NTNU
Norwegian University of
Science and Technology

Master's thesis

2021

Benjamin Fimreite

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Benjamin Fimreite

Utilizing Electromyography (EMG) as an Input Modality for Head-Mounted Displays (HMDs)

June 2021

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Utilizing Electromyography (EMG) as an Input Modality for Head-Mounted Displays (HMDs)

Benjamin Fimreite

Master of Science in Informatics
Submission date: June 2021
Supervisor: Michail Giannakos
Co-supervisor: Evangelos Niforatos

Norwegian University of Science and Technology
Department of Computer Science

Summary

Head-mounted displays are part of a rapidly expanding field of technology. Many companies have tried to commercialise so-called smart glasses, providing the wearer a heads-up display by superimposing an image on top of the real world. However, no effective method for interacting with such devices have been found. Many solutions have been attempted, from touch surfaces and controllers to eye tracking and camera tracking. These solutions suffer from being too conspicuous, leading to low social acceptability and thus low mainstream acceptance.

This study investigates whether nasal electromyography is viable as an input modality for head-mounted displays. By attaching inconspicuous electrodes to the bridge of the nose utilising the subtle movement of nasal flares, a prototype consisting of both hardware and software is developed to explore this idea.

To identify the viability of this approach, the prototype is compared against an established baseline consisting of a side-mounted touchpad. A user study is performed to collect objective and subjective data, which can be analysed to find an answer.

In its current implementation, nasal electromyography is not found to be a viable solution. The study does the groundwork, and suggests what to improve to further explore nasal electromyography as an input modality.

Sammendrag

Hodemonterte skjermer er en del av et hurtig voksende felt innenfor teknologi. Mange selskap har prøvd å kommersialisere såkalte smartbriller som tilbyr brukeren et heads-up display ved å overlappe et bilde over den virkelig verden. Ingen effektiv løsning for å interagere med slike enheter har dog blitt funnet. Mange løsninger har blitt forsøkt, fra berøringsflater og kontrollere til øyesporing og kamerasporing. Disse løsningene lider av å være for iøynefallende, som leder til lav sosial aksept og dermed lav akseptanse blant folk flest.

Dette studiet undersøker hvorvidt neseelektromyografi er en levedyktig inngangsmodalitet for hodemonterte skjermer. En prototype bestående av både maskinvare og programvare ble utviklet for å utforske denne ideen ved å feste lite iønefallende elektroder på nesebroen og å utnytte de subtile bevegelsene utført ved å blusse opp neseborene.

For å identifisere levedyktigheten til denne tilnærmingen blir prototypen sammenlignet med en etablert grunnlinje bestående av en berøringsflate montert på siden av smartbrillene. En brukertest blir gjennomført for å samle inn både objektive og subjektive data som kan analyseres for å svare på spørsmålet.

I dens nåværende implementasjon ble ikke neseelektromyografi funnet å være en levedyktig løsning. Dette studiet gjør grunnarbeidet og foreslår hva som kan forbedres for å utforske neseelektromyografi som en inngangsmodalitet videre.

Acknowledgement

This thesis was written as the final part of my degree in the Master of Science in Informatics programme at the Norwegian University of Science and Technology (NTNU), Trondheim in 2020/2021.

First and foremost, I would like to thank my supervisors, Michail Giannakos and Evangelos Niforatos. They have consistently provided thorough guidance and direction, in addition to the necessary hardware.

I would also like to thank the voluntary participants for the user study, without which this study could not have been finalised.

Finally, I would like to mention that this past year has been particularly rough for many people due to the ongoing COVID-19 pandemic. The support of my fiancée Marte and our son Felix and their ability to always put a smile on my face, no matter how difficult times are have been very valuable during this project.

