

The project titled “Real-time EEG Classification for Motor imagery based BCI Applications ” submitted by Fahim Abdullah, Roll No.: 201020114, Session: MS 2019-20, has been accepted as satisfactory in fulfillment of the requirement for the Masters degree in Computer Science and Engineering.

BOARD OF EXAMINERS

Dr. Md. Sujan Ali

Chairman

Professor

Department of Computer Science And Engineering

Jatiya Kabi Kazi Nazrul Islam University

.....

Signature

Internal Member 1:

Member

.....

Signature

Internal Member 2:

Member

.....

Signature

External Member:

Member

.....

Signature

CANDIDATE'S DECLARATION

We declare that this project is our own work and has not been submitted in any other form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Date:

Signature of the Candidate

.....

Fahim Abdullah

CONTENTS

List of Figures	v
List of Tables	v
Acknowledgements	vi
Abstract	vii
1 INTRODUCTION	1
1.1 Problem Definition	1
1.2 Project Aim	1
1.3 Social Impact	2
1.4 Deliverables	2
2 BACKGROUND STUDY	3
2.1 Background and Motivation	3
2.2 Selected Dataset	4
2.3 User Interface	5
2.4 Research Available Technologies	5
2.4.1 MATLAB	5
2.4.2 Python	6
2.4.3 Java	6
2.4.4 R	6
2.5 Chosen Technology	7
3 METHODOLOGY	8
3.1 Working Procedure	8
3.2 Preprocessing	8
3.2.1 Data Extraction	9
3.2.2 Feature Extraction	10

3.2.3	Feature Selection	10
3.3	Classification	10
3.3.1	Support-Vector Machine (SVM)	11
3.4	Result Representation	11
3.4.1	Accuracy	11
3.4.2	Kappa Value	12
3.4.3	Confusion Matrix	12
4	ANALYSIS	13
4.1	Introduction	13
4.2	Why do system fail?	13
4.3	Software Requirements	13
5	IMPLEMENTATION	15
5.1	Introduction	15
5.2	Implementation Methodology	15
5.3	Interface and results	16
6	CONCLUSION	21
6.1	Evaluation Of Entire Project	21
6.2	Future Work	22

List of Figures

3.1.1 Working procedure	9
3.4.1 Confusion matrix for binary classification	12
5.3.1 The screenshot of final UI	17
5.3.2 The screenshot of importing dataset	17
5.3.3 The screenshot of state of UI after importing dataset	18
5.3.4 The screenshot of final UI waving left hand	18
5.3.5 The screenshot of final UI waving right hand	19
5.3.6 The screen shot of final UI	19
5.3.7 Confusion matrix of MI classifier	20

Acknowledgements

We would like to thank the Department of Computer Science and Engineering for building up the base which was needed to complete the work. We specially thank our supervisor and teacher Dr. Md. Sujan Ali who has motivated and directed us to this field. Without his help, direction and advice this could never have been completed. Thanks to our friends, family and associates who assisted us with every way possible. We are grateful to every single person who has directly or indirectly assisted us in this endeavor.

Abstract

Brain computer interface (BCI) is an emanating technology that deals with brain aided control of computer or other devices. It exploits the electrical activity within the brain of the handicapped patients who have lost their abilities thanks to severe injuries. Brain activities associated with Motor imagery (MI) tasks are often captured through various acquisition methods. Among these procurement methods, Electroencephalogram(EEG) is taken into account as most substantial for non-invasive BCI systems, due to its excellent temporal resolution, usability, non-invasiveness, portability, and low set-up costs. BCI intends to revive those capabilities by creating an information flow pathway from the human brain to the external device that must be controlled. This system utilizes the brain activity of the disabled person and tries to assist and map their sensorymotor functions. For efficient and accurate mapping, it controls the bidirectional flow of electrical information. It empowers a weakened person to become independent of any external support to regulate any device around them. But for BCI to be effective the classification of motor imagery has to be in real time. Taking that a main objective of our project, we have built a system which can correctly classify all Motor imagery task from a continuous stream of data.

Chapter 1

INTRODUCTION

1.1 Problem Definition

Brain-Computer Interface (BCI) is a technology which enables direct communication with a machine by using brain signals. It exploits the electrical activity within the brain of the handicapped patients who have lost their abilities due to severe injuries. Brain activities associated with Motor imagery (MI) tasks are often captured through various acquisition methods. Among these procurement methods, Electroencephalogram (EEG) is taken into account as most substantial for non-invasive BCI systems, due to its excellent temporal resolution, usability, non-invasiveness, portability, and low set-up costs. Electroencephalography (EEG) is method to record electrical activity and an electro-physiological monitoring method of the brain. It is typically noninvasive, with the electrodes placed along the scalp and records the summation of Excitatory and Inhibitory postsynaptic potential. BCI intends to revive those capabilities by creating an information flow pathway from the human brain to the external device that must be controlled. This system utilizes the brain activity of the disabled person and tries to assist and map their sensorymotor functions. For efficient and accurate mapping, it controls the bidirectional flow of electrical information. It empowers a weakened person to become independent of any external support to regulate any device around them.

