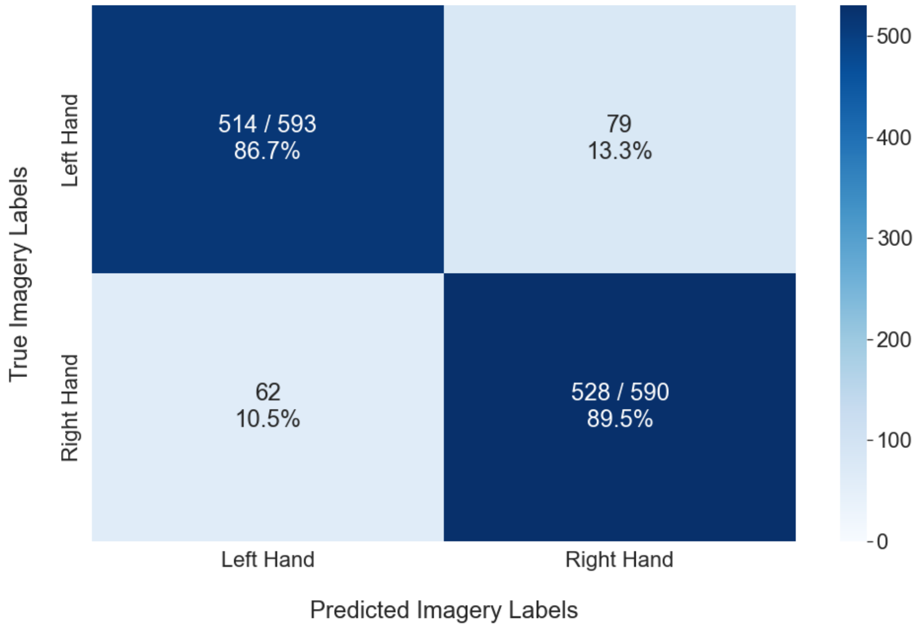


Figure 6.38. Confusion matrix of T x 1 x C input shape for BCI-IV-2A

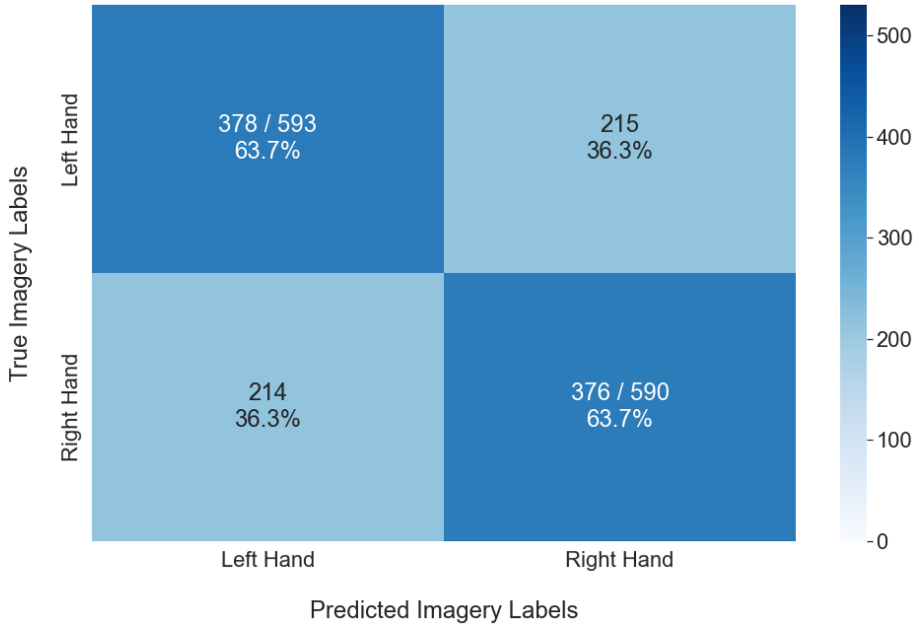


Figure 6.39. Confusion matrix of C x 1 x T input shape for BCI-IV-2A

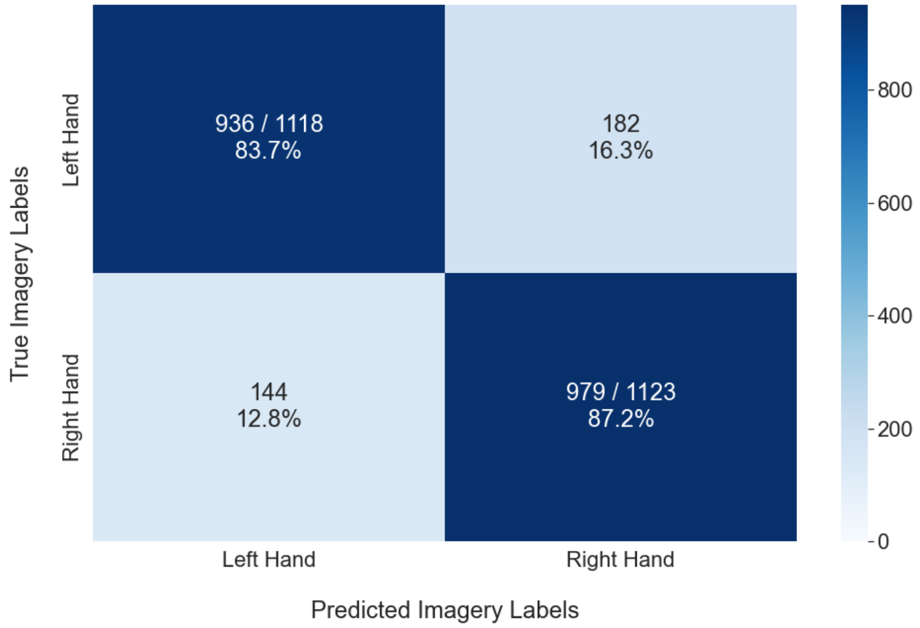


Figure 6.40. Confusion matrix of T x C input shape for BCI-IV-2B

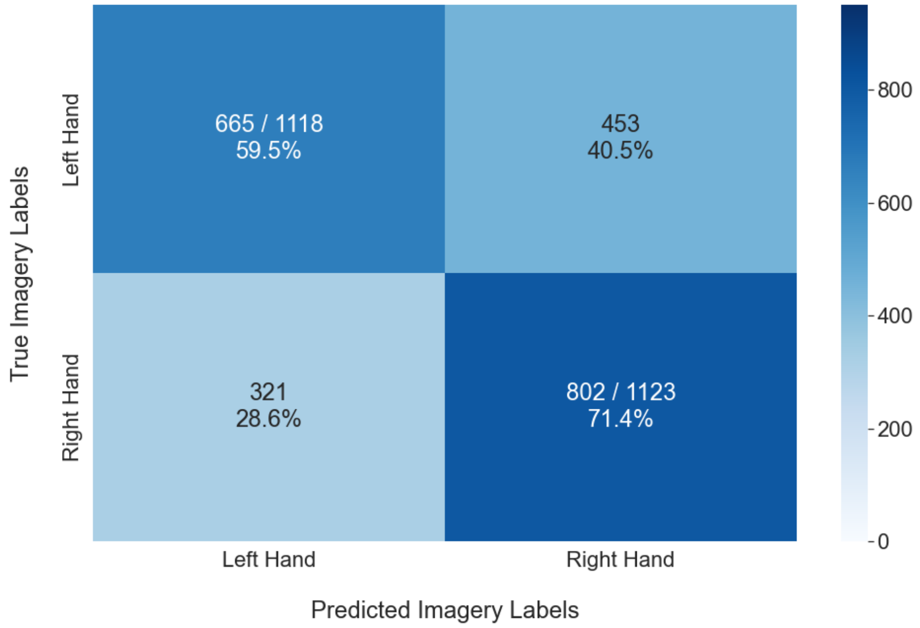


Figure 6.41. Confusion matrix of C x T input shape for BCI-IV-2B

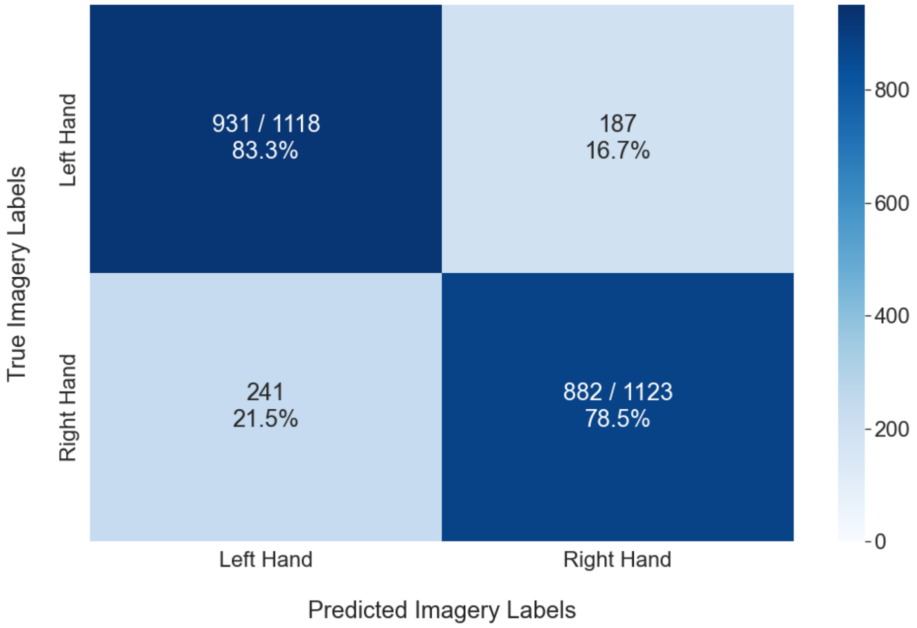


Figure 6.42. Confusion matrix of T x C x 1 input shape for BCI-IV-2B

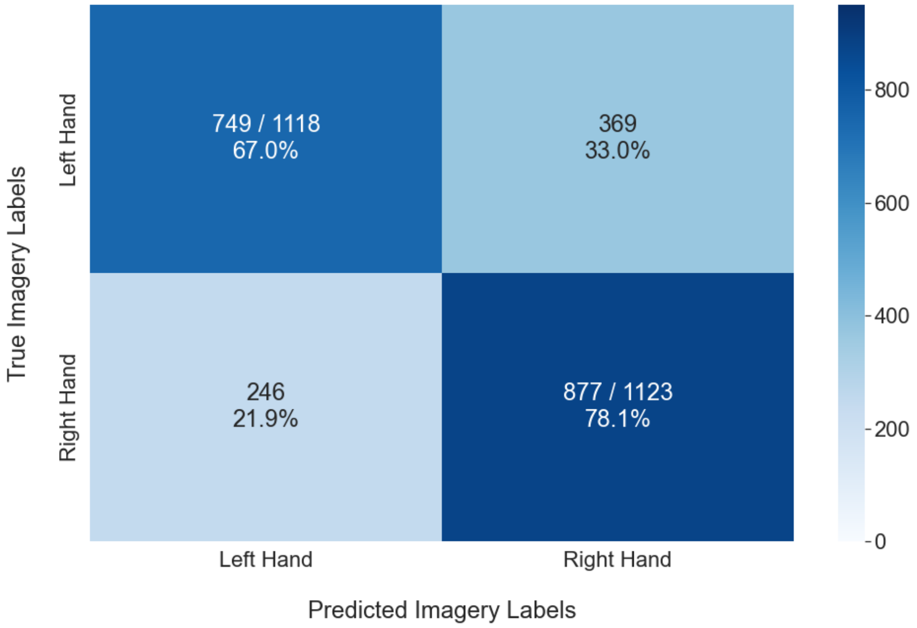


Figure 6.43. Confusion matrix of C x T x 1 input shape for BCI-IV-2B

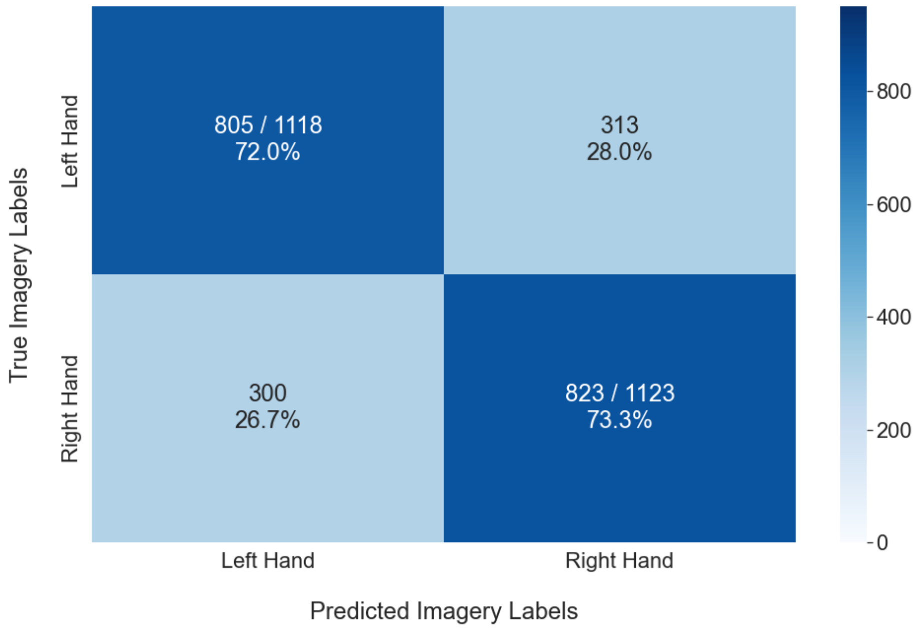


Figure 6.44. Confusion matrix of 1 x T x C input shape for BCI-IV-2B

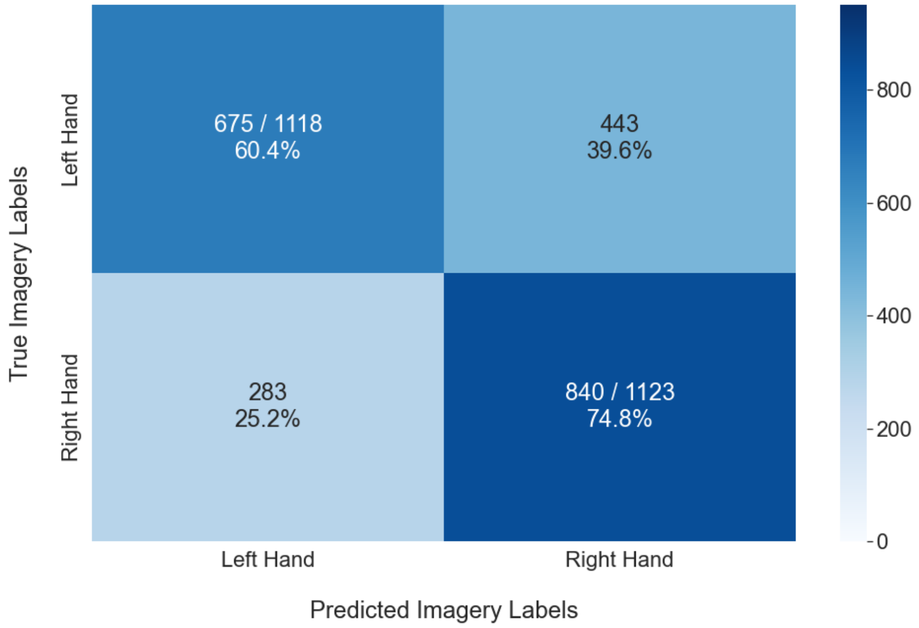


Figure 6.45. Confusion matrix of 1 x C x T input shape for BCI-IV-2B

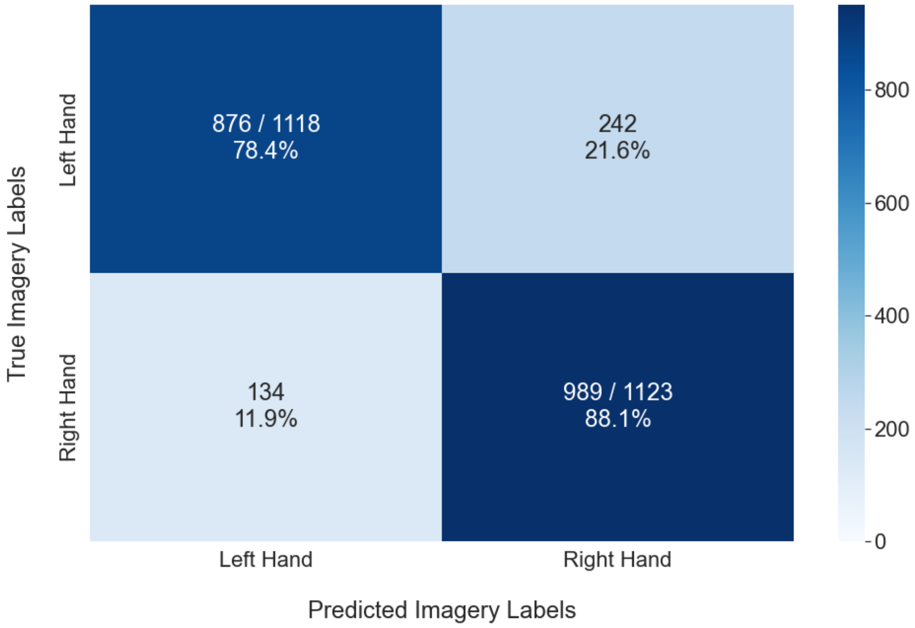


Figure 6.46. Confusion matrix of T x 1 x C input shape for BCI-IV-2B

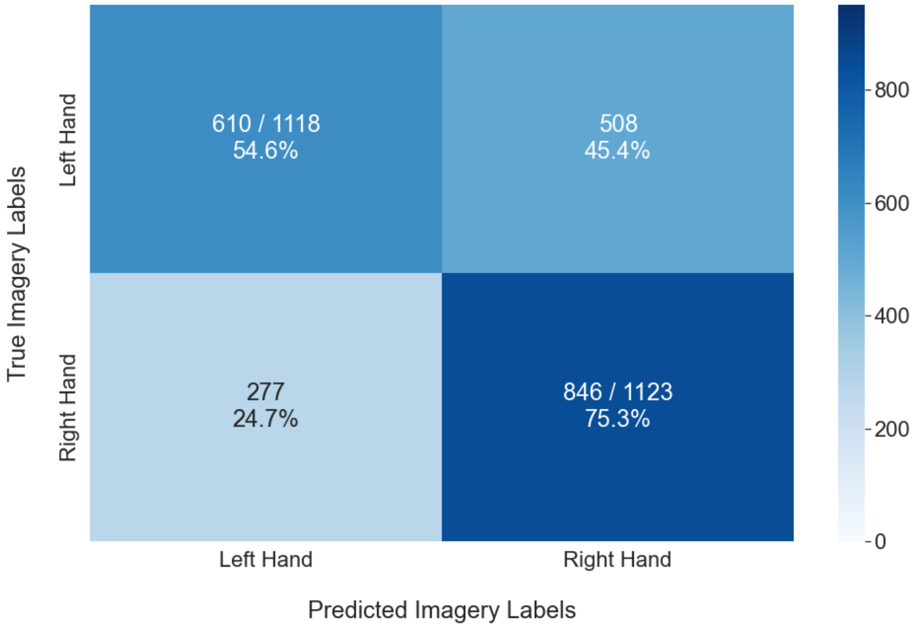


Figure 6.47. Confusion matrix of C x 1 x T input shape for BCI-IV-2B

Table 6.11. Confusion matrix values for BCI-IV-2A and BCI-IV-2B.

Input Shape	**BCI – IV – 2A**				**BCI – IV – 2B**			
	TL*	**FL***	**TR***	**FR***	**TL**	**FL**	**TR**	**FR**
T x C	536/593 (90.4%)	57 (9.6%)	521/590 (88.3%)	69 (11.7%)	936/1118 (83.7%)	182 (16.3%)	979/1123 (87.2%)	144 (12.8%)
C x T	384/593 (64.8%)	209 (35.2%)	333/590 (56.4%)	257 (43.6%)	665/1118 (59.5%)	453 (40.5%)	802/1123 (71.4%)	321 (28.6%)
T x C x 1	314/593 (53.0%)	279 (47.0%)	361/590 (61.2%)	229 (38.8%)	931/1118 (83.3%)	187 (16.7%)	882/1123 (78.5%)	241 (21.5%)
C x T x 1	440/593 (74.2%)	153 (25.8%)	480/590 (81.4%)	110 (18.6%)	749/1118 (67.0%)	369 (33.0%)	877/1123 (78.1%)	246 (21.9%)
1 x T x C	481/593 (81.1%)	112 (18.9%)	481/590 (81.5%)	109 (18.5%)	805/1118 (72.0%)	313 (28.0%)	823/1123 (73.3%)	300 (26.7%)
1 x C x T	329/593 (55.5%)	264 (44.5%)	299/590 (50.7%)	291 (49.3%)	675/1118 (60.4%)	443 (39.6%)	840/1123 (74.8%)	283 (25.2%)
T x 1 x C	514/593 (86.7%)	79 (13.3%)	528/590 (89.5%)	62 (10.5%)	876/1118 (78.4%)	242 (21.6%)	989/1123 (88.1%)	134 (11.9%)
C x 1 x T	378/593 (63.7%)	215 (36.3%)	376/590 (63.7%)	214 (36.3%)	610/1118 (54.6%)	508 (45.4%)	846/1123 (75.3%)	277 (24.7%)

6.13. Model Statistics for IS-EEG

The statistical values of the model were calculated according to the input shape structures for the datasets. The F1 score, which is the harmonic mean of the precision and recall values and the success of balancing the precision and recall values of the model, the cappa coefficient value and the standard deviation values for each input shape were obtained. These obtained values are given in table 6.12 for both data sets.

