2-1-2018

EMG-based eye gestures recognition for hands free interfacing

Ahmed Zahran

The American University in Cairo

School of Sciences and Engineering

EMG-BASED EYE GESTURES RECOGNITION FOR HANDS FREE INTERFACING

By

Ahmed Magdy Zahran

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Robotics, Control and Smart Systems (RCSS)

Under supervision of:

Prof. Dr. Mohamed Moustafa

Associate Professor, tenure, Computer Science and Engineering

August, 2017

DEDICATION

This thesis is dedicated to my parents, my father Eng. Magdy Zahran and my mother Eng. Amira Hussein who were always there for me, supporting me endlessly till I reached this step in my career. There are no words to express my gratitude for your patience and encouragement to confront all the challenges to achieve my goals. Thank you very much for everything you have done for me.

I also dedicate this thesis to my brother Eng. Hisham Zahran, whom I wish all the best in his studies and his career.

ACKNOWLEDGMENTS

First of all, I am deeply indebted to God who graced me with patience, wisdom, strength and who graced me with all the below mentioned people to fulfill that work.

I would like to thank deeply my respectful and great supervisor Prof. Dr Mohamed Moustafa for his time, support and encouragement to achieve this work. I am really grateful for all your efforts and for your keen to produce a well-established and profound work.

Also I would like to thank Prof. Dr Khaled El Ayat for his time, effort and support as well to follow up with that work and his keen and care to produce a profound work as well.

Additionally, I would like to thank Prof. Dr Maki Habib, the Program Director of the Robotics, Control and Smart Systems for being my instructor in several courses throughout this program and for his guidance and supervision on various previous projects from which I particularly learned how to make a proper documentation for your work, which helped a lot in the writing of this thesis.

I would like to also acknowledge HCI Lab, VIS Institute, University of Stuttgart namely Jakob Karlous, Mariam Hassib and Yomna Abdel Rahman for their guidance, support and providing the necessary facilities and equipments to complete this project.

Furthermore, I could not express my deep gratitude to my great friends Amr Nagaty and Mohamed Essam for their great support in the very hard times. Additionally my great supportive friends Eslam Yassin, Waleed Ashraf, Moustafa Moussa, Aesser El-Gindy, Mateen Sayed, Hamza Almschref, Alex Bartelt and all my great friends whom I was graced with and who motivated and encouraged me in the good and the hard times, Thank you very much for being there.

Last but not least, I would like to deeply express my gratitude to my managers namely Dipl.Ing. Ameer Attallah, Dipl.Ing.Jörg Holtfreter, Dr Erwin Christen, Mr Adrien Eyraud as well as my great Senior Dipl.Ing. Uwe Carstensen for their profound support, encouragement and patience to achieve that work and for their keen on my career more than anything else, I am really grateful and honored to work with you all.

ABSTRACT

This study investigates the utilization of an Electromyography (EMG) based device to recognize five eye gestures and classify them to have a hands free interaction with different applications. The proposed eye gestures in this work includes Long Blinks, Rapid Blinks, Wink Right, Wink Left and finally Squints or frowns. The MUSE headband, which is originally a Brain Computer Interface (BCI) that measures the Electroencephalography (EEG) signals, is the device used in our study to record the EMG signals from behind the earlobes via two Smart rubber sensors and at the forehead via two other electrodes. The signals are considered as EMG once they involve the physical muscular stimulations, which are considered as artifacts for the EEG Brain signals for other studies. The experiment is conducted on 15 participants (12 Males and 3 Females) randomly as no specific groups were targeted and the session was video taped for reevaluation. The experiment starts with the calibration phase to record each gesture three times per participant through a developed Voice narration program to unify the test conditions and time intervals among all subjects. In this study, a dynamic sliding window with segmented packets is designed to faster process the data and analyze it, as well as to provide more flexibility to classify the gestures regardless their duration from one user to another. Additionally, we are using the thresholding algorithm to extract the features from all the gestures. The Rapid Blinks and the Squints were having high F1 Scores of 80.77% and 85.71% for the Trained Thresholds, as well as 87.18% and 82.12% for the Default or manually adjusted thresholds. The accuracies of the Long Blinks, Rapid Blinks and Wink Left were relatively higher with the manually adjusted thresholds, while the Squints and the Wink Right were better with the trained thresholds. However, more improvements were proposed and some were tested especially after monitoring the participants actions from the video recordings to enhance the classifier. Most of the common irregularities met are discussed within this study so as to pave the road for further similar studies to tackle them before conducting the experiments. Several applications need minimal physical or hands interactions and this study was originally a part of the project at HCI Lab, University of Stuttgart to make a hands-free switching between RGB, thermal and depth cameras integrated on or embedded in an Augmented Reality device designed for the firefighters to increase their visual capabilities in the field.

Keywords: EMG, Electromyography, eye gesture, Blink, Wink, Squint, Frown, MUSE, Brain Computer Interface, BCI, Dynamic sliding window, Packet segmentation, Threshold Algorithm, Hands free interaction, Augmented Reality, Firefighters.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter introduces the problem definition and the motivation behind this project, then discusses the main purposes that this project will specifically tackle seeking to contribute to the solutions proposed in this field.

1.1 MOTIVATION

Controlling devices remotely with minimal interactions became a major topic of interest in research as well as providing tangible advantages to several real life applications. The minimization of the physical interaction of the user with the supporting devices such as the Virtual/Augmented Reality devices (i.e. Oculus Rift VR) will provide him with better capability and freedom to do other tasks which would rather require his kinesthetic interaction. For instance, several applications would rely on different ways of human hands free control and interaction such as Speech recognition, Electroencephalography (EEG) brain signals (Motor Imagery or Event Related Potentials such as P300), Electromyography (EMG) muscular signals, Electrocardiography (ECG) signals, Electrooculugram (EOG) signals recognition and more.

EMG signals are measured by surface or needle electrodes, as stated by the US National Library of Medicine Medical Subject Headings (MeSH) [1], that detect the electric potential in the muscular cells when they are physically, electrically or neurologically stimulated. Hence, the EMG signals are clear and reliable in terms of signal quality and could be easily identified for further classifications. When the measured EMG signals are put into consideration to control devices and applications, unlike the EEG signals, they do not require training or getting familiar with the way of how to produce them. Typical applications, which relied on the EMG signals, were produced either from hands, arms, legs muscular stimulations or eye gestures. Furthermore, some applications targeted the disabilities, as mentioned by Mangukiya *et al.* [2], who used the self stimulated muscle signals of the disabled person to control the movement of a robotic arm and it was claimed to help effectively in the restoration of the paralyzed arm functionality.

Nevertheless, EMG applications were extended to non medical applications and were integrated with other technologies as well. For instance, firefighters became in need to use more advanced tools to help them find the origin of the fire and the hot sources to extinguish them as quick as possible. In addition, they need to locate the calls for help and to find the possible safe paths to rescue the people under risk. Earlier studies, as conducted by Bennett *et al.* [3] and Wu *et al.* [4], suggested the development of firefighter helmets with an integrated light weight thermal camera to have clearer views through the darkness and smoke and the infrared wavelength is much

longer than for visible light as well. Recently, they are provided with more advanced aiding equipments such as the Augmented Reality devices proposed by Evangelista [5] at EPFL and the prototype of an an Oculus Rift device, designed at HCI Lab, VIS Institute at the University of Stuttgart as shown in Figure 1.1, with three integrated cameras (R.G.B, thermal and depth cameras) to increase the firefighters vision capabilities by changing the views. However, this made the tasks difficult to be all handled together at the same time especially that their hands are busy to handle the other important and main tasks. Hence, typical applications need more flexible robust hands free way to control such assistive devices during their work and that was the motiviation behind investigating the EMG based hands free control in this project.

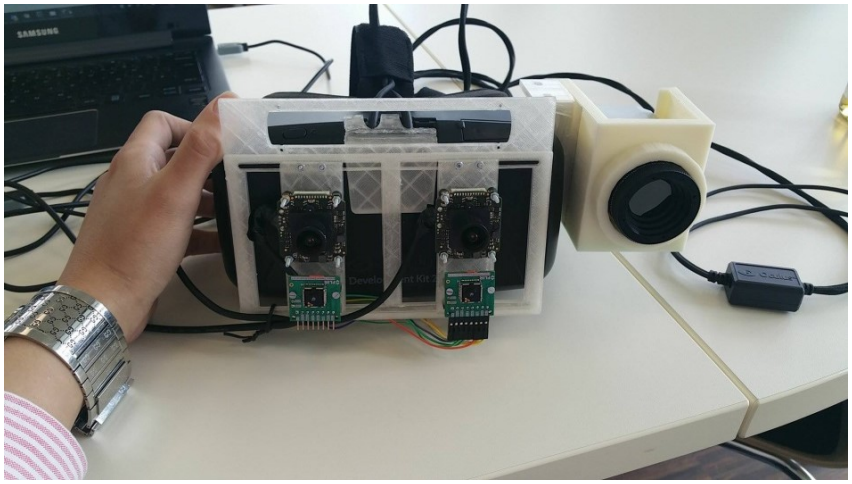

Figure 1.1: OCULUS RIFT with Thermal Imaging Camera and Depth Sensor (VIS)

1.2 AIM OF THE WORK

The aim of this thesis is to investigate the reliability and feasibility of using an EMG based hands free device to interface with different real outdoors applications. In the literature, several research studies were conducted exploring the use of either EEG signals from the brain or EMG signals from facial or eye gestures or muscular contractions or using other Signal types such as ECG and EOG or making hybrid systems. Various projects were motivated by the advancements in the Brain Computer Interfaces (BCIs) which are currently becoming more reliable and easier to use than ever before, especially the current Commercial BCIs which include both EEG and EMG signal acquisitions from the head scalp and include also demo software with classifiers for signals analysis as well as facial and eye gestures recognition. They are currently produced in simpler ways to plug and play directly, as well as being equipped with their signal amplification and conditioning components. In fact, most of the current commercial BCIs are operating on the Wireless Bluetooth communication and some devices became currently manufactured with dry electrodes instead of the traditional electrodes which always needed gel to be in good contact with the skin to read the acquired data accurately.

This work is intended to not only interface directly with application, but rather to be integrated with other technologies as well such as the Virtual and Augmented Reality technologies which are now evolving and became also commercialized to the public in several applications such as the Gaming and entertainment, an in more sophisticated and critical applications such as the Medical, Safety and Industrial Monitoring. Particularly, the future motivation of this project, as a use case, is to further investigate the integration of both technologies together (EMG and AR device) to be utilized in one of the interesting applications with the Firefighters, as discussed in Chapter 3 on page 32, in which the Fire-Fighter would be wearing an Augmented Reality device with three externally integrated Cameras (Thermal, RGB and Depth cameras) and he should be able to switch between the cameras to have different views and to be able to Zoom In and Out with the Depth Cam. But as mentioned at the beginning of this section, this thesis is intended to work and interface generically with other applications as well.

For instance, we had to define the problem and the constraints to narrow down our selection on the suitable technology and how would we later tackle its challenges. So, we are bounded by the following constraints:

• **Constraint 1:** Noisy and Disturbed Environment.
We aim to have a generic robust solution for outdoor applications in harsh environments, therefore it would be very hard to rely on technologies such as Voice/Speech Recognition, Brain EEG Signals. Additionally, it would be a great challenge to use camera tracking systems since they would not perform well under different lighting conditions and need more setup

• **Constraint 2:** Busy hands in the main operation.
As previously highlighted by the Fire-Fighters example that their hands are engaged in doing their main tasks and there is a need for a robust hands-free device to interface with the assistive technologies. Additionally, it is hard to aquire the EMG signals from the Limbs (hands or legs) as they are already under high load of work.

Hence, the utilization of an EMG based eye gestures recognition device became our scope of investigation in this project, since the eye gestures such as voluntary blinks, winks or squints (frowns) could be made intentionally on purpose and their signals are having more clear features. Additionally, unlike the EEG signals, they are not be affected by the internal signal disturbances such as inattentiveness, as well as the external stimulations such as the visual or auditory stimulations. On the otherside, the Brain EEG signals will not be reliable in such disturbed environments as mentioned in the first constraint. In addition, although the eye gesture signals are considered as artifacts when being interested to read the EEG signals only, however they are more pure and easier to recall and classify compared to the EEG when exploited as control signals as illustrated more and referenced in the next chapter.

Nevertheless, in the next chapter Chapter 2 on page 18, the different methods and technologies of the eye gestures extractions will be discussed to have an overview on their differences, advantages and disadvantages so as to show the reason which led us narrow our scope to test the EMG method of extraction and put it into consideration.

1.3 STRUCTURE OF THE THESIS

The work is structured as follows:

Chapter 2 – Background and Related Work: Overview on the evolution of the Brain Computer Interfaces, then a literature survey on the different Eye-based facial gestures extraction methodologies.

Chapter 3 – Concept and Proposed Solution: Proposed Solution, Hypotheses, Project Flow Diagram and the Data Collection Criteria

Chapter 4 – Experimental Methods: System Setup and Implementation of the user studies.

Chapter 5 – Results and Discussion: Analysis of the Results and Comparison with the related literature

Chapter 6 – Conclusion: Summary of the thesis and its contribution, Future work

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter provides an overview on some of the recent technologies related to the utilized devices in this work and their applications. Additionally, it summarizes some of the previous work related to the detection and the classification of the eye gestures. In fact, it compares the different detection techniques and sensors used to capture the eye gestures or the facial gestures in general whether considered as artifacts to be filtered out and eliminated or as clear signals which could be exploited for several applications. Since the device specifically used in this project is originally a Brain Computer Interface (MUSE headband), but is used to read the EMG signals from the eye gestures in our case, thus the Brain Computer Interfaces will be first demonstrated briefly, then an overview over the Brain EEG and the muscular EMG signals will be provided to differentiate between both of them. Furthermore, some previous work integrating the Brain Computer Interfaces with the Augmented/Virtual Reality devices such as the Oculus RIFT is briefly mentioned at the end of this chapter.

2.1 THE BRAIN COMPUTER INTERFACES

2.1.1 THE Evolution of Brain Computer Interfaces

The use of Brain Computer Interfaces (BCIs) is currently being widely investigated in several projects and studies aiming to adopt the concept of Hands free control over various applications. BCIs can read Brain Electroencephalography (EEG) signals directly from the head scalp. In fact, the first real experimental demonstration of acquiring the brain signals to control an application was not until 1999 when Chapin *et al.* [6] used the neural activity acquired invasively from the motor cortex of rats brain to control a robotic arm. However, the concept of using the EEG signals itself in control was earlier presented in the 1970s with a publication under the title of "Toward direct brain – computer communication" by Vidal [7].

For instance, advancements in neurology, signal processing techniques, sensor miniaturization and bio-materials have facilitated further developments into the invasive and noninvasive BCI interfaces. van Erp *et al.* [8] discussed the comparison between the different methods of capturing the brain activities, as well as the pros and cons of each method as summarized in Table 2.1 on the next page.

Hence, as van Erp *et al.* [8] contended that although the EEG signals are not ideal, however they are the simplest and currently the fittest to be read by the Brain Computer Interfaces. The Non-invasive BCIs are safe, wearable and do not require sophisticated setup. Some BCIs require conductive gel to avoid noisy measurements and others proposed working with dry

Table 2.1: Different techniques of Reading the Brain Activities

Hardware	acronym	Description	Advantages	Disadvantages	Invasive
Electroencephalography	EEG	Direct measure of Neuronal Activity	Cheapest & Portable	Poor Spatial Resolution	No
Magnetoencephalography	MEG	Direct measure of Neuronal Activity	Good Temporal Resolution	Poor Spatial Resolution	No
functional Magnetic Resonance Imaging	fMRI	Magnetically	Good Spatial Resolution	Poor Temporal Resolution	No
functional Near InfraRed Spectroscopy	fNIRS	Optically	Portable	Poor Temporal Resolution	No
Electrocorticography	ECoG	Sensors at Cortex level	Excellent Temporal & Spatial Resolution	Invasive – Medical purposes Only	Yes
Subcortical Electrode Arrays	SEA	Sensors Under Cortex level	Excellent Temporal & Spatial Resolution	Invasive – Medical purposes Only	Yes

electrodes instead. EEG signals are recorded with BCI electrodes placed on a user's scalp and capturing his brain activity. This technique compared to other methods of brain study such as the Magneto-encephalography– MEG, functional magnetic resonance imaging – fMRI, Near-Infrared spectroscopy NIRS), enables the evaluation of brain activity in real time, operates under various environmental conditions, and ensures relatively low implementation costs as stated by Holewa *et al.* [9].

The use of BCIs was conceptualized traditionally according to Wadsworth Center, as stated by Wolpaw *et al.* [10], as an alternative way to provide muscle free communication for the severely paralyzed patients. As shown in Figure 2.1, the brain-computer interface (BCI) is a data acquisition device which provides communication from the human brain to a computer or a machine, and the generated brain waves are called (EEG signals).

Recently, the definition of the BCIs was reformed according to Kothe [11] at the Swartz Center for Computational Neuroscience (SCCN) as the system which acquires the Bio-signal from a person and predicts the abstract aspect of the person's cognitive state as shown in Figure 2.2.

The traditional focus of the field on the medical applications for the severely disabled patients was demonstrated in typical applications such as the Brain Controlled Wheelchairs as controlled by Motor Imagery (thoughts) by Carlson, Tom and Millan [12] and by external stimulations such as visual oddball one used by Rebsamen *et al.* [13] to extract the P300 from the EEG signals. In addition, the Keyboard Characters' spellers controlled by the Brain were also earlier introduced by Farwell *et al.* [14] and still there are further studies to improve their speed and efficiency to better serve the disabled. In fact, the field currently became mature enough to

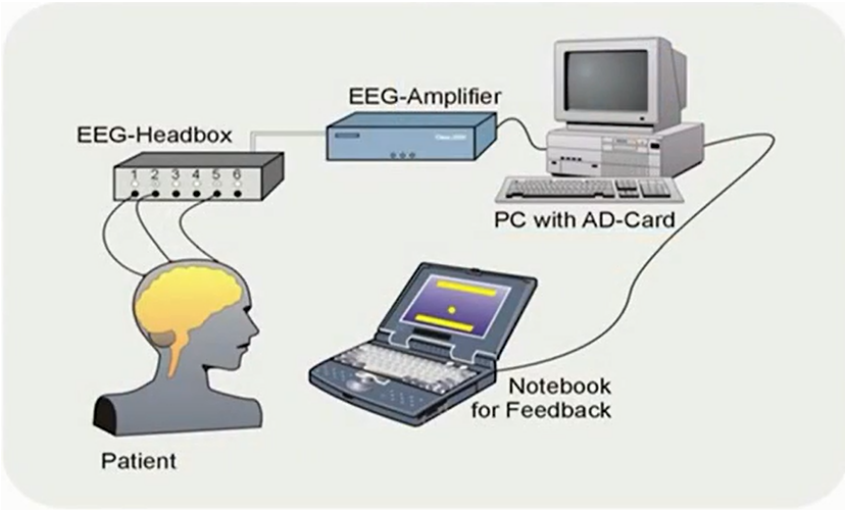

Figure 2.1: BCI System (Wadsworth Center) Wolpaw *et al.* [10]

Figure 2.2: Bio-signal acquisition, then Cognitive State Prediction SCCN Kothe [11]

have new roles in the non-medical applications span such as using the BCIs for gaming and entertainment, Devices control as well as user psychological state monitoring as highlighted in the review paper published by van Erp *et al.* [8].

For instance, some remarkable challenges in the BCIs are considered in terms of usability, reliability and consistency. In terms of the usability, BCIs should be operated easily and without an exhaustive training and long calibration. In addition, the reliability is to a great extent related to the signal processing challenges such as; Robustness to noise and mental distractions, Asynchronous and Continuous (Self-Paced) BCI which gives more flexibility to the user to control anytime. Furthermore, the interpretation of the signals acquired is one of the key challenges which relies on extracting the distinguishable features and effectively classifying them according to the intended application commands.

2.1.2 BCIs - EEG and EMG Signals Detection

One of the powerful advantages of the BCIs is that they can capture the EMG (Electromyography) signals as well, besides the EEG (Electroencephalography) signals. The kinesthetic movements and the muscular contractions can be normally detected from the EEG interpretation. However, BCIs which are having electrodes mounted on the forehead can simply detect the facial artifacts (e.g. eye gestures) on the EEG signal. In the previous work which focused on using the Brain EEG signals solely, these EMG artifacts were considered as contaminations that need to be

Figure 2.3: The Emotiv Epoc Neuroheadset

analyzed and filtered out from the acquired EEG Signal as Manoilov [15] referred. In fact, As Reyes *et al.* [16] suggested exploiting them for non-medical applications as they are normally clear and directly analyzed.

In next section, the different extractions types of the eye based facial gestures will be demonstrated to have an overview on the diversity of the used technologies to detect such gestures and their applications.

2.2 EYE GESTURES EXTRACTION METHODOLOGIES

In this section, the eye based facial gestures will be demonstrated to have an overview on the diversity of the used technologies to detect such gestures and their applications.

2.2.1 EXTRACTION using EEG - Electroencephalogram

In the previous section, one of the challenges confronting the acquisition of solely pure EEG Signals was the facial artifacts or the eye gestures which are normally considered as contaminants especially for medical diagnosis applications. Some references suggested analyzing those artifacts and considering them as good control signals for typical applications. For instance, Reyes *et al.* [16] used the Emotiv EPOC BCI Headset as shown in figure 2.3 to read EEG Signals, but focused on extracting and analyzing the signals evoking from six facial gestures: blink, left wink, right wink, raise brow, smile, and clench.

The signals were also processed in the Frequency domain, but the Smile and Clench were the only shown to be identifiable. The study was conducted on 12 participants (equally from both genders) in which the individual is instructed by the test director to do one gesture at a time to be recorded and analyzed separately. Reyes *et al.* [16] implemented three applications to test the system and as a prototype as well: Sticking Tongue Out, Home Control Application and

Patient Communicator. However, it was reported that their approach needed further advanced classification methods and more collected data for future work.

2.2.2 EXTRACTION using EMG - Electromyography

Other studies such as Chambayil *et al.* [17] classified the contamination of the EEG signals into two groups: technical and biological artifacts. The technical artifacts are generally due to the external power source noises or disturbances, while the biological artifacts are caused by Eye movements, ECG and EMG artifacts. Hence the Eye Blinks were considered as control signals instead of artifacts to select characters on a Virtual Keyboard application programmed in LabVIEW. They extracted the blinking feature by getting the Kurtosis coefficient and the amplitudes of the blinking signal. Figure 2.4 below shows the flowchart of the work starting with the Data acquisition of the EEG signals then the implementation of the Signal Processing and Feature Extractions methods and finally classifying the signals to execute the control signals which controls the device and the user is provided with visual feedback.

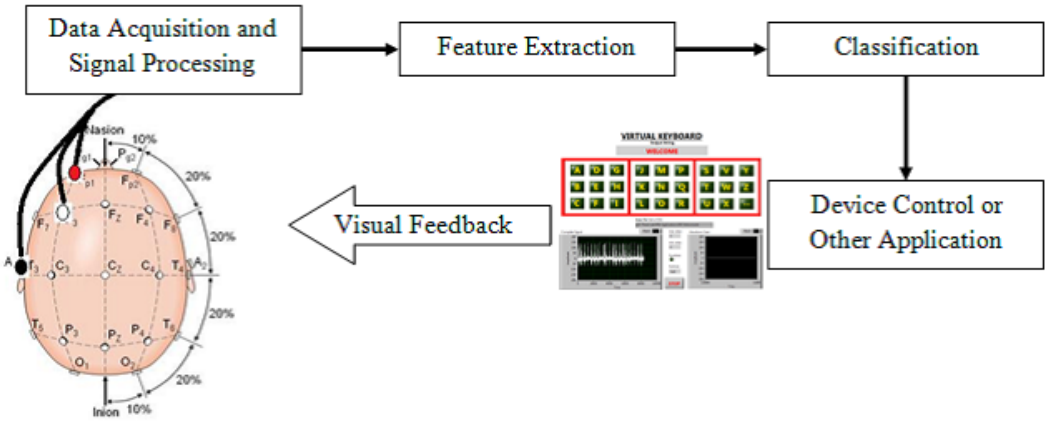

Figure 2.4: BCI system using Eye Gestures, Chambayil *et al.* [17]

The Virtual Keyboard application was designed as a "Three levels selectors" to make a single decision for a character after approximately 1 min. The keyboard characters [A-Z], in addition to the Space [–] were divided equally into three blocks. Eye blinking actions should be detected every 20s to narrow down the selections from the first stage on the blocks level, then the Second Stage on a Column of three Characters' level and finally a character out of them is selected. The selection of any of the three choices at each level (3 Blocks then 3 Columns and finally 3 Characters) is based on how many blinks were detected in the specified 20s time interval for each single stage selection as shown in figure 2.5. For example, if the user made 2 Blinks in the first 20s, then the 2nd block is selected and glowed. Afterwards, if he made 3 blinks in the second 20s then the 3rd Column (P - Q - R) is selected and finally if only one blink was detected in the third and last 20s interval, then the Character is "P". The 20 seconds interval specified

for every single stage selection is sub-divided into four 5 seconds clips in which the user could make one pure blink within.