1.2 Project Aim

In this study, we are proposing a method that can identify if any Motor imagery (MI) tasks are being performed by the subject on the real time and if so,

classify the MI task.

1.3 Social Impact

From this thesis we are expecting the following set of social impacts.

1. Implementation of BCI for real time usage.
2. Application control using thoughts.
3. Better and faster prosthetic control.
4. Rehabilitation of motor function damaged patients.
5. Rehabilitation of stroked or other paralyzed patients.

1.4 Deliverables

Our project's output will consist of the following deliverables:

1. A desktop application
2. A project report.

Chapter 2

BACKGROUND STUDY

2.1 Background and Motivation

Conventionally, BCIs are generally used for medical applications like neural control of prosthetic artificial limbs(1). Recent research often with noninvasive approaches based on electroencephalography(EEG) has opened up the possibility for novel BCIs focused on enhancing performance of healthy users(2; 3; 4).

To implement EEG-based real time BCIs, many data analysis methods have been investigated (5) and common spatial pattern (CSP) is one of the most effective methods. CSP was first introduced into the domain of real-time BCI in 2000 (6; 7), since then and until recently there are many papers published about the applications of CSP (8; 9; 10). Moreover, some advanced variations of CSP were also proposed, for example, Common Spatio Spectral Pattern (CSSP) which allows for individually tuned frequency filters at each electrode position (8), subband-CSP (SBCSP) aiming at solving the problem of fine tuning process for each subject (9), and filter bank-CSP(FBCSP) that can automatically select discriminative pairs of frequency bands and corresponding CSP features(10), etc. However, these methods are complex to be realized and thus, time consuming. In contrast, the original form is still quite effective and efficient. Yet, as its application is usually based on multi-channels, it requires a relatively long time for electrodes installation.

In neuroscience Electroencephalography (EEG) analysis has been a critical tool with applications in neuroscience, neural engineering and even commercial applications. To uncover relevant information for neural classification and neuroimaging, many of the analytical tools utilized in EEG studies have used machine learning. Due to the the supply of huge EEG data sets and advances in machine learning in recent times, have both led to the deployment of deep learning architectures,

especially within the analysis of EEG signals and in understanding the knowledge it's going to contain for brain functionality. The robust automatic classification of those signals is a crucial step towards making the utilization of EEG more practical in many applications and fewer reliant on trained professionals. The use of electroencephalography (EEG) data for motor imagery-based brain-computer interface (MI-BCI) has received a lot of attention in the past few decades, but the biggest challenge in BCI is obtaining reliable classification performance of the MI tasks in real time EEG data.

2.2 Selected Dataset

The dataset that is selected is dataset 1 of BCI competition IV. It is created for classification of continuous EEG without trial structure.

These datasets were obtained from healthy participants who performed motor imagery tasks without feedback during the entire session. Each participant executed two categories of motor imagery tasks, chosen from three categories: left hand, right hand, and foot (with the participant selecting the side, and optionally both feet), during the entire session without receiving feedback. The datasets collected were from healthy subjects.

During the first two runs of the session, visual cues in the form of arrows pointing left, right, or down were displayed on a computer screen for a duration of 4 seconds. The subjects were instructed to perform the motor imagery task corresponding to the cued direction. The visual cues were presented on a computer screen in the first two runs. Each cue was an arrow pointing left, right, or down and was displayed for 4 seconds. During this time, the subject was instructed to perform the cued motor imagery task. The cue periods were followed by a 2-second blank screen and a 2-second fixation cross displayed at the center of the screen. The fixation cross was shown superimposed on the cues, resulting in a total duration of 6 seconds. The data sets provided include complete marker information.

Continuous signals of 59 EEG channels and markers indicating the time points of cue presentation and corresponding target classes are provided as calibration data. The data is available in Matlab format (*.mat) with the following variables: cnt: the continuous EEG signals, size [time x channels]. The array is stored in datatype INT16. To convert the continuous EEG signals to uV values in Matlab, use the formula $\text{cnt} = 0.1 * \text{double}(\text{cnt})$.

mrk: structure of target cue information with fields (the file of evaluation data does not contain this variable).

pos: a vector indicating the positions of the cue in the EEG signals, given in unit sample and with a length of cues.

y: Vector of target classes with a length of cues, where -1 represents class one and 1 represents class two.

nfo: structure providing additional information with fields

fs: sampling rate,

clab: cell array of channel labels,

classes: Array of the names of the motor imagery classes

xpos: x-position of electrodes in a 2d-projection,

ypos: y-position of electrodes in a 2d-projection

2.3 User Interface

A user interface guides how a user interacts with a system. It should hide the complexities of the system allowing the user to operate the system easily and efficiently. An interface for a web application works in exactly the same way. Molly E Holzchlag classifies 5 features of Interface Design. These are metaphor; clarity; consistency; orientation; and navigation. Metaphor refers to the symbolic representation of areas of your site e.g. using familiar images for entry points, exits and windows in an environment. Clarity demands that every asset on your website should have a reason for being there, and that this reason should be apparent to the user. E.g. images and buttons should perform their perceived purpose. The third feature to consider is consistency, which states that the use of metaphors and navigation aids be uniformly used. Orientation means the user must know exactly where he or she is at every step of using your application. Aids to ensure this include titles, headers and footers etc. The final feature highlighted by Molly E Holzchlag is the importance of good navigation, which can coincide with layout design issues.