Table 6.12. Statistical values of BCI-IV-2A and BCI-IV-2B Models.

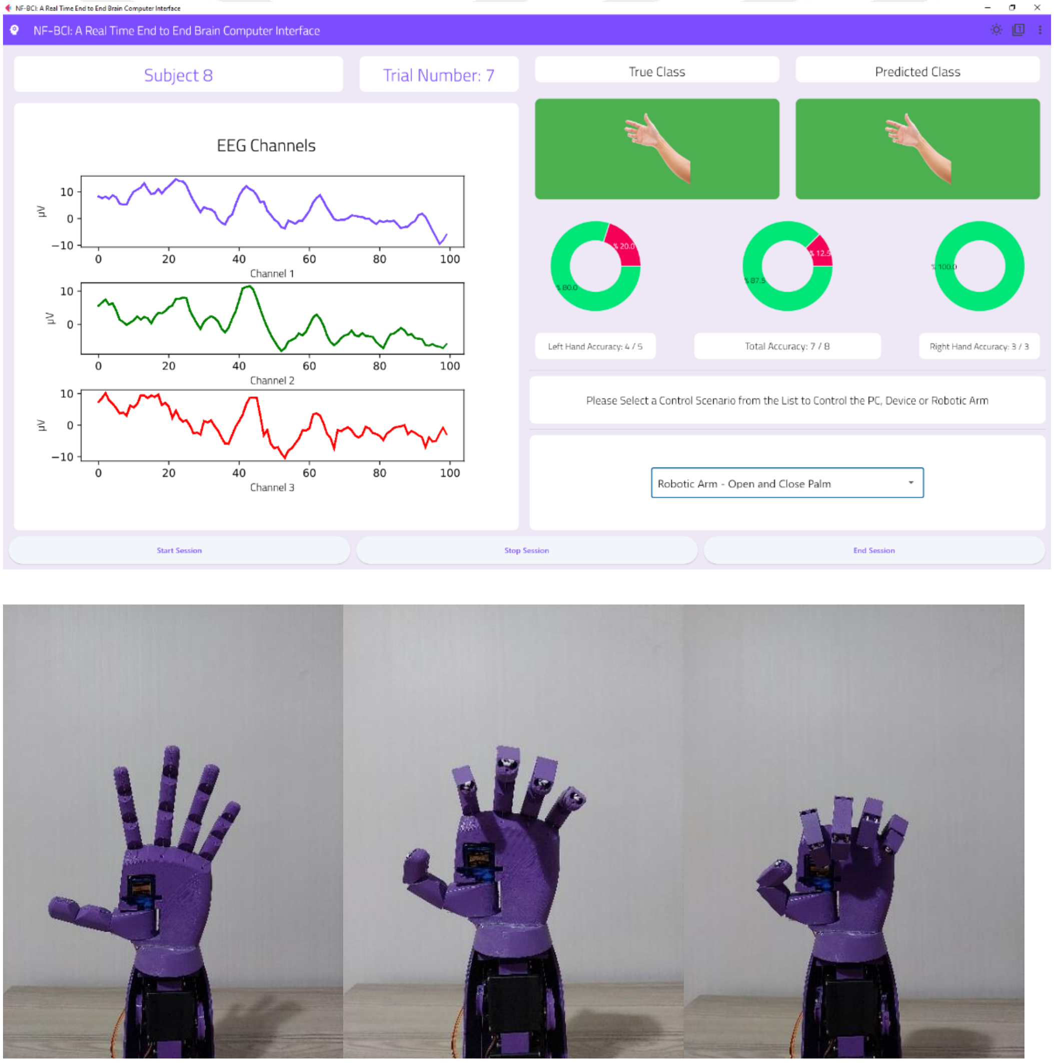
Input Shape	**BCI – IV – 2A**				**BCI – IV – 2B**			
	Accuracy	**F1 Score**	**Kappa**	**STD**	**Accuracy**	**F1 Score**	**Kappa**	**STD**
T x C	89.293	0.893	0.787	5.471	84.943	0.854	0.709	8.607
C x T	60.561	0.605	0.212	6.413	65.383	0.653	0.309	11.695
T x C x 1	57.114	0.570	0.141	5.704	80.318	0.809	0.618	10.083
C x T x 1	77.782	0.777	0.555	7.697	72.538	0.725	0.451	10.436
1 x T x C	81.376	0.813	0.626	6.376	72.747	0.726	0.453	12.269
1 x C x T	53.033	0.531	0.062	2.958	67.518	0.674	0.352	11.628
T x 1 x C	87.969	0.881	0.762	5.021	82.665	0.830	0.664	10.846
C x 1 x T	63.713	0.637	0.275	5.737	64.835	0.646	0.299	9.790

6.14. NF-BCI Control Scenario Results

In this section, the results of NF-BCI software to visualize and classify EEG signals and apply control scenarios according to classification results are given.

6.14.1. Robotic Arm - Open and Close Palm

In this scenario, it is aimed that the NF-BCI software will close and open all fingers on the robotic arm. When the classifier working on NF-BCI makes a right hand prediction, the palm of the robotic arm is closed. When the classifier makes a left hand prediction, the palm of the robotic arm is opened. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.48 and figure 6.49.



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 8

Trial Number: 7

True Class

Predicted Class

Left Hand Accuracy: 4 / 5

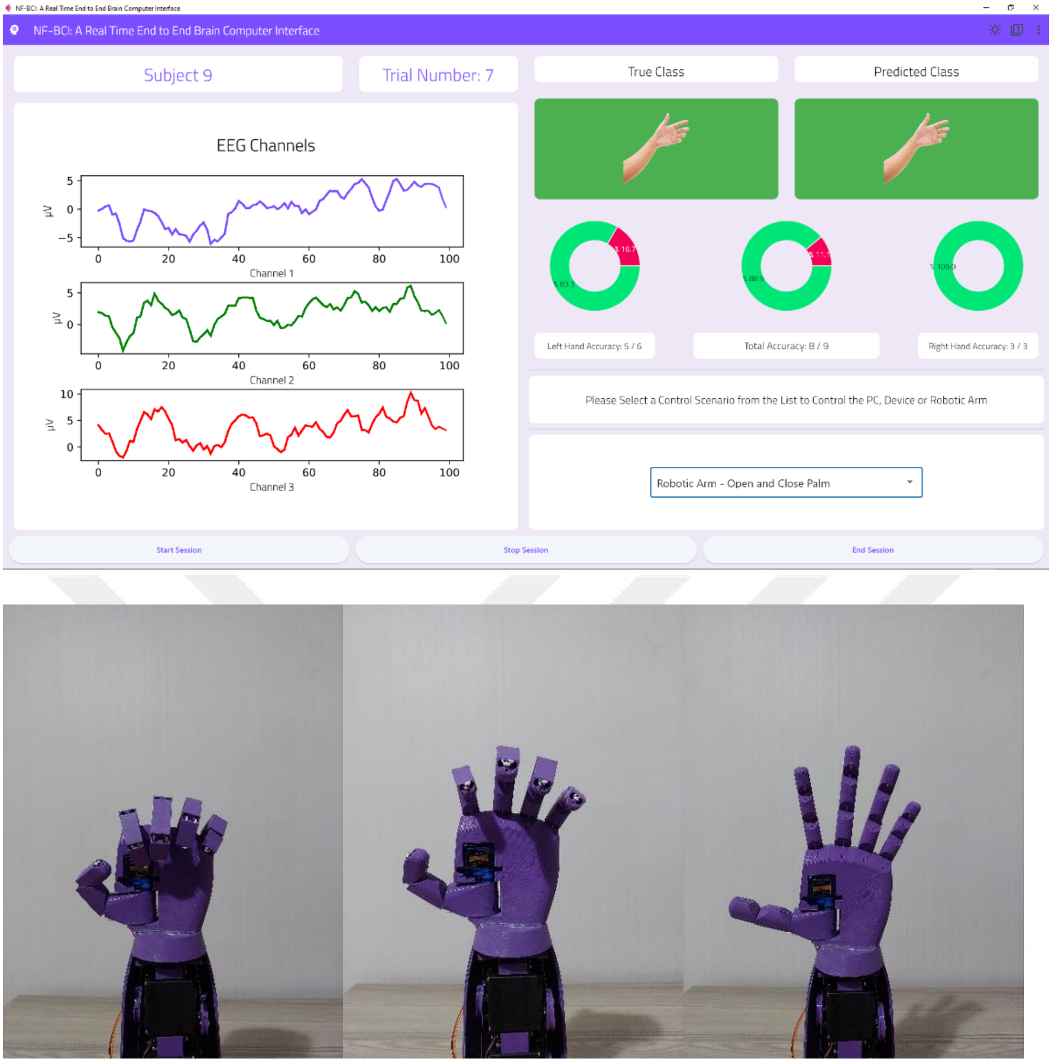
Total Accuracy: 7 / 8

Right Hand Accuracy: 3 / 3

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Open and Close Palm

Figure 6.48. Robotic arm – open and close palm scenario for right hand classification



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 9

Trial Number: 7

True Class

Predicted Class

Left Hand Accuracy: 5 / 6

Total Accuracy: 8 / 9

Right Hand Accuracy: 3 / 3

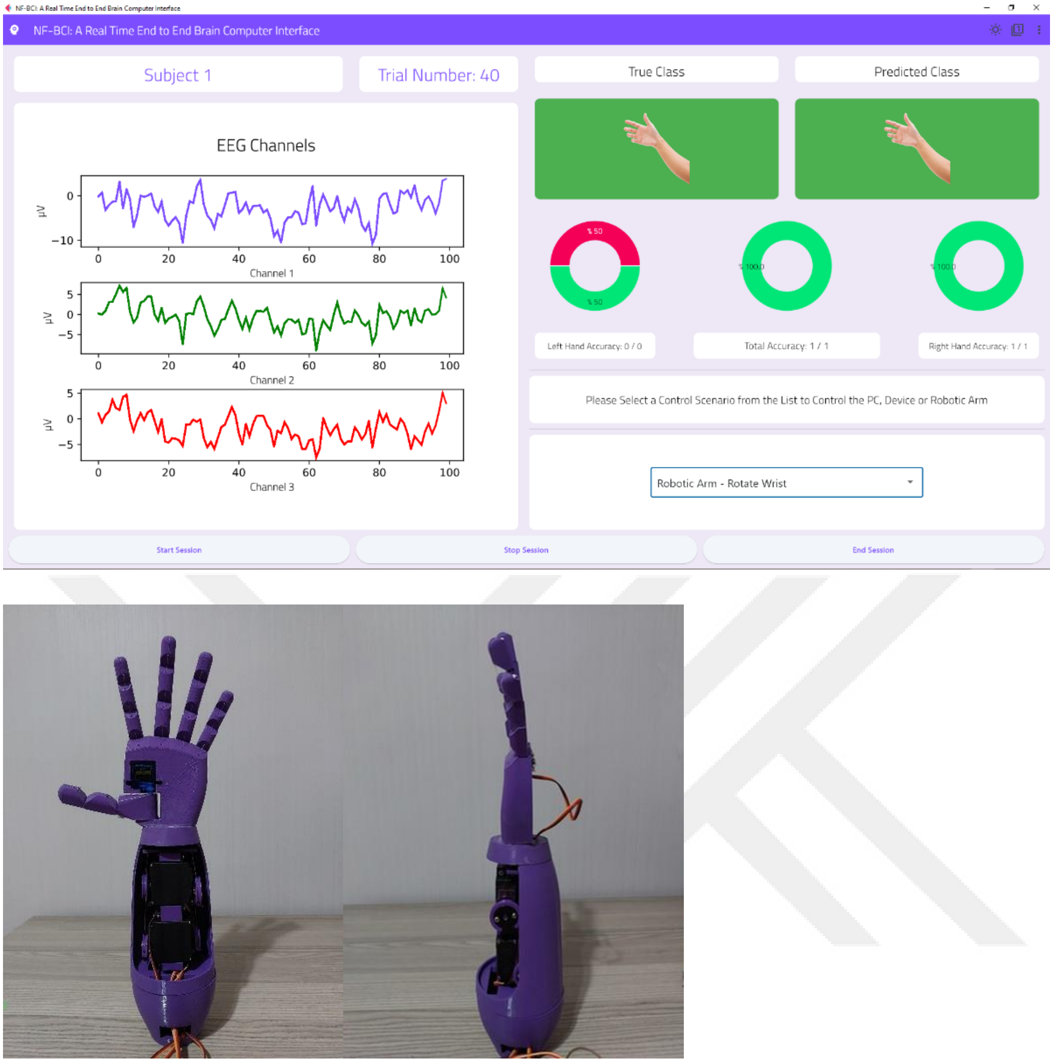
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Open and Close Palm

Figure 6.49. Robotic arm – open and close palm scenario for left hand classification

6.14.2. Robotic Arm – Rotate Wrist

In this scenario, the NF-BCI software is aimed to rotate the main body of the robotic arm. When the classifier working on NF-BCI makes a right hand prediction, the robotic arm body turns to the right. When the classifier makes a left hand prediction, the robotic arm body turns to the left. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.50 and figure 6.51.