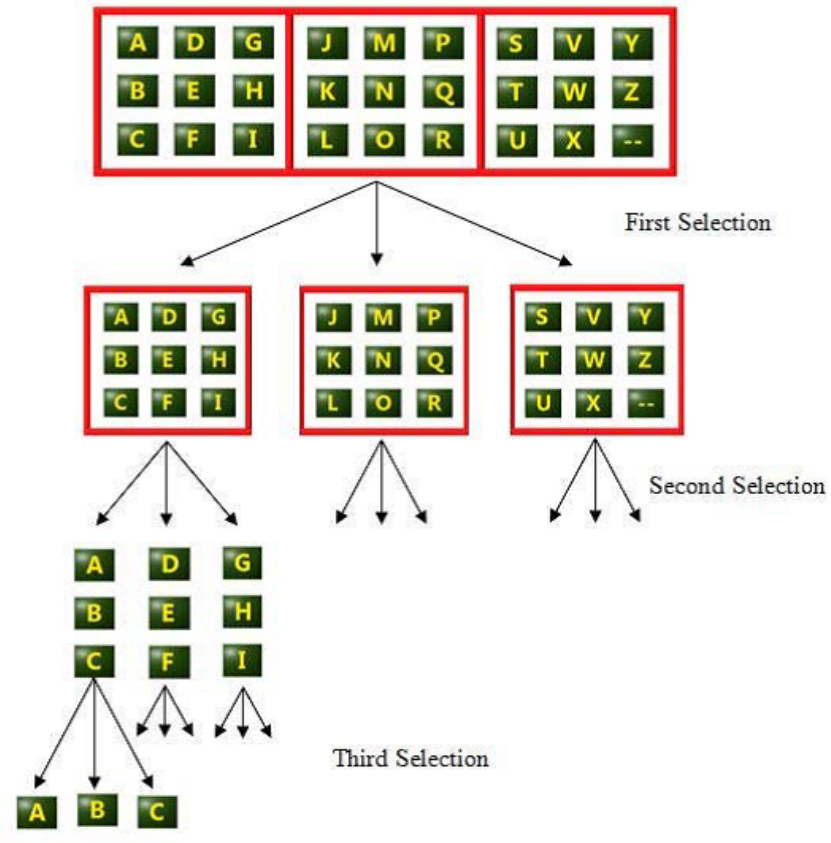

Figure 2.5: The Virtual Keyboard Character Selection Stages, Chambayil *et al.* [17]

Chambayil *et al.* [17] tested the system on 14 participants while being supine, lying on bed and relaxed. An Eye closed Calibration was made at the beginning of each test and the user study was instructed by a director. It was claimed that their Correct Spelling Rate was 1 Character per minute which was improved a lot compared to other studies. However, the reduction of the eye blink given time for detection was suggested as a future improvement.

2.2.3 EXTRACTION using EOG - Electrooculogram

Other studies used the Electrooculogram (EoG) method to build hands free control applications. In Valeriani *et al.* [18], the eye winks were exploited as the control signal to operate the Smart phones and they were measured from the forehead by two non-invasive electrodes. The motivation behind the use of eye winks as a Control signal was also due to the difficult situations in which both hands are busy (i.e. while driving). In addition, it has more advantage and robustness over the other operation alternatives of the smart phones such as the voice recognition especially in noisy and disturbed environments. The eye blinks were not targeted in

this application; hence the free placement of the electrodes as well as the potential comparison between them made it easier to avoid the false positives resulted from involuntary eye blinks.

The architecture of the system is demonstrated as shown Figure 2.6, in which the user is placing two electrodes above or on the sides of the eyes, and then signal amplification is done by the OpenBCI board which is connected via Bluetooth to a computer that makes the signal analysis and classifications to extract eye winks.

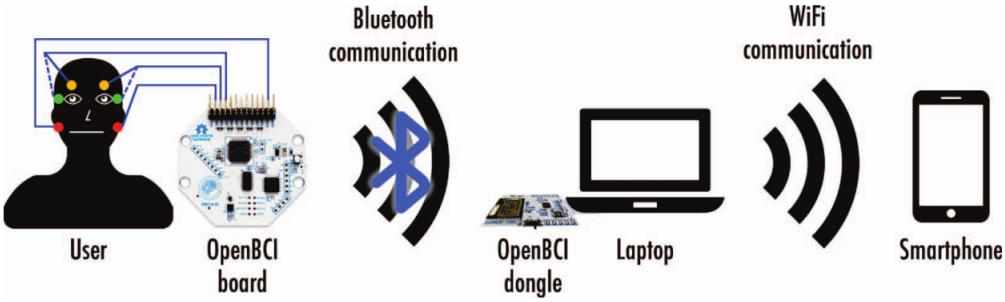

Figure 2.6: Architecture of the SystemValeriani *et al.* [18]

2.2.4 Extraction using Cameras

Further studies additionally detect the eye based facial gestures using cameras such as Missimer *et al.* [19] who designed a mouse pointer control using voluntary blinks and winks. They included the pointer motion and all types of mouse click in addition to dragging and dropping. An accuracy of 96.6% in the eye gesture classification was reported even without the need for a prior training. However, one of the limitations of this study is that the user has to be directly facing the camera and could not turn his head. Also the users wearing glasses are having unsatisfied results when exposed to different lighting conditions due to reflections. Furthermore, it was stated that the wink detection still needs more improvements to overcome the eye irregularities by the user such as winking with one eye and engaging the other eye involuntarily.

This section summarized and made an overview on the different technologies used to extract the eye based facial gestures. As previously mentioned, some limitations are obstructing us in this project to use some of these technologies such as the cameras as the user will be exposed to different lighting conditions, in addition to the noisy and disturbed environment which negatively affects some other very sensitive signals such as the Brain EEG signals.

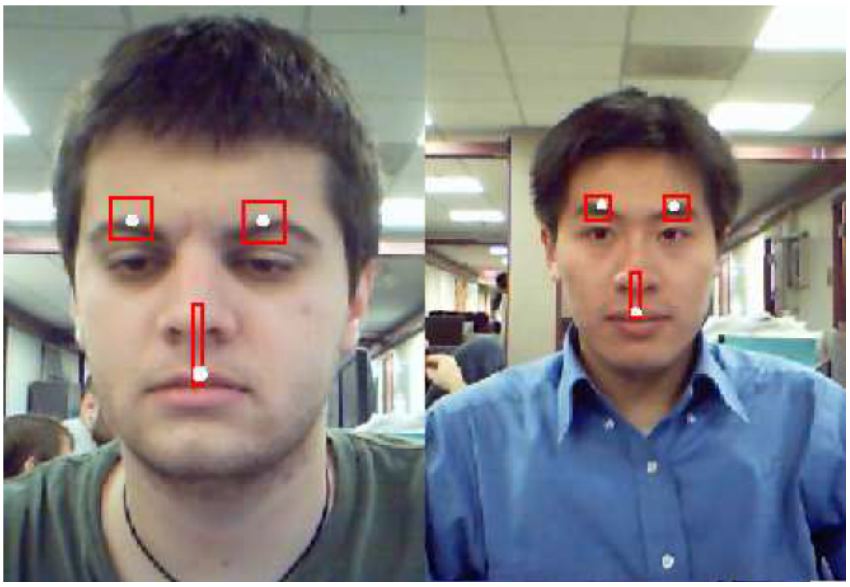

Figure 2.7: Outlined Rectangles spotting the Tracking points.Missimer *et al.* [19]

2.3 HYBRID APPROACHES USING EYE-BASED FACIAL GESTURES

2.3.1 EMG and EEG Signals - Emotiv BCI

George *et al.* [20] implemented an application to use the facial expressions and mid states to insert text acronyms into a message. The Emotiv EEG Headset as shown before in figure 2.3 was used as well as the Emotiv Control Panel Software to sense and capture the EEG data, then interpret and convert the expressions and the EEG data to the matching shortcut texts. The EMG data of the facial expressions is also acquired by the Emotiv headset such as blinking, smiling and winking.

The software of the Emotiv Control Panel has two packages in which the first (Expressiv suite) is responsible for the facial expressions as well as the muscles activity and the second (Cognitiv suite) captures the Event Related desynchronization (ERD) patterns. The flowchart of the system as demonstrated in Figure 2.8 on the following page starts by collecting the EEG data then checks and processes the mental and the facial expressions, which are classified under the ERD and the EMG respectively.

Some Commands for text acronyms were generated by the EEG mental state such as thinking of pushing something which will write down "L8R" (identified as the acronym of the word "Later"). In addition, the rest of the commands were matched with the predefined facial expressions which had their sensitivity adjusted to avoid false positives. Facial expression commands such as Blinking were used to write down "BTW" (identified as an acronym for the word "By the way"). Figure 2.9 shows the list of commands for the predefined text acronyms.

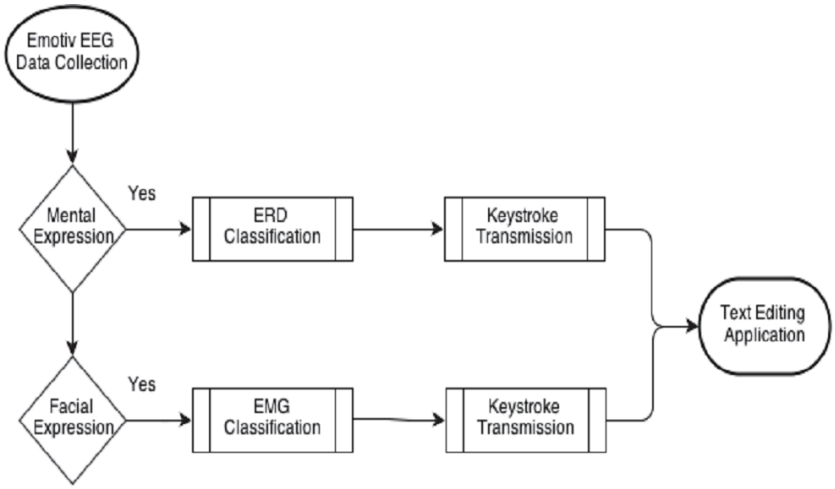

Figure 2.8: Flowchart for brain signal measurement & collection process George *et al.* [20]

Mental State				
Acronym	***Acronym Meaning***	***Test Sentence***	***Command***	***Percent Accuracy***
L8R	Later	I'll see you L8R	Push	63.04
TMI	Too much information	Dude, that's TMI.	Left	93.33
OIC	Oh, I see	OIC	Drop	70.73
XOXO	*Hugs and kisses*	I'll see you tonight dear. XOXO	Rotate Left	58.00
Facial Expression				
TTYL	Talk to you later	I'm busy right now, TTYL.	Look Left	100
LOL	Laughing out loud	Geologist will date anything. LOL	Look Right	94.11
BTW	By the way	BTW, I'm going to the store.	Blink	100
IDK	I don't know	IDK when I'm coming home.	Left wink	81.13
JK	Just kidding	I fell down the rabbit hole.	JK Right wink	100
GR8	Great	That's Gr8 you bring the drinks.	Raise Brow	100
POV	Point of view	Well that's my POV.	Clench Teeth	89.47
THX	Thanks	Thx for the invite.	Smile	85.00

Figure 2.9: List of Emotiv Commands for Text Acronyms George *et al.* [20]

As previously discussed, the sensitivity of the EMG data corresponding to the facial expressions can be adjusted from the Emotiv Control Panel software as shown in Figure 2.10 on the next page in order to avoid false positives or in other words, unintended but right signals generated causing the actions (commands) to be executed.

George *et al.* [20] discussed the results of each of the two approaches and concluded that the EMG based technique had higher accuracy with low false positives, while on the other hand

Figure 2.10: Expressiv EMG Sensitivities with keystroke association George *et al.* [20]

many false positives occurred in the ERD based technique. Hence, their attempt to combine between both techniques was relatively successful but still needs more accuracy. Also, they contended that their results were not that accurate due to some shortcomings in the software as well as the difficulty in the mental state to make the thoughts focusing only on the specified commands.

2.3.2 EOG and EEG Signals

Ma *et al.* [21] proposed a hybrid human machine interface as a novel to use the Electro-oculography (EOG) together with the EEG signals to control a robot. The aim was to integrate both techniques to have more robustness to the BCI-related applications. For instance, the EEG mode relied on the detection of Event related potentials such as the P300, N170, and VPP evoked from the stimulation based oddball paradigm. On the other side, it was mentioned that the eye movements such as Blinking, Winking, Gazing and Frowning (Squinting) lead to voltage fluctuations which are then detected by the EOG and classified using a thresholding algorithm. The efficiency of the proposed system was verified by conducting two experiments in which the first was a "Multifunctional Humanoid Robot Control" and the second was concerned with the control of four mobile robots as shown in figure 2.11.

The imposition of EOG signals on the acquired EEG signals from the BCI are considered as artifacts when dealing with only the EEG-based applications, as they are affected by the eye motion. Nevertheless, their high response speed and clarity make them feasible for use in typical applications and human machine interfaces which would allow the engagement of eye gestures for control.

(a) (b)

Figure 2.11: Experimental robots (a) Humanoid robot NAO (b) Mobile robot Kobuki Ma *et al.* [21]

Despite their highlighted advantages, Ma *et al.* [21] claimed that the number of outputs for an EOG-based system would be limited due to the limitation of the detected eye gestures by such a system. Additionally, their accuracy would be relatively poor compared to the video-based eye tracker. Finally, they cause fatigue over time and utilization as they rely on the muscular stimulation unlike the BCI. However, the use of the EOG in this project was dedicated for fast-response tasks and hybridized with the EEG as shown in figure 2.12 which had another independent role for menu-selection tasks.

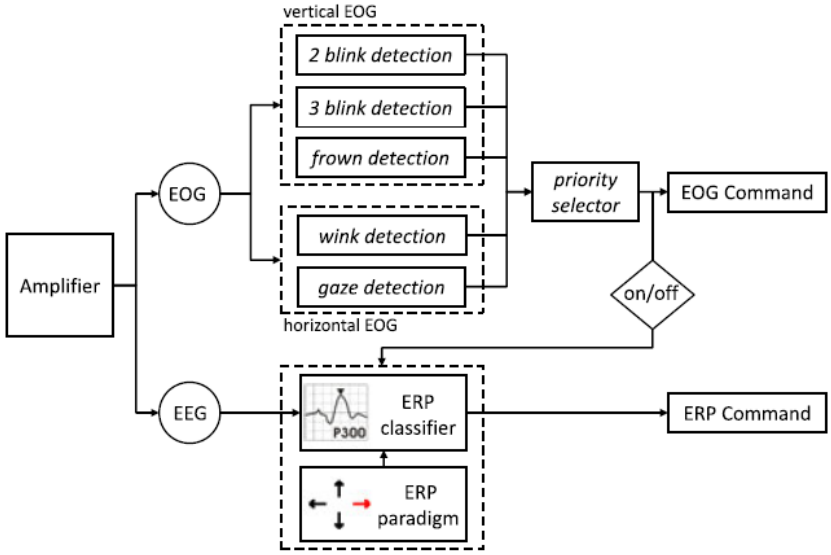

Figure 2.12: Hybrid EOG + EGG HMI proposed model Ma *et al.* [21]

2.4 BCI AND AUGMENTED/VIRTUAL REALITY APPLICATIONS

Typical applications in the literature combined the use of the Brain Computer Interfaces with the Virtual or Augmented Reality devices. This section will provide an overview on some of the previous related work integrating both technologies.

2.4.1 PsychicVR

The PsychicVR project implemented by Amores *et al.* [22] at the MIT Media lab integrated the MUSE BCI headband and the Oculus RIFT Virtual Reality device to increase attentiveness for entertainment. They introduced a 3D virtual environment where the user can feel the fantasy of possessing superhero powers when concentrating such as levitating himself, Telekinesis to allegedly levitate and affect surrounding objects as well as Pyrokinesis in which a person can intendedly create and control fire with his mind and finally an X-Ray vision as shown in Figure 2.13 depending on the concentration levels of the user.

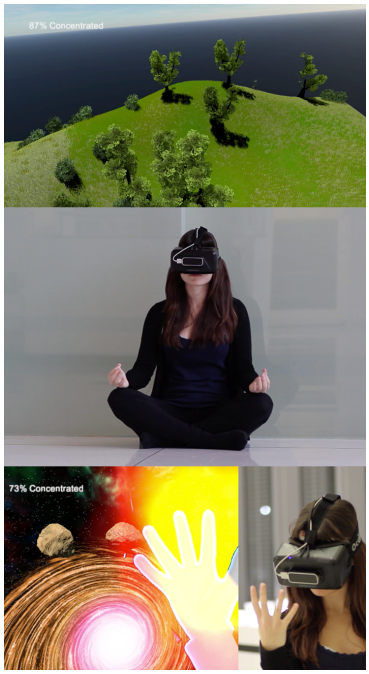

Figure 2.13: PsychicVR: Virtual Levitation & Pyrokinesis according to the user's Concentration levels, Amores *et al.* [22]

The Muse BCI was originally established and marketed as a Brain Computer Interface for Meditation and they even had desktop and mobile entertainment applications to meditate and increase your concentration levels. For this reason, PsychicVR exploited these features and read the EEG signals from the MUSE BCI and deployed it as a feedback to the Oculus RIFT to change the 3D virtual environment views accordingly. Hence, the higher level of concentration the user becomes, the more control to his virtual power he would gain.

2.4.2 PhysioVR

The PhysioVR was another project created by Muñoz *et al.* [23] introducing an open source framework for the mobile virtual reality (mVR) applications which integrated the Virtual Reality technology with several wearable devices such as the Brain Computer Interfaces, Heart Rate and EMG sensors.

Muñoz *et al.* [23] reported that the MUSE BCI, the Android Wearable Heart Rate sensor as well as the EMG Myo armband are the devices supported within their implemented framework as shown in Figure 2.14

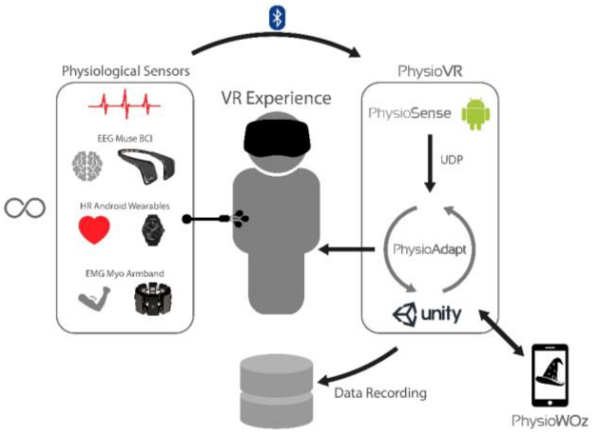

Figure 2.14: PhysioVR App connects the Sensors to the smarthphone via Bluetooth communication Muñoz *et al.* [23]

CHAPTER 3

CONCEPT AND PROPOSED SOLUTION

This chapter provides an overview of the proposed solution and the hypotheses that this project presented and aimed to tackle. Additionally, the flow diagram of the full big project will be demonstrated as well as our scope specifically in this project. Then the Open Sound Control (OSC) library, which the MUSE headband uses as its communication protocol, will be introduced briefly. Finally, the data collection criteria for this project will be proposed at the end of this chapter.

3.1 OVERVIEW ON THE PROPOSED SOLUTION

The challenges discussed in the problem definition section including the disturbed noisy environment and the targeted hands free interaction for users such as the firefighters, as well as interfacing our device with another technology such as the proposed manipulation of views for an AR device with equipped cameras directed us to investigate the idea of implementing the EMG based eye gestures control. In addition, we exploited the advantage of having recent Commercial headbands with electrodes which can capture both EEG and EMG signals, thus we used the simple Commercial Muse headband as shown in Figure 3.1 on the following page. However, EMG signals can be generally acquired using any lab EMG electrodes and data acquisition hardware.

The MUSE headband includes four channels: Two electrodes on the forehead muscles and two Smart-Sense rubber ear electrodes at the mastoids (behind the earlobe). The reason behind the utilization of the MUSE headband in our project, is its suitability and flexibility for the application purposes; for instance its size is relatively small, easy to wear and adjustable on the forehead with good contacts by the dry electrodes which require no Gel compared to some other headbands. Additionally, it is connected to the PC via Bluetooth wireless communication and equipped with the signal amplification and conditioning components. Although only four channels would be claimed as few or very limited, but our aim is to capture only 5 eye gestures and we are investigating the efficiency of recognizing those eye actions from only those 4 sensors in these positions since we do not want to load the user with sophisticated heavy devices with redundant electrodes which would also affect the overall performance.

The user should perform the following proposed eye gestures to recognize and classify them to be interfaced with applications such as the ongoing proposed prototype by HCI Lab, VIS Institute, University of Stuttgart to interface the Muse Headband with the Oculus RIFT to manipulate the Thermal, RGB and Depth Cameras views. In addition, the aim of this whole prototype is to

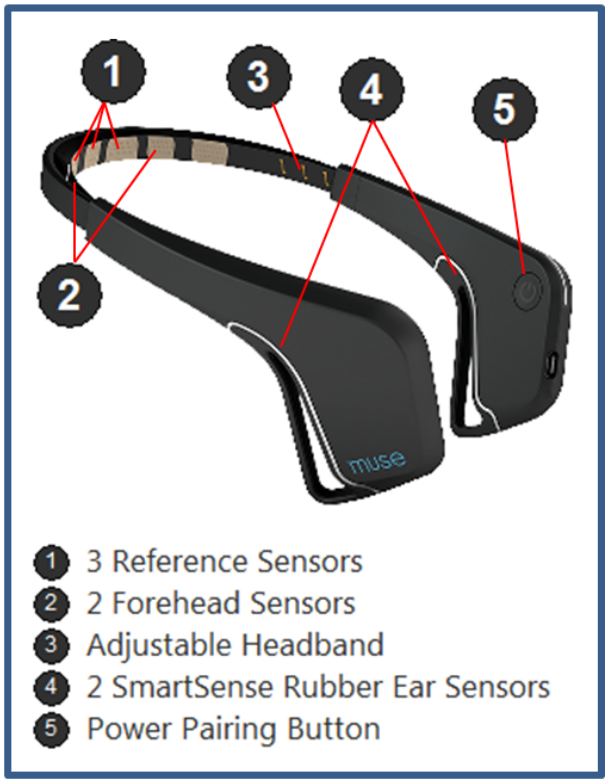

Figure 3.1: Commercial MUSE headband with 4 Channels

make the user have the feeling of gaining an extension to his eye (vision) capabilities as shown in Figure 3.2.

1. Blinking Rapidly
2. Long Blink
3. Winking Right
4. Winking Left
5. Squinting

The above listed actions are voluntarily done, but the challenge is to differentiate between them and the involuntarily eye blinks. For this reason, the Rapid Blinking or Long Blink are suggested to be investigated.

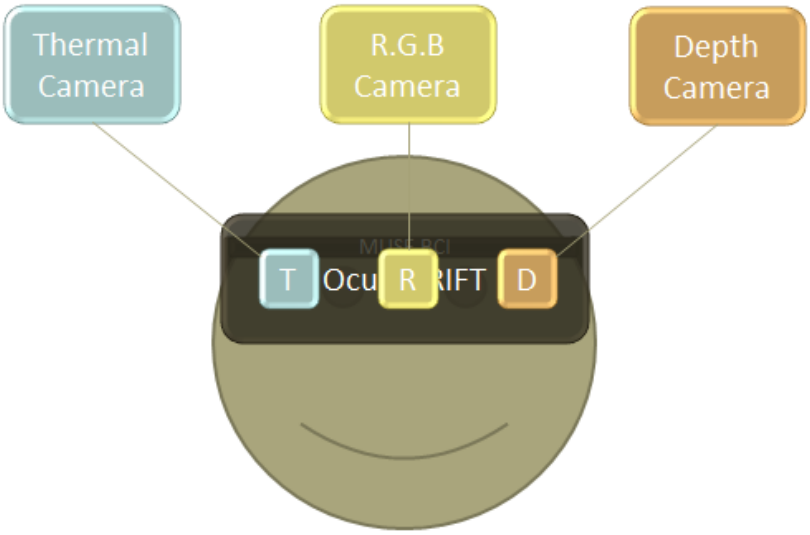

Figure 3.2: The user wears the Muse headband headset under the Prototype of the Oculus RIFT VR Device equipped with three external cameras.