Contents

List of Figures

List of Tables

List of Acronyms

ANN Artificial Neural Network. 5, 6

AR Augmented Reality. 4, 11

BCI Brain-Computer Interface. 11

CAD Computer-Aided Design. 10

CNN Convolutional Neural Network. 6, 9, 13, 15, 33

ECG Electrocardiography. 10, 11

EEG Electroencephalography. 8–11, 13

EMG Electromyography. 1, 3, 7–13, 20–22, 26–28, 32, 33

EOG Electrooculography. 7–9

HCI Human-Computer Interaction. 3, 4, 21

HMD Head-mounted display. 1, 3, 4, 6, 8, 24, 33

HUD Heads-Up Display. 8, 11

IDE Integrated Development Environment. 10

LED Light Emitting Diode. 8

ML Machine Learning. 3, 5

MLP Multilayer Perceptron. 6, 13, 16

NCS Nerve Conducting Study. 3

OS Operating System. 19

PaaS Platform as a Service. 11

ReLU Rectified Linear Unit. 13, 15

SUS System Usability Scale. 25, 27, 29

SVM Support Vector Machine. 7, 9

TLX Task Load Index. 25, 27

UEQ User Experience Questionnaire. 25–27, 30

UI User Interface. 12, 13, 19–22, 33

VR Virtual Reality. 4

1 Introduction

In a rapidly evolving digital world, technology moves fast. Head-mounted displays (HMDs) are a product working its way towards mainstream acceptance. Interacting with a HMDs like smart glasses has for long been a tantalising problem with so far no effective solution. Current approaches utilise either touch surfaces on the side of the HMD, hand-held or worn controllers in the form of joysticks and rings. Some are testing eye-tracking as an input modality. However, Electromyography (EMG) — measuring the electric activity generated when muscles contract — may be a valuable alternative to efficiently providing input to HMDs.

1.1 Motivation

A new and more efficient input modality for HMDs could spearhead further development of the field by making it more socially acceptable to wear in public. This study is an exciting possibility to guide the way forward, placing itself at the leading edge of development.

1.2 Research Questions

This thesis tries to answer the following research questions:

RQ1: *Is nasal EMG an effective method for interacting with HMDs?*

RQ2: *How effective is nasal EMG for interacting with HMDs?*

RQ3: *How effective are the individual nasal gestures for interacting with HMDs?*

1.3 Contributions

This study contributes an implementation of a novel type of accessible input modality for wearable devices based on nasal EMG and deep neural network classification. Compared to existing input modalities aimed at wearable devices, the prototype designed in this study is novel in the sense that no other study found during literature review utilises the nasalis muscle. Additionally, most other products use unsubtle approaches consisting of large devices placed conspicuously on the body. This leads to low social acceptability. The prototype of this study attempts to remedy this by utilising small electrodes hidden in the frame of an HMD. It is also more accessible to people with reduced motor function, for instance the paralysed, than most other products. A final situation in which nasal EMG could prove especially useful is during high intensity working situations in which the user has their hands busy elsewhere while being provided instructions in an HMD.

1.4 Report outline

This thesis consists of 6 chapters, summarised in the following outline.

Chapter 1: Introduction

This chapter presents the thesis context and motivation and defines the research questions.

Chapter 2: Background

This chapter presents background theory and different definitions relevant to this study, as well as related works.

Chapter 3: System Design and Implementation

This chapter presents the prototype development, all the way from design to implementation.

Chapter 4: Methodology

This chapter describes and motivates the methods used in this study.

Chapter 5: Results

This chapter presents the results of this study.

Chapter 6: Conclusion

This chapter presents a conclusion to this study, as well as answers to the research questions. Finally, it presents suggestions for future work.

2 Background

This chapter presents some definitions relevant to this study, as well as some previous work. Section 2.1 presents some relevant definitions. Section 2.2 lists some existing studies related to this study.

2.1 Definitions

This section presents some definitions relevant to this study. Section 2.1.1 describes the field of EMG. Section 2.1.2 presents the anatomy of the human nasal muscles. Section2.1.3 provides an introduction to the use of HMDs. Section 2.1.4 describes the field of Human-Computer Interaction (HCI). Finally, Section 2.1.5 describes the field of Machine Learning (ML).