Subject 1

Trial Number: 40

Left Hand Accuracy: 0 / 0

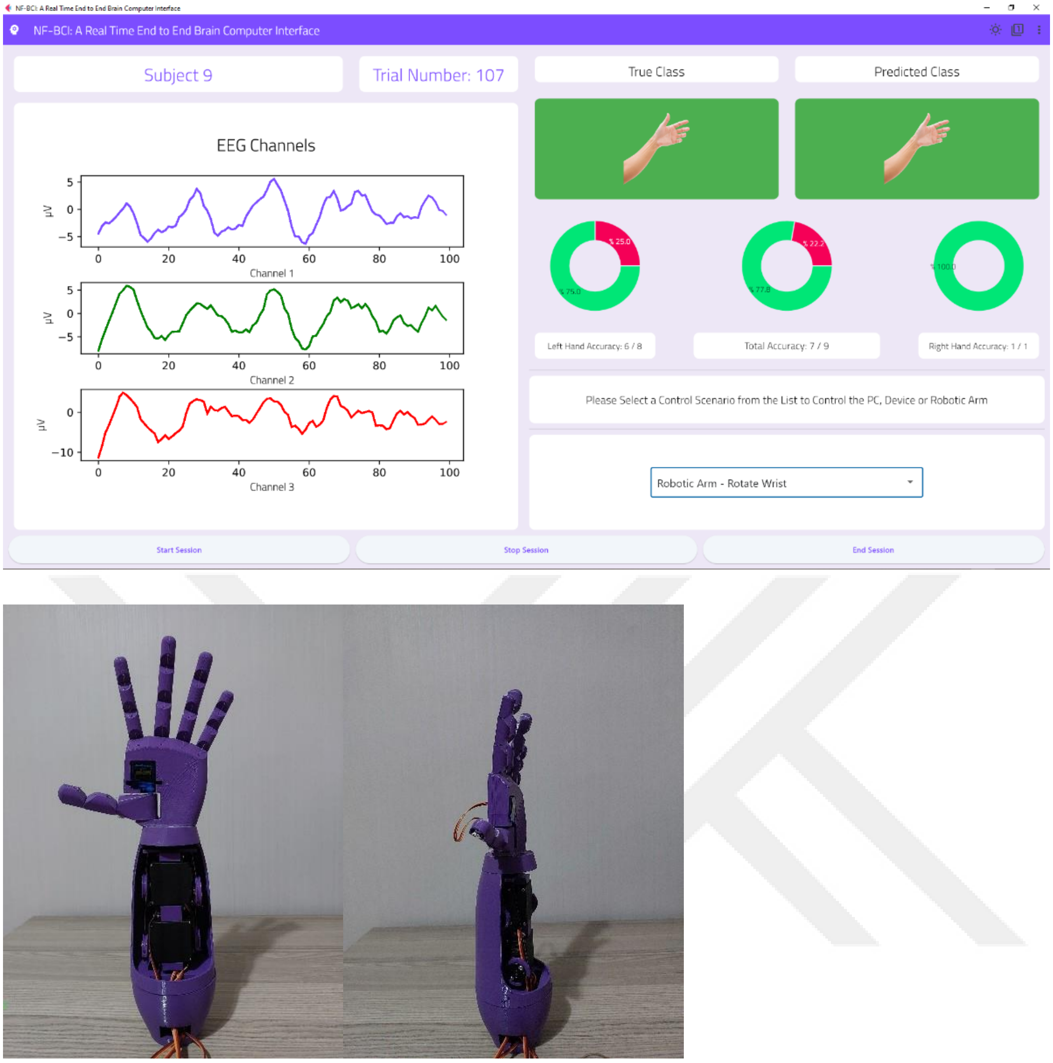
Total Accuracy: 1 / 1

Right Hand Accuracy: 1 / 1

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Rotate Wrist

Figure 6.50. Robotic arm – rotate wrist scenario for right hand classification



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 9

Trial Number: 107

True Class

Predicted Class

Left Hand Accuracy: 6 / 8

Total Accuracy: 7 / 9

Right Hand Accuracy: 1 / 1

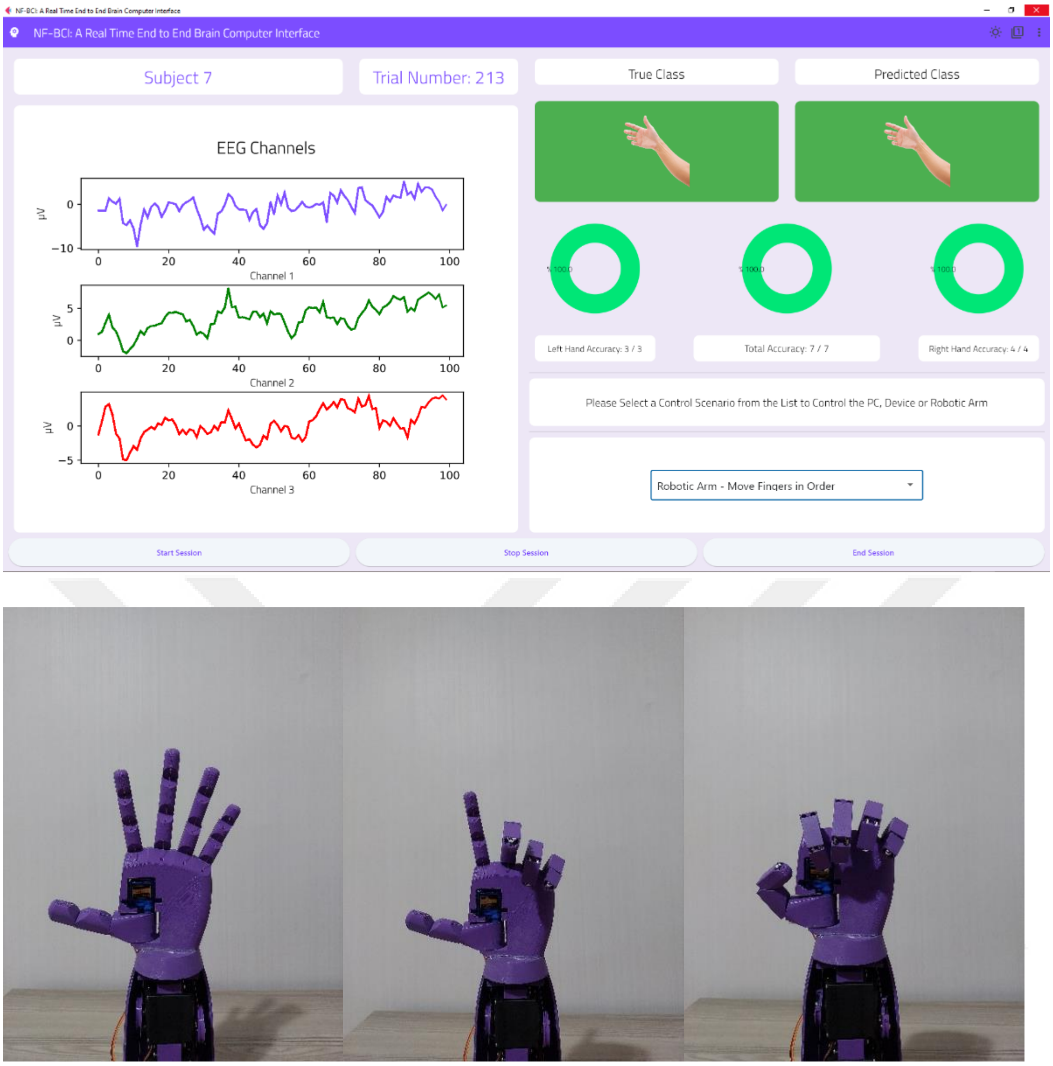
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Rotate Wrist

Figure 6.51. Robotic arm – rotate wrist scenario for left hand classification

6.14.3. Robotic Arm – Move Fingers in Order

In this scenario, NF-BCI software is aimed to close and open all fingers on the robotic arm in sequence. When the classifier working on NF-BCI makes a right hand prediction, all the fingers on the robotic arm are closed in sequence. When the classifier makes a left hand prediction, all fingers on the robotic arm are opened in sequence. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.52 and figure 6.53.



Subject 7

Trial Number: 213

Left Hand Accuracy: 3 / 3

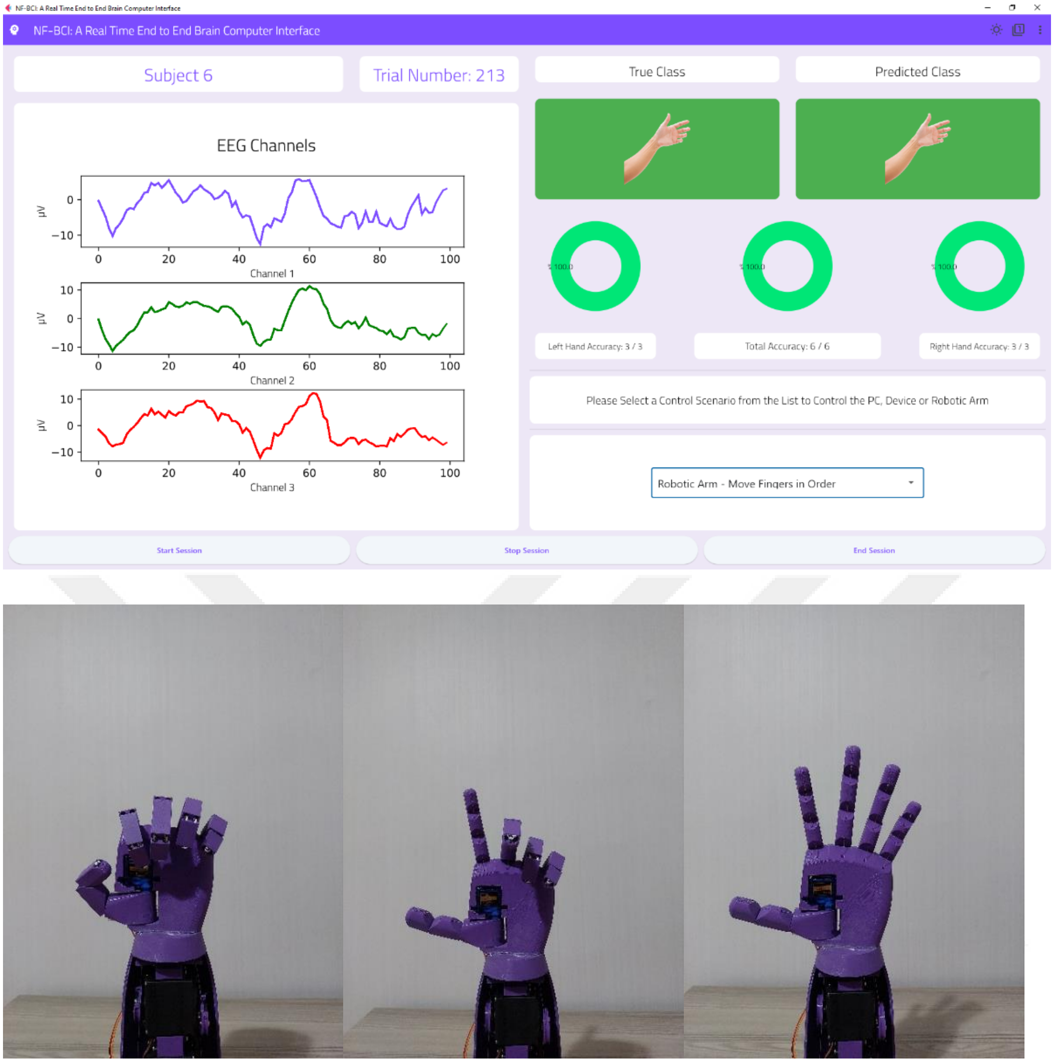
Total Accuracy: 7 / 7

Right Hand Accuracy: 4 / 4

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Move Fingers in Order

Figure 6.52. Robotic arm – move fingers in order scenario for right hand classification



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 6

Trial Number: 213

True Class

Predicted Class

Left Hand Accuracy: 3 / 3

Total Accuracy: 6 / 6

Right Hand Accuracy: 3 / 3

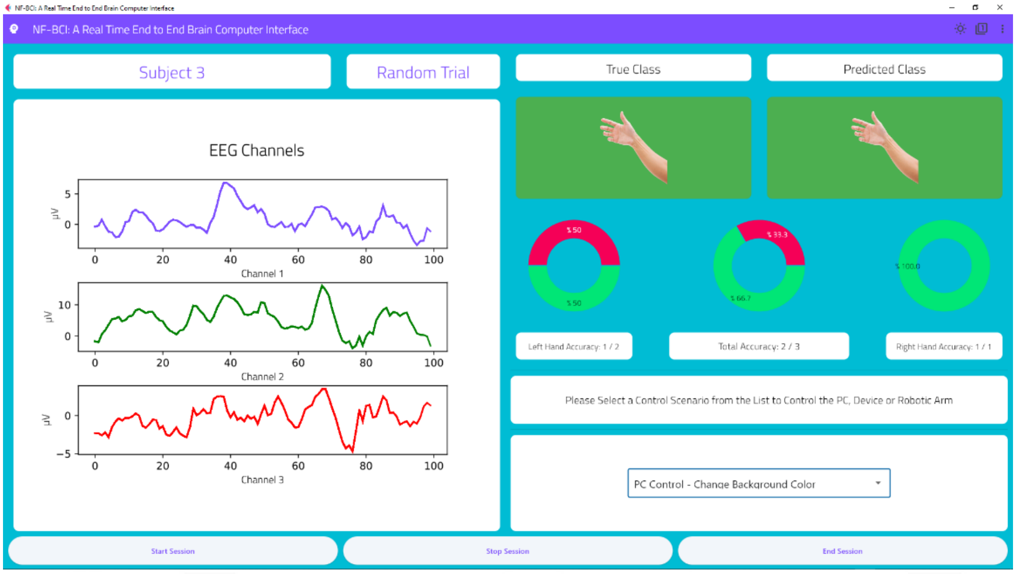
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

Robotic Arm - Move Fingers in Order

Figure 6.53. Robotic arm – move fingers in order scenario for left hand classification

6.14.4. PC Control – Change Background Color

In this scenario, the NF-BCI software is intended to change the color of its interface background. When the classifier working on NF-BCI makes a right hand prediction, the color of the interface background changes to a turquoise color. When the classifier makes left-hand prediction, the interface background color changes to purple. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.54 and figure 6.55.