Hence, if the user makes the following actions in Table 3.1, the views displayed to him on the Oculus RIFT screen should be changed accordingly:

Table 3.1: Proposed eye gestures and their corresponding required actions

Eye Gesture	Corresponding Executed Action
Wink Right	Switch to Thermal Camera (on the Right Hand Side)
Wink Left	Switch to Depth Camera (On his Left Hand Side)
Rapid Blink / Long Blink	Switch Back to RGB Camera (Normal View - Middle)
Squint	Zoom In and Zoom Out

3.2 HYPOTHESES

The proposed actions stated in Table 3.1 were initially viewed and tested on the MUSE Lab Software to find the possibility of getting significant features from each action. Referring to Ma *et al.* [21] who reported the use of a thresholding algorithm to extract the features of four different eye-gestures and classified them, as well as the initial views from the Muse Lab, the following hypotheses were introduced and subjected widely later to investigation with more participants to assure the generalization.

Since the features will not only be investigated on the signal shape level, but also on the channels level, thus Figure 3.3 on the next page defines the given labels for the dry electrode channels of the Muse headset to be easily identified prior the next illustrations. Channels 1 and 4 are attached to the head from the sides directly above the ear. Channels 2 and 3, as well as the reference (R), are attached to the forehead.

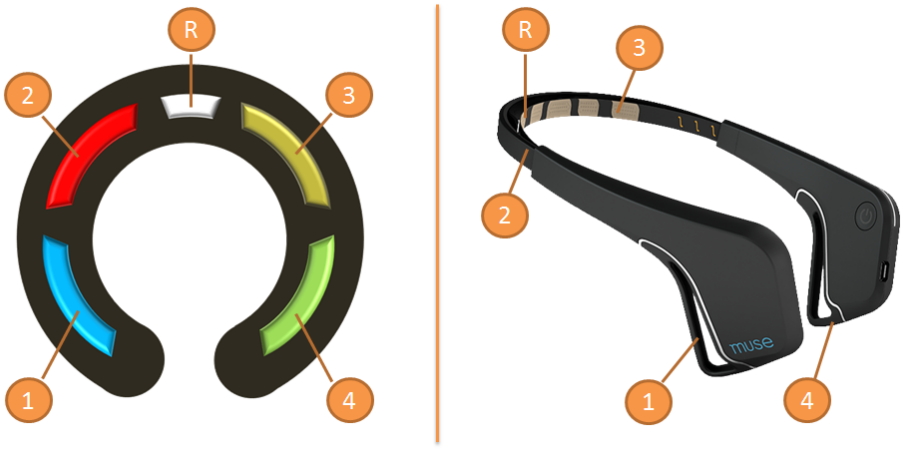

Figure 3.3: MUSE headband - Four Channels and Reference node

Figure 3.4 shows the Muse electrodes location on the scalp depending on the International 10-20 System for EEG tests. The channel electrodes TP9 and TP10 at the earpieces are adjustable, unlike the electrodes AF7 and AF8 on the front band which are fixed but they were designed on an average adult's head size.

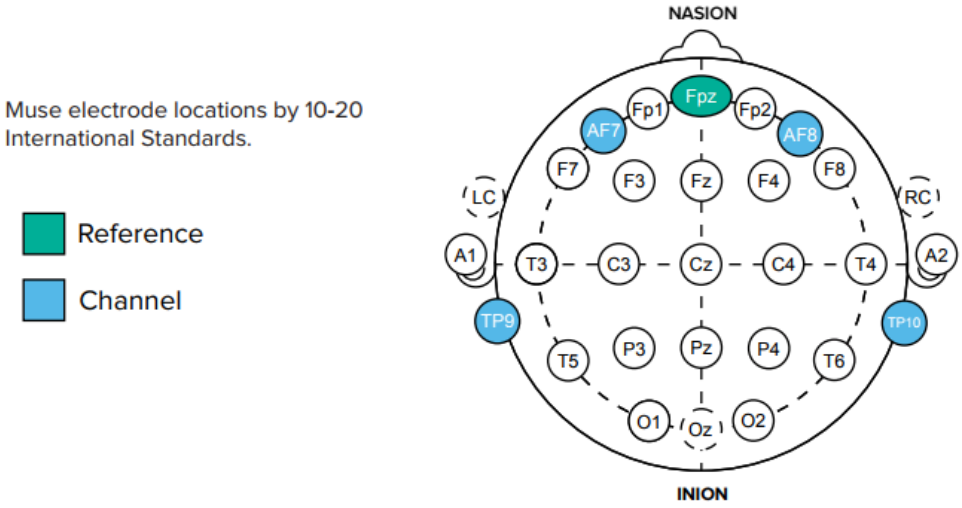

Figure 3.4: Muse Headband electrode locations

3.2.1 BLINKING

If the user is purely blinking, then positive (peak) and negative (valley) amplitudes will be clearly detected on the MUSE channels 1 and 4, as the Blink action was observed to be read from the mastoids (behind the earlobe). In addition, The forehead muscles do not contribute significantly to the blinking action, therefore Channels 2 and 3 have very weak detections. Figure 3.5 shows the Long period voluntary blinking which means the eye is closed for a while then released back again after a very short delay (less than or equal 1 second)

Long Blink:

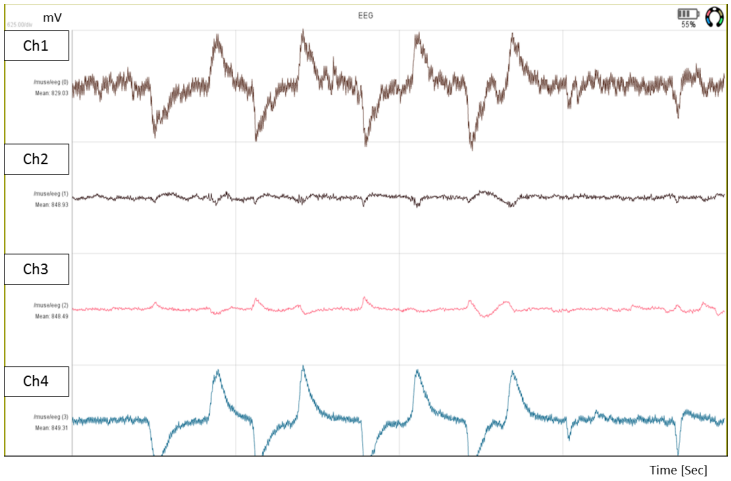

Figure 3.5: Voluntary Long Blinks recorded from the MUSE Lab

Figure 3.6 shows the voluntary rapid blinks which means the user is successively blinking very fast. The blink signals are also dominant on channels 1 and 4 only.

Rapid Blinking:

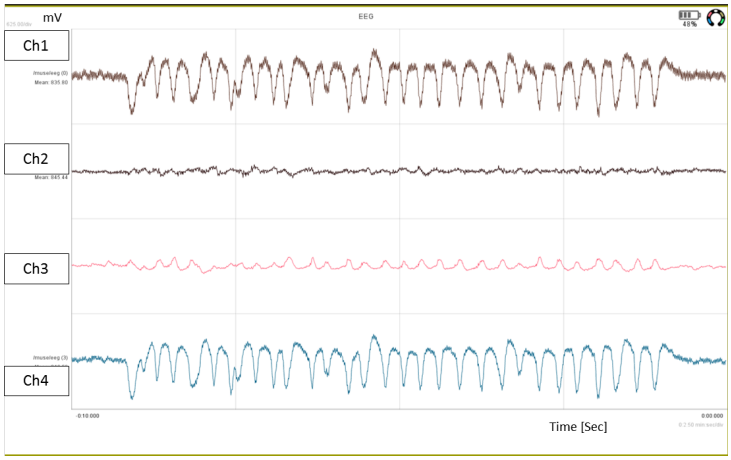

Figure 3.6: Voluntary Rapid Blinks recorded from the MUSE Lab

3.2.2 WINKING - Wink Left and Right

If the user is purely winking Right or left, then Channel 2 will read peaks and valleys in case of winking left as shown in Figure 3.7 and channel 3 will read the same in case of winking right as in Figure 3.8 as the muscles directly under Channels 2 or 3 were contracted.

Winking Left:

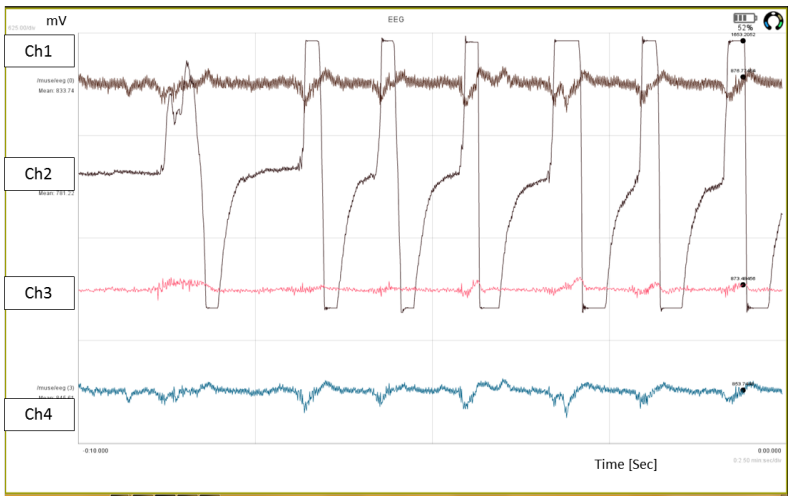

Figure 3.7: Wink Left - Channel (2) reads peaks and valleys - (MUSE Lab)

Winking Right:

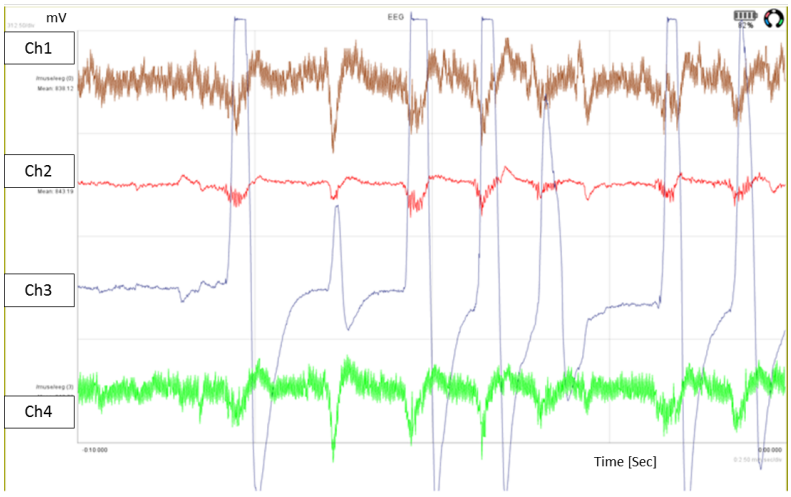

Figure 3.8: Wink Right - Channel (3) reads peaks and valleys - (MUSE Lab)

3.2.3 SQUINTING

In this application, Squinting is quite similar to frowning in which the forehead muscles become contracted and as stated before in Table 3.1, it should be used to Zoom In or Out as if the user is focusing on a target. Consequently Squinting is affecting Channels 2 and 3 together as shown in Figure 3.9

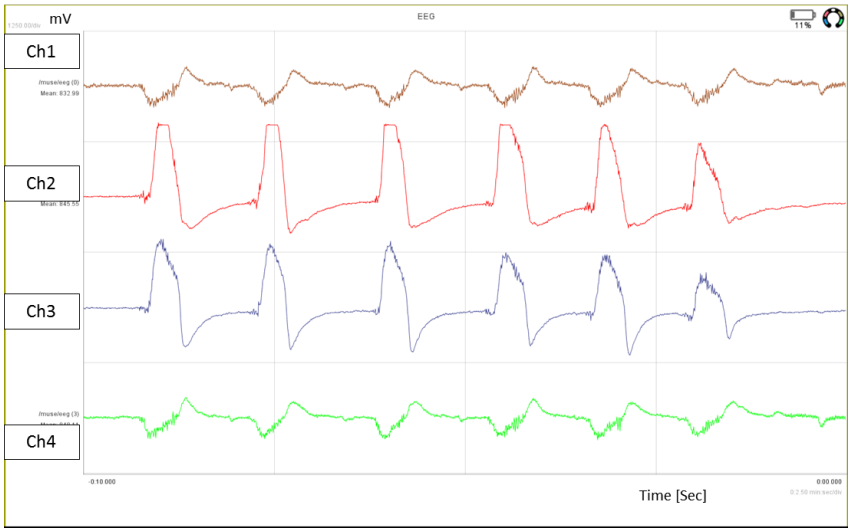

Figure 3.9: Squinting - Channels 2 and 3 are reading Amplitudes - (Muse Lab)

The challenge with Squinting seems to be higher as it relies on how good Channels 2 and 3 are, in contact with the forehead as it would read a Wink if any of the contacts is loose or much tighter than the other. Hence it is a coupled signal and it needs to be accurately classified to become distinguishable from the Winking effects. However, the key here is to correctly identify the amplitudes on channels 2 and 3 within a certain time interval as a Squint.

3.3 BLOCK DIAGRAM

This section provides an overview on the application flow diagram. Figure 3.10 demonstrates the whole process starting by the user who implements an eye gesture (i.e. Blink, Wink or Squint), then the Muse headband acquires the signals and send them as asynchronous packets to a PC or a Laptop via Bluetooth Communication. The Signal is processed and the features are extracted, then classified to a Predefined action as per Table 3.1. Consequently, the classified command should manipulate the displays on the Augmented Reality device with the integrated Cameras (Thermal, RGB and Depth cameras).

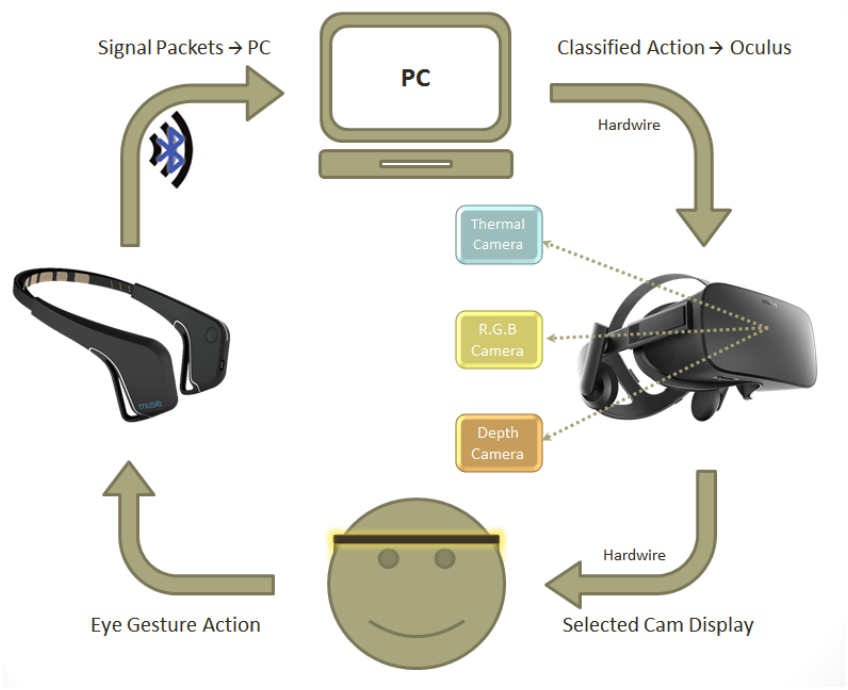

Figure 3.10: Overview on the Application Flow Diagram

As discussed in the previous section, since the proposed solution in this project is primarily focusing on the extraction of the eye gestures from the EMG signals of the MUSE electrodes, the implemented user studies and the data collected are basically analyzing the EMG signals acquired for each action per participant and classifying them. The MUSE headband sends several defined encoded OSC messages asynchronously in the form of packets to the PC via Bluetooth communication as in Figure 3.11 on the next page. These encoded OSC messages such as **/muse/eeg** or **/muse/acc** contain the channel values of the four EEG electrodes and the three axes of the accelerometer respectively. In addition, the Bluetooth wireless communication adds also to the advantages on which the MUSE was selected besides being simple and easy to wear and adjust, as it provides more flexibility and feasibility to the application.

The next section gives an overview on the Open Sound Control (OSC) Library and illustrates how the encoded messages are decoded to read the correct channel values.

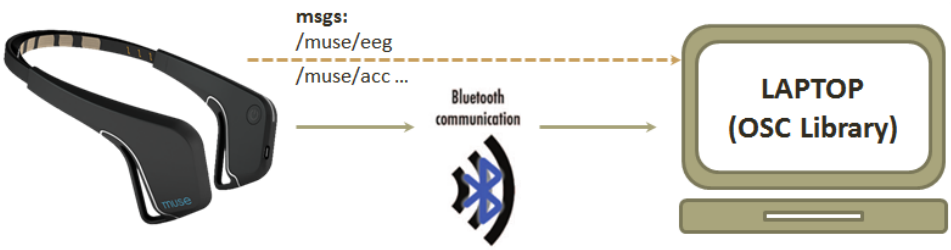

Figure 3.11: MUSE sends OSC messages with the channel values to a PC via Bluetooth

3.4 OPEN SOUND CONTROL (OSC) LIBRARY

The Open Sound Control (OSC) library is used by the MUSE headband to send the channel values and all the other status messages in the same form via bluetooth. In fact, the OSC library was originally established by Adrian Freed and Wright Wright [24] in 1997 as a network protocol to communicate between the computers and modern electronic musical instruments as well as other multimedia devices.

The OSC was designed as a 'Transport-dependent" protocol such that the data is transmitted in the format of a binary message and carried by the general-purpose and high speed network technologies such as the Ethernet. For instance, the OSC works with the UDP and the TCP/IP Internet protocols over Ethernet, wireless network and serial ports. In fact the use of the OSC had several advantages in terms of interoperability, flexibility and accuracy.

3.4.1 OSC Message Format

The transmission unit of the OSC is the OSC Packet, in which the OSC Client represents the application sending packets and the OSC Server is the one receiving them. Table 3.2 shows the different data types used in the OSC messages.

Table 3.2: OSC Messages Data Types

Data Type	Definition
int32 (i)	32-bit, big-endian, two's complement integer
float32 (f)	32-bit, big-endian, IEEE 754 floating point number
double (d)	64-bit, big-endian, IEEE 754 floating point number
blob (b)	int32 size count followed by that many bytes of binary data
string (s)	sequence of non-null ASCII characters, followed by a null character
boolean (T or F)	represented as either a T or F, no argument data encoded
null (N)	no bytes allocated in argument data
impulse (I)	event trigger, no bytes allocated in argument data
timetag (t)	64-bit, big-endian, fixed-point number in NTP Format
array	array of existing OSC-Arguments

As mentioned by [25], An OSC message comprises an OSC Address Pattern followed by an OSC Type Tag String followed by zero or more OSC Arguments. For instance, The OSC Address Pattern is an OSC-string beginning with the character '/' (forward slash). The OSC Type Tag String is defined as an OSC-string beginning with the character ',' (comma) followed by a sequence of characters matching exactly the sequence of OSC Arguments in the provided message. THe OSC Type Tag is considered as every character after the comma and it represents the type of the corresponding OSC Argument. The use of the starting comma for the OSC Type Tag Strings made it easier for the recipient of an OSC Message to know whether that OSC Message is lacking an OSC Type Tag String.

3.5 DATA COLLECTION CRITERIA

This section will describe the criteria planned to collect the useful data from the participants in the user studies, which will then be assessed to evaluate the efficiency of our approach.

For instance, the proposed eye gestures introduced in the first section were five actions including two types of Blinks: Long Blink and Rapid Blinking, in addition to two winking actions: Wink Right and Left, as well as Squinting. They need to be analyzed and evaluated for their feasibility and consistency among participants. Hence, several tests and trials were conducted to record these actions from random participants to check first the consistency of getting similar signals from each action per participant, so that each signal has independent clear features which can be easily relied on to design the classifier.

3.5.1 OFFLINE Data Collection

The Offline data collection was very important to record the signals at the beginning from the participants to find out the matched common features which could be relied on to build a robust classifier. It is proposed that we would have at least 15 random participants, since we are not targeting any groups, genders or ages.

The Offline data collection consists of 2 types of tests: Baseline and Core Tests

3.5.1.1 Baseline Tests

1. **Baseline Instructions test**: The participant should do no voluntary eye actions here, however he would be opening his eyes and reading a text with some instructions. His baseline signals will be recorded during this test for further calibration to the channels.

2. **Baseline Eye motion to extreme positions**: This test is only to know if the movement of the eyeballs would affect the signals or have any detected peaks or not. Of course, we are

seeking to have no stimulations from the eyeballs motion since this will lead to noise and signal perturbations. Hence, for this test the user would trace a cursor moving along the screen borders with his eyeballs only without moving his head.

3.5.1.2 Core Tests

The core tests include the following proposed actions: Rapid and Long Blinking, Winking Right and Left in addition to Squinting. The participant will be introduced to the following two subtest types for every action:

1. **Free Test**: The participant will start doing the instructed action repetitively after being asked to Start and will Stop when asked after a certain defined time period (around 8 seconds)

2. **Guided Test**: The participant will be asked to do the action in a relatively slow motion in which he will start the action (i.e. closing his eyes for Blinking), then he will wait or halt for a very small period (around 1 second) on that position and finally he releases the action (Open his eyes again). Each action is recorded for three times at least to enrich the collected data collected in case of any less efficiently or failed to do the action accurately trials.

The aim behind having the above two subtest types is to identify the best practice for each signal such as Blinking if it is preferable to be repetitively done (Rapid Blinking) as recorded in the "Free test" or to be slowly done (Long Blink) as in the "Guided Test" or both applies as voluntary distinctive actions. For the other actions, the free test and the guided tests would be relatively the same since the Winking and the Squinting actions are normally slower than Blinking. Thus we want only to analyze from the Free test how the repetitive signals would look like and if they will be interleaved. On the other side, the guided test guarantees that we would record also pure signals from the participants for further analysis and feature extractions.

In fact, the apparent features from the dummy tests on the MUSE Lab software as demonstrated in subsections 3.2.1, 3.2.2 and 3.2.3 show that the signals are apparently distinctive in terms of their shape, threshold levels, time interval and stimulated channels. Once the data are analyzed and the features for each type of action are extracted, we would pursue to the next step to make a feasible classifier to recognize the reliable eye actions which will be conceived in the next subsection.

3.5.2 ONLINE Data Collection, Analysis and Classification

This subsection will briefly provide an overview on the conceived classifier based on the preliminary signals recorded from the dummy trials in subsections 3.2.1, 3.2.2 and 3.2.3. As previously mentioned, the actions seem to have distinctive features by their apparent traits such

as the stimulated channels which simplifies a lot our classification and the other apparent features which need to be processed such as their shape, threshold levels and time period.

In terms of the stimulated channels, the Blinking actions appear solely on Channels 1 and 4, however the Squinting actions appear with high thresholds on Channels 2 and 3. If only 2 control signals are in our scope of interest, then the Blinking and Squinting can be inevitably easily classified with only simple thresholding on the Channels. The relatively challenging part appears by adding the Winking actions to the list of the desired control actions. However, they would still be quite distinguishable from the others as each wink appears on only one channel. As previously shown, the Wink Right appears with high thresholds on Channel 3 and the Wink Left appears on Channel 2 which make them having distinctive features on the channels level from the other actions especially the Blinking. In some cases, the eye gestures irregularities of some users are considered a challenge when trying to generalize the idea of classification based on the channels, since there would be some different responses when involving more muscles to do a certain action. For instance, if a user could not wink right normally only by stimulating his right forehead muscle and he involved the left forehead muscle as well to do it, then Channels 2 and 3 (left and right) would have significant detections which would be confused by the Squinting action.

The challenge is to select a suitable classification method for our application. As it seems from the dummy trials and as clarified above that some signals especially the Squinting and the Winking are not always having the same 100% exact signal responses due to the inconsistency of the contact positions and tightness of the electrodes, as well as the expected variations in the action implementation between the users. Nevertheless, the basic thresholding classifier with more advanced fine adjustments would be also investigated in our case, since the signals are still obviously distinguishable and as referred to Ma *et al.* [21] who applied the thresholding algorithm to detect Blinks, Winks, Frowns (Squints) and Gaze actions.

Furthermore, after the classifier implementation and running the online study, the proposed accuracy metric in this case is the confusion matrix to know the following:

1) How many performed actions were truly recognized and classified? (True positives)

2) How many non performed actions were truly not detected? (True negatives)

3) How many non performed actions were wrongly classified? (False Positives)

4) How many performed actions were not recognized? (False Negatives)

Additionally, it is inevitably crucial to ask the participants, at the beginning in the questionnaire form, about the actions that they can do and the actions they cannot naturally do or face any difficulty in doing it. They will check mark if they can Blink, Wink Right, Wink Left and Squint, so that the naturally difficult eye actions for any of them would be excluded from the analysis to

avoid any irrelevant results. It is preferable to have at least 15 participants as well for the online classification test to get more accurate results. In addition, the Latin Square approach [26] will be used to shuffle and randomize the order of the instructed actions for each participant to avoid any learning curves and to test the classifier regardless of whatever the order of actions is.