2.4 Research Available Technologies

Many technologies exist that could be used to develop the software. The major ones are listed below.

2.4.1 MATLAB

MATLAB allows for matrix manipulations, algorithm implementation, user interface creation, function and data plotting, and interfacing with programs written in

other languages such as C, C++, C#, Java, FORTRAN, and Python. While MATLAB is primarily used for numerical computing, it also offers an optional MuPAD symbolic engine toolbox for access to symbolic computing abilities. Additionally, the Simulink package adds graphical multi-domain simulation and model-based design for dynamic and embedded systems, making it a versatile tool for various implementations of machine learning algorithms and creating machine learning models.

2.4.2 Python

Python is considered to be in the first place in the list of all machine learning development languages due to the simplicity. The syntaxes belonging to python are very simple and may be easily learnt. Therefore, many machine learning algorithms can be easily implemented in it. Compared to other languages such as Java, C++, or Ruby, Python has a shorter development time. Python supports multiple programming paradigms, including object-oriented, functional, and procedural styles of programming. There are many libraries in Python that make doing tasks easier. For example: Numpy is a library for python that helps to solve many scientific computations. Also, there is Pybrain, which is for using machine learning in Python.

2.4.3 Java

Java is a suitable option for AI development.. Artificial intelligence has lot to try to to with search algorithms, artificial neural networks and genetic programming. Java provides several benefits such as easy usage, simplified debugging, package services, simplified handling of large-scale projects, graphical representation of data, and improved user interaction. It also has the incorporation of Swing and SWT (the Standard Widget Tool kit). These tools can enhance the visual appeal and complexity of graphics and interfaces.

2.4.4 R

R is one of the most effective language and environment for analysing and manipulating the data for statistical purposes. R is a powerful tool for producing high-quality graphics and visualizations, which are essential in data analysis and scientific research. With R, it is easy to create publication-quality plots that include mathematical symbols and formulas where necessary.. Apart from being a general purpose language R has a wide range of packages, including RODBC,

Gmodels, Class, and Tm, which are widely used in the field of machine learning.

2.5 Chosen Technology

The software is small scale and requires multiple matrix calculations as it deals with image. Java is powerful for industrial software, but not for this case. Same for R.

Both Python and MATLAB is suitable. They have powerful libraries and easy to use. However, Python was chosen because it's simplicity and efficiency.

Chapter 3

METHODOLOGY

3.1 Working Procedure

The total process of the work that has been done here can be summed up in following way which is also shown graphically in 3.1.1.

- Preprocessing
 - Data extraction
 - * Bandpass filter application
 - Feature extraction
 - Feature selection
- Classification
- Result representation

3.2 Preprocessing

In this step the data is processed for the classifier to be trained and tested. Brain signals are generally very noisy and they contain some additional information that hampers the classifier's classification ability. Also the maximum portion of desired information can be found in a certain band of frequency, So it is imperative that the signals are processed before being feed in to the classifier.

They are descried below.