Left Hand Accuracy: 1 / 2

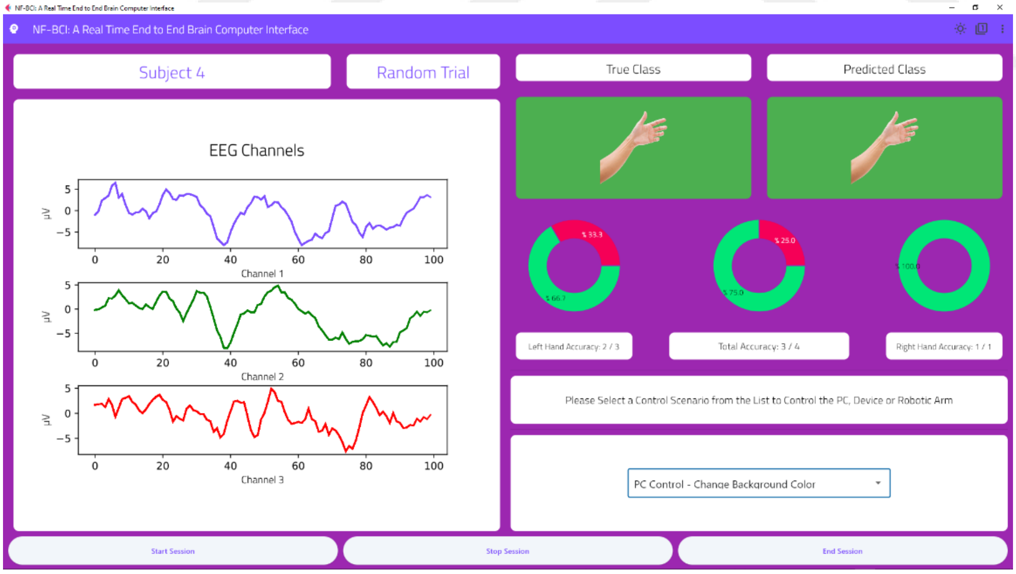
Total Accuracy: 2 / 3

Right Hand Accuracy: 1 / 1

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

PC Control - Change Background Color

Figure 6.54. PC control – change background color scenario for right hand classification



Left Hand Accuracy: 2 / 3

Total Accuracy: 3 / 4

Right Hand Accuracy: 1 / 1

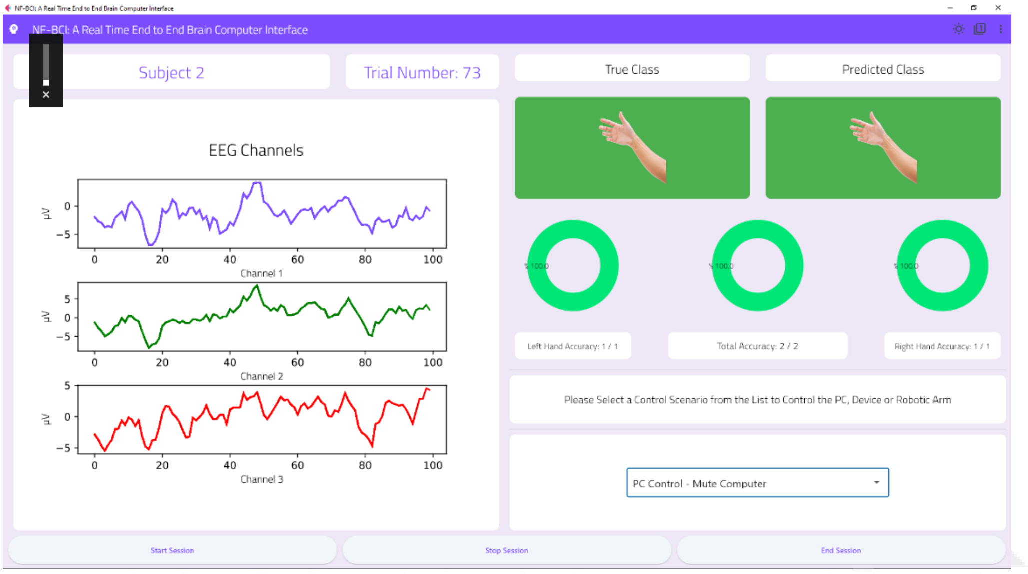
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

PC Control - Change Background Color

Figure 6.55. PC control – change background color scenario for left hand classification

6.14.5. PC Control – Mute Computer

In this scenario, it is aimed that the NF-BCI software will mute and unmute the computer it is running on. When the classifier working on NF-BCI makes a right hand prediction, the computer is muted. When the classifier makes a left-hand prediction, the computer's volume is turned up. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.56 and figure 6.57.



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 2

Trial Number: 73

Left Hand Accuracy: 1 / 1

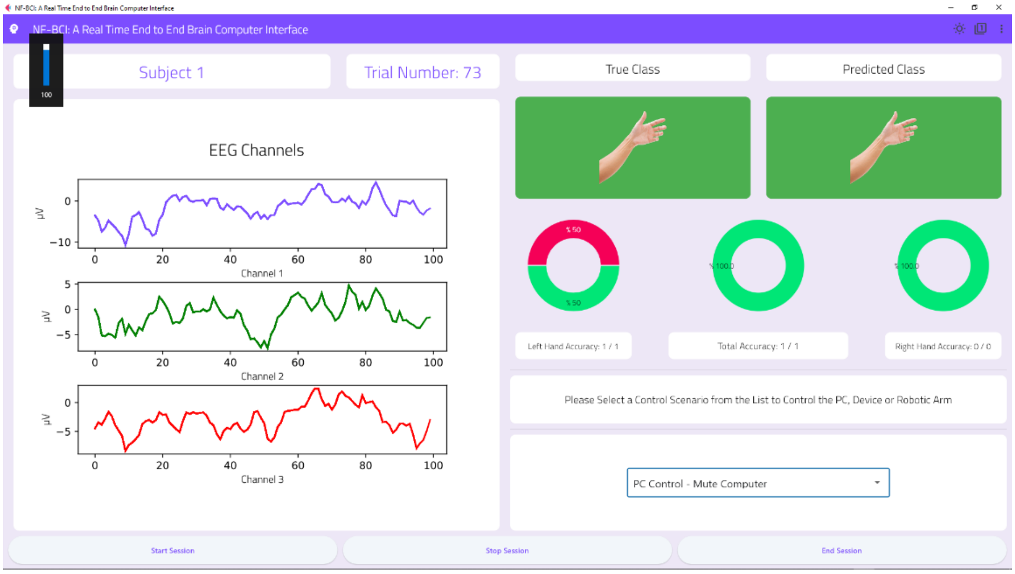
Total Accuracy: 2 / 2

Right Hand Accuracy: 1 / 1

Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

PC Control - Mute Computer

Figure 6.56. PC control – mute computer scenario for right hand classification



NF-BCI: A Real Time End to End Brain Computer Interface

Subject 1

Trial Number: 73

Left Hand Accuracy: 1 / 1

Total Accuracy: 1 / 1

Right Hand Accuracy: 0 / 0

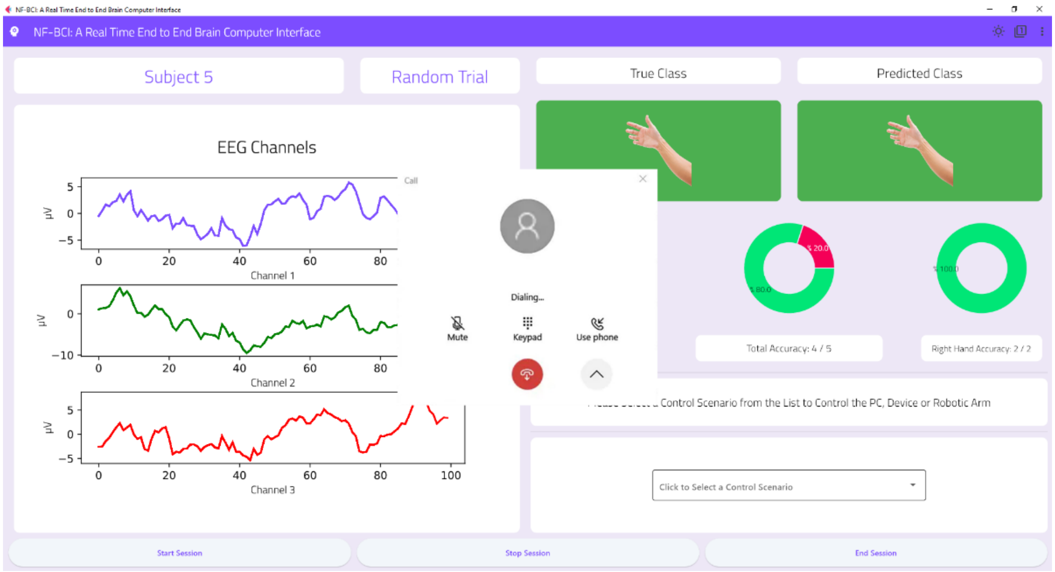
Please Select a Control Scenario from the List to Control the PC, Device or Robotic Arm

PC Control - Mute Computer

Figure 6.57. PC control – mute computer scenario for left hand classification

6.14.6. PC Control – Call Emergency

In this scenario, it is aimed that the NF-BCI software opens and closes the phone application of the computer it is running on. When the classifier working on NF-BCI makes a right hand prediction, the phone application of the computer is opened. Quick search definitions can be made through this application. When the classifier makes a left-hand prediction, the computer's phone application is closed. The results obtained when this scenario is applied while NF-BCI is running are given in figure 6.58.



Total Accuracy: 4 / 5

Right Hand Accuracy: 2 / 2

... Control Scenario from the List to Control the PC, Device or Robotic Arm

Click to Select a Control Scenario

Figure 6.58. PC control – call emergency scenario for right hand classification

7. DISCUSSION AND CONCLUSIONS

7.1. Discussion for IS-EEG

In this study, the datasets were converted into 2D and 3D matrix sizes, and these conversion processes provided a total of 8 different input shapes, 2 different combinations in 2D structure and 6 different combinations in 3D structure. Afterwards, 2D and 3D CNN models were developed to train 2D and 3D input shape structures. These models have the same type of layers, same number of layers and same parameters. The raw data prepared in the input shape structures described in the previous sections were given as direct input to the designed CNN models without any signal preprocessing.

In this section, the results obtained during and after training the datasets with 8 different input shape structures were compared with each other, the pros and cons of the models were presented and the results were evaluated.

The models trained with raw data, without applying any preprocessing methods, showed a classification success between 53.03% and 89.29% for the BCI-IV-2A dataset and between 65.38% and 84.94% for the BCI-IV-2B dataset. Classification accuracies according to the input shape change are shown in figure 7.1.

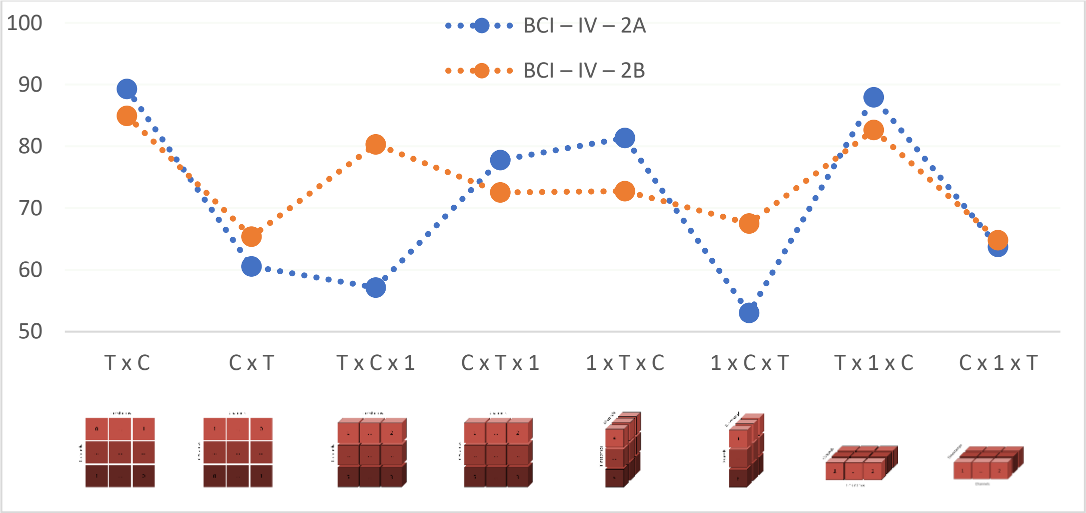


Figure 7.1. Average accuracy values for each input shape.

In addition to these, it has been observed that the different input shapes and the number of channels in the dataset are also effective on the epoch times. Epoch times were measured in the range of 2-23 seconds for BCI-IV-2A and 4-10 seconds for BCI-IV-2B. The epoch times obtained according to the input shape change are shown in figure 7.2.

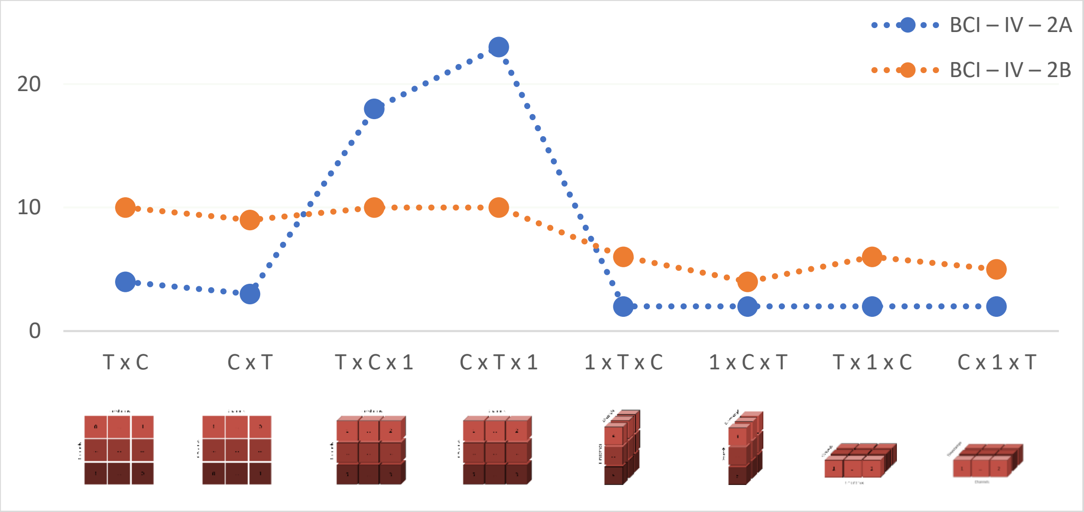


Figure 7.2. Epoch times for each input shape.

With the input shape in the T x C structure, which achieved the best accuracy for the BCI-IV-2A dataset, an average success rate of 89.293% and a standard deviation of 5.471 were obtained. With the input shape in T x C structure, which achieved the best accuracy for the BCI-IV-2B dataset, an average success rate of 84.943% and a standard deviation of 8.607 were obtained. These values obtained with the developed models reveal the high classification success and robustness of the models.

In both datasets, the average accuracy values span a wide range (range value 36.26 for 2A dataset and 20.10 for 2B dataset). It has been observed that some input shapes cause high levels of overfitting during training (figure 6.17, figure 6.21, figure 6.23; figure 6.25, figure 6.29, figure 6.31). Although these input shapes have very high training accuracy values, their validation values are quite low. Input shapes have been observed to have a significant effect on epoch times too (table 6.10). In both datasets, the epoch times are spread over a wide range (range value 21 for dataset 2A and range value 6 for dataset 2B). As a result, it has been determined that input shape structures are a very important parameter in the performance of CNN models (table 6.8, table 6.9).

In both datasets, the highest accuracy values were achieved with T x C and T x 1x C input shapes respectively (figure 7.1). Researchers can train their models by selecting appropriate input shapes for 2D and 3D CNN architectures if needed. In both datasets, it has been observed that the changes in the accuracy and epoch time values obtained as the input shape changes are parallel (figure 7.1, figure 7.2). However, it cannot be concluded that a given input shape will achieve the same success in every CNN model and every dataset to be used (figure 7.1). For this reason, it was concluded that researchers should try different input shape structures for the CNN models they set up.

In the BCI-IV-2A dataset, both higher average accuracy values and lower standard deviations were obtained compared to the 2D dataset (table 6.12). When the training and validation graphs were examined, it was observed that the models trained with the 2A dataset learned faster and reached lower loss values than the models trained with the 2B dataset (figure 6.16 - 6.23, figure 6.24 – figure 6.31). The decisive reason for this is that the 2A dataset has 22 EEG channels, while the 2B dataset has 3 channels.

To deal with the possible covariate shift, we used batch normalization layers along the path of the data. Instead of normalizing all the data outside the model and giving it as input to the model, we started the model with batch normalization. In this way, each batch entering the model was normalized only within itself, not all data together. The data was then propagated through layers. When the data enters a new layer, it is processed here and the 0 mean and unit variance distributions change. In order to avoid covariate shifts that may arise as a result of these changes, we used a batch normalization layer after each convolutional layer and ensured that the data were normalized throughout the training in the model. Since each batch entering the model may show a very different distribution from the previous or next batches, we have kept the feature values of the batches on the same scale in order to achieve high classification success with a more robust and stable model against possible covariate shifts.