3.6 MAIN CONTRIBUTIONS

In this work, several contributions were proposed to investigate their performance in this study and for further studies as well.

3.6.1 DYNAMIC Sliding Window Algorithm with Segmented Packets

For instance, this work introduces another method to segment the data prior to the feature extraction and classification. It is also relying on the sliding window concept to check a group of data together, however the window size is dynamic. In studies such as Ma *et al.* [21] and Nakanishi *et al.* [27], fixed window lengths were used to extract the eye gestures. Even Chambayil *et al.* [17] fixated a relatively huge time interval (around 5s) for the user to blink within and to count the number of blinks after 20s to decide on a selection in one out of three stages so as to finally decide on a selected character in the virtual keyboard application.

In other different studies such as Noor *et al.* [28], the dynamic sliding window method was introduced as a new approach to recognize the user's physical activity by reading single tri-axial accelerometers. Their technique proposed that the window size starts by an initial value, then it is automatically expanded by half its original size when the activity signal length is longer than the boundaries of the initial window till a maximum size limit stated by three expansions.

In our approach, the idea would be close to the implementation of Noor *et al.* [28], but in this work, the dynamic sliding window with Packets is proposed to instead capture the eye gestures from the users so as to handle the duration variations of the gesture implementation between them. The data are acquired from the MUSE headband point by point per channel in the "Data Acquisition" loop, then stacked together in a "Packets Generation Loop" which segments the data till reaching the defined packet size. Then the Packets are sent and queued to the last loop which makes the Packets Management to decide currently how many packets to process, then extracts the features and finally makes the classification. The decision on the number of packets to be processed relies on whether a feature out of the defined features for an action gesture was detected within this packet or not. If yes, then the feature extraction function makes a feedback to the Packets Management function to increment the number of packets to be processed in the next loop. In other words, the next loop will have data points equal to the Packets Size multiplied by the number of Packets.

3.6.2 EMG-BASED Blinks Detection behind the Earlobe

Various studies such as Ma *et al.* [21] and Skotte *et al.* [29] relied on the detection of the blinks from the Electrooculography (EOG) signals by placing the electrodes above, below or on the sides of the eyes to detect the other gestures as well such as the winks such as Valeriani *et al.* [18]. Other studies relied on the image processing of the eye movements to deduce the corresponding actions such as the Long and Short Blinks such as Królak *et al.* [30] and Morris *et al.* [31] which use face and eye tracking algorithms to capture the eye blinks.

Other studies such as Yang *et al.* [32] used a similar commercial bluetooth headset from Neuro Sky, which is originally a Brain Computer Interface as well, to detect the EMG signals from the eye blinks. The Neurosky headset includes four sensors where three of them were attached on the Left earlobe itself and the remaining is fixed on the forehead on the left side as shown in figure 3.12.

Figure 3.12: Neurosky headset - Three Sensors on the left ear and the fourth on the forehead leftside Yang *et al.* [32]

In our approach, the detection of the blinks is proposed to be captured from the region behind the earlobe using the MUSE headband, which includes two rubber sensors at the right and the left ear similarly in that region, as shown previously in Figure 3.1 on page 33.

CHAPTER 4

EXPERIMENTAL METHODS

This chapter demonstrates the data collection criteria, devices configurations, System setup and the testing environments for the implemented user studies. It includes a full detailed description on how the user study test was established and accomplished. Additionally it demonstrates the unqiue features added to the implemented testing tool to ensure the accuracy, the unification among all the participants and the randomness of the running test order to avoid any expected learning curves. LabVIEW is the programming environment used to build these User Study tools and it showed high efficiencies in the acquisition of the data, its analysis and visualization.

4.1 SYSTEM SETUP

Recalling from the previous section, the core device used in the proposed system to detect and recognize the eye gestures is the Muse headband shown in Figure 4.1 which is a non-invasive sensing device that captures the electroencephalography (EEG) as well as the Electromyography (EMG) signals. It includes a 3-axis accelerometer with 10 bits resolution as well as enabling the detection of motion for further classifications.

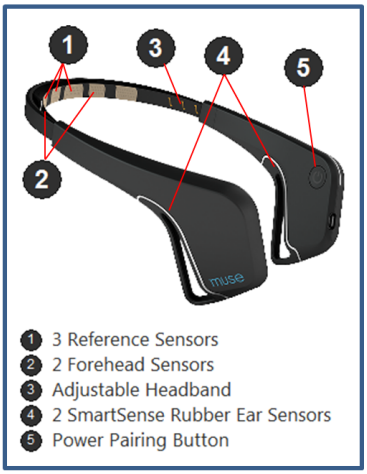

Figure 4.1: MUSE headband with 4 Channels

Recalling that the Muse headband is a dry electrode based Brain Computer interface having three Reference Sensors and four channel electrodes TP9, AF7, AF8, and TP10. The two Forehead sensors are located at AF8 and AF10 as well as the other two TP9 and TP10 are located on the sides directly above the ears. Data are sent via Wireless Bluetooth communication. In this project, the MUSE is used to detect and classify the EMG-based eye gestures by exploiting its four dry electrodes positioned behind the ears and on the forehead.

The System Setup was further demonstrated under the Block Diagram Section 3.3 on page 39, Chapter 3 on page 32. Figure 4.2 shows an overview on the main procedures of the user study which will be discussed in the following sections.

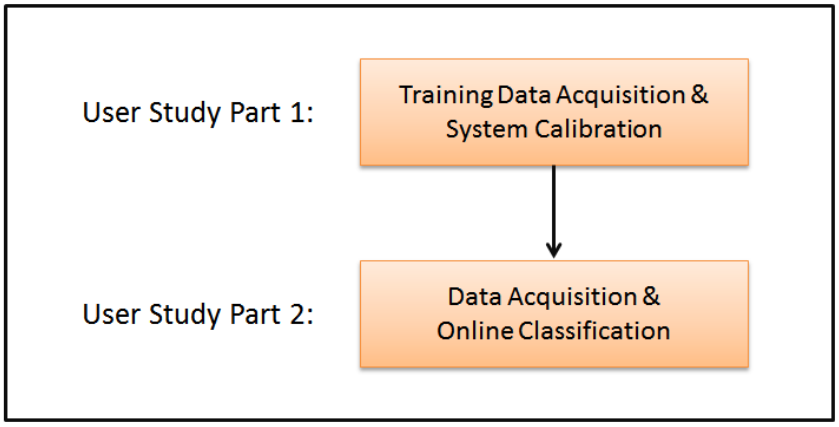

Figure 4.2: Overview on the main procedures of the User Study

4.2 PART 1: TRAINING DATA ACQUISITION AND CALIBRATION

This section demonstrates the training data collection criteria including the baseline tests which were used for offsetting and checking the influence of the eye movements on the signal. Then, the calibration of the system and its implementation will be discussed afterwards. Figure 4.3 shows the hierarchy of the topics of this section.

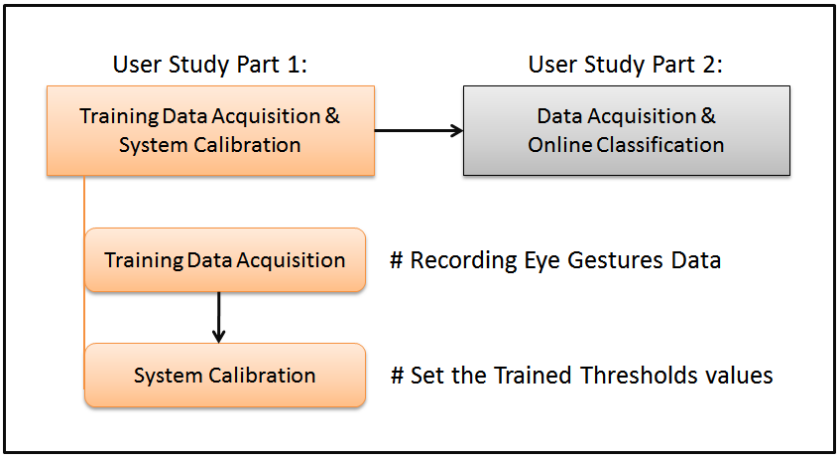

Figure 4.3: Training Data Acquisition and System Calibration section

One of the main challenges taken into consideration was the ability to conduct the user study on all the participants under the same circumstances. The external conditions were already met, but the challenging part was how we would be able to run the test itself especially that we not only need to ensure the randomness of the sequential order of the commanded instructions,

but also to provide all the participants with the same time and conditions to implement every action as well. The randomness of the commands order was essential to claim that our proposed solution has no learning curve relying on the order of actions. Hence, this led us to implement the training application (programmed in LabVIEW) with editable Voice Narrated commands which can be generically applied for further researches as well and to categorize each type of action as shown in Figure 4.4 and further discussed.

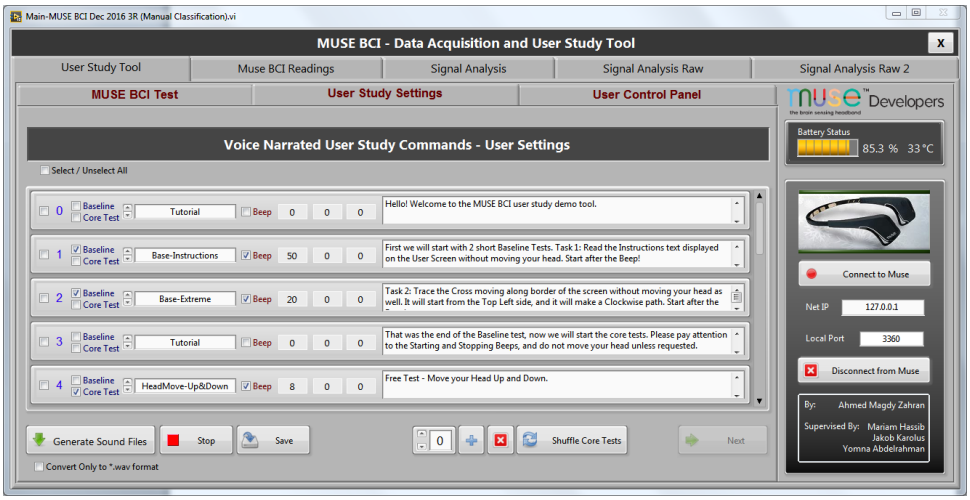

Figure 4.4: Training Data Collection with Voice Narrated Commands - LabVIEW

4.2.1 TRAINING Data Collection

Recalling the proposed methods in the Data Collection criteria in section 3.5, Chapter 3 to detect and classify the eye gestures, the test study commands were categorized to Baseline and Core Tests.

The Baseline tests include the following two commands:

1. **Baseline Instructions test**: The user test study will start with this part in which participant will see a black screen with some text to read as shown in figure Figure 4.5 on the next page. He will only read the text without doing any intended (voluntary) eye actions without moving his head. These Baseline signals will be recorded during this test for further calibration to the channels.

2. **Baseline Eye motion to extreme positions**: This test is conducted after reading the instruction text in the first part, to check the effect of the eyeballs movement on the captured signals (if any) and to filter any noises out to avoid false stimulations. Hence, in this test the user would trace a small white fixation cross (+) moving along the screen borders in a rectangular path with his eyeballs only without moving his head as shown in figure Figure 4.6 on the following page. It will make only one cycle and record the signals during this time for further analysis.

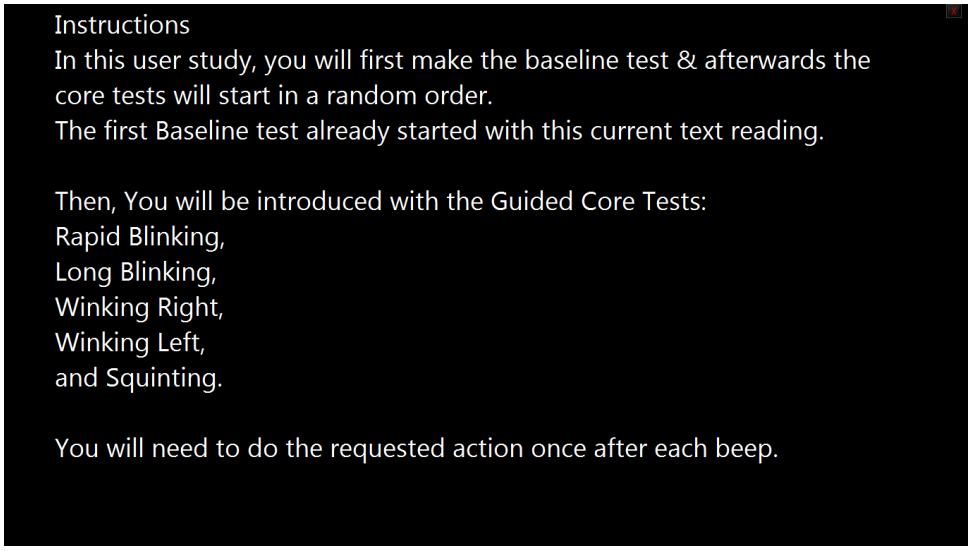

Figure 4.5: Baseline Instructions window - No intended Eye actions.

Figure 4.6: Baseline eye motion test to extreme positions

The Core tests include the following commands:
1. Guided-Blink : Trained by Long Blink to detect the highest and lowest amplitudes.
2. Guided-WinkRight
3. Guided-WinkLeft
4. Guided-Squint

As previously mentioned in the Data criteria section 3.5, Chapter 3, the core tests include all the required eye gesture actions to be recorded from the participants. Accordingly, to standardize the core tests among all and during the same test, a black screen with a fixation cross as shown in Figure 4.7 on the next page is displayed to the participants to avoid any external visual disturbances which would make any signal perturbations.

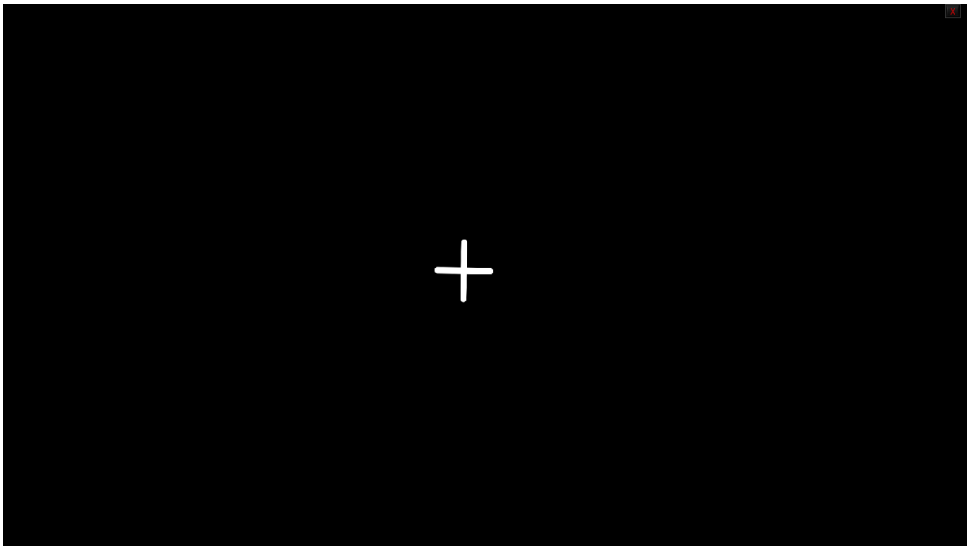

Figure 4.7: Core tests screen display

The participants were instructed by the voice narration as well to do every action for multiple times. They were instructed first to do a certain action, but they performed it once after hearing each beep. They were introduced with three beeps for every action, so as to record and save at least three performances of each eye gesture signal. The advantage behind conducting the training session this way is not only to standardize the test among all the participants, but also to save the timestamps of the beep occurrences with the action name, then match those periods with the recorded channels data to be split into actions data as shown in Figure 4.8 on the following page for further processing.

The following Table 4.1 demonstrates a real sample data from the recorded instructions whether it was a new action "Start" or an action end "Stop" as well as the Beep orders and their timestamps:

4.2.2 SYSTEM Calibration

Although more training data would provide more accuracy, however it was preferred to do the calibration with three actions per gesture so as not to affect the online test results due to fatigue since the participant starts the online test afterwards. The training session duration is around 3 minutes for each participant before running the online classification test.

After the Training data acquisition and saving the data for a participant, the full data were reloaded and subdivided according to the saved actions, then the required thresholds for each action were calculated and automatically set as shown in Figure 4.11 before conducting the online test in which the user will be doing the instructed actions and the classifier will execute and display the classification results in real time. The trained thresholds were calculated by averaging the peaks and valleys of the recorded trained signals per gesture. Channels 1 and

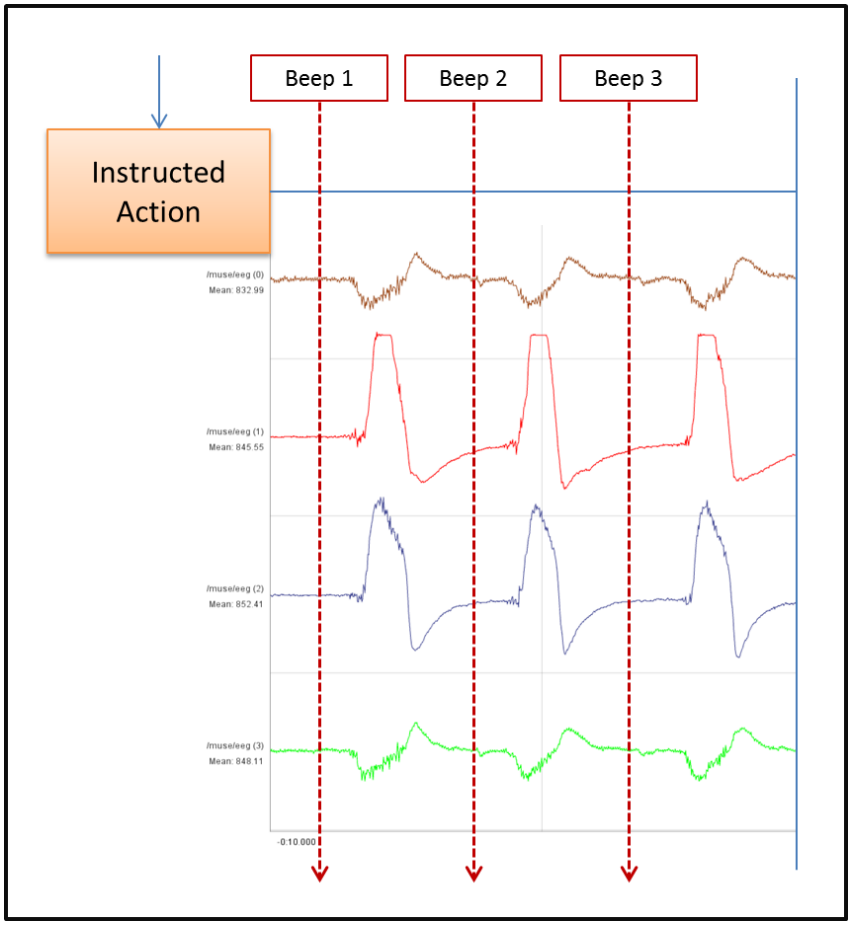

Figure 4.8: Calibration: Single Action is performed once after each beep

4 were set according to the average of Blinks. On the other side, channels 2 and 3 were set according to the maximum value of all the averaged peaks and valleys of the squint and winks trained data. This approach was subjected to investigation, however it could be altered for better classifications after the analysis of the experiment observations.

As previously mentioned, the instructions were saved with their timestamps in a csv file independent from the Technical Data Management System (TDMS) measurements file acquired during the whole session. Hence, both files are first loaded as shown in Figure 4.9 on page 54. The TDMS file format stands for Technical Data Management System and it was developed by National Instruments to have flexible and organized file storage, self-scaling hybrid data index, and an interactive post-processing environment [33].

Then the marked instructions were matched with the measurements data and the performed actions data were captured and clustered with the marked timestamps and indexes to deal with each action independently as shown in Figure 4.10 on page 54.

After the preparation of the actions data cluster and relying on our trials and assumptions, the Channel thresholds were set accordingly. By observation, the Blinks were the only actions appearing on Channel 1 and 4 with high significance and since there are 2 Blinking actions in

Table 4.1: Sample Recorded instructions: Start, Stop & the Beep with their timestamps

Test Instruction	Timestamp Occurrence
Base-Instructions_C-01/07_Start-00/01	5:13:20.326 PM 4/1/2017
Base-Instructions_C-01/07_Stop-01/01	5:13:30.326 PM 4/1/2017
Guided-WinkLeft_C-03/07_Start-00/04	5:13:43.000 PM 4/1/2017
Guided-WinkLeft_C-03/07_Beeps-01/04	5:13:44.015 PM 4/1/2017
Guided-WinkLeft_C-03/07_Beeps-02/04	5:13:48.413 PM 4/1/2017
Guided-WinkLeft_C-03/07_Beeps-03/04	5:13:52.806 PM 4/1/2017
Guided-WinkLeft_C-03/07_Stop-04/04	5:13:57.107 PM 4/1/2017
Guided-Squint_C-04/07_Start-00/04	5:14:00.548 PM 4/1/2017
Guided-Squint_C-04/07_Beeps-01/04	5:14:01.550 PM 4/1/2017
Guided-Squint_C-04/07_Beeps-02/04	5:14:05.944 PM 4/1/2017
Guided-Squint_C-04/07_Beeps-03/04	5:14:10.320 PM 4/1/2017
Guided-Squint_C-04/07_Stop-04/04	5:14:14.621 PM 4/1/2017
Guided-Blink_C-05/07_Start-00/04	5:14:18.884 PM 4/1/2017
Guided-Blink_C-05/07_Beeps-01/04	5:14:19.885 PM 4/1/2017
Guided-Blink_C-05/07_Beeps-02/04	5:14:24.191 PM 4/1/2017
Guided-Blink_C-05/07_Beeps-03/04	5:14:28.582 PM 4/1/2017
Guided-Blink_C-05/07_Stop-04/04	5:14:33.006 PM 4/1/2017
Guided-WinkRight_C-06/07_Start-00/04	5:14:40.360 PM 4/1/2017
Guided-WinkRight_C-06/07_Beeps-01/04	5:14:41.362 PM 4/1/2017
Guided-WinkRight_C-06/07_Beeps-02/04	5:14:45.664 PM 4/1/2017
Guided-WinkRight_C-06/07_Beeps-03/04	5:14:50.090 PM 4/1/2017
Guided-WinkRight_C-06/07_Stop-04/04	5:14:54.391 PM 4/1/2017

this project comprising the Long Blink with a Negative (valley) followed by a positive (peak) amplitude and the Rapid Blinking action with more than two successive negative amplitudes but obviously smaller than the Long Blink amplitudes, thus Channels 1 and 4 were set with the average value of all the peaks and valleys of the trained Blinking data. On the other side, the maximum value of the average values among the Winking Right and Left as well as the Squinting actions was set for the Peak Thresholds and the Valley Thresholds for Channels 2 and 3.