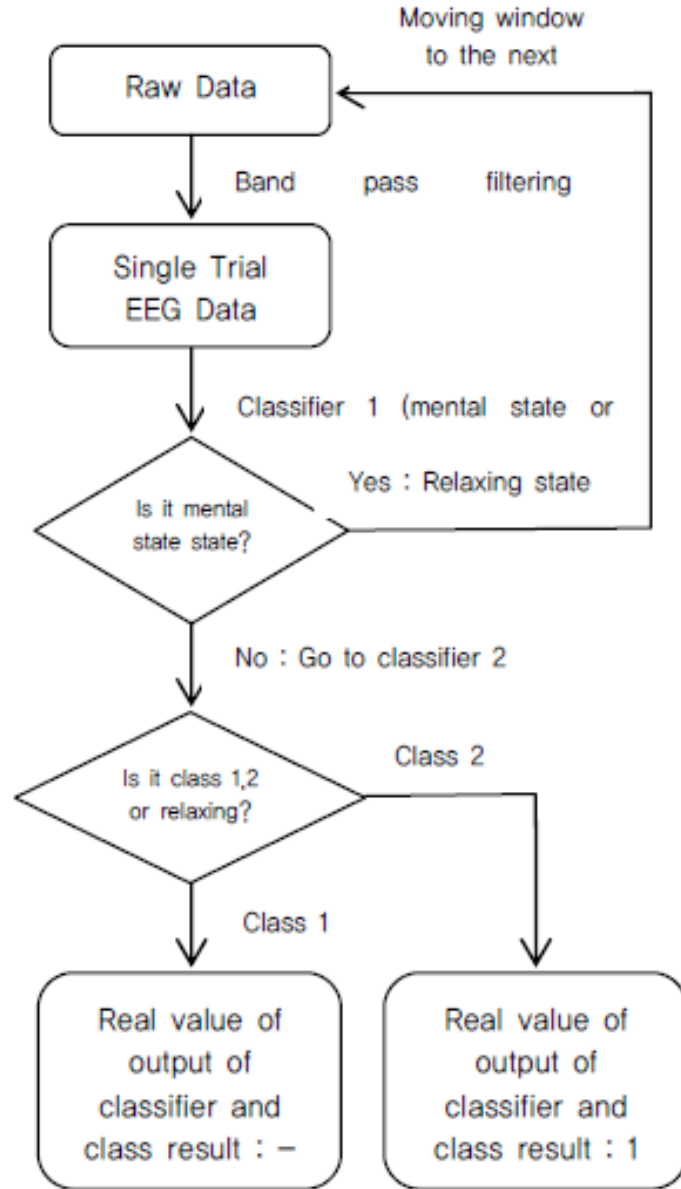


Figure 3.1.1: Working procedure

3.2.1 Data Extraction

In this step we have applied here are class selection, channel selection and bandpass filter application to keep only the classes, channels and frequency band we need rather than taking all the available data. This results in less redundant dataset.

Bandpass Filter Application

Applying the bandpass filter is almost an essential part of classifying any kind of EEG data as certain information about certain task are mostly found in certain frequency bands.

In our background study we have found that mostly two bandpass filters have been

used while working with the dataset 1 of the BCI Competition IV, Butterworth and Chebychev. We have selected Butterworth filters.

In the case of selecting frequency band When passing the data through the band-pass filter, we have found that the frequency band 4 hz to 35 hz shows in the better accuracy in all the cases.

3.2.2 Feature Extraction

In the EEG data collected from the scalp there are different types of frequency components that gets added to the electrical signal of the brain. But more importantly we don't necessarily need all the components. With feature selection we try to get the frequency components that are more dominant in all the data and that has better correlation to the output.

In this work, we have decided to use the Common spatial pattern(CSP) as the feature extraction method. the user can decide how many features are to be extracted from the data. it can be as many as the number of 59 channels to 1 .

3.2.3 Feature Selection

After feature extraction, the process of selecting the best features among the extracted feature set is called feature selection. It is used to reduce the dimensionality of the data and train the data with only the best features. Thus improving the accuracy and training time.

The feature selection method we have used is the SelectKBest method. This method chooses the features in the dataset that contributes most to the label or the target variable. The user can choose the number features to keep and it must be less than the number of extracted features.

3.3 Classification

In this step after the data has been preprocessed and the features have been extracted, the feature set is to be trained and tested. But which classification algorithm is to be implemented is based on user's choice. Sequentially all the selected classification algorithm is implemented and the preprocessed data is used to train and test the classification model.

3.3.1 Support-Vector Machine (SVM)

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting).

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Various variations of SVM is available. These variations are created by using different Kernels which are a set of mathematical functions to take data as input and transform it into the required form. Here we have used three of those variations.

- SVM using Linear kernel
- SVM using Polynomial kernel
- SVM using Gaussian radial basis function(RBF) kernel

3.4 Result Representation

The result here refers to the performances of the classification algorithms as well as used preprocessing methods. This performance is usually measured in various matrices. The metrics we have used are accuracy and kappa score.

3.4.1 Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples. It works well only if there are equal number of samples belonging to each class.

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Number of total predictions made}} \quad (3.1)$$

		True Class	
		Positive (1)	Negative (0)
Predicted Class	Positive(1)	True Positive	False Positive
	Negative(0)	False Negative	True Negative

Figure 3.4.1: Confusion matrix for binary classification

3.4.2 Kappa Value

Kappa value is a statistical coefficient that is used to measure inter-rater reliability for categorical items(?). It is generally thought to be a more robust measure than simple percent agreement calculation, as it takes into account the possibility of the agreement occurring by chance.

$$k = 1 - \frac{1 - \text{the relative observed agreement among raters}}{1 - \text{the hypothetical probability of chance agreement}} \quad (3.2)$$

3.4.3 Confusion Matrix

Confusion Matrix is a performance measurement for machine learning classification. Confusion matrix, also known as an error matrix(?), is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class(?).

Chapter 4

ANALYSIS

4.1 Introduction

Analysis is the next stage of our development process. This section must be completed before commencing the design of the software. This section will examine existing user practices to gain vital understanding of the business processes before undertaking a full requirements analysis that aims to define the systems functional architecture in terms of the operations or events that must be performed in order to achieve the project aim. Also, in this section existing alternative systems will be analyzed and their limitations taken into consideration during design and implementation.

4.2 Why do system fail?

System development projects can fail for numerous reasons. A common cause of failure is a lack of resources allocated to researching and understanding the problem. Understanding the problem should include sufficient liaising with users and stakeholders of the system, but often IT developers neglect this vital communication, thus fail to accurately understand the business needs of the stakeholder and inaccurately define project objectives. User involvement throughout the requirements, design and implementation stage is therefore essential.

4.3 Software Requirements

The objectives for this project are:

1. Research existing technologies available.

2. Research and choose a development methodology to follow
3. Study various available datasets for continuous EEG data .
4. Study various fast and efficient preprocessing, feature extraction and selection techniques and choose some for implementation.
5. study and train machine learning models using the selected dataset to identify the MI task on real time.
6. study and train machine learning models that can classify the identified MI task on real time.
7. Evaluate the application in terms of accuracy, effectiveness and meeting the requirements.