In models with high classification accuracy, we predict that the model establishes a relationship between different channels, as EEG channels are scanned one by one depending on time during the training of input shapes. In this way, we predict that higher classification accuracies are obtained by scanning the relationships between the EEG channels of the test data entering the models, and that the models are more stable and invariant against time-dependent feature changes. In models with low classification accuracy, we predict that model scan channels together in a single time period during training and establishing relationships between time periods rather than between channels. In this way, we predict that models looking for relationships between certain time periods give low test accuracy values against nonstationary EEG signals whose time-dependent properties are highly variable.

Large amounts of training data are needed to achieve high classification success using deep neural networks. However, collecting large amounts of motor imagery signals is quite challenging. Pre-training and experience are needed to collect these signals. In addition, the subjects' inability to keep their attention at the same level during the signal acquisition session, fatigue and the presence of environmental disrupting factors make this process even more difficult.

In order to overcome this problem, researchers have used data augmentation methods that create new data and increase existing dataset systematically. Data augmentation can reduce the tendency of overfitting and increase the classification success and stability of the model. It can also enable models trained with nonstationary EEG data to make fewer mistakes, be invariant and increase robustness when encountering new test data [226,227,228]. Signal acquisition processes can take days, weeks and even months, and data augmentation applications have shown very promising results compared to the results obtained as a result of these signal acquisition processes [229,230] These gains may also reduce the time and funds that researchers allocate to collect signals [195].

Our CNN model, IS-EEG, which was trained with raw data without signal preprocessing methods, achieved accuracy values close to or better than many state-of-the-art models with the correct input shape and correct CNN architecture [159,172,174,220,224,225,231-234]. Accordingly, it has been shown that by establishing the right CNN architecture and choosing the right input shape structure, feature extraction of EEG motor imagery signals can be done successfully with raw data. Thus, the loss of possible distinguishing features in sensitive EEG signals is prevented and much faster and more successful results can be obtained by saving time and effort.

7.2. Discussion for NF-EEG

Almost all of the existing EEG based motor imagery classification methods have applied a wide variety of signal preprocessing methods to a greater or lesser extent and have developed their models in this way. At the same time, various data augmentation methods have been developed to enlarge the limited datasets. However, signal preprocessing methods are largely based on human knowledge and experience, and the development and implementation of these methods causes a great loss of labor and time. In addition, inappropriate or incorrectly applied signal preprocessing methods greatly affect and reduce the classification success. However, since there may be unknown relations and patterns in the dataset, it is not possible to perform preprocessing operations to detect these patterns without having this pattern information.

For the reasons mentioned above, we developed the NF-EEG model, which trains and classifies directly with raw data without applying any signal preprocessing, and with this model, we achieved higher classification success than state-of-the-art models with or without signal preprocessing. In addition, we increased the classification success with input reshaping and data augmentation methods. Along with all this, we have reduced the amount of effort and time required to achieve a notable classification result by automating the EEG signal classification processes, which are very difficult to analyze and to extract features, and also require a lot of effort and time.

7.3. Conclusions

With this study, we outperformed many state-of-the-art models by achieving a classification success of 93.56% in the two-class BCI Competition IV 2A dataset, a classification success of 88.40% in the two-class BCI Competition IV 2B dataset, and a classification success of 81.05% in the four-class BCI Competition IV 2A dataset by developing NF-EEG model. We have shown that high classification successes can be achieved with the right model architecture, appropriate hyperparameter selection and efficient input shape structure by using raw data without applying signal preprocessing methods.

NF-EEG is also emerging as an option for the development and implementation of real-time BCI systems. It is generalizable, robust, and fast with high classification accuracies, low standard deviation rates, and fast evaluation times which needed for these systems.

In this thesis, also the effect of input shape structures on the performance of CNN models used in the classification of EEG motor imagery signals is shown with quantitative data. The obtained results showed that even if researchers develop correct CNN models during classification of EEG signals, if they choose the inaccurate input shape, they can achieve poor classification success. This may cause researchers to disrupt the correct models they have established and may increase the time and energy they will spend to build a model. In order to prevent this situation, researchers can increase the classifier performance by trying different input shape structures and minimize the losses they may experience due to this parameter.

Training and classification with raw signals can let researchers to develop real-time fast, reliable and robust models with high accuracy values. Since signal preprocessing methods are not used, possible data loss that may occur in these processes can be pre-vented. In addition, pre-training processes can be shortened by automating feature extraction processes. Along with these advantages, there are also some disadvantages. Deep learning models need a large amount of data and the difficulty of collecting EEG signals may lead to the need to develop data augmentation methods. In addition, hyperparameter optimization may be required for newly designed artificial neural network models to achieve high classification success, which may cause time and computational costs.

These advantages and disadvantages create some future research opportunities. Performance gains can be achieved quickly by applying fine tuning and hyperparameter optimization to deep learning models that are previously designed and used by researchers. In addition, new neural network layers and architectures can be designed to detect EEG features accurately and quickly, and the right parameters for models can be searched. By developing data augmentation methods, it is possible to obtain the required large amount of data. Also, previously or newly developed robust and fast models can be used and tested in real-time systems.

In addition to all these, with a correctly constructed CNN model, high classification successes can be achieved without the need for laborious and time-consuming signal preprocessing to extract distinctive features from EEG signals. With the CNN model developed in this study and using raw data, close or better results were obtained than the models that achieved high classification success using signal preprocessing.

It is predicted that this developed methodology will be useful in the development and use of real-time brain-computer interfaces, as it does not need signal preprocessing, can train the system quickly with new data, and is robust with low standard deviation values.

The NF-BCI software emerged as a program with the capabilities to visualize and classify brain signals in real time, and to control the generated robotic arm by connecting with a microcontroller. While this software can use NF-EEG and IS-EEG models in the background, it offers a flexible structure where different classifier models can be integrated. It can also connect with the computer it is working on and a microcontroller. With this connection, it can control the computer or an external device it is connected to via a microcontroller. These control scenarios can be pre-designed and selected via the interface. It also stands out as a research platform for researchers who want to test a dataset or classifier model with its ability to visualize the signals entering the system and display real-time classification results. The real-time performance of the CNN models and interface software developed with the produced robotic arm has been tested. Control signals coming from the interface have been successfully applied and computer and robotic arm control scenarios have been fulfilled. These control scenarios can be adapted to the needs of patients with basic care needs or adapted to the needs of healthy people who want to gain environmental control ability and new skills, and desired scenarios can be realized.

By combining all these developed models, methods and software, an end-to-end brain-computer interface operating in real time has been developed. This interface fulfilled the defined tasks and enabled the thesis to reach its goals. In addition, the NF-BCI interface is a platform open to development with its flexible structure. It is planned to add new modules to this interface in future studies. Some of these modules are modules that allow users to upload their own datasets and classifier models and provide signal acquisition support for various devices.

However, it is clear that there is still a long way to go before the robust implementation of brain computer interface control scenarios. Sensitive brain signals from users can change very quickly and these signals deteriorate very quickly, making it very difficult to maintain continuous and robust control scenarios. Against these rapid changes, the system needs to be integrated quickly. To overcome this problem, various security layers can be developed in software and certain security layers and protocols can be added between signals from users and control tasks. Failure to perform control operations without passing these security layers can prevent false control operations or undesired control scenarios. This field of study also emerges as a crucial area of progress for brain-computer interfaces.

REFERENCES

[1] Meng, J., Zhang, S., Bekyo, A., Olsoe, J., Baxter, B., & He, B. (2016). Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks. *Scientific Reports, 6*(1), 38565. https://doi.org/10.1038/srep38565

[2] Liu, Z., Shore, J., Wang, M., Yuan, F., Buss, A., & Zhao, X. (2021). A systematic review on hybrid EEG/fNIRS in brain-computer interface. *Biomedical Signal Processing and Control*, *68*, 102595. https://doi.org/10.1016/j.bspc.2021.102595

[3] Bhagat, N. A., Venkatakrishnan, A., Abibullaev, B., Artz, E. J., Yozbatiran, N., Blank, A. A., ... & Contreras-Vidal, J. L. (2016). Design and optimization of an EEG-based brain machine interface (BMI) to an upper-limb exoskeleton for stroke survivors. *Frontiers in neuroscience, 10*, 122. https://doi.org/10.3389/fnins.2016.00122

[4] Aldayel, M., Ykhlef, M., & Al-Nafjan, A. (2020). Deep learning for EEG-based preference classification in neuromarketing. *Applied Sciences, 10*(4), 1525. https://doi.org/10.3390/app10041525

[5] He, Y., Eguren, D., Azorín, J. M., Grossman, R. G., Luu, T. P., & Contreras-Vidal, J. L. (2018). Brain–machine interfaces for controlling lower-limb powered robotic systems. *Journal of neural engineering, 15*(2), 021004. https://doi.org/10.1088/1741-2552/aaa8c0

[6] Nacy, S. M., Kbah, S. N., Jafer, H. A., & Al-Shaalan, I. (2016). Controlling a servo motor using EEG signals from the primary motor cortex. *American Journal of Biomedical Engineering, 6*(5), 139-146. https://doi.org/10.5923/j.ajbe.20160605.02

[7] Al-Qaraawi, S., Croock, M. S., & Alawi, S. H. (2018). Electroencephalography Signals Based Face Interaction for Directional Control System. International *Journal of Computing and Network Technology, 6*(2). https://doi.org/10.12785/ijcnt/060203

[8] Thut, G., & Miniussi, C. (2009). New insights into rhythmic brain activity from TMS–EEG studies. *Trends in cognitive sciences, 13*(4), 182-189. https://doi.org/10.1016/j.tics.2009.01.004

[9] Oblique view of male head showing electrode placement on the scalp for an EEG (Electroencephalogram); SOURCE: Original; M_Brain_skull_EEG.mb

[10] FitMind (2021). Brainwaves: Altered States & Technologies. https://fitmind.com/blog-collection/brainwaves-in-meditation-brain-wave-frequencies

[11] Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics, 65*(2), 413. https://doi.org/10.1103/RevModPhys.65.413

[12] Khalighi, M., Vahdat, B. V., Mortazavi, M., Hy, W., & Soleimani, M. (2012, May). Practical design of low-cost instrumentation for industrial electrical impedance tomography (EIT). In *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings* (pp. 1259-1263). IEEE. https://doi.org/10.1109/I2MTC.2012.6229173.

[13] NCCR Evolving Language & Human Neuroscience Platform (2023). Geneva at the heart of the human brain. https://evolvinglanguage.ch/geneva-at-the-heart-of-the-human-brain/

[14] Fatema, U. (2019). Motor Imagery Signal Classification using EEG and ECoG signal for Brain Computer Interface. *International IEEE/EMBS Conference on Neural Engineering*, NER 2019

[15] Abdulwahab, S. S., Khleaf, H. K., Jassim, M. H., & Abdulwahab, S. (2020). A Systematic Review of Brain-Computer Interface Based EEG. *Iraqi J. Electr. Electron. Eng, 16*(2), 1-10. https://doi.org/10.37917/ijeee.16.2.9

[16] Wang, M., Li, R., Zhang, R., Li, G., & Zhang, D. (2018). A wearable SSVEP-based BCI system for quadcopter control using head-mounted device. *Ieee Access, 6*, 26789-26798. https://doi.org/10.1109/ACCESS.2018.2825378

[17] Blausen Medical. (2014). Medical gallery of Blausen medical 2014. *WikiJournal of Medicine, 1*(2), 1-79. https://doi.org/10.15347/wjm/2014.010

[18] Homer, M. L., Nurmikko, A. V., Donoghue, J. P., & Hochberg, L. R. (2013). Sensors and decoding for intracortical brain computer interfaces. *Annual review of biomedical engineering, 15,* 383-405. https://doi.org/10.1146/annurev-bioeng-071910-124640

[19] Wolpaw, J.R. and Wolpaw, E.W. (2012). *Brain-Computer interfaces: Principles and Practice*. New York: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195388855.001.0001

[20] Menon, V., & Crottaz-Herbette, S. (2005). Combined EEG and fMRI studies of human brain function. *International review of neurobiology, 66,* 291-321. https://doi.org/10.1016/S0074-7742(05)66010-2

[21] Elliott, M. L., Knodt, A. R., Ireland, D., Morris, M. L., Poulton, R., Ramrakha, S., ... & Hariri, A. R. (2020). What is the test-retest reliability of common task-functional MRI measures? New empirical evidence and a meta-analysis. *Psychological science, 31*(7), 792-806. https://doi.org/10.1177/095679762091678

[22] Hong, K. S., Khan, M. J., & Hong, M. J. (2018). Feature extraction and classification methods for hybrid fNIRS-EEG brain-computer interfaces. *Frontiers in human neuroscience, 12,* 246. https://doi.org/10.3389/fnhum.2018.00246

[23] Khan, M. J., & Hong, K. S. (2017). Hybrid EEG–fNIRS-based eight-command decoding for BCI: application to quadcopter control. *Frontiers in neurorobotics, 11,* 6. https://doi.org/10.3389/fnbot.2017.00006

[24] Shimadzu Europa (2022). LABNIRS. https://www.shimadzu.eu/labnirs

[25] Witkowski, M., Cortese, M., Cempini, M., Mellinger, J., Vitiello, N., & Soekadar, S. R. (2014). Enhancing brain-machine interface (BMI) control of a hand exoskeleton using electrooculography (EOG). *Journal of neuroengineering and rehabilitation, 11*(1), 1-6. https://doi.org/10.1186/1743-0003-11-165

[26] Soekadar, S. R., Witkowski, M., Vitiello, N., & Birbaumer, N. (2015). An EEG/EOG-based hybrid brain-neural computer interaction (BNCI) system to control an exoskeleton for the paralyzed hand. *Biomedical Engineering/Biomedizinische Technik, 60*(3), 199-205. https://doi.org/10.1515/bmt-2014-0126

[27] Heath, J. (2017). Basics of bandpass filters. https://www.analogictips.com/basics-of-bandpass-filters/

[28] Everything RF (2021). What is a Notch Filter? https://www.everythingrf.com/community/what-is-a-notch-filter

[29] Zelko, J. (2020). Implementation of 1 Dimensional Filtering Using Median Filter. https://github.com/JuliaDSP/DSP.jl/issues/343

[30] Nguyen, K. A., Wang, Y., Li, G., Luo, Z., & Watkins, C. (2019). Realtime tracking of passengers on the london underground transport by matching smartphone accelerometer footprints. *Sensors, 19*(19), 4184. https://doi.org/10.3390/s19194184

[31] Zargari Marandi, R., & Sabzpoushan, S. H. (2015). Qualitative modeling of the decision-making process using electrooculography. *Behavior research methods, 47*, 1404-1412. https://doi.org/10.3758/s13428-014-0549-9

[32] Barina, D. (2010). Signal denoising by wavelet transform thresholding. https://en.m.wikipedia.org/wiki/File:Wavelet_denoising.svg

[33] Ghosh, S. (2019). Principal Component Analysis in R – Walk Through. https://dimensionless.in/wp-content/uploads/2019/07/pca2.png

[34] Parietal (2021). Faster independent component analysis for real data. https://team.inria.fr/parietal/research/statistical-and-machine-learning-methods-for-large-scale-data/faster-independent-component-analysis-for-real-data/

[35] Tariq, M., Trivailo, P. M., & Simic, M. (2019). Classification of left and right foot kinaesthetic motor imagery using common spatial pattern. *Biomedical Physics & Engineering Express, 6*(1), 015008. https://doi.org/10.1088/2057-1976/ab54ad

[36] Lawhern, V., Hairston, W. D., McDowell, K., Westerfield, M., & Robbins, K. (2012). Detection and classification of subject-generated artifacts in EEG signals using autoregressive models. *Journal of neuroscience methods, 208*(2), 181-189. https://doi.org/10.1016/j.jneumeth.2012.05.017

[37] Lai, S. M., Zhang, Z., Hung, Y. S., Niu, Z., & Chang, C. (2011). A chromatic transient visual evoked potential based encoding/decoding approach for brain–computer interface. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 1*(4), 578-589. https://doi.org/10.1109/JETCAS.2011.2178734

[38] Sharma, R. K. (2017, April). EEG signal denoising based on wavelet transform. In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)* (Vol. 1, pp. 758-761). IEEE. https://doi.org/10.1109/ICECA.2017.8203645

[39] Gluth, S., Rieskamp, J., & Büchel, C. (2013). Deciding not to decide: computational and neural evidence for hidden behavior in sequential choice. *PLoS computational biology, 9*(10), e1003309. https://doi.org/10.1371/journal.pcbi.1003309

[40] Luck, S. J. (2012). Event-related potentials. https://doi.org/10.1037/13619-028

[41] Pfaff, D. W., Volkow, N. D., & Rubenstein, J. L. (Eds.). (2022). *Neuroscience in the 21st century: from basic to clinical*. Springer Nature. https://doi.org/10.1007/978-3-030-88832-9

[42] Duhamel, P., & Vetterli, M. (1990). Fast Fourier transforms: a tutorial review and a state of the art. *Signal processing, 19*(4), 259-299. https://doi.org/10.1016/0165-1684(90)90158-U

[43] Kuremoto, T., Obayashi, M., Mabu, S., & Kobayashi, K. (2017). Mental task recognition by EEG Signals: A novel approach with ROC analysis. *Human-Robot Interaction-Theory and Application,* 65-78. https://doi.org/10.5772/intechopen.71743

[44] Mahato, S., & Paul, S. (2019). Electroencephalogram (EEG) signal analysis for diagnosis of major depressive disorder (MDD): a review. *Nanoelectronics, Circuits and Communication Systems: Proceeding of NCCS 2017*, 323-335. https://doi.org/10.1007/978-981-13-0776-8_30

[45] Haider, A. (2021, May). A brief review of signal processing for EEG-based BCI: Approaches and opportunities. In *2021 IEEE International Conference on Electro Information Technology (EIT)* (pp. 389-394). IEEE. https://doi.org/10.1109/EIT51626.2021.9491924

[46] Yang, X. (2020). Linear Discriminant Analysis, Explained. https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b

[47] Shoka, A., Dessouky, M., El-Sherbeny, A., & El-Sayed, A. (2019). Literature review on EEG preprocessing, feature extraction, and classifications techniques. *Menoufia J. Electron. Eng. Res, 28*(1), 292-299. https://doi.org/10.21608/mjeer.2019.64927

[48] Toth, G. (2023). Linear and Quadratic Discriminant Analysis. https://www.datasklr.com/select-classification-methods/linear-and-quadratic-discriminant-analysis

[49] Quitadamo, L. R., Cavrini, F., Sbernini, L., Riillo, F., Bianchi, L., Seri, S., & Saggio, G. (2017). Support vector machines to detect physiological patterns for EEG and EMG-based human–computer interaction: a review. *Journal of neural engineering, 14*(1), 011001. https://doi.org/10.1088/1741-2552/14/1/011001

[50] Shetty, S. H., Shetty, S., Singh, C., & Rao, A. (2022). Supervised Machine Learning: Algorithms and Applications. *Fundamentals and Methods of Machine and Deep Learning: Algorithms, Tools and Applications,* 1-16. https://doi.org/10.1002/9781119821908.ch1

[51] Bhavsar, H., & Panchal, M. H. (2012). A review on support vector machine for data classification. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1*(10), 185-189.