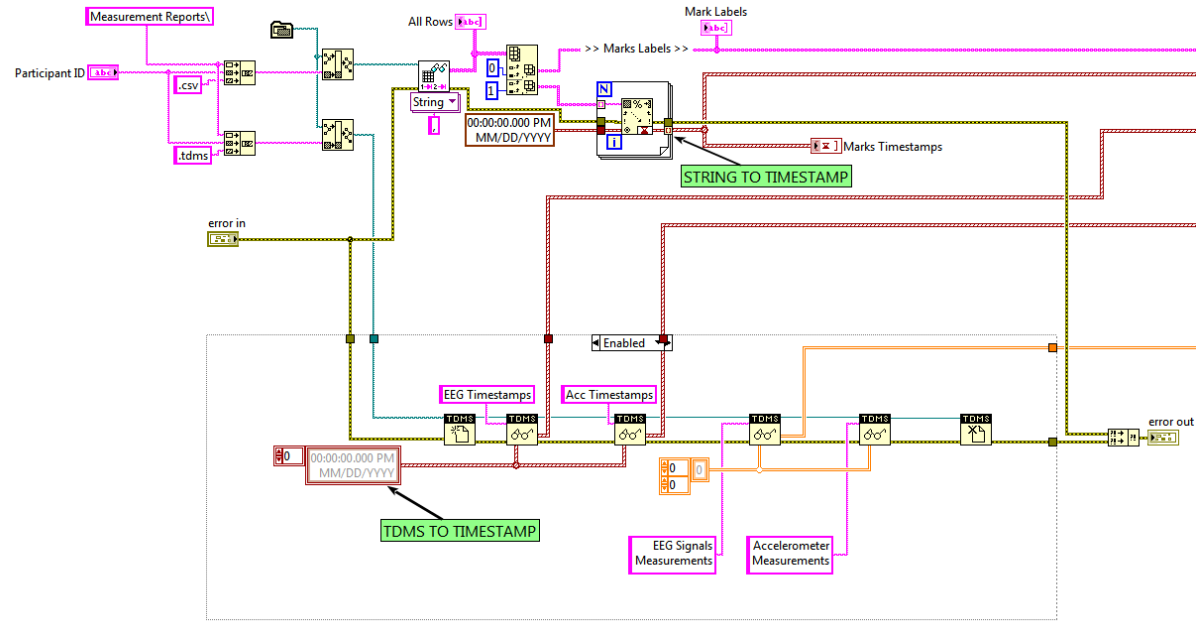

Figure 4.9: Code: Load Instructions and Measurements files and Read their data

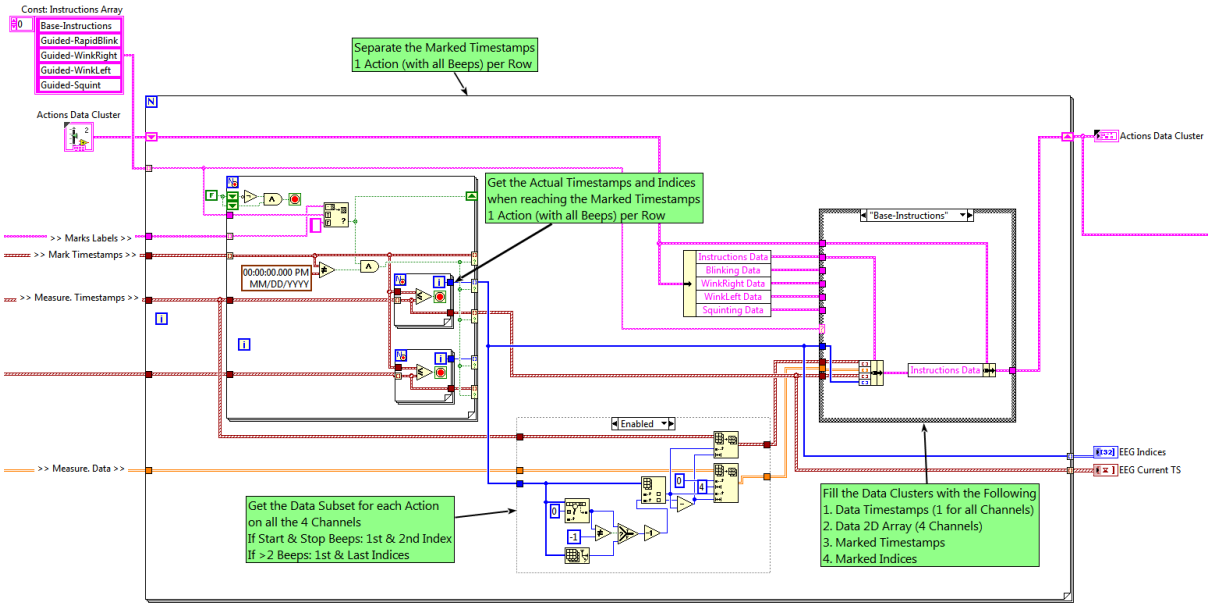

Figure 4.10: Code: Matching the Instructions Marks with the Measurements Data

4.3 PART 2: DATA ACQUISITION AND ONLINE CLASSIFICATION

In this study, 15 Participants (P01 to P15) consisting of 12 Males and 3 Females were involved in the experiment without being targeted on purpose from any groups. However, they are mostly students as the study was conducted at the university campus. After the collection and saving of the Training data from each participant, the thresholds were automatically adjusted before running the online test as illustrated previously. They were simply asked to perform the actions

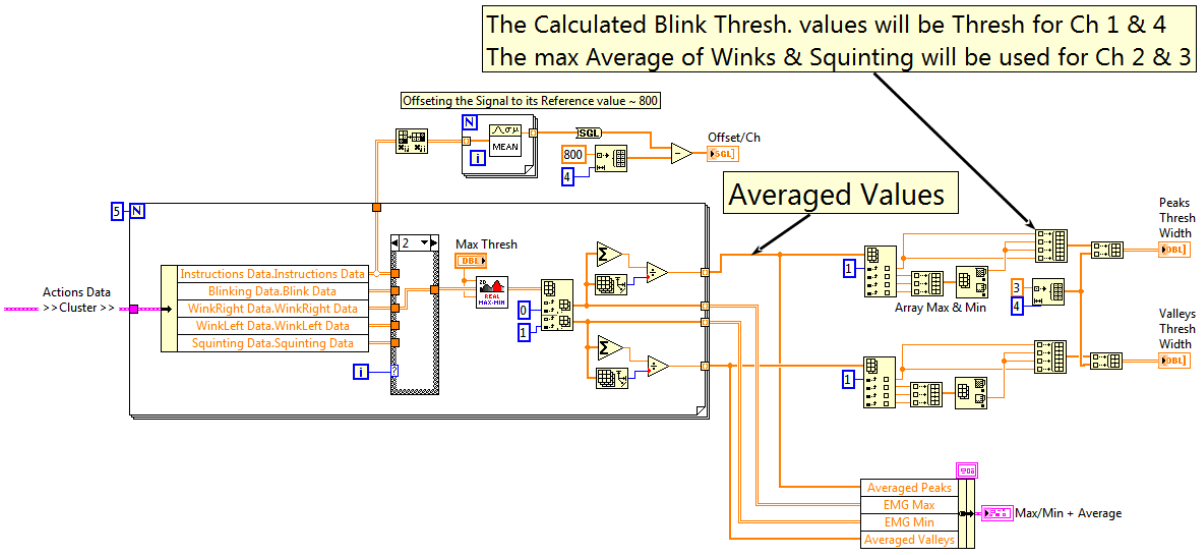

Figure 4.11: Code: Setting Thresholds for all the Channels from the trained data

as per the session moderator instruction and the classifier output was displayed and saved as well as the data acquired from each channel for the whole session.

4.3.1 TEST preparation and Conditions

The participants were asked before running the test about the actions which they could not perform so as to exclude them from our test and analysis. Hence, 4 out of the 15 participants (P03, P05, P08 and P10) reported, in the "Participation sheet", their inability to perform the wink right and 1 participant (P15) could not wink left. However, some participants did not report any inability at the beginning, but it was found that they could not perform the actions in the normal right way and this will be further discussed in the Discussion section, in the next chapter.

The participants were sitting infront of the Black screen and concentrating on the centered fixation cross while doing the test as shown in Figure 4.12. Additionally, after their consent, their face was Video taped during the session so as to check their eye gestures performance later while analyzing the data.

Furthermore, every participant was introduced with 15 shuffled eye actions events during the online test that were instructed one by one by the experiment moderator. Each eye action was repeated three non consecutive-times during the session so as to have more precision. The whole session was video recorded and the data were saved in another measurements file (different from the Training data measurement file) and finally, the classifier results were saved in an CSV format file in case of executing any action, regardless of the participant's performance to get all the results and identify the true and false detected actions. The 15 events were shuffled from

Figure 4.12: System Setup: Laptop is running the test and recording the videos and above is the black screen with the fixation cross

one participant to another according to the Latin Square approach as shown in Figure 4.13 on page 57.

4.3.2 DATA Acquisition and Parsing OSC Values

The test starts by applying and testing the trained thresholds. Also a Manual offseting feature was added in case of observing the signal fluctuating higher or lower than the Reference mean point which is around 800 for the MUSE EEG Channels.

Recalling from the Block Diagram Section 3.3 on page 39, Chapter 3 on page 32, the MUSE headband sends OSC messages with the values of the channels under a message address such as (/muse/eeg) over Bluetooth communication. The "NI OSC Library for LabVIEW by National Instruments" [34] was used to decode and translate the OSC messages sent by the MUSE to be capable of reading the values in SINGLE FLOAT32 format as shown in Figure 4.14 on the following page.

The communication to the MUSE headband can be established via UDP or TCP protocols. For instance, the UDP was used in our application for simplicity. In LabVIEW the communication is first established with the Localhost IP and any available Port through the "UDP Open" function, then "UDP Read" function outputs the data received in String format and the Maximum size or

MUSE BCI Eye-based Facial Gestures - User Study 2 - RANDOMIZED ACTIONS (Latin Square 15 x 15)

Participants	Actions															DONE
P01	Long_B	W_Right	Squint	Long_B	Rapid_B	W_Left	Squint	Long_B	W_Right	Rapid_B	W_Right	Rapid_B	W_Left	W_Left	Squint	
P02	Rapid_B	Long_B	W_Left	Squint	Rapid_B	Squint	W_Right	Long_B	Long_B	W_Right	Squint	W_Left	Rapid_B	W_Left	W_Right	
P03	Long_B	Long_B	W_Left	W_Right	Squint	Long_B	W_Left	Squint	W_Right	W_Left	Rapid_B	Squint	Rapid_B	W_Right	Rapid_B	
P04	W_Left	W_Right	Rapid_B	Squint	W_Right	Long_B	W_Left	W_Left	Rapid_B	Squint	Long_B	Rapid_B	W_Right	Long_B	Squint	
P05	W_Right	W_Right	Squint	Rapid_B	Rapid_B	Squint	Rapid_B	W_Left	Long_B	Long_B	W_Left	W_Right	Long_B	Squint	W_Left	
P06	W_Left	W_Left	Long_B	W_Right	Squint	Squint	W_Right	W_Right	Rapid_B	Long_B	Rapid_B	Squint	Long_B	Rapid_B	W_Left	
P07	W_Right	Rapid_B	Rapid_B	Long_B	W_Left	Rapid_B	Long_B	W_Right	W_Left	Squint	Squint	W_Right	Squint	W_Left	Long_B	
P08	Long_B	Long_B	W_Left	W_Right	W_Right	Rapid_B	Squint	Rapid_B	Squint	W_Left	Rapid_B	W_Left	Squint	Long_B	W_Right	
P09	Squint	W_Left	Squint	W_Left	Long_B	W_Right	W_Right	Long_B	W_Left	Rapid_B	Long_B	Rapid_B	Rapid_B	W_Right	Squint	
P10	W_Right	Squint	Long_B	W_Left	Squint	W_Right	W_Left	Rapid_B	Squint	W_Left	Long_B	W_Right	Long_B	Rapid_B	Rapid_B	
P11	W_Left	Squint	W_Right	Long_B	Long_B	W_Left	Squint	Rapid_B	Rapid_B	W_Right	W_Right	Long_B	W_Left	Squint	Rapid_B	
P12	Rapid_B	Squint	W_Right	Squint	W_Left	Rapid_B	Long_B	Squint	W_Right	Long_B	W_Left	Long_B	W_Right	Rapid_B	W_Left	
P13	Rapid_B	W_Left	Rapid_B	W_Left	Long_B	W_Right	Long_B	W_Right	Squint	Rapid_B	W_Right	W_Left	Squint	Squint	Long_B	
P14	Squint	Rapid_B	W_Right	Rapid_B	W_Left	Long_B	Rapid_B	W_Left	Long_B	Squint	W_Left	Squint	W_Right	Long_B	W_Right	
P15	Squint	Rapid_B	Long_B	Rapid_B	W_Right	W_Left	Rapid_B	Squint	W_Left	W_Right	Squint	Long_B	W_Left	W_Right	Long_B	

Figure 4.13: Participants shuffled actions summary according to the Latin Square approach

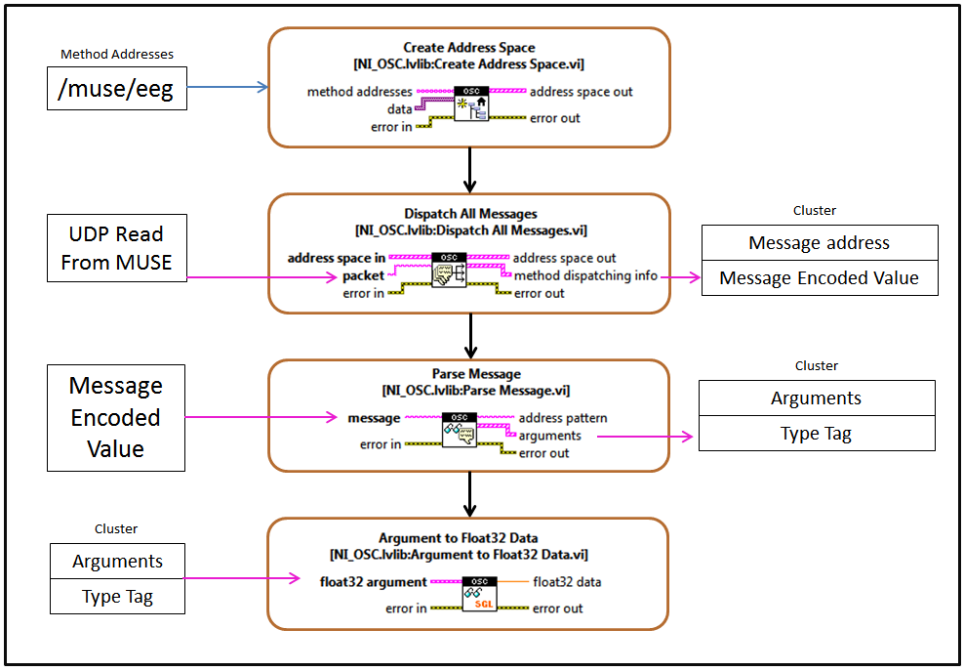

Figure 4.14: NI OSC functions parse the message values sent from the Muse

number of bytes to read must be defined as well. The Datagram size of the MUSE headband is around 1536 bytes. Finally, after the test, the UDP communication must be closed using the "UDP Close" function. Figure 4.15 on page 58 demonstrates the block diagram of a demo example depicting the UDP communication with the MUSE headband.

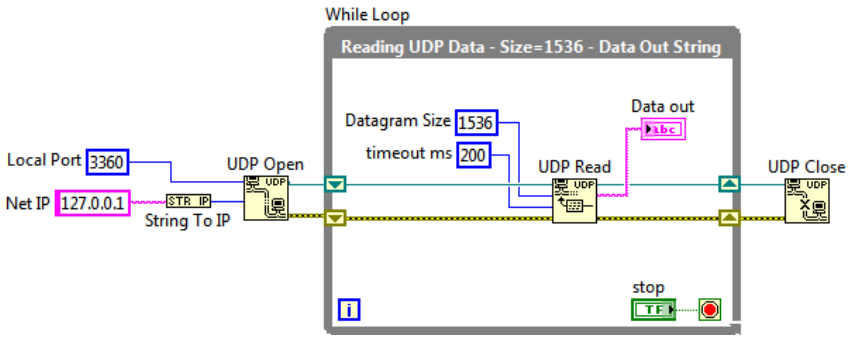

Figure 4.15: MUSE headband UDP Communication and Data Read in LabVIEW(Demo Example)

4.3.3 FEATURE Extraction

After the data acquisition and receiving the floating point values per channel, the data should be saved and analyzed by extracting the features defined for every eye gesture and executing a classification decision. Recalling the signal shapes of the proposed eye gestures from Chapter 3 on page 32, the features of every action were defined and tested.

The features were defined in terms of the following parameters:
1. Channels: The action appeared significantly on which channel?
2. Amplitudes: Peaks (positive amplitudes) and Valleys (negative amplitudes)
3. Shape: defining a signal by its shape implies to know how it starts and how it ends
4. Time Interval: if a starting feature was detected, it should have a timeout

Recalling Ma *et al.*[21], the thresholding algorithm was utilized to detect similar eye gestures including the blinks, winks and squints. It was also stated that the thresholding provided significantly better recall than the pattern matching for the blinks and winks. Hence, this work aimed to investigate the thresholding since the signals were clear and distinguishable. In addition, more advanced machine learning techniques need more time and participants to train the classifier first.

4.3.3.1 Feature 1: Channels

As described before, the Blinks are mostly appearing with very high significance on Channel 1 and 4 as shown in Figure 4.16 on the next page and they are relatively the same or very close as both channels sense the EEG or EMG stimulations from the back of the ear. By observation, the other actions (Winks and Squints) do not have high influence on channel 1 and 4 compared to the other channels 2 and 3 in which the Squinting affects both 2 and 3 since the action is stimulated from the forehead and the Winking Left and Right affect Channel 2 and 3 respectively as shown in Figure 4.17 on page 60

4.3.3.2 Feature 2: Amplitudes

The amplitudes of the Blinks were observed generally among the participants to be above the mean point 900 on the Peaks level and below 750 on the Valleys level, however the Rapid Blink amplitudes were smaller than the Long Blink as they were faster and this compromises on the signal strength or power. However the Squints and Winks on channels 2 and 3 are having relatively higher amplitudes that were generally above 1300 and below 300. These threshold values are all estimates, since every user is having a calibration session prior to the test to calculate his training based thresholds and also they could be manually adjusted by observation. Therefore they are User-dependent values.

4.3.3.3 Feature 3: Shape

As shown in Figure 4.16 on the next page, the Long Blink action was observed to start first with a valley (-ve amplitude) then makes a peak (+ve amplitude) afterwards. However, the Rapid Blinks are not always having high peaks, but they would cross the valley threshold in a repetitive action and their frequency are higher than the Long Blinks. On the contrary, the Squints and Winks start with a high peak first then a valley as shown in Figure 4.17 on the following page.

4.3.3.4 Feature 4: Time Interval

The Long Blink duration is relatively around 1 sec and the Rapid Blinks are faster (nearly around 5 times per second). However, the Squints and the Winks could have longer duration since they are implemented by stimulating the forehead muscles and the user would halt on the action for moments then release back. This was also considered when building the classifier to start detecting an action from the first feature (i.e Wink Right: Peak on Channel 3), then it waits till either the second feature is detected (i.e Wink Right: Valley on Channel 3) to make a Wink Right Classification or that the duration timed out to reset the Classifier and do nothing.

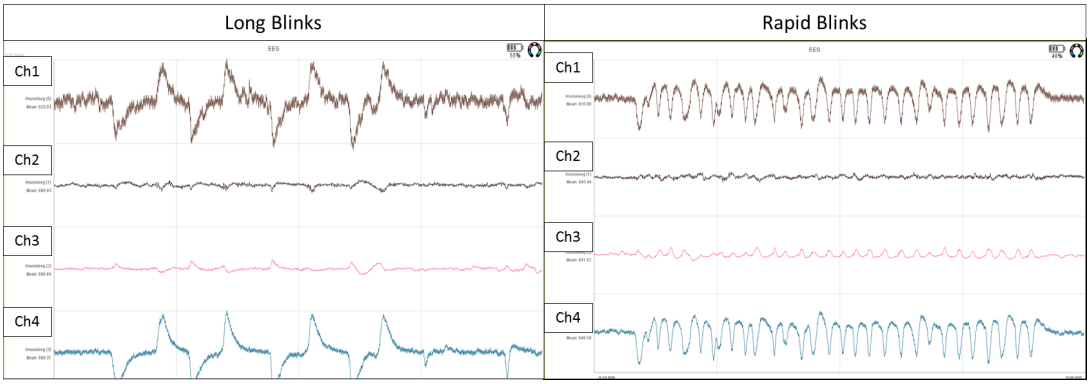

Figure 4.16: Long and Rapid Blinks features

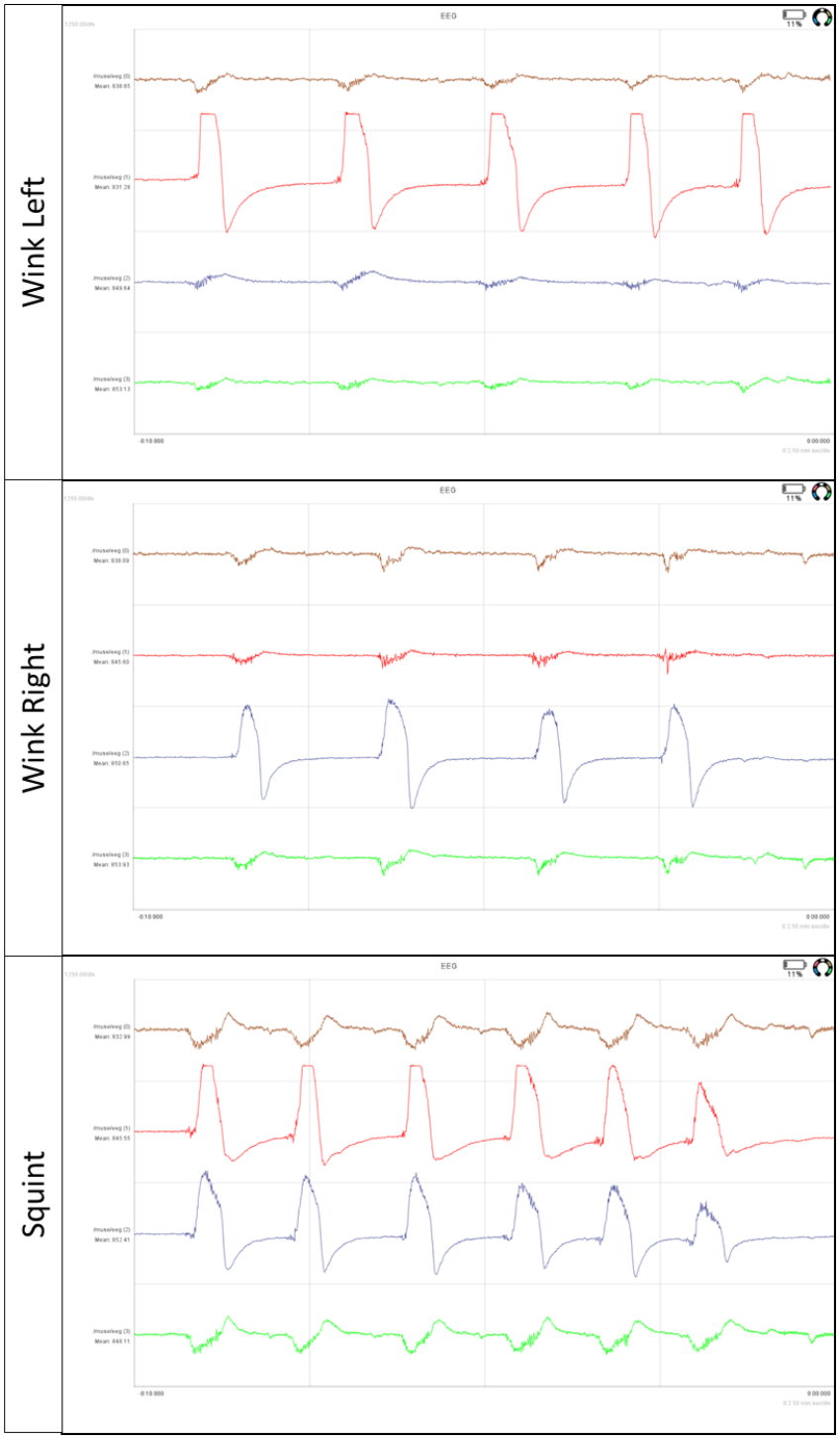

Figure 4.17: Winks and Squints have high amplitudes on Channels 2 and 3

4.3.4 DATA Management

In fact, the data need to be stored for a time period or specific number of samples to have an array of values for every channel to be capable of analyzing all these values together. However, it was not efficient and it was totally redundant to analyze the data Point by Point every time new points were received or to shift a sliding window point by point (as in a queue to add new data point and remove the oldest one) as this caused the system to be very slow and laggy. In addition there was a huge delay between the current performed actions and their classifications, thus it was totally unpractical. This was resolved by implementing a dynamic data packeting method.

4.3.4.1 Online Dynamic Sliding Window with Data Packets

The aim behind using this data management method is to save the memory space and CPU usage as well as executing the classified commands promptly without delays. In order to understand the theory of operation behind this concept, the block diagram of the system is first demonstrated as shown in Figure 4.18 on page 61 to have a wider overview of the procedures. As described, three main loops are responsible for following the Data operations: Data acquisition, Packets Generation, then Packets Managements in addition to Features Extraction and Classification.