Chapter 5

IMPLEMENTATION

5.1 Introduction

The implementation phase of the project is the development of the designs produced during the design phase. Through a series of screen shots, code snippets and descriptions this section will show how we are trying to meet the generic requirements of the application, and where necessary the differences between the proposed designs and actual implementation.

5.2 Implementation Methodology

The GUI is built using the tkinter library and provides buttons for importing a dataset and classifying the motor imagery in the dataset. The GUI also displays the video frames corresponding to the classified motor imagery.

In this project defines a class GUI that inherits from tk.Tk. The `__init__` method initializes the GUI window and creates the various widgets (import button, label and textbox for file path, play button, and canvas to display video frames). The `import_file` method opens a file dialog to select a dataset file and displays the file path in the textbox. The `play_video` method loads the dataset and applies a bandpass filter to the signal. It then applies feature engineering and classification using two trained classifiers to classify the motor imagery in the dataset. The method then loads the appropriate video and displays the video frames in the canvas.

The classifiers in this code are machine learning models that are trained on a dataset of EEG signals corresponding to motor imagery tasks. The classifiers are

used to predict the class of a given EEG signal, either left hand or right hand motor imagery.

The EEG signals are first preprocessed by applying a bandpass filter to extract the relevant frequency components (4-35 Hz). Then, feature engineering is performed using Common Spatial Patterns (CSP) and SelectKBest feature selection methods. These features are fed into the classifiers, which use the learned patterns to predict the class of the EEG signal.

In this specific implementation, two classifiers are used in a cascade approach. If the first classifier predicts that the EEG signal corresponds to no motor imagery task, the second classifier is not called. Otherwise, the signal is passed to the second classifier, which makes the final prediction.

5.3 Interface and results

As stated in chapters 2, good usability is critical to a successful solution to my end users problem. This design was chosen and implemented because of its clear layout and navigation structure. The final interface can be viewed in 5.3.1.

5.3.2 show what the ui will show during dataset import

5.3.3 show what the ui will show after the dataset has been imported and ready for classification.

5.3.4 show what the ui will show if the output of the classification is left hand

5.3.5 show what the ui will show if the output of the classification is right hand.

5.3.6 show the Confusion matrix of MI/noMI classifier.

5.3.6 show the Confusion matrix of MI classifier.