[52] Saini, A. (2023). Guide on Support Vector Machine (SVM) Algorithm. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[53] Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. S. (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data, 7*(4), 221-248. https://doi.org/10.1089/big.2018.0175

[54] Toussaint, G. T. (2002). Proximity graphs for nearest neighbor decision rules: recent progress. *Interface, 34*.

[55] Pal, N. (2020). K-Nearest Neighbors(K-NN) algorithm from scratch with a hands-on example (in R). https://insightimi.wordpress.com/2020/03/01/k-nearest-neighboursk-nn-algorithm-from-scratch-with-a-hands-on-example-in-r/

[56] Su, Y. (2021, January). A Review on Adaptive Classifiers for BCI Classification. *In Proceedings of the 2021 International Conference on Bioinformatics and Intelligent Computing* (pp. 303-307). https://doi.org/10.1145/3448748.3448797

[57] Congedo, M., Barachant, A., & Bhatia, R. (2017). Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces, 4*(3), 155-174. https://doi.org/10.1080/2326263X.2017.1297192

[58] Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. Journal of neural engineering, 15(3), 031005. https://doi.org/10.1088/1741-2552/aab2f2

[59] Zou, J., Han, Y., & So, S. S. (2009). Overview of artificial neural networks. *Artificial neural networks: methods and applications*, 14-22. https://doi.org/10.1007/978-1-60327-101-1_2

[60] Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer, 29*(3), 31-44. https://doi.org/10.1109/2.485891

[61] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature, 521*(7553), 436-444. https://doi.org/10.1038/nature14539

[62] Ebrahimi, Z., Loni, M., Daneshtalab, M., & Gharehbaghi, A. (2020). A review on deep learning methods for ECG arrhythmia classification. *Expert Systems with Applications: X, 7,* 100033. https://doi.org/10.1016/j.eswax.2020.100033

[63] Lee, J. G., Jun, S., Cho, Y. W., Lee, H., Kim, G. B., Seo, J. B., & Kim, N. (2017). Deep learning in medical imaging: general overview. *Korean journal of radiology*, *18*(4), 570-584. https://doi.org/10.3348/kjr.2017.18.4.570

[64] IBM Data and AI Team (2023). AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference? https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

[65] Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 1-40. https://doi.org/10.1186/s40537-016-0043-6

[66] Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, *11*(2), 40. https://doi.org/10.3390/technologies11020040

[67] Shalbaf, A., Bagherzadeh, S., & Maghsoudi, A. (2020). Transfer learning with deep convolutional neural network for automated detection of schizophrenia from EEG signals. *Physical and Engineering Sciences in Medicine, 43,* 1229-1239. https://doi.org/10.1007/s13246-020-00925-9

[68] Regan, D. (1989). Evoked potentials and evoked magnetic fields in science and medicine. *Human brain electrophysiology*, 59-61.

[69] Wang, Y., Wang, R., Gao, X., Hong, B., & Gao, S. (2006). A practical VEP-based brain-computer interface. *IEEE Transactions on neural systems and rehabilitation engineering*, *14*(2), 234-240. https://doi.org/10.1109/TNSRE.2006.875576

[70] Bin, G., Gao, X., Wang, Y., Hong, B., & Gao, S. (2009). VEP-based brain-computer interfaces: time, frequency, and code modulations [Research Frontier]. *IEEE Computational Intelligence Magazine,* *4*(4), 22-26. https://doi.org/10.1109/MCI.2009.934562

[71] Markand, O. N., & Markand, O. N. (2020). Visual evoked potentials. *Clinical Evoked Potentials: An Illustrated Manual*, 83-137. https://doi.org/10.1007/978-3-030-36955-2_3

[72] Birbaumer, N., Elbert, T., Canavan, A. G., & Rockstroh, B. (1990). Slow potentials of the cerebral cortex and behavior. *Physiological reviews, 70*(1), 1-41. https://doi.org/10.1152/physrev.1990.70.1.1

[73] Hinterberger, T., Schmidt, S., Neumann, N., Mellinger, J., Blankertz, B., Curio, G., & Birbaumer, N. (2004). Brain-computer communication and slow cortical potentials. *IEEE Transactions on Biomedical Engineering, 51*(6), 1011-1018. https://doi.org/10.1109/TBME.2004.827067

[74] Farwell, L. A., & Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology, 70*(6), 510-523. https://doi.org/10.1016/0013-4694(88)90149-6

[75] Donchin, E., & Smith, D. B. (1970). The contingent negative variation and the late positive wave of the average evoked potential. *Electroencephalography and clinical Neurophysiology, 29*(2), 201-203. https://doi.org/10.1016/0013-4694(70)90124-0

[76] Ravden, D., & Polich, J. (1999). On P300 measurement stability: habituation, intra-trial block variation, and ultradian rhythms. *Biological psychology, 51*(1), 59-76. https://doi.org/10.1016/S0301-0511(99)00015-0

[77] Jeannerod, M. (1995). Mental imagery in the motor context. *Neuropsychologia, 33*(11), 1419-1432. https://doi.org/10.1016/0028-3932(95)00073-C

[78] Pfurtscheller, G., Neuper, C., Flotzinger, D., & Pregenzer, M. (1997). EEG-based discrimination between imagination of right and left hand movement. *Electroencephalography and clinical Neurophysiology, 103*(6), 642-651. https://doi.org/10.1016/S0013-4694(97)00080-1

[79] Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE, 89*(7), 1123-1134. https://doi.org/10.1109/5.939829

[80] Blankertz, B., Sannelli, C., Halder, S., Hammer, E. M., Kübler, A., Müller, K. R., ... & Dickhaus, T. (2010). Neurophysiological predictor of SMR-based BCI performance. *Neuroimage, 51*(4), 1303-1309. https://doi.org/10.1016/j.neuroimage.2010.03.022

[81] Bai, O., Rathi, V., Lin, P., Huang, D., Battapady, H., Fei, D. Y., ... & Hallett, M. (2011). Prediction of human voluntary movement before it occurs. *Clinical Neurophysiology, 122*(2), 364-372. https://doi.org/10.1016/j.clinph.2010.07.010

[82] Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *sensors, 12*(2), 1211-1279. https://doi.org/10.3390/s120201211

[83] Fazel-Rezai, R., & Abhari, K. (2009). A region-based P300 speller for brain-computer interface. *Canadian Journal of Electrical and Computer Engineering, 34*(3), 81-85. https://doi.org/10.1109/CJECE.2009.5443854

[84] Furdea, A., Halder, S., Krusienski, D. J., Bross, D., Nijboer, F., Birbaumer, N., & Kübler, A. (2009). An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology, 46*(3), 617-625. https://doi.org/10.1111/j.1469-8986.2008.00783.x

[85] Halder, S., Rea, M., Andreoni, R., Nijboer, F., Hammer, E. M., Kleih, S. C., ... & Kübler, A. (2010). An auditory oddball brain–computer interface for binary choices. *Clinical Neurophysiology, 121*(4), 516-523. https://doi.org/10.1016/j.clinph.2009.11.087

[86] Klobassa, D. S., Vaughan, T. M., Brunner, P., Schwartz, N. E., Wolpaw, J. R., Neuper, C., & Sellers, E. W. (2009). Toward a high-throughput auditory P300-based brain–computer interface. *Clinical neurophysiology, 120*(7), 1252-1261. https://doi.org/10.1016/j.clinph.2009.04.019

[87] Kübler, A., Furdea, A., Halder, S., Hammer, E. M., Nijboer, F., & Kotchoubey, B. (2009). A brain–computer interface controlled auditory event-related potential (P300) spelling system for locked-in patients. *Annals of the New York Academy of Sciences, 1157*(1), 90-100. https://doi.org/10.1111/j.1749-6632.2008.04122.x

[88] Hanafiah, Z. M., Taib, M. N., & Hamid, N. H. A. (2010, December). EEG pattern of smokers for Theta, Alpha and Beta band frequencies. In *2010 IEEE Student Conference on Research and Development (SCOReD)* (pp. 320-323). IEEE. https://doi.org/10.1109/SCORED.2010.5704025

[89] Di, W., Zhihua, C., Ruifang, F., Guangyu, L., & Tian, L. (2010, July). Notice of retraction: Study on human brain after consuming alcohol based on EEG signal. In *2010 3rd International Conference on Computer Science and Information Technology* (Vol. 5, pp. 406-409). IEEE. https://doi.org/10.1109/ICCSIT.2010.5564084

[90] Ek, Z., Akg, A., & Bozkurt, M. R. (2013). The classificaton of eeg signals recorded in drunk and non-drunk people. *International Journal of Computer Applications*, *68*(10). https://doi.org/10.5120/11619-7018

[91] Sriraam, N. (2012, February). EEG based detection of alcoholics using spectral entropy with neural network classifiers. In *2012 International Conference on Biomedical Engineering (ICoBE)* (pp. 89-93). IEEE. https://doi.org/10.1109/ICoBE.2012.6178961

[92] Malar, E., Gauthaam, M., & Chakravarthy, D. (2011). A novel approach for the detection of drunken driving using the power spectral density analysis of EEG. *International Journal of Computer Applications, 21*(7), 10-14. https://doi.org/10.5120/2525-3436

[93] Lin, C. T., Tsai, S. F., & Ko, L. W. (2013). EEG-based learning system for online motion sickness level estimation in a dynamic vehicle environment. *IEEE transactions on neural networks and learning systems, 24*(10), 1689-1700. https://doi.org/10.1109/TNNLS.2013.2275003

[94] Wei, C. S., Chuang, S. W., Wang, W. R., Ko, L. W., Jung, T. P., & Lin, C. T. (2011). Implementation of a motion sickness evaluation system based on EEG spectrum analysis. In *2011 IEEE International Symposium of Circuits and Systems (ISCAS)* (pp. 1081-1084). IEEE. https://doi.org/10.1109/ISCAS.2011.5937757

[95] Bagchi, S., & Chattopadhyay, M. (2012). An easy-to-adopt approach for regular and routine monitoring of the consciousness level of human brain of stayed alone sick person. In *2012 Sixth International Conference on Sensing Technology (ICST)* (pp. 698-703). IEEE. https://doi.org/10.1109/ICSensT.2012.6461767

[96] Bi, L., Fan, X. A., & Liu, Y. (2013). EEG-based brain-controlled mobile robots: a survey. *IEEE transactions on human-machine systems, 43*(2), 161-176. https://doi.org/10.1109/TSMCC.2012.2219046

[97] Barbosa, A. O., Achanccaray, D. R., & Meggiolaro, M. A. (2010, May). Activation of a mobile robot through a brain computer interface. In *2010 IEEE International Conference on Robotics and Automation* (pp. 4815-4821). IEEE. https://doi.org/10.1109/ROBOT.2010.5509150

[98] Ang, K. K., Guan, C., Chua, K. S. G., Ang, B. T., Kuah, C., Wang, C., ... & Zhang, H. (2010, August). Clinical study of neurorehabilitation in stroke using EEG-based motor imagery brain-computer interface with robotic feedback. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 5549-5552). IEEE. https://doi.org/10.1109/IEMBS.2010.5626782

[99] King, C. E., Wang, P. T., Mizuta, M., Reinkensmeyer, D. J., Do, A. H., Moromugi, S., & Nenadic, Z. (2011, August). Noninvasive brain-computer interface driven hand orthosis. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5786-5789). IEEE. https://doi.org/10.1109/IEMBS.2011.6091432

[100] Meyer, T., Peters, J., Brötz, D., Zander, T. O., Schölkopf, B., Soekadar, S. R., & Grosse-Wentrup, M. (2012, October). A brain-robot interface for studying motor learning after stroke. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4078-4083). IEEE. https://doi.org/10.1109/IROS.2012.6385646

[101] Ang, K. K., Guan, C., Phua, K. S., Wang, C., Teh, I., Chen, C. W., & Chew, E. (2012, August). Transcranial direct current stimulation and EEG-based motor imagery BCI for upper limb stroke rehabilitation. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 4128-4131). IEEE. https://doi.org/10.1109/EMBC.2012.6346875

[102] Tung, S. W., Guan, C., Ang, K. K., Phua, K. S., Wang, C., Zhao, L., ... & Chew, E. (2013, July). Motor imagery BCI for upper limb stroke rehabilitation: An evaluation of the EEG recordings using coherence analysis. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 261-264). IEEE. https://doi.org/10.1109/EMBC.2013.6609487

[103] Selvam, V. S., & Shenbagadevi, S. (2011, August). Brain tumor detection using scalp EEG with modified wavelet-ICA and multi layer feed forward neural network. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 6104-6109). IEEE. https://doi.org/10.1109/IEMBS.2011.6091508

[104] Sharanreddy, M., & Kulkarni, P. K. (2013). Detection of primary brain tumor present in EEG signal using wavelet transform and neural network. *Int J Biol Med Res, 4*(1), 2855-9.