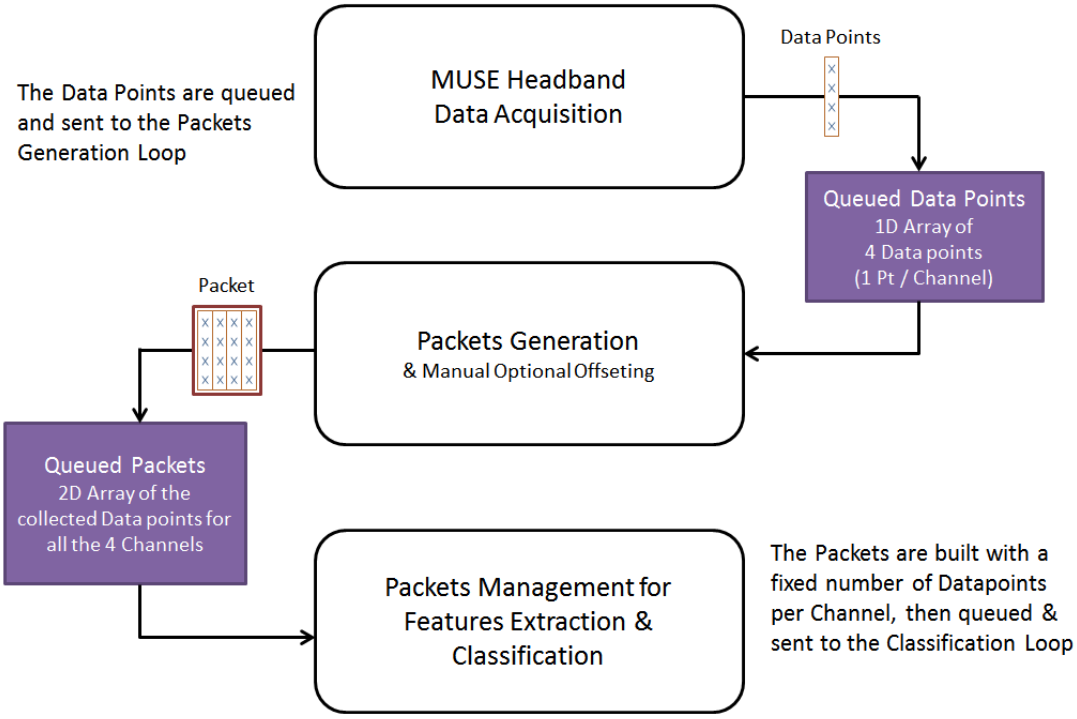

Figure 4.18: Packets Generation, Data Management and Classification architecture

It starts by the acquisition of the data points from the MUSE headband (Point by Point per Channel) and putting them in a queue for the next loop to regenerate and collect them in Packets.

A Packet includes a fixed number of data points per channel determined by a given period of time. For example, if the desired time period for a packet was set to 0.1 sec, around 32 data points would be collected per channel to form a complete packet, then the whole packet will be sent (queued) to the third loop further management, feature extractions and classification. Certainly, the number of collected data points relies on the acquisition rate and the loop speed.

As previously stated, the memory space was saved by getting rid of the unwanted data whenever the packet has no features, as well as saving the CPU usage by executing the feature extraction and classification function once per packet and wait till the next packet is built and delete the last packet when it does not contain any features. In case a packet had any detected features, it is kept in the memory and the new packets are added on and counted upon forming a group of packets, till either reaching a certain number of Packets without getting any second feature, then it will timeout and all of the collected packets will be deleted or else a second feature was detected within the new added packets and it matched one of the actions, then it will be classified immediately and afterwards, the classifier will reset and start collecting again packet per packet.

The following figure 4.19 describes how and when the packets are generated and sent to the Feature extraction and Classification function to decide whether to delete this packet or to continue adding for the next loops till a specified timeout.

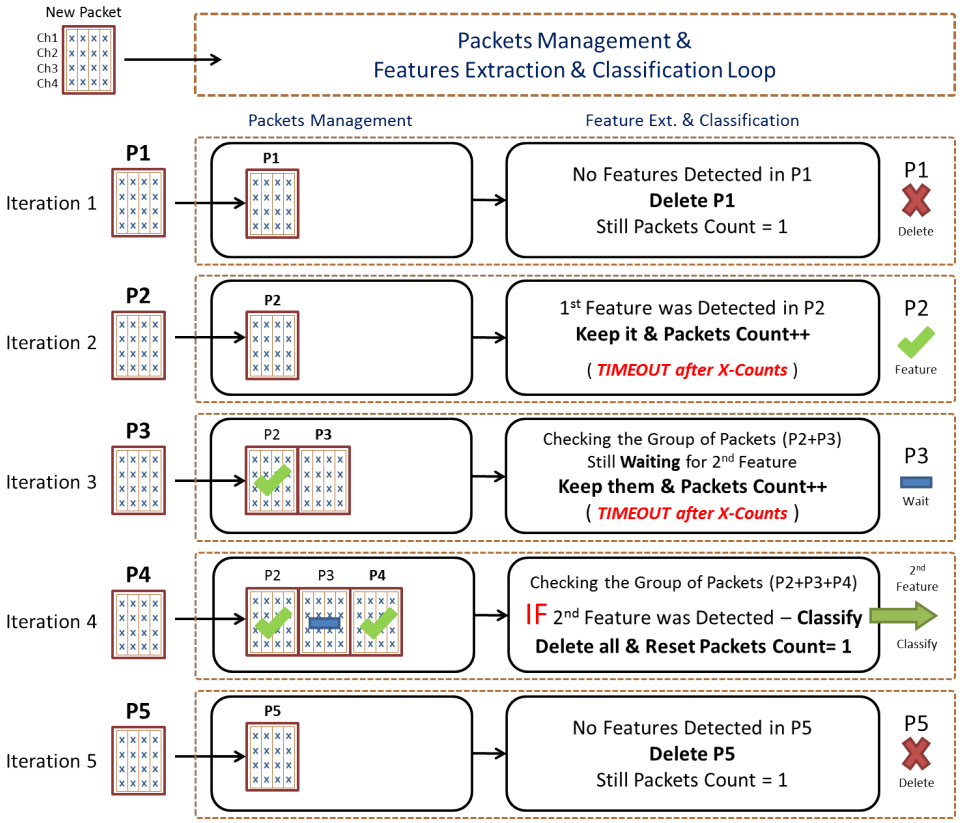

Figure 4.19: Packets Management, Feature Extraction and Classification Criteria

4.3.5 CLASSIFICATION

The following example shown in figure 4.20 describes more the theory of operation of the system. As previously mentioned, the packets are checked one by one and if it has no Starting features, it will be totally dismissed and if a Packet had a Starting feature, it will be kept and the Packet counter will increment which implies expanding the Data Array size to add additional packets without dismissing the previous ones until the the Data array reaches the maximum number of Packets (size) or the Second feature was detected within the permissible size. This method is analogous to the queuing method, but the queue size is dynamic from 1 to a maximum predefined size.

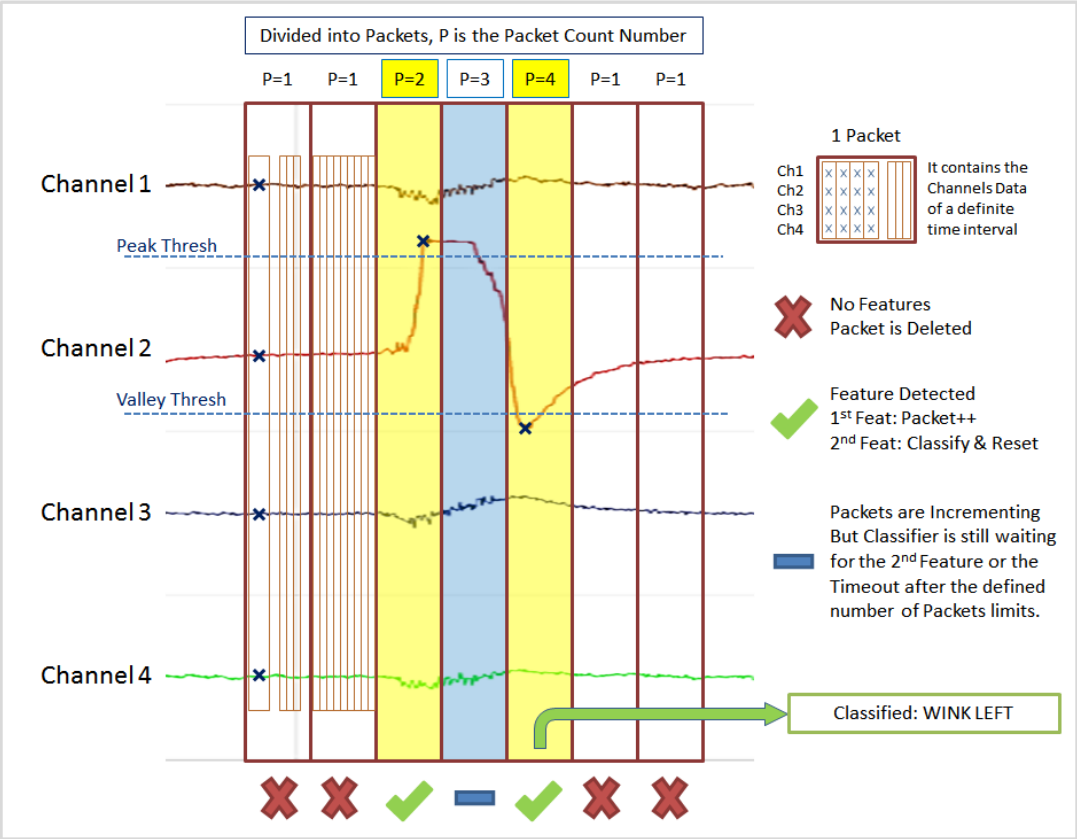

Figure 4.20: Example: Feature Extraction and Classification from Packets

The Packets are checked Channel by Channel for the predefined features. Channel 4 was ignored since it was nearly identical to Channel 1 as they are symmetric and there are no side actions affecting them rather than the blink which is performed symmetrically on the two sides.

Figure 4.21 on page 64 depicts the whole block diagram of the core feature extraction by thresholding and the ascending order of checking the eye gestures according to their classification priorities, which is represented from the least priority Blinks to the highest priority Wink Right or mainly Squint. The reason behind the actions priorities is to avoid the wrong classification to minor actions instead of the major performed actions. Practically, sometimes the large amplitude

signals such as the squint (on channels 2 and 3) would accompany large amplitudes on the irrelevant channels 1 and 4 which would make classification conflicts with the Long Blink action. In this case, the Squinting is having more priority than the Long Blink and it will reset the Long Blink Packet counter to 1 and it will increment its own to 2 and it will keep counting up till either finding the Second Squinting feature or it will timeout after 10 Packets. The packet is a 2D array including 4 rows with each Channel data. Channel 1 and 4 are equivalent since they both read the blinking action which is a symmetric action, thus Channel 4 was excluded for faster processing and the first 3 channels were considered in 3 iterations. Each iteration is responsible to pass the corresponding channel data to the Threshold detector function, then check its features and make a classification.

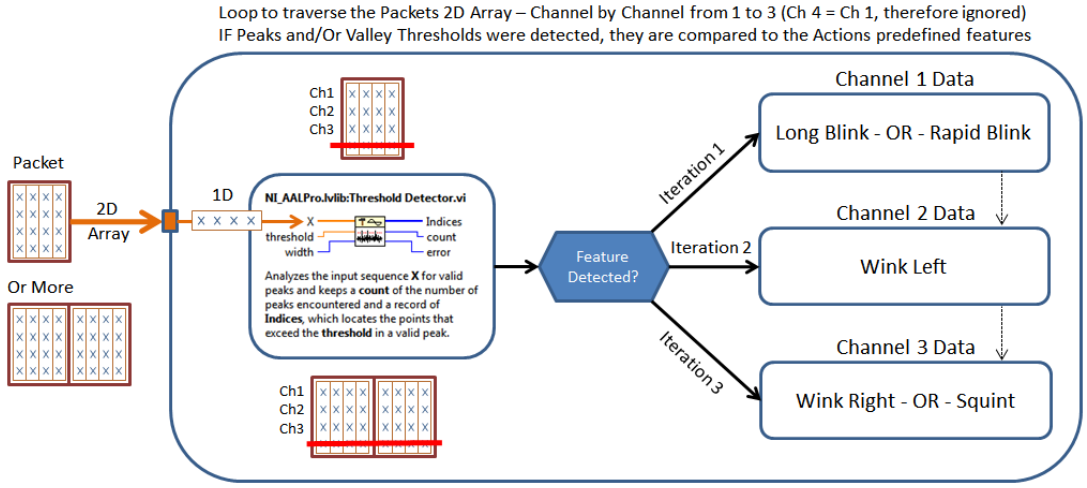

Figure 4.21: Feature Extraction and Classification function architecture

Furthermore, every eye gesture had its own feature conditions. The defined intended Blinks in our system are having more unique and identifiable features than the involuntary blinks which have small amplitudes on Channels 1 and 4 as shown in Figure 4.22 on the next page.

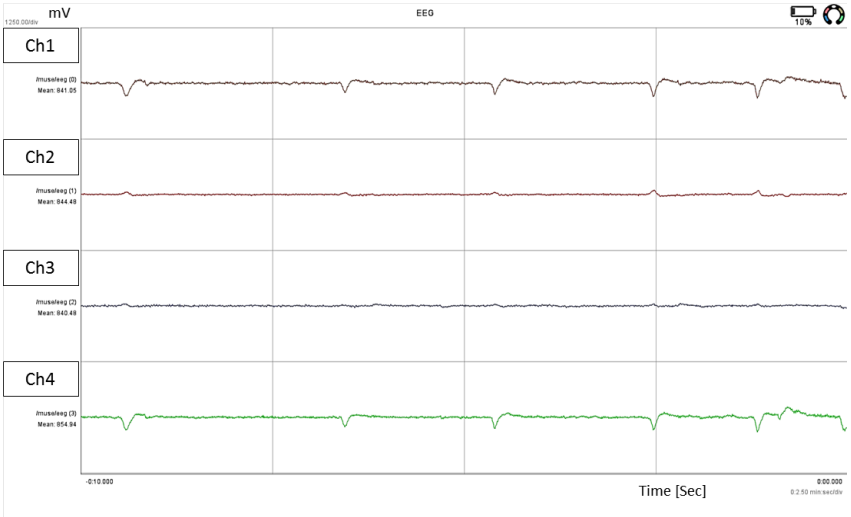

Figure 4.22: Normal Involuntary Blinks with very low amplitudes

For instance, we will define the trained or manually adjusted Peak and Valley thresholds variables as PTh1, PTh2, PTh3, PTh4 and VTh1, VTh2, VTh3, VTh4 respectively for the Channels from 1 to 4. Two thresholding functions were used in our implementation, one for the Peaks and the other for the Valleys but the Data points as well as the valley threshold values were negated to be used with the Threshold Detector function as shown in Figure 4.23.

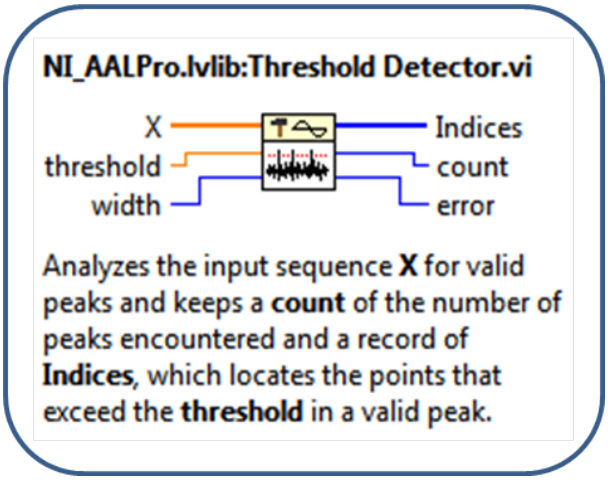

Figure 4.23: LabVIEW: Threshold Detector function for Peaks with crossing points more than or equal to the provided width value.

The Threshold Detector function has 3 inputs and 3 Outputs as follows in Table 4.2:

Table 4.2: Threshold Detector function inputs and outputs

Type	Variable	Description
Input	X	1D Data Input per Channel. X size must be greater than Width
	Threshold	Level that the number of peak points must be greater than or equal to the Width
	Width	Minimum number of samples above the threshold value for the crossing peak to be considered valid
Output	Indices	Array containing the starting indices of all the detected valid peaks from the 1D data array.
	Count	Number of detected valid peaks within the given 1D data array

In fact, the Width input in our test was set to minimum 3 samples to be considered a valid peak, so as to assure that it is a Real peak and not an outlier or noise. The counts output is considered the trigger that confirms the detection of a feature when it is more than Zero as it counts the Peaks within the provided Channel Data (Packet or Group of Packets after the first detection). Since the **Indices** provide the starting index of all the detected Peaks or valleys with the data negated, thus we used a Threshold Detector function for the Peak and another function for the valleys, then we compare both indices to check whether a valley was received before a Peak for

the Long blinks or the contrary for the Winks and Squints. All the actions were restricted to be classified with more than 2 Packets having small sizes (around 32 or 40 points in 0.1 sec) to be guaranteed as a gesture. If all the features were detected together in only one small packet, then it is dismissed as it would be considered as a distortion.

4.3.5.1 Long Blink Classification Conditions

1. Checking on Channel 1
2. Peak is detected - Current Threshold > PTh1, Hence (Peak Threshold Counts>0)
3. Valley is detected - Current Threshold < VTh1, Hence (Valley Threshold Counts>0)
4. Peak Index is larger than the Valley Index (Peak comes after a Valley)
5. Number of Packets (Count) > 2 (For small packet sizes as 32 points, around 0.1 sec according to our specified rate)

4.3.5.2 Rapid Blink Classification Conditions

1. Checking on **Channel 1**
2. No Peaks are detected (Counts<=0)
3. At least 2 successive Valleys are Detected - Current Threshold < VTh1
4. Number of Packets (Count) > 2 (For small packet sizes as 32 points, around 0.1 sec according to our specified rate)

4.3.5.3 Wink Left Classification Conditions

1. Checking on **Channel 2**
2. Peak is detected - Current Threshold > PTh2, Hence (Peak Threshold Counts>0)
3. Valley is detected - Current Threshold < VTh2, Hence (Valley Threshold Counts>0)
4. Peak Index is smaller than the Valley Index (Peak comes before a Valley)
5. Number of Packets (Count) > 2 (For small packet sizes as 32 points, around 0.1 sec according to our specified rate)

4.3.5.4 Wink Right Classification Conditions

1. Checking on **Channel 3**
2. Peak is detected - Current Threshold > PTh3, Hence (Peak Threshold Counts>0)
3. Valley is detected - Current Threshold < VTh3, Hence (Valley Threshold Counts>0)
4. Peak Index is smaller than the Valley Index (Peak comes before a Valley)
5. Number of Packets (Count) > 2 (For small packet sizes as 32 points, around 0.1 sec according to our specified rate)
6. **No Detection from Channel 2 (No Wink Left)**

4.3.5.5 Squint Classification Conditions

1. Checking on **Channel 3**
2. Peak is detected - Current Threshold > PTh3, Hence (Peak Threshold Counts>0)
3. Valley is detected - Current Threshold < VTh1, Hence (Valley Threshold Counts>0)
4. Peak Index is smaller than the Valley Index (Peak comes before a Valley)
5. Number of Packets (Count) > 2 (For small packet sizes as 32 points, around 0.1 sec according to our specified rate)
6. **A Full Detection occurred at Channel 2 as well (Wink Left = 1)**

The Squinting action is depicted as the detection of two full gestures simultaneously on Channels 2 and 3 starting with a Peak and ending with a valley. So it can be analogous to the detection of a Wink Right and a Wink Left at the same time. For this reason, one of the Winks has to be checked first as the (Wink Left here in our case) then the second Wink (Wink Right) is checked afterwards. As shown in figure 4.24, if a Wink Left shape was detected, then the classification would either be a Wink left or a Squint in case a Wink Right shape was detected on Channel 3. Otherwise, the detection will either be a Wink Right on case Channel 3 if an action was detected or else nothing is detected.

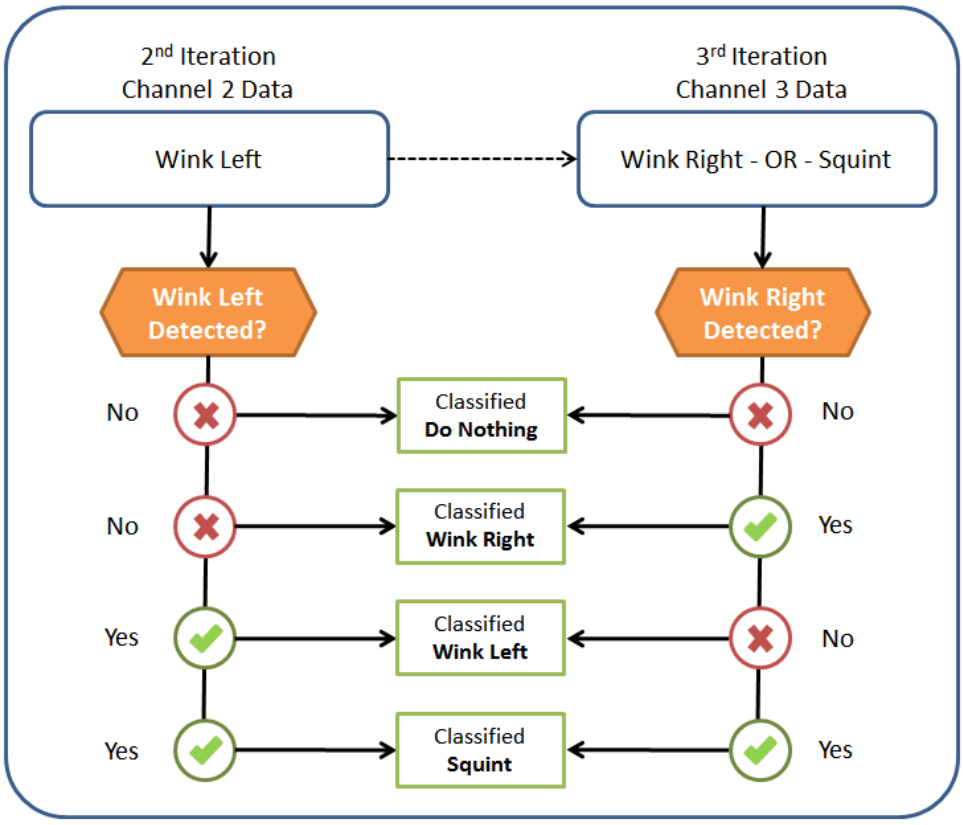

Figure 4.24: Features Extraction on Channel 2 and 3 to classify a Wink or a Squint

CHAPTER 5

RESULTS AND DISCUSSION

In this chapter, the results of the conducted user study will be presented in terms of accuracy, precision and the rates of hits and misses. Additionally, the results will be discussed for the overall collected data to evaluate the classifier performance. Also, some specific performances and disturbances which affected the results will be discussed and how the signals would be better recognized according to the observations and the evaluation of the performances.

5.1 UNDERSTANDING THE CONFUSION MATRIX TERMINOLOGY

This sub section will provide a brief description on the terminologies used in the analysis to better understand the results. For instance, confusion matrix as described by Provost *et al.* [35] includes the actual and predicted classifications decided by a Classifier to describe its performance on the provided data set.

Moreover, according to Visa *et al.* [36], it was illustrated as the matrix with size n x n showing the predicted classification or as in our case the instructed or actual performed action versus the actual resulted classification, in which n represents the number of independent classes. The entries of the Confusion matrix are as shown in Figure 5.1 on page 69 which is an example table for the classification of two classes (n=2).

	PREDICTED NEGATIVE	PREDICTED POSITIVE
ACTUAL NEGATIVE	a	b
ACTUAL POSITIVE	c	d

Figure 5.1: Example of a Confusion Matrix for Two-Classes Classification Visa *et al.* [36]

The following points describe the meaning of those entries:

- [a] represents the number of **True Negative (TN)** predictions
- [b] represents the number of **False Positive (FP)** predictions
- [c] represents the number of **False Negative (FN)** predictions
- [d] represents the number of **True Positive (TP)** predictions

In other words, the Confusion Matrix variables will be labeled in our results by TP, TN, FP and FN. According to our application, they are depicted as follows:

• **True Positive (TP)**: Instructed/Performed Action and Got Detected/Classified
• **False Positive (FP)**: NOT Instructed/Performed Action, But Got Detected/Classified
• **True Negative (TN)**: NOT Instructed/Performed Action and NOT Detected/Classified
• **False Negative (FN)**: Instructed/Performed Action, BUT NOT Detected/Classified

5.1.1 DATA Analysis

After constructing the Confusion Matrix and deducing the True Positive and Negative, as well as the False Positives and Negatives for all the classes, the data can now be analyzed for the following performance measures with their defined equations according to Fawcett [37]:

5.1.1.1 Sensitivity, Recall, Hit rate, or True Positive Rate (TPR)

For an Action class, this defines the Ratio between the number of its **Correct Detections when performed** to the total number of times this action was **Performed**. In other words, it is the rate of having true classifications, when actually performed. It is calculated as follows in equation (5.1):

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{FNR} \tag{5.1}$$

5.1.1.2 Specificity or True Negative Rate (TNR)

For an Action class, this defines the Ratio between the number of its **Correct Non-Detections when Not performed** to the total number of times this action was **Not Performed**. In other words, it is the rate of having true No or Negative classifications, when truly Not performed. It is calculated as follows in equation (5.2):

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{FPR} \tag{5.2}$$

5.1.1.3 Fall Out or False Positive Rate (FPR)

For an Action class, this defines the Ratio between the number of its **False Detections when Not performed** to the total number of times this action was **Not Performed**. In other words, it is the rate of having Classifications for this action, although is was actually Not performed. It is calculated as follows in equation (5.3):

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} = 1 - \mathrm{TNR} \tag{5.3}$$

5.1.1.4 Miss Rate or False Negative Rate (FNR)

For an Action class, this defines the Ratio between the number of its **Non Detections when actually performed** to the total number of times this action was **Performed**. In other words, it is the rate of having No classifications, although is was actually performed. It is calculated as follows in equation (5.4):

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} = \frac{\mathrm{FN}}{\mathrm{TP} + \mathrm{FN}} = 1 - \mathrm{TPR} \tag{5.4}$$

5.1.1.5 Accuracy

The accuracy of an action determines how often the classifier was correct to classify this action, when it was performed and correctly classified and when it was not performed and correctly not classified. The accuracy is calculated as follows in equation (5.5):

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{5.5}$$

5.1.1.6 Precision or Positive Predictive Value

The precision of an action determines how often the classifier was correct to classify this action when performed from all its detections. The precision is calculated as follows in equation (5.6):

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.6}$$

5.1.1.7 F1 Score

The F1-score is the weighted average or harmonic mean of the precision and true positive rate (Recall). The F1-score is calculated as follows in equation (5.7):

$$\mathrm{F}_1 = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{5.7}$$

5.2 USER STUDY CLASSIFICATION RESULTS

This section demonstrates the overall results of the designed classifier used in the user study for the programmatically adjusted Thresholds from the Training Data and the results of the Default or Manually adjusted Thresholds which were tested with some participants as well.