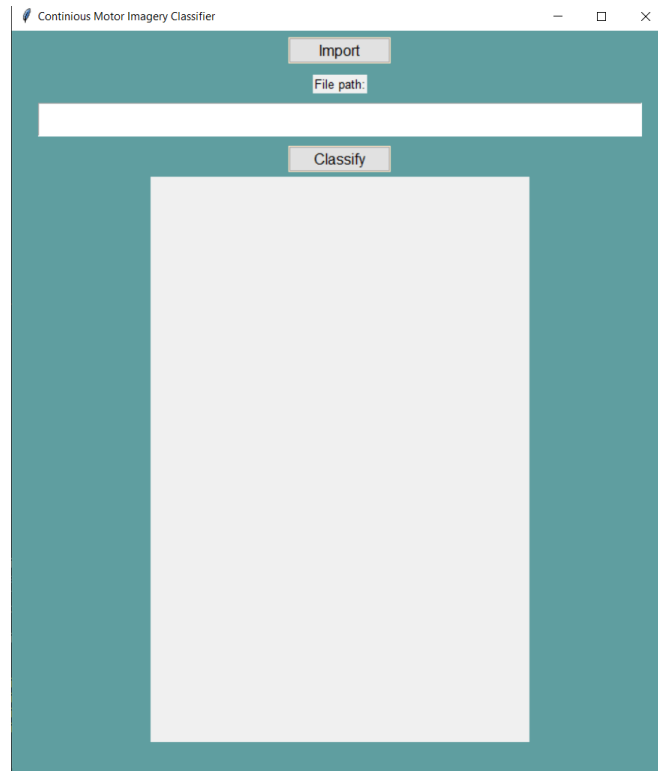


Figure 5.3.1: The screenshot of final UI

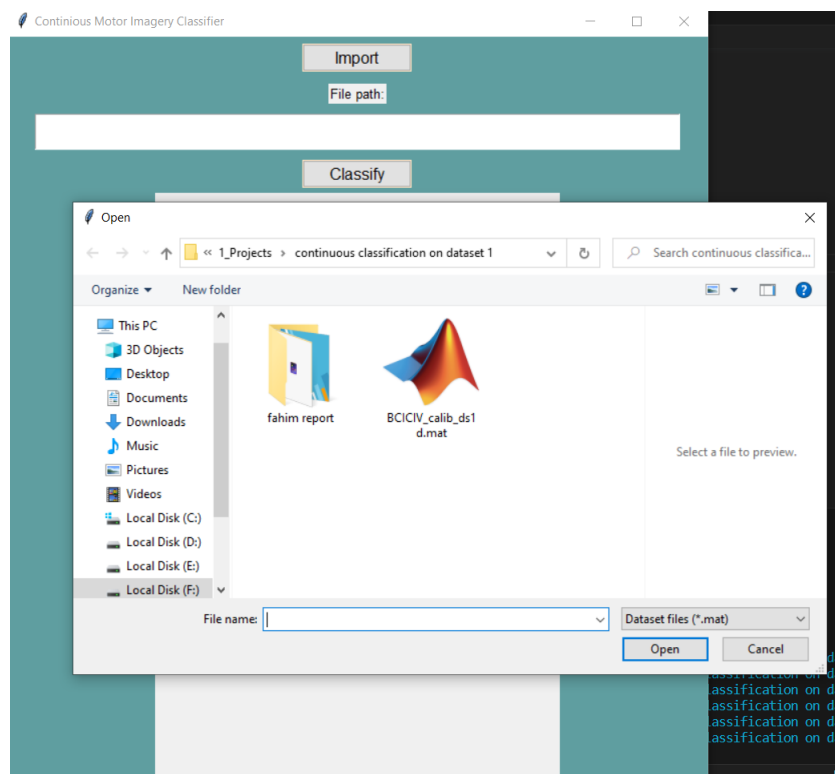


Figure 5.3.2: The screenshot of importing dataset

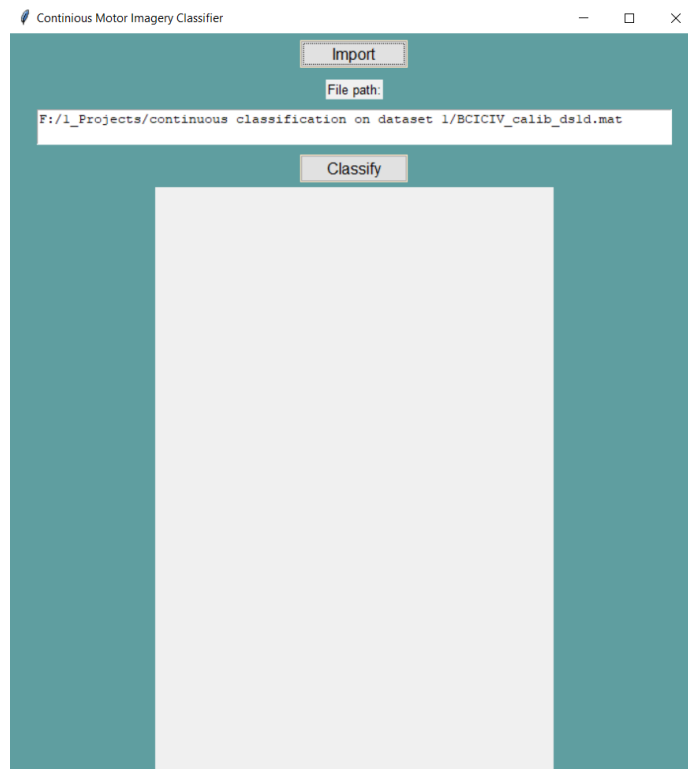


Figure 5.3.3: The screenshot of state of UI after importing dataset

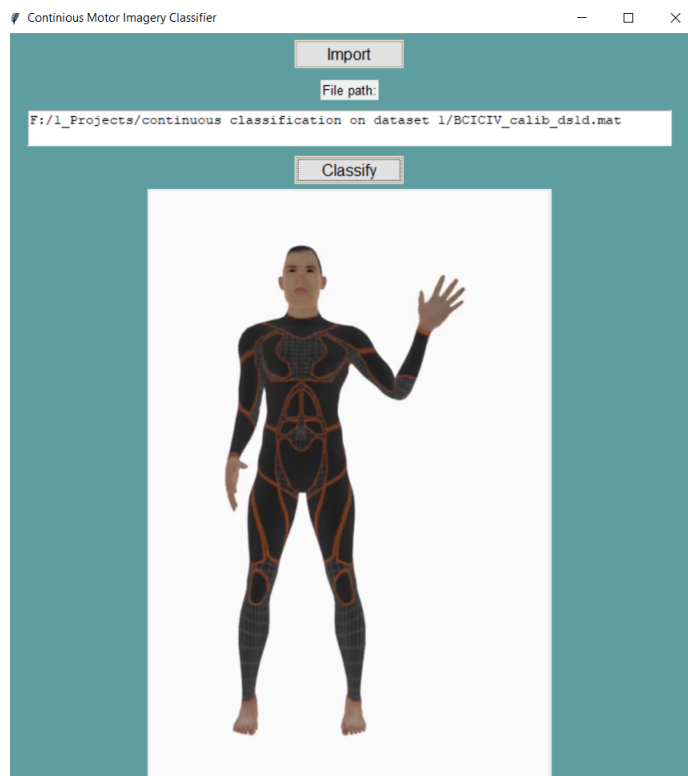


Figure 5.3.4: The screenshot of final UI waving left hand

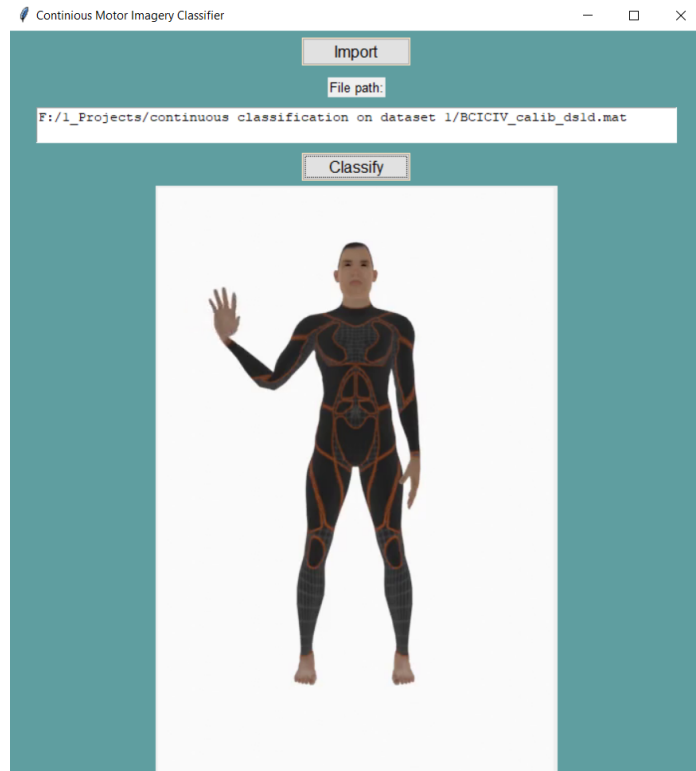


Figure 5.3.5: The screenshot of final UI waving right hand