[105] Poulos, M., Felekis, T., & Evangelou, A. (2012). Is it possible to extract a fingerprint for early breast cancer via EEG analysis? *Medical hypotheses, 78*(6), 711-716. https://doi.org/10.1016/j.mehy.2012.02.016

[106] Sharanreddy, M., & Kulkarni, P. K. (2013). Automated EEG signal analysis for identification of epilepsy seizures and brain tumour. *Journal of medical engineering & technology, 37*(8), 511-519. https://doi.org/10.3109/03091902.2013.837530

[107] Liang, S. F., Shaw, F. Z., Young, C. P., Chang, D. W., & Liao, Y. C. (2010, August). A closed-loop brain computer interface for real-time seizure detection and control. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 4950-4953). IEEE. https://doi.org/10.1109/IEMBS.2010.5627243

[108] Fadzal, C. W. N. F. C. W., Mansor, W., & Khuan, L. Y. (2011). Review of brain computer interface application in diagnosing dyslexia. In *2011 IEEE Control and System Graduate Research Colloquium* (pp. 124-128). IEEE. https://doi.org/10.1109/ICSGRC.2011.5991843

[109] Navarro, A. A., Ceccaroni, L., Velickovski, F., Torrellas, S., Miralles, F., Allison, B. Z., ... & Faller, J. (2011). Context-awareness as an enhancement of brain-computer interfaces. In *Ambient Assisted Living: Third International Workshop, IWAAL 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings 3* (pp. 216-223). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21303-8_30

[110] Peng, H., Hu, B., Qi, Y., Zhao, Q., & Ratcliffe, M. (2011, May). An improved EEG de-noising approach in electroencephalogram (EEG) for home care. In *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops* (pp. 469-474). IEEE. https://doi.org/10.4108/icst.pervasivehealth.2011.246021

[111] Venthur, B., Blankertz, B., Gugler, M. F., & Curio, G. (2010, October). Novel applications of BCI technology: psychophysiological optimization of working conditions in industry. In *2010 IEEE International Conference on Systems, Man and Cybernetics* (pp. 417-421). IEEE. https://doi.org/10.1109/ICSMC.2010.5641772

[112] Roy, R. N., Bonnet, S., Charbonnier, S., & Campagne, A. (2013, July). Mental fatigue and working memory load estimation: interaction and implications for EEG-based passive BCI. In *2013 35th annual international conference of the IEEE Engineering in Medicine and Biology Society* (EMBC) (pp. 6607-6610). IEEE. https://doi.org/10.1109/EMBC.2013.6611070

[113] Duru, D. G., Duru, A. D., Barkana, D. E., Sanli, O., & Ozkan, M. (2013, November). Assessment of surgeon's stress level and alertness using EEG during laparoscopic simple nephrectomy. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 452-455). IEEE. https://doi.org/10.1109/NER.2013.6695969

[114] Ke, J., Zhang, M., Luo, X., & Chen, J. (2021). Monitoring distraction of construction workers caused by noise using a wearable Electroencephalography (EEG) device. *Automation in Construction, 125,* 103598. https://doi.org/10.1016/j.autcon.2021.103598

[115] Villalba-Diez, J., Zheng, X., Schmidt, D., & Molina, M. (2019). Characterization of industry 4.0 lean management problem-solving behavioral patterns using EEG sensors and deep learning. *Sensors*, *19*(13), 2841. https://doi.org/10.3390/s19132841

[116] Lin, C. T., Lin, B. S., Lin, F. C., & Chang, C. J. (2012). Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking. *IEEE Systems Journal*, *8*(2), 363-370. https://doi.org/10.1109/JSYST.2012.2192756

[117] Ou, C. Z., Lin, B. S., Chang, C. J., & Lin, C. T. (2012, July). Brain computer interface-based smart environmental control system. In *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (pp. 281-284). IEEE. https://doi.org/10.1109/IIH-MSP.2012.74

[118] Dong, Y., Hu, Z., Uchimura, K., & Murayama, N. (2010). Driver inattention monitoring system for intelligent vehicles: A review. *IEEE transactions on intelligent transportation systems*, *12*(2), 596-614. https://doi.org/10.1109/TITS.2010.2092770

[119] Coetzer, R. C., & Hancke, G. P. (2009, September). Driver fatigue detection: A survey. In AFRICON 2009 (pp. 1-6). IEEE. https://doi.org/10.1109/AFRCON.2009.5308101

[120] Pritchett, S., Zilberg, E., Xu, Z. M., Karrar, M., Burton, D., & Lal, S. (2011, November). Comparing accuracy of two algorithms for detecting driver drowsiness—Single source (EEG) and hybrid (EEG and body movement). In *7th International Conference on Broadband Communications and Biomedical Applications* (pp. 179-184). IEEE. https://doi.org/10.1109/IB2Com.2011.6217916

[121] Dahal, N., Nandagopal, N., Nafalski, A., & Nedic, Z. (2011, November). Modeling of cognition using EEG: a review and a new approach. In *TENCON 2011-2011 IEEE Region 10 Conference* (pp. 1045-1049). IEEE. https://doi.org/10.1109/TENCON.2011.6129270

[122] The Bitbrain Team (2018). Nissan’s Brain-to-Vehicle technology communicates our brains with vehicles. https://www.bitbrain.com/blog/nissan-brain-to-vehicle-technology

[123] Papanastasiou, G., Drigas, A., Skianis, C., & Lytras, M. (2020). Brain computer interface based applications for training and rehabilitation of students with neurodevelopmental disorders. A literature review. *Heliyon, 6*(9). https://doi.org/10.1016/j.heliyon.2020.e04250

[124] Huang, Z., Javaid, A., Devabhaktuni, V. K., Li, Y., & Yang, X. (2019). Development of cognitive training program with EEG headset. IEEE Access, 7, 126191-126200. https://doi.org/10.1109/ACCESS.2019.2937866

[125] Sorudeykin, K. A. (2010). An educative brain-computer interface. *arXiv preprint arXiv:1003.2660*. https://doi.org/10.48550/arXiv.1003.2660

[126] Johnston, S. J., Boehm, S. G., Healy, D., Goebel, R., & Linden, D. E. (2010). Neurofeedback: A promising tool for the self-regulation of emotion networks. *Neuroimage,* *49*(1), 1066-1072. https://doi.org/10.1016/j.neuroimage.2009.07.056

[127] Zotev, V., Phillips, R., Yuan, H., Misaki, M., & Bodurka, J. (2014). Self-regulation of human brain activity using simultaneous real-time fMRI and EEG neurofeedback. *NeuroImage, 85,* 985-995.

[128] Marquez BY, Alanis A, Lopez MA, Magdaleno-Palencia JS. Sport education based technology: Stress measurement in competence. In: e-Learning and e-Technologies in Education (ICEEE), 2012 International Conference on. IEEE; 2012. p. 247–52. https://doi.org/10.1016/j.neuroimage.2013.04.126

[129] Birbaumer, N., Ruiz, S., & Sitaram, R. (2013). Learned regulation of brain metabolism. *Trends in cognitive sciences,* *17*(6), 295-302. https://doi.org/10.1016/j.tics.2013.04.009

[130] Vecchiato, G., Astolfi, L., Fallani, F. D. V., Salinari, S., Cincotti, F., Aloise, F., ... & Babiloni, F. (2009, September). The study of brain activity during the observation of commercial advertsing by using high resolution EEG techniques. In *2009 Annual International Conference of the IEEE Engineering in Medicine*

and Biology Society (pp. 57-60). IEEE. https://doi.org/10.1109/IEMBS.2009.5335045

[131] Yoshioka, M., Inoue, T., & Ozawa, J. (2012, June). Brain signal pattern of engrossed subjects using near infrared spectroscopy (NIRS) and its application to TV commercial evaluation. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1-6). IEEE. https://doi.org/10.1109/IJCNN.2012.6252752

[132] Vecchiato, G., Babiloni, F., Astolfi, L., Toppi, J., Cherubino, P., Dai, J., ... & Wei, D. (2011, October). Enhance of theta EEG spectral activity related to the memorization of commercial advertisings in Chinese and Italian subjects. In *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)* (Vol. 3, pp. 1491-1494). IEEE. https://doi.org/10.1109/BMEI.2011.6098615

[133] Tan, D., & Nijholt, A. (2010). *Brain-computer interfaces and human-computer interaction*. Springer London. https://doi.org/10.1007/978-1-84996-272-8_1

[134] Royer, A. S., Doud, A. J., Rose, M. L., & He, B. (2010). EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies. *IEEE Transactions on neural systems and rehabilitation engineering, 18*(6), 581-589. https://doi.org/10.1109/TNSRE.2010.2077654

[135] Bonnet, L., Lotte, F., & Lécuyer, A. (2013). Two brains, one game: design and evaluation of a multiuser BCI video game based on motor imagery. *IEEE Transactions on Computational Intelligence and AI in games, 5*(2), 185-198. https://doi.org/10.1109/TCIAIG.2012.2237173

[136] Khalifa, W., Salem, A., Roushdy, M., & Revett, K. (2012, May). A survey of EEG based user authentication schemes. In *2012 8th International Conference on Informatics and Systems (INFOS)* (pp. BIO-55). IEEE.

[137] Revett, K., Deravi, F., & Sirlantzis, K. (2010, September). Biosignals for user authentication-towards cognitive biometrics? In *2010 International Conference on Emerging Security Technologies* (pp. 71-76). IEEE. https://doi.org/10.1109/EST.2010.32

[138] Su, F., Zhou, H., Feng, Z., & Ma, J. (2012, March). A biometric-based covert warning system using EEG. In *2012 5th IAPR International Conference on*

Biometrics (ICB) (pp. 342-347). IEEE. https://doi.org/10.1109/ICB.2012.6199830

[139] Nakanishi, I., Ozaki, K., & Li, S. (2012, September). Evaluation of the brain wave as biometrics in a simulated driving environment. In *2012 BIOSIG-Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)* (pp. 1-5). IEEE.

[140] Pan, H., Mi, W., Lei, X., & Deng, J. (2020). A closed-loop brain–machine interface framework design for motor rehabilitation. *Biomedical Signal Processing and Control, 58,* 101877. https://doi.org/10.1016/j.bspc.2020.101877

[141] Johnson, N. N., Carey, J., Edelman, B. J., Doud, A., Grande, A., Lakshminarayan, K., & He, B. (2018). Combined rTMS and virtual reality brain–computer interface training for motor recovery after stroke. *Journal of neural engineering*, *15*(1), 016009. https://doi.org/10.1088/1741-2552/aa8ce3

[142] Riener, R. (2016). The Cybathlon promotes the development of assistive technology for people with physical disabilities. *Journal of neuroengineering and rehabilitation, 13,* 1-4. https://doi.org/10.1186/s12984-016-0157-2.

[143] He, B., Baxter, B., Edelman, B. J., Cline, C. C., & Wenjing, W. Y. (2015). Noninvasive brain-computer interfaces based on sensorimotor rhythms. Proceedings of the IEEE, 103(6), 907-925. https://doi.org/10.1109/jproc.2015.2407272.

[144] Al-Saegh, A., Dawwd, S. A., & Abdul-Jabbar, J. M. (2021). Deep learning for motor imagery EEG-based classification: A review. *Biomedical Signal Processing and Control, 63,* 102172. https://doi.org/10.1016/j.bspc.2020.102172.

[145] Abualsaud, K., Mahmuddin, M., Saleh, M., & Mohamed, A. (2015). Ensemble classifier for epileptic seizure detection for imperfect EEG data. *The Scientific World Journal, 2015*. https://doi.org/10.1155/2015/945689

[146] Tsui, C. S. L., Gan, J. Q., & Hu, H. (2011). A self-paced motor imagery based brain-computer interface for robotic wheelchair control. *Clinical EEG and neuroscience, 42*(4), 225-229. https://doi.org/10.1177/155005941104200407.

[147] Ng, D. W. K., Soh, Y. W., & Goh, S. Y. (2014, December). Development of an autonomous BCI wheelchair. In *2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI)* (pp. 1-4). IEEE. https://doi.org/10.1109/CIBCI.2014.7007784

[148] Heo, J., & Yoon, G. (2020). EEG studies on physical discomforts induced by virtual reality gaming. *Journal of Electrical Engineering & Technology, 15,* 1323-1329. https://doi.org/10.1007/s42835-020-00373-1

[149] Pires, G., Torres, M., Casaleiro, N., Nunes, U., & Castelo-Branco, M. (2011, November). Playing Tetris with non-invasive BCI. In *2011 ieee 1st international conference on serious games and applications for health (segah)* (pp. 1-6). IEEE. https://doi.org/10.1109/SeGAH.2011.6165454

[150] van de Laar, B., Gürkök, H., Bos, D. P. O., Poel, M., & Nijholt, A. (2013). Experiencing BCI control in a popular computer game. *IEEE Transactions on Computational Intelligence and AI in Games, 5*(2), 176-184. https://doi.org/10.1109/tciaig.2013.2253778.

[151] Bonnet, L., Lotte, F., & Lécuyer, A. (2013). Two brains, one game: design and evaluation of a multiuser BCI video game based on motor imagery. *IEEE Transactions on Computational Intelligence and AI in games, 5*(2), 185-198. https://doi.org/10.1109/tciaig.2012.2237173.

[152] Alazrai, R., Abuhijleh, M., Alwanni, H., & Daoud, M. I. (2019). A deep learning framework for decoding motor imagery tasks of the same hand using EEG signals. *IEEE Access, 7,* 109612-109627. https://doi.org/10.1109/access.2019.2934018.

[153] Zhu, X., Li, P., Li, C., Yao, D., Zhang, R., & Xu, P. (2019). Separated channel convolutional neural network to realize the training free motor imagery BCI systems. *Biomedical Signal Processing and Control, 49,* 396-403. https://doi.org/10.1016/j.bspc.2018.12.027.

[154] Prochazka, A., Kukal, J., & Vysata, O. (2008, March). Wavelet transform use for feature extraction and EEG signal segments classification. In *2008 3rd International symposium on communications, control and signal processing* (pp. 719-722). IEEE. https://doi.org/10.1109/ISCCSP.2008.4537317

[155] Zabidi, A., Mansor, W., Lee, Y. K., & Fadzal, C. C. W. (2012, September). Short-time Fourier Transform analysis of EEG signal generated during imagined writing. In *2012 international conference on system engineering and technology (ICSET)* (pp. 1-4). IEEE. https://doi.org/10.1109/ICSEngT.2012.6339284

[156] Bagh, N., & Reddy, M. R. (2020). Hilbert transform-based event-related patterns for motor imagery brain computer interface. *Biomedical Signal Processing and Control, 62*, 102020. https://doi.org/10.1016/j.bspc.2020.102020.

[157] Edelman, B. J., Baxter, B., & He, B. (2015). EEG source imaging enhances the decoding of complex right-hand motor imagery tasks. *IEEE Transactions on Biomedical Engineering, 63*(1), 4-14. https://doi.org/10.1109/tbme.2015.2467312.

[158] Tabar, Y. R., & Halici, U. (2016). A novel deep learning approach for classification of EEG motor imagery signals. *Journal of neural engineering, 14*(1), 016003. https://doi.org/10.1088/1741-2560/14/1/016003.

[159] Luo, J., Feng, Z., Zhang, J., & Lu, N. (2016). Dynamic frequency feature selection based approach for classification of motor imageries. *Computers in biology and medicine, 75*, 45-53. https://doi.org/10.1016/j.compbiomed.2016.03.004.

[160] Lee, H. K., & Choi, Y. S. (2019). Application of continuous wavelet transform and convolutional neural network in decoding motor imagery brain-computer interface. *Entropy, 21*(12), 1199. https://doi.org/10.3390/e21121199.

[161] Li, M. A., Zhu, W., Liu, H. N., & Yang, J. F. (2017). Adaptive feature extraction of motor imagery EEG with optimal wavelet packets and SE-isomap. *Applied Sciences, 7*(4), 390. https://doi.org/10.3390/app7040390.

[162] Faust, O., Hagiwara, Y., Hong, T. J., Lih, O. S., & Acharya, U. R. (2018). Deep learning for healthcare applications based on physiological signals: A review. *Computer methods and programs in biomedicine, 161*, 1-13. https://doi.org/10.1016/j.cmpb.2018.04.005.

[163] Saa, J. F. D., & Çetin, M. (2012). A latent discriminative model-based approach for classification of imaginary motor tasks from EEG data. *Journal of neural engineering, 9*(2), 026020. https://doi.org/10.1088/1741-2560/9/2/026020.

[164] Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2008, June). Filter bank common spatial pattern (FBCSP) in brain-computer interface. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 2390-2397). IEEE. https://doi.org/10.1109/IJCNN.2008.4634130

[165] Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2012). Mutual information-based selection of optimal spatial–temporal patterns for single-trial EEG-based BCIs. *Pattern Recognition, 45*(6), 2137-2144. https://doi.org/10.1016/j.patcog.2011.04.018.

[166] Craik, A., He, Y., & Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of neural engineering, 16*(3), 031001. https://doi.org/10.1088/1741-2552/ab0ab5.

[167] Zhang, X., Yao, L., Huang, C., Sheng, Q. Z., & Wang, X. (2017). Intent recognition in smart living through deep recurrent neural networks. In *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24* (pp. 748-758). Springer International Publishing. https://doi.org/10.1007/978-3-319-70096-0_76

[168] Zhang, Z., Sun, J., & Chen, T. (2022). A new dynamically convergent differential neural network for brain signal recognition. *Biomedical Signal Processing and Control, 71,* 103130. https://doi.org/10.1016/j.bspc.2021.103130

[169] Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering, 15*(5), 056013. https://doi.org/10.1088/1741-2552/aace8c.

[170] Sharma, R., Kim, M., & Gupta, A. (2022). Motor imagery classification in brain-machine interface with machine learning algorithms: Classical approach to multi-layer perceptron model. *Biomedical Signal Processing and Control, 71,* 103101. https://doi.org/10.1016/j.bspc.2021.103101.