2.1.1 Electromyography

"Electromyography is the study of muscle function through the inquiry of the electrical signal the muscles emanate", as defined by Basmajian and De Luca. EMG is often used in clinical contexts to aid in diagnosing neuromuscular diseases, usually in combination with Nerve Conducting Studies (NCSs), measuring the conducting function of the nerves. EMG can be split into two main groups, intramuscular EMG and surface EMG. These are further described in the following sections.

In more recent years EMG has found several new use cases. One such use case is the control of prosthetic limbs, where an artificial device replacing a missing body part is controlled via surface EMG. Electrodes are placed on the remaining muscles of the original limb. Castellini and Van Der Smagt (2009) discuss advancements in this field. In combination with ML, they find improvements in accuracy in advanced hand prosthetics.

2.1.1.1 Intramuscular Electromyography

Intramuscular EMG involves placing needles in the muscles. There can be one needle inserted through the skin into a muscle with one surface electrode as reference, or two needles inserted, referenced to each other. Because needles are placed directly in desired muscle, intramuscular EMG provide highly accurate readings, are able to differentiate the signals from the desired muscles to adjacent muscles, and are not limited to superficial muscles.

2.1.1.2 Surface Electromyography

Surface EMG involves placing electrodes on the skin near the muscles. The electrodes can either be a pair, or a more complex network. At least a pair is needed, as surface EMG involves measuring the potential difference between the two. Surface EMG was selected for this study for its ease of use, while maintaining enough signal sensitivity.

2.1.2 Nasal Muscles

The nose is a part of the complex human facial anatomy. There are seven muscles in the nose, as seen in Figure 1 (Sugawara) and in Letourneau and Daniel (1988). Bruintjes et al. (1996) perform surface EMG on six of these muscles, showing how the muscles activate both during breathing and voluntary facial gestures. Of the gestures tested, they find that flaring shows the highest EMG-activity. Flaring activates four of the six tested muscles more than 50% of their maximum, the maximum being measured when grimacing.

For this study, the transverse part of the nasalis muscle was selected for a few main reasons. Firstly, it is a comparatively large nasal muscle. This makes electrode placement easier, and makes fitting the electrodes in the frame of the HMD feasible. It also provides good signal strength. Secondly, the muscle is not shown to be particularly active during breathing, reducing the risk of involuntary activation. Finally, and most importantly, this is a muscle many humans can control by flaring their nose. This is essential for the prototype to be useful as an input device.