As previously mentioned in Chapter 4, the user study was conducted on 15 participants (P01 to P15) including 12 Males and 3 Females. In addition, 4 out of the 15 participants (P03, P05, P08 and P10) reported, in the "Participation sheet", their inability to perform the wink right and 1 participant (P15) could not wink left. The Discussion section will include the observation of the test with some other participants who had some challenges as well

Table 5.1 is only a dummy example to show the Confusion Matrix analysis criteria for the different classification cases of the predefined actions. The X marks are the check marks showing the performance conditions such as the true positive (TP) case when the action is classified as performed for the Long Blink and this implies that the other cases had true negatives (TN) for this event as they were not performed and correctly not detected.

Table 5.1: An EXAMPLE only to show the different Confusion Matrix cases (TP, FP, TN, FN) with the Performed and Classified Actions

Action Class (Positive Classes)		**Long Blink**				**Rapid Blink**				**Wink Right**				**Wink Left**				**Squint**			
Performed Action (Predicted)	**Classified Action (Actual Result)**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**
Long Blink	<*Long Blink* >	X						X				X				X				X	
Rapid-Blink	<*Long Blink* >		X						X			X				X				X	
Wink Right	<*No Detect* >			X				X					X			X				X	
Wink Left	<*Wink Left* >			X				X				X		X						X	
Squint	<*Wink Left* >			X				X				X			X						X
<No Actions >	<*Rapid Blink* >			X			X					X				X				X	

For instance, Table 5.2 shows the confusion matrix results of Participant P06 with the calculated Trained Thresholds for Channels 1 and 4 set to (PTh1 = PTh4 = 1008) for the Peaks and (VTh1 = VTh4 = 616) for the valleys. In addition, Channels 2 and 3 thresholds were set to 1420 and 142 for the peaks (PTh2, Pth3) and valleys (VTh2, VTh3) respectively. Each participant is instructed to do each action at least 3 times during the whole session. Some actions were repeated more times in case he did not do it in the right way or to double check the robustness of classifying an action. Participant P06 was instructed to do 5 Long Blinks, 4 Squints and 3 times in the other actions.

Table 5.3 summarizes the results of the confusion matrix for participant P06 and figures 5.2 and 5.3 show his performance results in terms of Recall, Fall out, Miss Rate, Specificity, Accuracy, Precision and F1 Score.

Table 5.2: Participant P06 Confusion matrix results with Trained Thresholds

Action Class (Positive Classes)		Long Blink				Rapid Blink				Wink Right				Wink Left				Squint			
Performed Action (Predicted)	**Classified Action (Actual Result)**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**	**TP**	**FP**	**TN**	**FN**
Long Blink	*Long Blink*	X						X				X				X				X	
Wink Left	*Wink Left*		X						X			X				X				X	
Wink Left	*No Detect*			X				X					X			X				X	
Wink Right	*Wink Right*			X				X				X		X						X	
Squint	*Squint*			X				X				X			X						X
Squint	*Squint*			X				X				X				X		X			
Wink Right	*Wink Right*			X				X		X						X				X	
Wink Right	*Wink Right*			X				X		X						X				X	
Rapid Blink	*Rapid Blink*			X		X						X				X				X	
Long Blink	*Long Blink*	X						X				X				X				X	
Rapid Blink	*Rapid Blink*			X		X						X				X				X	
Squint	*Wink Right*			X				X			X					X					X
Squint	*Squint*			X				X				X				X		X			
Long Blink	*No Detect*				X			X				X				X				X	
Rapid Blink	*Rapid Blink*			X		X						X				X				X	
Wink Left	*Wink Left*			X				X				X		X						X	
Long Blink	*No Detect*				X			X				X				X				X	
Long Blink	*Long Blink*	X						X				X				X				X	

Table 5.3: Participant P06 Confusion Matrix Results per Gesture class for Trained Thresholds

Confusion Matrix Variables	True		False	
	Positive	**Negative**	**Positive**	**Negative**
Long Blink	3	13	0	2
Rapid Blink	3	15	0	0
Wink Right	3	14	1	0
Wink Left	3	15	0	0
Squint	3	14	0	1

Figure 5.2: Participant P06 - Performance measures in terms of TPR, TNR, FPR and FNR

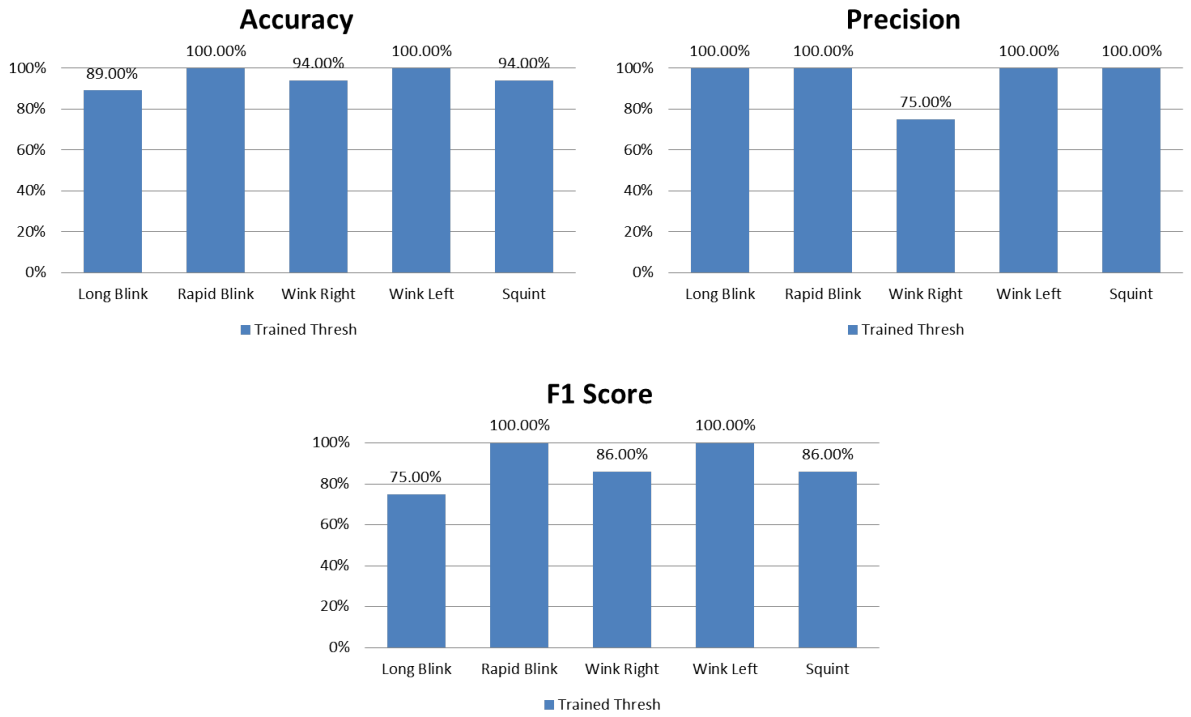

Figure 5.3: Participant P06 - Accuracy, Precision and F1 Score Performance measures

5.2.1 TRAINED Thresholds Test Results

Around 303 eye gesture events were captured from all the participants during the user study to test their adjusted trained thresholds. Table 5.4 on page 74 includes the overall results from all the participants for the Long Blink, Rapid Blinks, Wink Right, Wink Left and Squint action classes.

Table 5.4: Confusion Matrix Results per each Gesture class for Trained Thresholds

Confusion Matrix Variables	True		False	
	Positive	Negative	Positive	Negative
Long Blink	12	244	12	35
Rapid Blink	63	210	13	17
Wink Right	19	261	9	14
Wink Left	23	255	2	23
Squint	81	195	13	14

Table 5.5 and 5.6 demonstrate the analysis of the confusion matrix to measure the classifier performance with the Trained Thresholds.

Table 5.5: Confusion Matrix Analysis TPR, TNR, FPR and FNR for Trained Thresholds

Confusion Matrix Analysis Terms	Recall TPR	Specificity TNR	Fall Out FPR	Miss Rate FNR
Long Blink	25.53%	95.31%	4.69%	74.47%
Rapid Blink	78.75%	94.17%	5.83%	21.25%
Wink Right	57.58%	96.67%	3.33%	42.42%
Wink Left	50.00%	99.22%	0.78%	50.00%
Squint	85.26%	93.75%	6.25%	14.74%

Table 5.6: Confusion Matrix Analysis Accuracy, Precision and F1 Score for Trained Thresholds

Confusion Matrix Analysis	Accuracy	Precision	F1 Score
Long Blink	84.49%	50.00%	33.80%
Rapid Blink	90.10%	82.89%	80.77%
Wink Right	92.41%	67.86%	62.30%
Wink Left	91.75%	92.00%	64.79%
Squint	91.09%	86.17%	85.71%

5.2.2 DEFAULT/MANUALLY Adjusted Thresholds Test Results

Around 287 eye gesture events were captured from some (not all) participants during the user study to test the Classifier with some Default or Manually adjusted thresholds. The Default Thresholds were of fixed values for (PTh1 & PTh4 = 950), (Pth2 & PTh3 = 1400), (VTh1 & VTh4 = 700) and (VTh2 & VTh3 = 550). Table 5.7 on the following page includes the overall results from all the participants for the Long Blink, Rapid Blinks, Wink Right, Wink Left and Squint action classes.

Table 5.7: Confusion Matrix Results per each Gesture class for Default Thresholds

Confusion Matrix Variables	True		False	
	Positive	**Negative**	**Positive**	**Negative**
Long Blink	26	4	230	27
Rapid Blink	85	19	177	6
Wink Right	17	7	246	17
Wink Left	17	6	255	9
Squint	62	12	198	15

Tables 5.8 and 5.9 demonstrate the analysis of the confusion matrix to measure the classifier performance with the Default Thresholds.

Table 5.8: Confusion Matrix Analysis TPR, TNR, FPR and FNR for Default/Manually adjusted Thresholds

Confusion Matrix Analysis Terms	Recall TPR	Specificity TNR	Fall Out FPR	Miss Rate FNR
Long Blink	49.06%	98.29%	1.71%	50.94%
Rapid Blink	93.41%	90.31%	9.69%	6.59%
Wink Right	50.00%	97.23%	2.77%	50.00%
Wink Left	65.38%	97.70%	2.30%	34.62%
Squint	80.52%	94.29%	5.71%	19.48%

Table 5.9: Confusion Matrix Analysis Accuracy, Precision and F1 Score for Default/Manually adjusted Thresholds

Confusion Matrix Analysis	Accuracy	Precision	F1 Score
Long Blink	89.20%	86.67%	62.65%
Rapid Blink	91.29%	81.73%	87.18%
Wink Right	91.64%	70.83%	58.62%
Wink Left	94.77%	73.91%	69.39%
Squint	90.59%	83.78%	82.12%

Additionally, the comparative results of the Trained and Default thresholds are shown in figures 5.4 and 5.5

Figure 5.4: Trained and Default Thresholds Performance Measures (TPR, TNR, FPR, FNR)

Figure 5.5: Trained and Default Performance Measures (Accuracy, Precision & F1 Score)

The Default or manually adjusted thresholds were better in some cases, since the trained ones were calculated from the average of peaks and valleys of only three signals per gesture, which increased the chances of deviations for the trained threshold values, in case of the occurrence of irregular eye gesture performances during the training.

5.3 OBSERVATION AND DISCUSSION

This section discusses the results and observation differences between this study and some comparable studies with relatively similar conditions.

5.3.1 OBSERVATIONS and Comparable Studies

5.3.1.1 Comparative Study 1: Rapid Blinks, Wink Right & Left and Squint(Frown)

From the data represented in Tables 5.5 and 5.6 from the Trained Thresholds and the Tables 5.8 and 5.9, the Recall or as also called Sensitivity, Hit rate or the True positive rate is relatively high for the Squinting action with 85.26% and 80.52% for the Training and the Default Thresholds respectively. Both Recall values for the Squint are significantly higher than the one tested by Ma *et al.* [21] who had a comparable study with only 13 subjects and reported a Recall of 77% for Squints or frowns as he named it.

In our case, the Blinks are detected from the two MUSE headband sensors behind the ears and not from the forehead and this is considered as an advantage in our case as we rely on independent channels, thus the channel number is also considered as a Classification parameter. In very special incidents, this channel would be affected slightly by the Winks or Squints and it depends more on how the electrodes are located but those actions were given higher priority to avoid such a case. Nevertheless, this is unlike Ma *et al.* [21] who relied on only 1 vertical electrode on the forehead to read the Blinks and the Squints.

Additionally, Ma *et al.* [21] reported that the Thresholding Algorithm was very reliable for Blinks, however it was below expectations for Squints. Unlike the results in this work, the thresholding algorithm was shown promising since they were on independent channels and even in our model, the Blinking signal was unlike the others, as it starts by a negative followed by a positive amplitude. But Ma *et al.* [21] had the same shape of the signal starting with a positive then followed by a negative amplitude for all his proposed actions (Blinks, Winks, Frown (Squint) and Gaze) as he had the sensors on different locations. Hence, further processing such as the first differentiation, as implemented by Ma *et al.* [21] to the signals, was not needed in our case since they were obviously distinguishable for direct thresholding.

Furthermore, in order to make a one to one comparison between this work and the work done by Ma *et al.* [21], the Long Blinks were excluded from our results, since the Rapid Blinks were only the common actions related to the Blink gestures between the two studies. The results were reevaluated and it was found that the Rapid Blinks Recall in our case reached 96% with the default thresholds which was typically the same as the other study.

5.3.1.2 Comparative Study 2: Normal, Double Blinks and 1 Wink action

Another study by Nakanishi *et al.* [27] proposed the recognition of involuntary normal blinks, voluntary double blinks and a Wink action at the dominant side (left or right) according to each user. The EOG signals were measured to control the trigger switch combined with the BCIs. It was stated that the Normal blinks and the Wink duration were around 400ms, but the Double blink was relatively greater around 600ms as it includes two successive blinks.

The study conducted their experiments on 8 participants only relied on the Peak thresholds extractions within fixed defined window intervals. The participants were given a Rest time between the sessions and the number of trials for all the actions were stated as 100 times per subject. The study utilized a support vector machine (SVM) to classify the signals. They used also the confusion matrix approach to analyze and compute the results of the SVM. They reported a high accuracy for the Wink action of 97.69%, however there were 6.58% mis-classifications for the Double blink with the Normal one due to the difficulty of blinking twice quickly for some subjects and others blinked three times instead.

However, they conducted an Online experiment and relied on the Peak thresholds extractions within fixed defined window intervals around 600, 700, and 800 ms for each action, in addition to 100 and 200 ms shift amount. It was stated that the classification are resulted per window, however the outputs are decided after having two successive windows with the same result. They evaluated the system by measuring defining the Detection accuracy as the percentage ratio between the Correct outputs and the Total number of outputs. A maximum detection accuracy of 93.77% was reported for the overall performance using a window size of 800ms and the Shifting amount was 200ms. Also some irregularities and slow performances were reported in this study.

5.3.2 OBSERVATIONS with Participants

It was observed that some Participants, who claimed their ability to do normally some actions especially the winks, were found that they could in a way perform the action but not considered as the normal mainstream definition of this action. The recorded videos were rechecked for all the participants to highlight all the challenges affecting the test.

5.3.2.1 Winks and Squints

For instance, Participants (P09 and P14) were implementing the Winks from their cheek muscles rather than the forehead ones. This made quite false results or no detections as the test was mainly relying on the forehead muscular contractions to classify the Winks and the Squints. Subject P09 was even making the Squints from his cheeks as well which resulted in several False Negatives. However, we considered this incident and included it normally in our results as it needs further investigation. For instance, Ma *et al.* [21] reported also some participants such as S7 in their project who had some irregular eye movements which were hard to recognize.

5.3.2.2 Rapid Blinks

Some Participants were observed to perform the Rapid Blinks sometimes too fast in some events which results in very low peaks such as (P14). On the other hand, others such as (P11) had sometimes high peaks which crossed the Peak threshold for Channel 1 (PTh1) and was considered as Long Blink in the Default Thresholds as he missed all the Rapid Blinks from the Trained Thresholds as the Valley threshold for Channel 1 (VTh1) was set very low. The thresholds may have needed to be manually adjusted in such a case.

Additionally, some other participants were making involuntary Rapid Blinks during the test, they were correctly detected by the classifier as it was designed to detect at least 2 successive negative amplitudes. However, this means the Rapid Blinks detection has to be designed after at least 4 successive valleys to make sure it was intended. This modification was implemented

later in the classifier and tested on some actions from the offline data and shown promising for more accurate results.

5.3.2.3 Long Blinks

Away from the Classifier confusion sometimes with the Long Blinks, but the Long Blinks were also observed hard to be successively implemented as they would cause fatigue. But as an eye gesture, it can perform well if better distinguished from the Rapid Blinks (not only from the amplitudes) and the thresholds on Channel 1 or 4 would suitably adjusted.

5.3.3 LIMITATIONS

The first significant limitation was the challenge against generalizing the system due to the differences between the ways of implementing the eye gestures. Hence, more powerful algorithms and machine learning techniques for more number of participants are still required to make the classifier more capable of accurately and correctly classifying the signals.

Additionally, the second challenge was the electrodes locations and positioning as these also have a big influence on the readings especially the locations. However, the electrodes had to be positioned on the forehead in a suitable way such that Channel 2 and 3 could be able to detect significantly the muscular contractions.

It is also important to note that using a headband where all the electrodes are fixed along, had also some disadvantages with few participants, as they would implement an action especially the Winks or Squints in a strong way which would move the headband, thus affect the readings of the sensors behind the ear as they experienced external disturbances due to dislocation during the implementation of the action. Although that was not the mainstream case, but may be investigating a system with independently assembled sensors would provide better results.

Furthermore, the fatigue is also one of the most common challenges in such interfaces when the muscular stimulations are done successively in a short time. Hence, the system should be used for applications which do not require fast repetitive interactions such as Menu Selection or such as changing the views of a display screen as in the proposed project on the Augmented Reality device's display. Ma *et al.* [21] reported that it there is a trade off between accuracy that requires longer training time and fatigue which affects the experience of the user. For typical applications, such as the switching of an augmented reality device displays for the firefighters, fatigue of their forehead muscles would be less as the switching would not be frequently done in a short time. However, their overall physical fatigue should be tested against their performance of doing the eye gestures.

5.4 IMPROVEMENTS ON THE CLASSIFIER PERFORMANCE

From the observations of the video recordings of the participants, as well as the presented results. Several modifications were noted, applied and tested on the Offline data to improve the classifier performance.

For instance, from the confusion matrix results, the Long blink was wrongly classified as Rapid Blink in eight incidents, which means false negatives for the Long Blinks class and false positives for the Rapid Blink class. This problem was found to be mainly due to the low features set for the Rapid Blinks in the experiment by only detecting two consecutive valleys (-ve amplitudes). Sometimes the involuntary blinks of some participants were crossing the valley threshold set for channels 1 and 4 (VTh1, VTh4), thus when the participant makes an involuntary blink just before implementing a Long Blink gesture as shown in Figure 5.6 on page 81, the classifier becomes confused and classifies this as Rapid Blink as it detected Valley 1 & 2 as shown and detected by Participants P10 and P11.

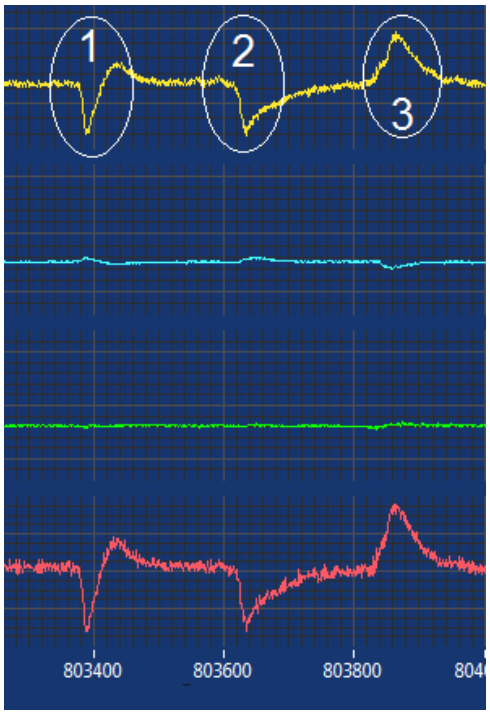

Figure 5.6: Long Blink starting from (2-3) confused with Rapid Blink due to a preceding involuntary Blink at (1)

However, generally even if the user made some involuntary blinks which crossed the defined valley thresholds, their implementation rate is very low compared to the Rapid Blinks. For this reason, in the normal cases, the involuntary blinks are dismissed, since their second valley feature will not be detected within the defined maximum range of packets, unless the user blinks twice rapidly which was also found in several incidents.

On the other hand, some other incidents were also found that the classifier decided a Long Blink, although it was originally a Rapid Blink. It was observed and found that some participants actually started the Rapid Blink action by a relatively strong and Long blink then proceeded with the rest of the blinks quicker and lower in amplitudes. In this case the classifier was confused as the Long Blink was classified first.

Consequently, the signals were first filtered with a Low Pass filter of 20 Hz Cut-Off frequency at a 250 Hz Sampling frequency, which is approximately equal to the data acquisition frequency of the MUSE headband. Thus the high frequencies were filtered out to making the signal smooth to decrease the fluctuations around the threshold limits, as well as getting ready to check its frequency domain within this range.

Then the Rapid Blinks features were redefined and extended as follows:

Rapid Blink Classification - Common Conditions
1. Checking on **Channel 1**
2. No Peaks are detected (Counts<=0)
3. Number of Packets (Count) > 2 (For small packet sizes as 32 points around 0.1 sec according to our specified rate)

The Fast Fourier Transform (FFT) was applied on the signal of Channel 1 to switch to the frequency domain, then the RMS values of the frequencies higher than the fundamental frequency were calculated and compared to some tested thresholds by trials.

Rapid Blink Classification - Modified Conditions
4. At least 3 or 4 consecutive Valleys are Detected - Instead of only 2
OR If the
5. RMS Value of frequency Range of the signal between 3-8 Hz > 4000
6. RMS Value of signal at 1Hz < 8000 (Trials)

The fourth point solved the confusion of classifying an involuntary blink then a Long Blink as a Rapid Blink, as the condition is now more strictly constrained on the time domain thresholds to three or four negative crossings (valleys).