[171] Carvalho, S. R., Cordeiro Filho, I., De Resende, D. O., Siravenha, A. C., De Souza, C. R., Debarba, H., ... & Boulic, R. (2017, October). A deep learning approach for classification of reaching targets from EEG images. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (pp. 178-184). IEEE. https://doi.org/10.1109/SIBGRAPI.2017.30

[172] Lu, N., Li, T., Ren, X., & Miao, H. (2016). A deep learning scheme for motor imagery classification based on restricted Boltzmann machines. *IEEE transactions on neural systems and rehabilitation engineering, 25*(6), 566-576. https://doi.org/10.1109/tnsre.2016.2601240.

[173] Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*. https://doi.org/10.48550/arXiv.1511.06448

[174] Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., ... & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human brain mapping, 38*(11), 5391-5420. https://doi.org/10.1002/hbm.23730

[175] Padfield, N., Zabalza, J., Zhao, H., Masero, V., & Ren, J. (2019). EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. *Sensors, 19*(6), 1423. https://doi.org/10.3390/s19061423.

[176] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis, 42*, 60-88. https://doi.org/10.1016/j.media.2017.07.005.

[177] Tibor Schirrmeister, R., Gemein, L., Eggensperger, K., Hutter, F., & Ball, T. (2017). Deep learning with convolutional neural networks for decoding and visualization of eeg pathology. *arXiv e-prints, arXiv-1708*. https://doi.org/10.48550/arXiv.1708.08012

[178] Ron-Angevin, R., Velasco-Álvarez, F., Fernández-Rodríguez, Á., Díaz-Estrella, A., Blanca-Mena, M. J., & Vizcaíno-Martín, F. J. (2017). Brain-Computer Interface application: auditory serial interface to control a two-class motor-imagery-based wheelchair. *Journal of neuroengineering and rehabilitation*, *14*(1), 1-16. https://doi.org/10.1186/s12984-017-0261-y.

[179] Yu, H., Lu, H., Wang, S., Xia, K., Jiang, Y., & Qian, P. (2019). A general common spatial patterns for EEG analysis with applications to vigilance detection. *IEEE Access*, *7*, 111102-111114. https://doi.org/10.1109/access.2019.2934519.

[180] Li, Y., Zhang, X. R., Zhang, B., Lei, M. Y., Cui, W. G., & Guo, Y. Z. (2019). A channel-projection mixed-scale convolutional neural network for motor imagery EEG decoding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*(6), 1170-1180. https://doi.org/10.1109/tnsre.2019.2915621.

[181] Zhang, D., Yao, L., Zhang, X., Wang, S., Chen, W., Boots, R., & Benatallah, B. (2018, April). Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32, No. 1). https://doi.org/10.1609/aaai.v32i1.11496

[182] Dai, G., Zhou, J., Huang, J., & Wang, N. (2020). HS-CNN: a CNN with hybrid convolution scale for EEG motor imagery classification. *Journal of neural engineering*, *17*(1), 016025. https://doi.org/10.1088/1741-2552/ab405f.

[183] Xu, G., Shen, X., Chen, S., Zong, Y., Zhang, C., Yue, H., ... & Che, W. (2019). A deep transfer convolutional neural network framework for EEG signal classification. *IEEE Access*, *7*, 112767-112776. https://doi.org/10.1109/access.2019.2930958.

[184] Zhao, X., Zhang, H., Zhu, G., You, F., Kuang, S., & Sun, L. (2019). A multi-branch 3D convolutional neural network for EEG-based motor imagery classification. *IEEE transactions on neural systems and rehabilitation engineering*, *27*(10), 2164-2177. https://doi.org/10.1109/tnsre.2019.2938295.

[185] Rim, B., Sung, N. J., Min, S., & Hong, M. (2020). Deep learning in physiological signal data: A survey. *Sensors*, *20*(4), 969. https://doi.org/10.3390/s20040969.

[186] Tayeb, Z., Erçelik, E., & Conradt, J. (2017, May). Decoding of motor imagery movements from EEG signals using SpiNNaker neuromorphic hardware. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 263-266). IEEE. https://doi.org/10.1109/NER.2017.8008341

[187] Niranjani, A. N., & Sivachitra, M. (2017, December). Motor imagery signal classification using spiking neural network. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 901-904). IEEE. https://doi.org/10.1109/ISS1.2017.8389309

[188] Majidov, I., & Whangbo, T. (2019). Efficient classification of motor imagery electroencephalography signals using deep learning methods. *Sensors, 19*(7), 1736. https://doi.org/10.3390/s19071736.

[189] Khan, S. H., Hayat, M., & Porikli, F. (2019). Regularization of deep neural networks with spectral dropout. *Neural Networks, 110*, 82-90. https://doi.org/10.1016/j.neunet.2018.09.009.

[190] Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., ... & Boudoukh, G. (2017, February). Can FPGAs beat GPUs in accelerating next-generation deep neural networks? In *Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays* (pp. 5-14). https://doi.org/10.1145/3020078.3021740

[191] Djemal, R., Bazyed, A. G., Belwafi, K., Gannouni, S., & Kaaniche, W. (2016). Three-class EEG-based motor imagery classification using phase-space reconstruction technique. *Brain sciences, 6*(3), 36. https://doi.org/10.3390/brainsci6030036.

[192] Shahid, S., Sinha, R. K., & Prasad, G. (2010). Mu and beta rhythm modulations in motor imagery related post-stroke EEG: a study under BCI framework for post-stroke rehabilitation. *Bmc Neuroscience, 11*(1), 1-2. https://doi.org/10.1186/1471-2202-11-s1-p127.

[193] Sciaraffa, N., Di Flumeri, G., Germano, D., Giorgi, A., Di Florio, A., Borghini, G., ... & Aricò, P. (2022). Validation of a light EEG-based measure for real-time stress monitoring during realistic driving. *Brain sciences*, *12*(3), 304. https://doi.org/10.3390/brainsci12030304

[194] He, H., & Wu, D. (2019). Transfer learning for brain–computer interfaces: A Euclidean space data alignment approach. *IEEE Transactions on Biomedical Engineering, 67*(2), 399-410. https://doi.org/10.1109/tbme.2019.2913914.

[195] Lashgari, E., Ott, J., Connelly, A., Baldi, P., & Maoz, U. (2021). An end-to-end CNN with attentional mechanism applied to raw EEG in a BCI classification task. *Journal of Neural Engineering*, *18*(4), 0460e3. https://doi.org/10.1088/1741-2552/ac1ade

[196] Yang, B., Duan, K., Fan, C., Hu, C., & Wang, J. (2018). Automatic ocular artifacts removal in EEG using deep learning. *Biomedical Signal Processing and Control, 43*, 148-158. https://doi.org/10.1016/j.bspc.2018.02.021

[197] Zhang, C., Kim, Y. K., & Eskandarian, A. (2021). EEG-inception: an accurate and robust end-to-end neural network for EEG-based motor imagery classification. *Journal of Neural Engineering*, *18*(4), 046014. https://doi.org/10.1016/j.bspc.2018.02.021

[198] Alotaiby, T., El-Samie, F. E. A., Alshebeili, S. A., & Ahmad, I. (2015). A review of channel selection algorithms for EEG signal processing. *EURASIP Journal on Advances in Signal Processing*, *2015*, 1-21. https://doi.org/10.1186/s13634-015-0251-9

[199] da Cruz, J. R., Chicherov, V., Herzog, M. H., & Figueiredo, P. (2018). An automatic pre-processing pipeline for EEG analysis (APP) based on robust statistics. *Clinical Neurophysiology,* *129*(7), 1427-1437. https://doi.org/10.1016/j.clinph.2018.04.600

[200] N Mashhadi, N., Khuzani, A. Z., Heidari, M., & Khaledyan, D. (2020, October). Deep learning denoising for EOG artifacts removal from EEG signals. In *2020 IEEE Global Humanitarian Technology Conference (GHTC)* (pp. 1-6). IEEE. https://doi.org/10.1109/GHTC46280.2020.9342884.

[201] Val-Calvo, M., Álvarez-Sánchez, J. R., Ferrández-Vicente, J. M., & Fernández, E. (2019). Optimization of real-time EEG artifact removal and emotion estimation for human-robot interaction applications. *Frontiers in Computational Neuroscience, 13,* 80. https://doi.org/10.3389/fncom.2019.00080

[202] Cao, L., Wu, H., Chen, S., Dong, Y., Zhu, C., Jia, J., & Fan, C. (2022). A Novel Deep Learning Method Based on an Overlapping Time Window Strategy for Brain–Computer Interface-Based Stroke Rehabilitation. *Brain Sciences, 12*(11), 1502. https://doi.org/10.3390/brainsci12111502

[203] Lun, X., Yu, Z., Chen, T., Wang, F., & Hou, Y. (2020). A simplified CNN classification method for MI-EEG via the electrode pairs signals. *Frontiers in Human Neuroscience, 14,* 338. https://doi.org/10.3389/fnhum.2020.00338

[204] Dose, H., Møller, J. S., Iversen, H. K., & Puthusserypady, S. (2018). An end-to-end deep learning approach to MI-EEG signal classification for BCIs. *Expert Systems with Applications, 114,* 532-542. https://doi.org/10.1016/j.eswa.2018.08.031

[205] Hajinoroozi, M., Mao, Z., Jung, T. P., Lin, C. T., & Huang, Y. (2016). EEG-based prediction of driver's cognitive performance by deep convolutional neural network. *Signal Processing: Image Communication, 47,* 549-555. https://doi.org/10.1016/j.image.2016.05.018

[206] Shen, Y., Lu, H., & Jia, J. (2017). Classification of motor imagery EEG signals with deep learning models. In *Intelligence Science and Big Data Engineering: 7th International Conference, IScIDE 2017, Dalian, China, September 22-23, 2017, Proceedings 6* (pp. 181-190). Springer International Publishing. https://doi.org/10.1007/978-3-319-67777-4_16

[207] Raza, H., Rathee, D., Zhou, S. M., Cecotti, H., & Prasad, G. (2019). Covariate shift estimation based adaptive ensemble learning for handling non-stationarity in motor imagery related EEG-based brain-computer interface. Neurocomputing, 343, 154-166. https://doi.org/10.1016/j.neucom.2018.04.087

[208] Sugiyama, M., Krauledat, M., & Müller, K. R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research, 8*(5).

[209] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering, 22*(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

[210] Miladinović, A., Ajčević, M., Jarmolowska, J., Marusic, U., Colussi, M., Silveri, G., ... & Accardo, A. (2021). Effect of power feature covariance shift on BCI spatial-filtering techniques: A comparative study. *Computer Methods and Programs in Biomedicine, 198,* 105808. https://doi.org/10.1016/j.cmpb.2020.105808

[211] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). pmlr. https://doi.org/10.48550/arXiv.1502.03167

[212] Brunner C, Leeb R, Müller-Putz G, Schlögl A and Pfurtscheller G BCI Competition 2008—Graz data sets 2A and 2B (Graz: Institute for Knowledge Discovery) (http://bbci.de/ competition/iv/)

[213] Swiderski, B., Osowski, S., Gwardys, G., Kurek, J., Slowinska, M., & Lugowska, I. (2022). Random CNN structure: tool to increase generalization ability in deep learning. *Eurasip journal on image and video processing, 2022*(1), 3. https://doi.org/10.1186/s13640-022-00580-y.

[214] Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., ... & Kulić, D. (2017, November). Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction* (pp. 216-220). https://doi.org/10.1145/3136755.3136817

[215] McFarland, D. J., Miner, L. A., Vaughan, T. M., & Wolpaw, J. R. (2000). Mu and beta rhythm topographies during motor imagery and actual movements. *Brain topography, 12*, 177-186. https://doi.org/10.1023/A:1023437823106

[216] Weber, E., & Doppelmayr, M. (2016). Kinesthetic motor imagery training modulates frontal midline theta during imagination of a dart throw. *International journal of Psychophysiology, 110*, 137-145. https://doi.org/10.1016/j.ijpsycho.2016.11.002.

[217] Liu, Y. H., Lin, L. F., Chou, C. W., Chang, Y., Hsiao, Y. T., & Hsu, W. C. (2019). Analysis of electroencephalography event-related desynchronisation and synchronisation induced by lower-limb stepping motor imagery. *Journal of Medical and Biological Engineering, 39*, 54-69. https://doi.org/10.1007/s40846-018-0379-9.

[218] Lotte, F., & Guan, C. (2010). Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering, 58*(2), 355-362. https://doi.org/10.1109/tbme.2010.2082539.

[219] Raza, H., Cecotti, H., Li, Y., & Prasad, G. (2016). Adaptive learning with covariate shift-detection for motor imagery-based brain–computer interface. *Soft Computing, 20*, 3085-3096. https://doi.org/10.1007/s00500-015-1937-5.

[220] Gaur, P., Pachori, R. B., Wang, H., & Prasad, G. (2015, July). An empirical mode decomposition based filtering method for classification of motor-imagery EEG signals for enhancing brain-computer interface. In *2015 International joint conference on neural networks (IJCNN)* (pp. 1-7). IEEE. https://doi.org/10.1109/IJCNN.2015.7280754

[221] Gaur, P., Pachori, R. B., Wang, H., & Prasad, G. (2018). A multi-class EEG-based BCI classification using multivariate empirical mode decomposition based filtering and Riemannian geometry. *Expert Systems with Applications, 95*, 201-211. https://doi.org/10.1016/j.eswa.2017.11.007.

[222] Shahid, S., & Prasad, G. (2011). Bispectrum-based feature extraction technique for devising a practical brain–computer interface. *Journal of neural engineering, 8*(2), 025014. https://doi.org/10.1088/1741-2560/8/2/025014.J.NeuralEng.8

[223] Zheng, Q., Zhu, F., & Heng, P. A. (2018). Robust support matrix machine for single trial EEG classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26*(3), 551-562. https://doi.org/10.1109/tnsre.2018.2794534.

[224] Sakhavi, S., Guan, C., & Yan, S. (2018). Learning temporal information for brain-computer interface using convolutional neural networks. *IEEE transactions on neural networks and learning systems, 29*(11), 5619-5629. https://doi.org/10.1109/tnnls.2018.2789927.

[225] Mane, R., Chew, E., Chua, K., Ang, K. K., Robinson, N., Vinod, A. P., ... & Guan, C. (2021). FBCNet: A multi-view convolutional neural network for brain-computer interface. *arXiv preprint arXiv:2104.01233*. https://doi.org/10.48550/arXiv.2104.01233

[226] Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., & Kennedy, P. J. (2016, July). Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)* (pp. 4368-4374). IEEE. https://doi.org/10.1109/IJCNN.2016.7727770

[227] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM, 64*(3), 107-115. https://doi.org/10.1145/3446776

[228] Zhang, Z., Duan, F., Sole-Casals, J., Dinares-Ferran, J., Cichocki, A., Yang, Z., & Sun, Z. (2019). A novel deep learning approach with data augmentation to classify motor imagery signals. *IEEE Access, 7*, 15945-15954. https://doi.org/10.1109/ACCESS.2019.2895133

[229] McFarland, D. J., Sarnacki, W. A., & Wolpaw, J. R. (2010). Electroencephalographic (EEG) control of three-dimensional movement. *Journal of neural engineering, 7*(3), 036007. https://doi.org/10.1088/1741-2560/7/3/036007

[230] Suwannarat, A., Pan-Ngum, S., & Israsena, P. (2018). Comparison of EEG measurement of upper limb movement in motor imagery training system. *Biomedical engineering online, 17*(1), 1-22. https://doi.org/10.1186/s12938-018-0534-0

[231] Ang, K. K., Chin, Z. Y., Wang, C., Guan, C., & Zhang, H. (2012). Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in neuroscience, 6,* 39. https://doi.org/10.3389/fnins.2012.00039

[232] Liu, T., & Yang, D. (2021). A densely connected multi-branch 3D convolutional neural network for motor imagery EEG decoding. *Brain Sciences, 11*(2), 197. https://doi.org/10.3390/brainsci11020197

[233] Jiang, Q., Zhang, Y., & Zheng, K. (2022). Motor imagery classification via kernel-based domain adaptation on an SPD manifold. *Brain Sciences, 12*(5), 659. https://doi.org/10.3390/brainsci12050659

[234] Gao, S., Yang, J., Shen, T., & Jiang, W. (2022). A Parallel Feature Fusion Network Combining GRU and CNN for Motor Imagery EEG Decoding. *Brain Sciences, 12*(9), 1233. https://doi.org/10.3390/brainsci12091233

[235] Gross, R. (2017). Humanoid Robotic Torso PROTO1. https://www.myminifactory.com/object/3d-print-48754

LIST OF PUBLICATIONS FROM THIS THESIS

Arı E, Taçgın E. Input Shape Effect on Classification Performance of Raw EEG Motor Imagery Signals with Convolutional Neural Networks for Use in Brain-Computer Interfaces. Brain Sci. 2023 Jan 31;13(2):240. doi: 10.3390/brainsci13020240. PMID: 36831784; PMCID: PMC9954790.