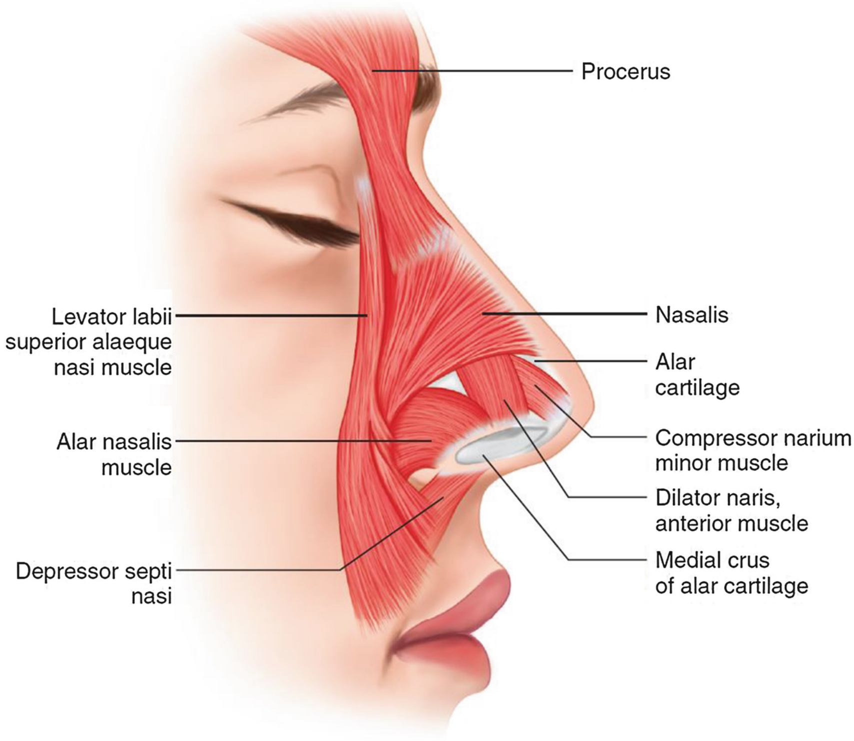

Figure 1: Nasal muscles

2.1.3 Head-Mounted Displays

HMDs are display devices worn on the head via a headband or a helmet. They typically consist of one or two small displays which when mounted to the head are placed directly in front of the eyes. They typically also include some form of tracking, allowing the user to navigate through a virtual space by moving their head. HMDs have had their use in the military since the 1960's (Feddersen 1962), displaying tactical information, often in aircraft or helicopters. When the technology advanced, it also found use medically (Birkfellner et al. 2002), where Augmented Reality (AR), in which computer generated images are drawn on top of the real world, has be utilised as an aid during operations. Even more recently, HMDs have be commercialised to the public, here usually for gaming or video. When gaming, a Virtual Reality (VR)-HMD is generally used. These HMDs provide a larger field of view, while also completely obscuring the real world, allowing the user to fully immerse themselves in the game world. AR is also used commercially, but is still in the early stages of mainstream adoption.

2.1.4 Human-Computer Interaction

HCI is the design and use of interfaces between humans and computers. The goal is to design effective, efficient, easy and enjoyable interfaces, such that the user can have the best possible time when interacting with computers. HCI is a multidisciplinary field, involving computer science, human psychology, design, and more.

2.1.5 Machine Learning

ML is a subset of artificial intelligence. In cases where it is challenging for a human to manually create the algorithms needed to solve a problem, it can be beneficial to rather program the computer to learn the required steps itself. To achieve this, the computer is taught how to build a mathematical model based on sample data in order to make predictions or decisions without being explicitly programmed to perform the task (Zhang 2020). The sample data, often called the training set, consists of labelled or unlabelled previously observed data, based on the approach taken. The three main approaches are supervised learning, unsupervised learning, and reinforcement learning. Reinforcement learning is mainly concerned with how the computer can best learn which actions to take in order to maximise its rewards in a given environment. This is not relevant to this study, and as such will not be further explained. Supervised learning and unsupervised learning, further explained below, were both considered. Ultimately, supervised learning was selected as the problem is best described as a classification problem.

Supervised Learning

Supervised learning is characterised by the training data being labelled. This involves the programmer telling the computer what the desired output given an input should be. The goal then is to learn the general rule mapping the input to this desired output. The computer makes predictions, and is corrected when giving wrong answers. This is done iteratively, until the computer achieves a satisfactory prediction accuracy. Supervised learning solves two categories of problems well; classification problems, in which the input is to be mapped to one of a limited set of output values, and regression problems, in which the input is to be mapped to any numerical value within a continuous range. One disadvantage with supervised learning is that it generally requires a large set of training data in order for the computer to learn how to generalise well.

Unsupervised Learning

Unsupervised learning is characterised by the training data not being labelled. The computer is then forced to learn the underlying structure of the data. This is useful to discover hidden groupings within the data. Unsupervised learning solves two categories of problems well; clustering problems, in which the data is grouped into clusters based on similarity, and association problems, where the goal is to discover rules defining the relationship between the items in the data set. The benefits of unsupervised learning when comparing it to supervised learning is that generating a large set of training data is simpler as there is no need to label the data. Also, unsupervised learning is useful in the cases where the programmer themselves don't know which classes are relevant.