The RMS values of Points 5 and 6 were for testing purposes only and to provide a proof of concept for the proposed modification, however it still needs to be verified and more accurately calculated with more tests. However, once these two conditions are met for Channel 1 or 4, then the Classifier executes a Rapid Blink as shown in figure 5.7

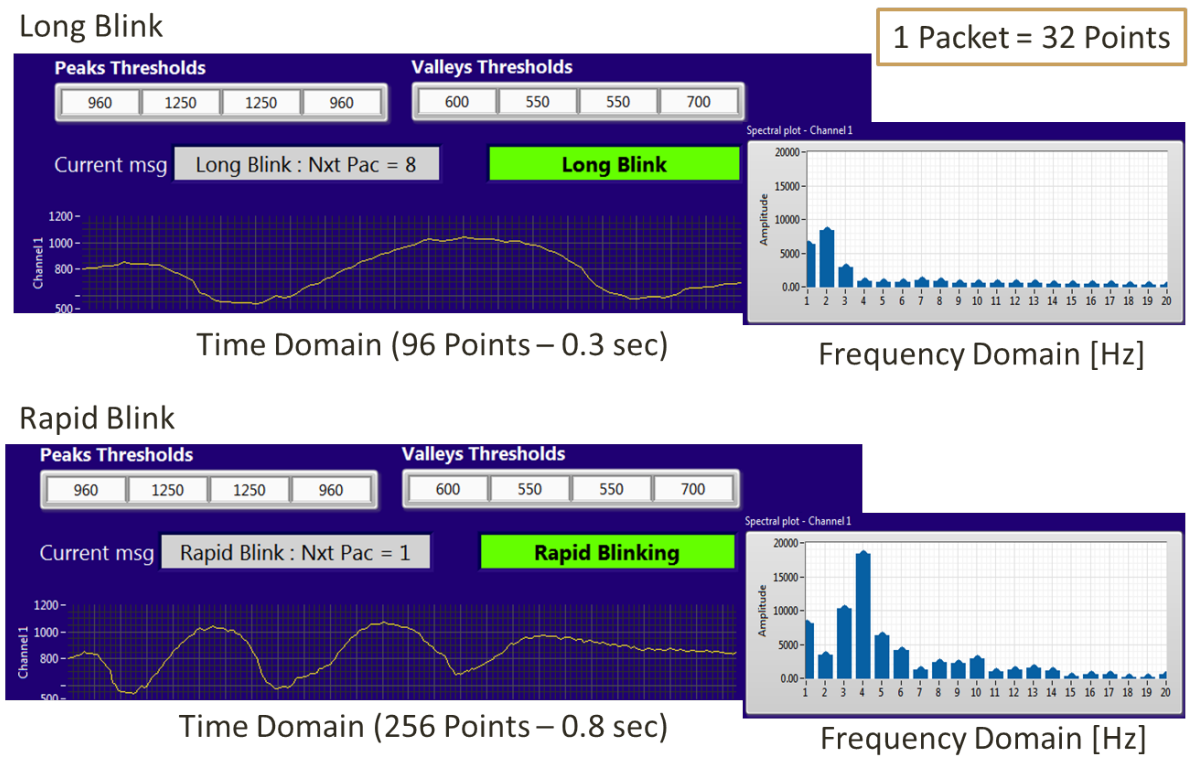

Figure 5.7: After Long Blink Detection, Classifier halts and checks the possibility of Rapid Blinks in the next 4 packets

As previously mentioned, the confusion of classifying a Long Blink instead of a Rapid Blink is due to a peak within the progressing blinking signals which were not yet finalized to prove the action as a Rapid Blink gesture. Therefore, to avoid this confusion, the Long Blink is not directly executed when classified, but rather delayed for one iteration, in which the classifier halts and feedbacks the current Packets count plus four more packets (Pcout + 4) to be processed all together in the next iteration to check whether the signal has higher frequencies which would most likely be a Rapid Blink or otherwise it is confirmed as a Long Blink as shown in figure 5.8.

The Packet size is very important to be pre-adjusted not to be very small to faster process the data and to avoid early classifications of irregularities (if an action signal had several irregular peaks or valleys). Also the Packet size should not be so big to read the whole within one packet as we made a constraint of at least two packets to classify a signal to make sure it is not a noise. Additionally, the greater the Packet size, the more memory space needed to process these data together. Hence, in our case, we tested the system with a Packet duration of 0.1 sec which collected around 32 or 40 points according to the MUSE headband acquisition rate and the Packets Generation Loop rate.

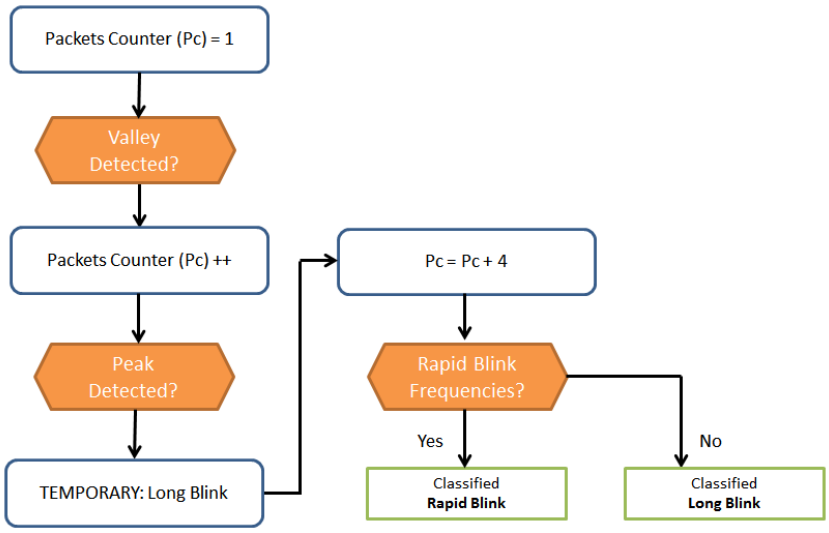

Figure 5.8: Flowchart illustrating the approach to tackle the Rapid Blink confusion with the Long Blink.

Figure 5.9 shows also the differences between the spectral plots for Channel 1 and Channel 2 on the occurrence of a Rapid Blink.

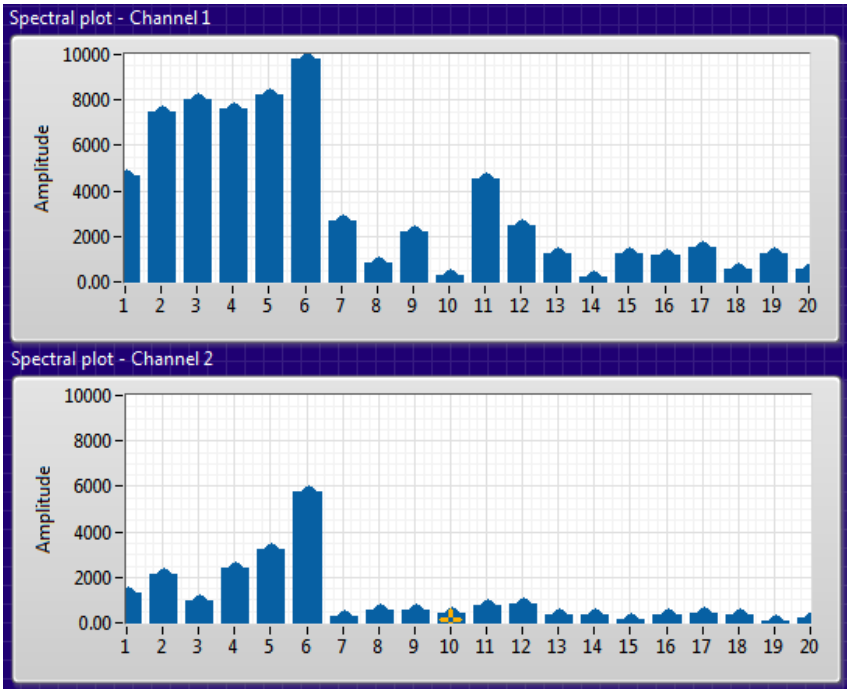

Figure 5.9: Spectral Plot for Channels 1 and 2 - Rapid Blink Detection in Frequency Domain

CHAPTER 6

CONCLUSION

This chapter provides a summary on the work done as well as the results of this study. Additionally it presents the contributions and outcomes of this work that would benefit further research tackles in this scope. Finally, the future work for further modifications and potential applications will be presented at the end of this chapter.

6.1 SUMMARY

This study proposed and investigated the idea of utilizing the eye gestures stimulations as a way of interfacing, to minimize the physical interactions via hands free control over different applications. The proposed eye gestures in this work were the Long Blinks, Rapid Blinks, Wink Right, Wink Left and the Squints, and the detection of these gestures was based on the Electromyography-EMG signals measured by 4 electrodes from the MUSE headband. The acquisition of the signals from the MUSE, in addition to the Feature extraction and Classification functions were all programmed in LabVIEW. The experiment was conducted on 15 participants who had satisfactory performance on most of the actions especially the Rapid Blinks and the Squints. Some challenges were confronted due to the irregular eye gestures of some subjects who could not perform some actions such as the Winks and Squints in the normal way. Additionally, as reported by Ma *et al.* [21] as well, these tests are muscle dependent and therefore they can cause fatigue when being performed for longer repetitive utilization.

6.2 CONTRIBUTIONS

This sections summarizes the contributions provided throughout this work. It will introduce the directly dedicated contributions to the study in the first sub section, then it will present general purpose contributions for further research projects.

6.2.1 Specific Study Contributions

Regarding the data management and classification, an **Online Dynamic Sliding Window with Segmented Packets** approach was designed to faster and more efficiently process the data and extract the features with more flexibility for relatively longer or shorter gesture durations. The Packet size can be manually adjusted in Seconds or Data points if required. It would be also used with different classification algorithms as it piles up the packets once it detects the first defined feature in the current Packet and checks the next packets for possible features in the rest

of the signal. It will either find a stopping feature and classify this signal directly or it would reach the maximum defined number of packets. This implies that there was no actual predefined action implemented and in this case, the window size is set back to 1 Packet. For the Thresholds algorithm, the first feature is either a Peak (+ve amplitude) or a Valley (-ve amplitude) in the time domain, but for further work, the features would be amplitudes or ranges in the frequency domain.

In addition, various studies used the Electrooculography (EOG) signals to detect the eye gestures by placing electrodes above, below or on the sides of the eyes. However, it was observed that few studies relied on the Electromyography (EMG) signals to acquire the eye gestures especially the blink from behind the earlobe, although the signals were found clear, distinguishable and independent from the other gestures as highlighted throughout this study.

6.2.2 General Purpose Contributions

Furthermore, aside from the core of the study and its results, this work provided several new tools that would be used for further research projects who aim to do the following:

6.2.2.1 MUSE BCI Commmunication Platform using LabVIEW

To program and communicate with the MUSE BCI headband using LabVIEW for further EEG-based or EMG-based applications, as this work provided a new non-existing platform, (as even checked on the MUSE website Tom [38] itself), to program and communicate with the MUSE headband in LabVIEW using the NI OSC Library which translates and reads all the message streams sent from the MUSE over Bluetooth communication.

6.2.2.2 Voice Narration-based User Study Platform

To establish a user study with standardized test conditions to collect consistent data from all the participants. This work provides a platform to programmatically direct the participants with the required instructions via a Voice Narration system with definite and adjustable time intervals. The session director has the ability to add or remove instructions and to shuffle their order at any time as well. Additionally, the instruction duration can be set by either a "Total Time Duration in seconds" or by "Beep cycles" if more than a reading for the same instruction is required. The tool saves the timestamps along with their description of the "Start", "End" and "Beep Occurrences" for every instruction, which are considered as the markups for the measurement data for further direct accessing, processing and analysis. This tool was also programmed in LabVIEW and can be produced independently in executable format for general purposes.

6.3 FUTURE WORK

This section summarizes our vision for the future work, potential applications and modifications to be added to improve the EMG-based eye gestures recognitions and to be capable of better utilizing this technology in Real life applications.

6.3.1 Future Improvements on System Performance

From the observations and the study results, several challenges should be tackled in the future work. For instance, the irregularities in the eye gestures need some special focus not only on the classification algorithm but also on the number of electrodes and their placements. In fact, Ma *et al.* [21] was using two horizontal electrodes on the sides of the eyes to detect the Winks, although the Winks in our case are detected from the two electrodes above the eye brows. In case of irregularities, the winks were the most difficult to correctly get identified, because they are not always purely implemented. For instance, some participants when instructed to either Wink Right or Left, would slightly accompany the other eye to the winking eye which made the classifier sometimes got confused if it was a Wink or a Squint.

As a proposal for investigation, the idea of adding two side electrodes along with the two forehead ones need to explored on different participants to check if more independent distinguishable features between the Wink Right, Wink Left and the Squints could be acquired or there are no huge differences. In general, the test need also to be conducted on more subjects to collect as many common features as possible and accordingly better robust classifiers would be designed.

The Rapid Blinks and the Long Blinks need to be trained independently especially after the tested improvements in the classifier as mentioned in 5.4, 5, since more features were defined for the Rapid Blinks to better recognize it from its frequency components range as well.

6.3.1.1 Sensors Independent Fixation

On the other side, it was observed that some subjects would do some actions such as the Winks or Squints strongly from the forehead that it would affect the readings of the other electrodes as they are all fixed-distanced together along the MUSE headband and thus they would move and cause perturbations or noises. Hence, it provides better results and performance if the EMG sensors (electrodes) were independently assembled and tested accordingly.

6.3.2 Potential Applications

On the application level, this work was originally dedicated to the Firefighters project at HCI Lab, University of Stuttgart to implement a hands-free interaction via the MUSE headband to

switch the views between the RGB, Thermal and depth cameras integrated with an Oculus RIFT device by performing different classified eye gestures. This prototypes a conceptual device comprising the Augmented Reality (AR) technology with the mentioned cameras embedded, in addition to the integration of the EMG electrodes under the AR device. The Oculus RIFT prototype was planned to be tested as an application in this project, however the system was not yet ready. Additionally, it was also preferred to test the performance of the EMG Signals from the MUSE headband as well as the classifier without any external disturbances, especially that the prototype of the Oculus RIFT was heavy and tight over the headband, thus it would affect currently the collected data.

The Muse headband had also 3 axis accelerometers which were also read and tested as raw data. They would also be exploited in a sense that would decrease the wrong execution of actions especially the blinks and the Squints, in case the user had unintentionally done the gesture while walking, moving or running. For instance, if the accelerometers detected that the user was standing or in stationary mode for a moment, it would allow the system to work and classify the gestures as intended actions. It is illogical that the user would be intentionally implementing a command gesture while moving his head due to any reason. Hence, the combination of the accelerometer to this application works as an Enable/Disable to the system.

BIBLIOGRAPHY

[1] US National Library of Medicine Medical Subject Headings (MeSH), *Electromyography*, 2016. [Online]. Available: https://meshb.nlm.nih.gov/record/ui?name=Electromyography (cit. on p. 13).

[2] Y. Mangukiya, B. Purohit, K. George, "Electromyography(EMG) sensor controlled assistive orthotic robotic arm for forearm movement," in *2017 IEEE Sensors Appl. Symp.*, IEEE, 2017, pp. 1–4, ISBN: 978-1-5090-3202-0. DOI: 10.1109/SAS.2017.7894065. [Online]. Available: http://ieeexplore.ieee.org/document/7894065/ (cit. on p. 13).

[3] M. V. Bennett, I. Matthews, "Life-saving uncooled IR camera for use in firefighting applications," *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 2744, pp. 549–554, 1996. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1018602 (cit. on p. 13).

[4] C. Wu, S. Feng, K. Li, S. Pan, J. Su, W. Jin, "Helmet-mounted uncooled FPA camera for use in firefighting applications," *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 4077, pp. 234–237, 2000, ISSN: 0277786X. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=911275 (cit. on p. 13).

[5] S. Evangelista, *Augmented reality for firefighters*. [Online]. Available: https://actu.epfl.ch/news/augmented-reality-for-firefighters/ (cit. on p. 14).

[6] J. K. Chapin, K. A. Moxon, R. S. Markowitz, M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nat. Neurosci.*, vol. 2, no. 7, pp. 664–670, 1999. [Online]. Available: http://www.nature.com/neuro/journal/v2/n7/abs/nn0799_664.html (cit. on p. 18).

[7] J. J. Vidal, "Toward direct brain-computer communication," *Annu. Rev. Biophys. Bioeng. 1973*, vol. 2, no. 1, pp. 157–180, 1973. [Online]. Available: http://www.annualreviews.org/doi/pdf/10.1146/annurev.bb.02.060173.001105 (cit. on p. 18).

[8] J van Erp, F. Lotte, M Tangermann, "Brain-Computer Interfaces: Beyond Medical Applications," *Computer (Long. Beach. Calif).*, vol. 45, no. 4, pp. 26–34, 2012, ISSN: 0018-9162. DOI: 10.1109/MC.2012.107 (cit. on pp. 18, 20).

[9] K. Holewa, A. Nawrocka, "Emotiv EPOC neuroheadset in brain - Computer interface," *Proc. 2014 15th Int. Carpathian Control Conf. ICCC 2014*, pp. 149–152, 2014. DOI: 10.1109/CarpathianCC.2014.6843587 (cit. on p. 19).

[10] J. R. Wolpaw, D. J. McFarland, T. M. Vaughan, "Brain-computer interface research at the Wadsworth Center," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 222–226, 2000, ISSN: 10636528. DOI: 10.1109/86.847823 (cit. on pp. 19, 20).

[11] C. A. Kothe, *Introduction to Modern Brain-Computer Interface Design*, 2013. [Online]. Available: ftp://sccn.ucsd.edu/pub/bcilab/lectures/01_Introduction.pdf (cit. on pp. 19, 20).

[12] J. d. R. Carlson, Tom and Millan, "Brain-controlled wheelchairs: a robotic architecture," *IEEE Robot. Autom. Mag.*, vol. 20, no. 1, pp. 65–73, 2013 (cit. on p. 19).

[13] B. Rebsamen, C. Guan, H. Zhang, C. Wang, C. Teo, M. H. Ang, E. Burdet, "A brain controlled wheelchair to navigate in familiar environments," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, pp. 590–598, 2010, ISSN: 15344320. DOI: 10.1109/TNSRE.2010.2049862 (cit. on p. 19).

[14] L. A. Farwell, E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, 1988, ISSN: 00134694. DOI: 10.1016/0013-4694(88)90149-6 (cit. on p. 19).

[15] P. Manoilov, "Eye-blinking artefacts analysis," *Proc. 2007 Int. Conf. Comput. Syst. Technol. - CompSysTech '07*, p. 1, 2007. DOI: 10.1145/1330598.1330654. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1330598.1330654 (cit. on p. 21).

[16] C. E. Reyes, J. L. C. Rugayan, C. Jason, G. Rullan, C. M. Oppus, G. L. Tangonan, "A study on ocular and facial muscle artifacts in EEG signals for BCI applications," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, 2012, ISSN: 2159-3442. DOI: 10.1109/TENCON.2012.6412241 (cit. on p. 21).

[17] B. Chambayil, R. Singla, R. Jha, "Virtual keyboard BCI using Eye blinks in EEG," *2010 IEEE 6th Int. Conf. Wirel. Mob. Comput. Netw. Commun. WiMob'2010*, pp. 466–470, 2010. DOI: 10.1109/WIMOB.2010.5645025 (cit. on pp. 22, 23, 44).

[18] D. Valeriani, A. Matran-Fernandez, "Towards a wearable device for controlling a smartphone with eye winks," *2015 7th Comput. Sci. Electron. Eng. Conf. CEEC 2015 - Conf. Proc.*, pp. 41–46, 2015. DOI: 10.1109/CEEC.2015.7332697 (cit. on pp. 23, 24, 45).

[19] E. Missimer, M. Betke, "Blink and wink detection for mouse pointer control," *Proc. 3rd Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA '10*, p. 1, 2010. DOI: 10.1145/1839294.1839322. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1839294.1839322 (cit. on pp. 24, 25).

[20] K. George, A. Iniguez, H. Donze, "Automated sensing, interpretation and conversion of facial and mental expressions into text acronyms using brain-computer interface technology," *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, pp. 1247–1250, 2014, ISSN: 10915281. DOI: 10.1109/I2MTC.2014.6860944 (cit. on pp. 25–27).

[21] J. Ma, Y. Zhang, A. Cichocki, F. Matsuno, "A novel EOG/EEG hybrid human-machine interface adopting eye movements and ERPs: Application to robot control," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 3, pp. 876–889, 2015, ISSN: 15582531. DOI: 10.1109/TBME.2014.2369483 (cit. on pp. 27, 28, 34, 43–45, 58, 77–80, 86, 88).

[22] J. Amores, X. Benavides, P. Maes, "PsychicVR: Increasing Mindfulness by Using Virtual Reality and Brain Computer Interfaces," *Proc. 2016 CHI Conf. Ext. Abstr. Hum. Factors Comput. Syst.*, p. 2, 2016. DOI: 10.1145/2851581.2889442. [Online]. Available: http://doi.acm.org/10.1145/2851581.2889442 (cit. on p. 29).

[23] J. E. Muñoz, T Paulino, H Vasanth, K Baras, "PhysioVR: A novel mobile virtual reality framework for physiological computing," *2016 IEEE 18th Int. Conf. e-Health Networking, Appl. Serv.*, no. September, pp. 1–6, 2016. DOI: 10.1109/HealthCom.2016.7749512 (cit. on p. 30).

[24] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 03, p. 193, 2005, ISSN: 1355-7718. DOI: 10.1017/S1355771805000932 (cit. on p. 40).

[25] National Instruments, *Getting Started With the NI OSC Library For LabVIEW*, 2013. [Online]. Available: http://www.ni.com/white-paper/14398/en/ (cit. on p. 41).

[26] R. A. Bailey, "More about Latin Squares," in *Des. Comp. Exp.* Cambridge University Press, 2008, ch. 9, pp. 152–161, ISBN: 978-0-521-68357-9. [Online]. Available: http://www.maths.qmul.ac.uk/$\sim$rab/DOEbook/ (cit. on p. 44).

[27] M. Nakanishi, Y. Mitsukura, Y. Wang, Y.-T. Wang, T.-P. Jung, "Online Voluntary Eye Blink Detection using Electrooculogram," *IEICE Proceeding Ser.*, vol. 1, no. October, pp. 114–117, 2014, ISSN: 2188-5079. DOI: 10.15248/proc.1.114. [Online]. Available: http://www.proceeding.ieice.org/summary.php?id=p1_0_114&year= (cit. on pp. 44, 78).

[28] M. H. M. Noor, Z Salcic, K. I.-k. Wang, "Dynamic Sliding Window Method for Physical Activity Recognition Using a Single Tri-axial Accelerometer," pp. 102–107, 2015 (cit. on p. 44).

[29] J. H. Skotte, J. K. Nøjgaard, L. V. Jørgensen, K. B. Christensen, G. Sjøgaard, "Eye blink frequency during different computer tasks quantified by electrooculography," *Eur. J. Appl. Physiol.*, vol. 99, no. 2, pp. 113–119, 2007, ISSN: 14396319. DOI: 10.1007/s00421-006-0322-6 (cit. on p. 45).

[30] A. Królak, P. Strumiłło, "Eye-blink detection system for human-computer interaction," *Univers. Access Inf. Soc.*, vol. 11, no. 4, pp. 409–419, 2012, ISSN: 16155289. DOI: 10.1007/s10209-011-0256-6 (cit. on p. 45).

[31] T. Morris, P. Blenkhorn, F. Zaidi, "Blink detection for real-time eye tracking," *J. Netw. Comput. Appl.*, vol. 25, no. 2, pp. 129–143, 2002, ISSN: 10848045. DOI: 10.1016/S1084-8045(02)90130-X (cit. on p. 45).

[32] S. W. Yang, C. S. Lin, S. K. Lin, C. H. Lee, "Design of virtual keyboard using blink control method for the severely disabled," *Comput. Methods Programs Biomed.*, vol. 111, no. 2, pp. 410–418, 2013, ISSN: 01692607. DOI: 10.1016/j.cmpb.2013.04.012. [Online]. Available: http://dx.doi.org/10.1016/j.cmpb.2013.04.012 (cit. on p. 45).

[33] National Instruments, *From Raw Data to Engineering Results: The NI Technical Data Management Solution (TDMS)*, 2015. [Online]. Available: http://www.ni.com/white-paper/7389/en/ (cit. on p. 52).

[34] ——, *NI OSC Library for LabVIEW - National Instruments*. [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/211428 (cit. on p. 56).

[35] F. Provost, R. Kohavi, "On Applied Research in Machine Learning," *Mach. Learn.*, vol. 30, no. 2/3, pp. 127–132, 1998, ISSN: 08856125. DOI: 10.1023/A:1007442505281. [Online]. Available: http://link.springer.com/10.1023/A:1007442505281 (cit. on p. 69).

[36] S. Visa, B. Ramsay, A. Ralescu, E. Van Der Knaap, "Confusion matrix-based feature selection," *CEUR Workshop Proc.*, vol. 710, pp. 120–127, 2011, ISSN: 16130073 (cit. on p. 69).

[37] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006, ISSN: 01678655. DOI: http://dx.doi.org/10.1016/j.patrec.2005.10.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786550500303X (cit. on p. 70).

[38] Tom, *Grabbing data from MuseIO: A few simple examples of Muse OSC Servers - Muse Developers*, 2015. [Online]. Available: http://developer.choosemuse.com/research-tools-example/grabbing-data-from-museio-a-few-simple-examples-of-muse-osc-servers (cit. on p. 87).

All links were last followed on August 16, 2017.