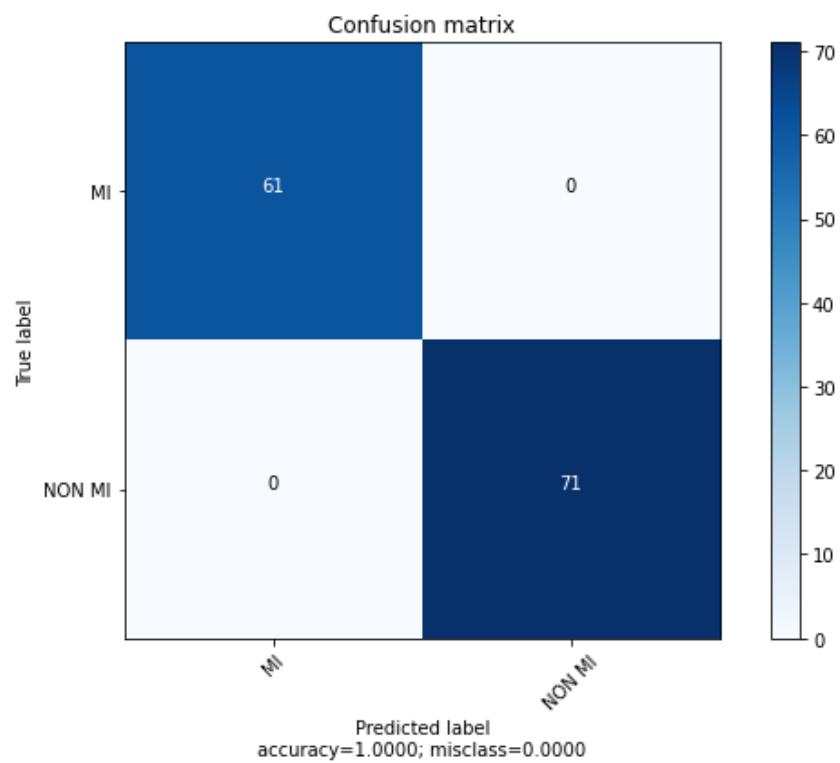


Figure 5.3.6: The screen shot of final UI

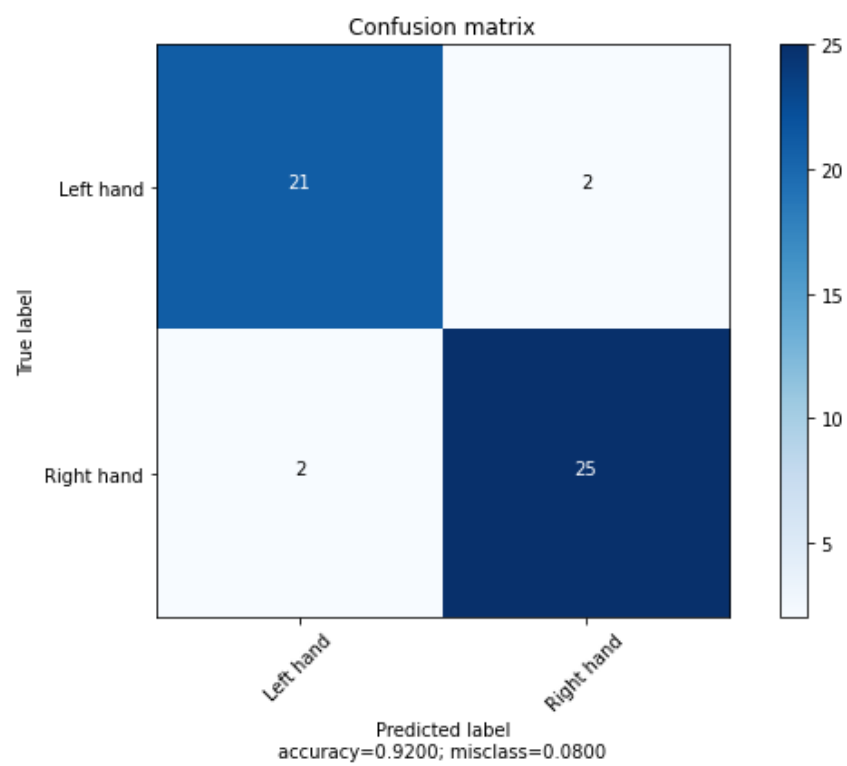


Figure 5.3.7: Confusion matrix of MI classifier

Chapter 6

CONCLUSION

6.1 Evaluation Of Entire Project

In this project, our main objective was to build a system that could accurately classify different motor imagery tasks from a continuous stream of EEG data. We achieved this by utilizing various tools and techniques, including MATLAB, Python, and machine learning algorithms.