2.1.5.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a subset of ML and are meant to simulate the biological neural networks of animal brains (Jain et al. 1996). ANNs are made up of processing units, or neurons, each taking some input and providing an output. The output is calculated by taking the weighted sum of all the inputs and adding a bias, before running the result through one of many possible activation functions. These neurons are typically organised into layers. The first layer, or input layer, takes input in the form of external data, such as from an image. Following this are typically one or more hidden layers, which take input from some or all of the neurons in the previous layer and propagates the output to the next layer. The final layer, or output layer, is responsible for answering the problem.

Training in an ANN consists of updating the weights of the connections between the neurons through a process called backpropagation. When the algorithm gives an answer, a loss function is used to calculate the cost, or how wrong it was. Backpropagation then uses the derivative, or gradient, of the loss function with respect to the weights to update the weights one layer at a time, starting at the output layer. When training, one has to be careful with overfitting. This is when the network learns the training data well but is not able to generalise it, leading to a low accuracy on the validation data.

ANNs also include a number of hyperparameters; parameters decided manually before training.

The network is not able to optimise these parameters itself. Hyperparameters include parameters like number of nodes in the hidden layers, number of hidden layers, learning rate, batch size and number of epochs. The learning rate decides how fast the network should update the weights. A high learning rate results in faster learning, but also introduces the possibility of not converging on a solution. A lower learning rate results in slower learning, but allows for smoother convergence. As such, a balance must be found. Batch size decides how many samples are given to the network between weight updates. The optimal number depends on the network and the training set, but some power of two is generally used. Number of epochs decides the number of times the entire data set is iterated over in total during training. Here, a higher number is generally better, but validation accuracy should be monitored to avoid overfitting.

2.1.5.2 Multilayer Perceptrons

Multilayer Perceptrons (MLPs) are a simple type of ANN containing an input layer, at least one hidden layer and one output layer. MLPs are characterised by being fully connected. This means that all nodes are connected to the output of all the nodes in the previous layer. MLPs are suitable for classification and regression problems because of their flexibility when learning how to map input to output.

2.1.5.3 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are another type of ANN. In addition to the fully connected layers of MLPs, CNNs include two additional main building blocks called convolutional layers and pooling layers. The convolutional layer is the core of CNNs. They consist of learnable parameters in the form of filters. These filters have a small receptive field, meaning they see only part of the input. The pooling layers are responsible for down-sampling the output of the previous layer, reducing the spatial size. This has several benefits. It reduces the number of parameters while also increasing generalisation and reducing overfitting. This comes at a very low cost, as knowing the exact location of a feature is generally unimportant compared to knowing its rough location relative to other features.

There are several benefits of using CNNs over MLPs. The biggest in general is the reduction of parameters (Albawi et al. 2017). In MLPs the number of parameters increase very quickly as more neurons are introduced because of the full connectivity. CNNs get around this issue thanks to the small reception field of the neurons. In addition to this, through a concept called parameter sharing, the network is able to share weights between certain neurons in the convolutional layer. This is possible because of the assumption that if a feature is useful to compute at a specific spatial region, then it probably is useful to compute the same feature across all other possible spatial regions (O'Shea and Nash 2015).

CNNs are typically used for image classification because of their ability to easily learn how to recognise whether an object is present in the image, regardless of where in the image it appears. However, they can also be adapted to work with one-dimensional input.

2.2 Related Work

This work presented in this section consists of different implementations of wearable input devices, designed to improve communication between humans and wearable devices like HMDs. In general, they utilise some number of sensors to recognise gestures of different parts of the body. A summary can be seen in table 1.

2.2.1 Argot

The *Argot* is a wearable, one-handed input device designed to let the user type without the use of a keyboard (Peshock et al. 2014). It achieves this through 15 conductive surfaces placed strategically on a glove which can be pressed together to enable multi-tap (Gong and Tarasewich 2005) typing.

2.2.2 Smart Wristband

The *Smart Wristband* is a wearable input device for smartglasses (Ham et al. 2014). It combines touch and motion by use of a touch screen panel and an inertial measurement unit to control smartglasses' interface accurately and quickly. Users in the study were able to use their finger on the touchpad to select objects and their wrist motion control the screen. The researchers note that in addition to this, five kinds of interactions have been implemented; point and click, navigation, program switch-over, zoom in and out, undo and redo.

2.2.3 Belt

The *Belt* is an unobtrusive input device for wearable displays (Dobbelstein et al. 2015). It incorporates a touch surface encircling the user's hip. The touch surface consists of ten general zones, which can be mapped to different actions such as opening an application. The study tries to find the social implications of such a device. It found that users were comfortable using the zones near their front pockets in public, and less comfortable using the zones behind their back or directly in front.

2.2.4 TIMMi

TIMMi is a textile input device for mobile interactions like smartphones and smartglasses (Yoon et al. 2015). It utilises the dexterity of the thumb and index fingers to implement 14 different interactions. The interactions are based on tapping on of three different spots on and bending the index finger. The study argues that this is a more socially acceptable input device, compared to e.g. a tactile input device mounted to the frame of the smartglasses.

2.2.5 zSense

zSense is a shallow depth gesture recognition system intended to provide input to smart wearables (Withana et al. 2015). It utilises infrared sensors and emitters to recognise gestures, and reports 94.8% gesture recognition accuracy across all configurations. zSense uses a software classifier, consisting of two SVMs and a Bayes network, to first estimate finger location before determining the exact gestures.

2.2.6 Itchy Nose

Itchy Nose is a wearable input device consisting of EOG sensors placed around the nose (Lee et al. 2018), supporting subtle and discreet interactions. It allows for personalised per user training, using a random decision forest classifier. The study proposes three different input gestures, but does not mention any results of a user test.

2.2.7 Make-a-Face

Make-a-Face is a wearable device which utilises EMG sensors on the lower half of the face, allowing users to interact with a computer system using their mouth, tongue or cheeks (Nakao et al. 2018). It uses logistic regression to classify the extracted features into five classes.

2.2.8 Earable TEMPO

The *earable TEMPO* is an earphone-type interface developed for hands-free interaction, designed to be operated by pushing the tongue against the roof of the mouth (Taniguchi et al. 2018). It uses a combination of a Light Emitting Diodes (LEDs) and a phototransistor to measure shape variations in the ear when the user presses their tongue against the roof of their mouth. The earable TEMPO shows high accuracy and precision, but when walking the average recall dropped to 48%, signifying a high degree of false positives.

2.2.9 Motion Capture Glove

The *Motion Capture Glove* is a glove utilising conductive textile and a motion sensor to recognise hand gestures(Yasuhiro and Lei 2018). The conductive textile allows for detecting both bending and touching of the gloves. They are detected by reading voltages when bending the index finger, or when touching the thumb to one of four pre-defined areas on the index- and middle-fingers. The system reached an overall accuracy of 93,88%.

2.2.10 Mind the Tap

Mind the Tap is a system which explores foot-based interaction with HMDs (Müller et al. 2019). Using optical tracking of the feet, the user can interact with a semicircular interaction grid visualised in the HoloLens AR-glasses. By testing different configurations of direct versus indirect visualisations, as well as number of rows and columns of buttons, the study concluded that foot-taps provide a viable interaction technique for HMDs. The results favoured dividing the buttons into columns over rows, as it provided better accuracy.

2.2.11 Playing games with your mouth

In *Playing games with your mouth*, the researchers create an interface for controlling video games with the mouth using EMG (Liao et al. 2019). Rather than being designed to replace the mouse and keyboard, it is designed to complement it. In a user study, EMG-electrodes were attached to the users' cheeks, and they were asked to bite left or right when asked to lean left or right respectively in a test program. A group of inexperienced players expressed the opinion that EMG made it more fun to play, while a group of experienced players generally preferred mouse and keyboard. Both groups reached a lower average response time.