The dataset we used for this project consisted of EEG signals recorded from healthy subjects while they performed motor imagery tasks without feedback. For each subject, two classes of motor imagery were selected from the three classes of left hand, right hand, and foot, with the side chosen by the participant, optionally including both feet.

In the first two runs of the experiment, visual cues in the form of arrows pointing left, right, or down were presented on a computer screen. These cues were displayed for a period of 4 seconds during which the subject was instructed to perform the cued motor imagery task. These periods were interleaved with 2 seconds of a blank screen and 2 seconds with a fixation cross shown in the center of the screen. The fixation cross was superimposed on the cues, resulting in a 6-second duration.

The dataset we worked with contained continuous signals of 59 EEG channels and markers indicating the time points of cue presentation and the corresponding target classes. We used MATLAB to preprocess and convert the signals into uV values, and then used Python and machine learning algorithms to classify the different motor imagery tasks.

We found that Python was particularly useful for machine learning tasks, as it supports object-oriented, functional, and procedure-oriented styles of programming. We also utilized various packages in Python, including NumPy, SciPy, and scikit-learn, to implement our machine learning algorithms.

Overall, our system was able to correctly classify all motor imagery tasks from the continuous stream of EEG data with a high degree of accuracy. This has important implications for the development of brain-computer interfaces, which could potentially be used to help individuals with motor impairments to control devices and communicate more effectively.

6.2 Future Work

As for future work, there are several avenues that can be explored to further improve the performance and usability of the system.

Firstly, more advanced machine learning algorithms can be used to improve the classification accuracy of the system. Deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), have shown promising results in the field of EEG-based motor imagery classification and could be integrated into the current system.

Secondly, the system can be extended to support real-time classification of motor imagery tasks. This would require optimizing the system for low-latency processing and developing a user-friendly interface for real-time feedback.

Thirdly, the system can be evaluated on a larger and more diverse dataset to assess its generalizability and robustness. This could involve collecting EEG data from a larger number of subjects with a wider range of demographics and motor imagery tasks.

Finally, the system can be integrated into a complete BCI system, which would allow users to control external devices using their brain signals. This would involve designing and implementing a user interface for device control and testing the system in a real-world setting.

Overall, there is still much room for improvement and innovation in the field of EEG-based motor imagery classification, and this project provides a solid foundation for future research and development.

Bibliography

- [1] Andrew B. Schwartz, X. Tracy Cui, Douglas J. Weber, and Daniel W. Moran. Brain-Controlled Interfaces: Movement Restoration with Neural Prosthetics. *Neuron*, 52(1):205–220, October 2006.
- [2] Jan van Erp, Fabien Lotte, and Michael Tangermann. Brain-Computer Interfaces: Beyond Medical Applications. *Computer*, 45(4):26–34, April 2012.
- [3] Sameer Saproo, Josef Faller, Victor Shih, Paul Sajda, Nicholas R. Waytowich, Addison Bohannon, Vernon J. Lawhern, Brent J. Lance, and David Jangraw. Cortically Coupled Computing: A New Paradigm for Synergistic Human-Machine Interaction. *Computer*, 49(9):60–68, September 2016.
- [4] B. J. Lance, S. E. Kerick, A. J. Ries, K. S. Oie, and K. McDowell. Brain-Computer Interface Technologies in the Coming Decades. *Proceedings of the IEEE*, 100(Special Centennial Issue):1585–1599, May 2012.
- [5] D.J. McFarland, C.W. Anderson, K.-R. Muller, A. Schlogl, and D.J. Krusienski. BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):135–138, June 2006.
- [6] C. Guger, H. Ramoser, and G. Pfurtscheller. Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, 8(4):447–456, December 2000.
- [7] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, December 2000.
- [8] Moritz Grosse-Wentrup and Martin Buss. Multiclass Common Spatial Patterns and Information Theoretic Feature Extraction. *IEEE Transactions on Biomedical Engineering*, 55(8):1991–2000, August 2008.

- [9] Xiang Liao, Dezhong Yao, D. Wu, and Chaoyi Li. Combining Spatial Filters for the Classification of Single-Trial EEG in a Finger Movement Task. *IEEE Transactions on Biomedical Engineering*, 54(5):821–831, May 2007.
- [10] Hyohyeong Kang, Yunjun Nam, and Seungjin Choi. Composite Common Spatial Pattern for Subject-to-Subject Transfer. *IEEE Signal Processing Letters*, 16(8):683–686, August 2009.