2.2.12 EEGlass

The *EEGlass* is an EEG-eyeware prototype (Vourvopoulos et al. 2019). In the study, the researchers investigate EEG-electrodes in an HMD form factor as an everyday alternative to established EEG systems. They find that the *EEGlass* demonstrates variations in frequency and amplitude similar to the Enobio 8, an established EEG system, only significantly differing in brain activity linked to upper limb motor-action due to electrode placement. The *EEGlass* is also able to use EOG-signals to detect eye movements in four basic directions, potentially useful for navigating Heads-Up Displays (HUDs).

Reference	**Input**	**Used sensor(s)**	**Number of recognised gestures**	**Classifier**
Peshock et al. 2014	Hand (Glove)	Magnetic Conductive Contacts	15[1]	-
Ham et al. 2014	Wrist (Wristband)	Touch Screen Panel, Inertial Measurement Unit	5	-
Dobbelstein et al. 2015	Hip (Belt)	Touch Surface	∼10	-
Yoon et al. 2015	Thumb and Index Fingers	Piezoresistive Fabric	14	Two-Phase- and Polynomial Regression
Withana et al. 2015	Hand (Gestures)	Infrared Sensors and Emitters	9[2]	SVMs and Bayes Network
Lee et al. 2018	Nose	EOG	3	Random Decision Forest
Nakao et al. 2018	Mouth	EMG	5	Logistic Regression
Taniguchi et al. 2018	Ear Canal/Roof of Mouth	Earphone-type Optical Sensor	1	Correlation Coefficient with Ground Truth
Yasuhiro and Lei 2018	Hand (Glove)	Inertial and Conductive	9	Dynamic Time Warping
Müller et al. 2019	Foot	Optical tracking	18[3]	SVMs
Liao et al. 2019	Cheeks	EMG	3	-
Vourvopoulos et al. 2019	Skull	EEG	4	-
This thesis	Nose	EMG	3	CNN

Table 1: Summary of related work in wearable input devices

[1] 15 conductive surfaces combines to simulate a full keyboard
[2] In the experiment containing the highest amount of gestures
[3] Max, practical limit is lower

3 System Design and Implementation

This chapter presents the system design and implementation of the prototype. Section 3.1 is included to give a short description of the development tools and why they were chosen. Section 3.2 details the hardware used in the study, while Section 3.3 presents the software developed.

3.1 Resources

The following tools and libraries were selected to aid development, in some cases due to their necessity, in others because of their convenience in regards to availability and ease of use.

3.1.1 FreeCAD

FreeCAD is a free and open-source 3D parametric modelling application. It is used for Computer-Aided Design (CAD) modelling, wherein a designer is aided by a computer program when designing models.

3.1.2 Ultimaker S5

The Ultimaker S5 is a 3D printer; a machine able to construct functional real world prototypes of 3D models. 3D printers typically use synthetic polymers like plastic. The Ultimaker S5 includes dual extrusion, allowing the use of two materials in one print.

3.1.3 PyTorch

PyTorch is a Python machine learning library designed to combine usability and speed, providing a Pythonic programming style and hardware accelerated tensor computations (Paszke et al. 2019). Other choices include libraries like Tensorflow and Keras, but PyTorch was selected due to a mix of familiarity, its focus on fast training and easy debugging, and detailed documentation.

3.1.4 BrainFlow

BrainFlow is a library intended to obtain, parse and analyse EEG, EMG, Electrocardiography (ECG) and other kinds of data from biosensors (Parfenov 2021).

3.1.5 PyBluez

PyBluez is a Python extension module allowing access to system Bluetooth resources (Haung 2019). As such, it enables the creation of a Bluetooth connection between a machine and the smartglasses.

3.1.6 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for developing native Android applications. It was built by Google in cooperation with JetBrains (Ducrohet et al. 2013).

3.1.7 Node.js

Node.js is a Javascript runtime, allowing Javascript to be executed outside of a web browser. It is typically used in back-end programming, creating dynamic content server-side before sending it to a browser (Surhone et al. 2010).

3.1.8 MongoDB

MongoDB is a document-oriented NoSQL database. This entails data being stored as JSON-like documents, which has the advantage of being both easily human readable and writable, while also being easily computer parsed and generated. MongoDB provides a cloud service, allowing access across devices.

3.1.9 Heroku

Heroku is a Platform as a Service (PaaS) for deploying and running apps. A developer builds an application and deploys it to Heroku, which then serves the app and handles user requests.

3.2 Hardware

This section introduces the different hardware components of the prototype.

3.2.1 OpenBCI Ganglion

OpenBCI is an open-source programmable Brain-Computer Interface (BCI)-platform meant to make BCI affordable and available. OpenBCI specifically is made to perform EMG, EEG and ECG by measuring muscle, brain or cardiac electrical activity respectively. OpenBCI provides two different bio-sensing devices, the Ganglion and the Cyton bio-sensing boards. The main differences include number of channels and sampling rate, in addition to how they communicate with computers. The Cyton provides eight channels sampled at 250Hz. It utilises an RFDuino module to communicate with a USB-dongle plugged into the computer, intended to improve data rates compared to a standard Bluetooth connection. The Ganglion however provides only four channels sampled at 200Hz, not allowing for as complex electrode setups as the Cyton does. It also uses the standard Bluetooth 4.n protocol, allowing connection to any device supporting this protocol. A Bluetooth USB-dongle is still provided for compatibility with other devices. The sampling rate of both boards can be increased by attaching a WiFi Shield, transmitting data over WiFi to get past the limit set by the Bluetooth bandwidth.

The Ganglion was selected as the best fit for this study for two reasons. Firstly, no more than two channels would be needed. Secondly, using the standard Bluetooth protocol provides more flexibility and is easier to work with.

3.2.2 Vuzix Blade

Vuzix is a technology firm focused on the development and sale of wearable display technology, primarily AR. Their main product line, the M-Series glasses, are currently intended for enterprise use. Some solutions they provide include tele-medicine, where for example a paramedic can receive expert advice from a hospital-based doctor without pausing patient care, and manufacturing, where AR can help training employees. Vuzix also provides a different line of product, namely the Vuzix Blade. The Vuzix Blade looks like a regular pair of glasses and are designed for comfortable all-day use, both in enterprise and for prosumers. A prosumer is an individual who both produces and consumes, the term being a portmanteau of producer and consumer. Embedded in the glasses are a see-through colour screen, layering a HUD on top of the real world.

3.2.3 Electrodes

To gather nasal EMG signals, the prototype incorporates a number of gold-plated electrodes connected to the Ganglion board, which were placed on the skin of the nose. Before sessions a conductive gel had to be applied to the electrodes in order to improve skin connection. The electrodes then had to be cleaned when removed. During the earliest stages of development, electrodes were simply taped to the surface skin of the nasalis muscle using medical tape. This proved to be a cumbersome process when repeated often. In addition to being a time consuming process, this process provided some difficulties achieving a consistent electrode placement. Even slightly different electrode placements can result in different signals, making classification less reliable. A more practical solution was required.

FreeCAD was used to create a 3D model of a custom nose bridge. The model was realised using the Ultimaker S5. The nose bridge was attached to the Vuzix Blade screwing it in in the place of the included replaceable nose bridge. The custom nose bridge accommodated space for two electrodes, which were attached using double sided and medical tape. This ensured a consistent electrode placement between sessions.

The prototype saw two different major implementations of electrode placement. At first, two electrodes were placed on either side of the nose. The two electrode pairs each formed a channel, for a total of two channels. With this configuration, one second of data resulted in 400 individual data points, 200 from each channel. The physical placement of the electrodes made for a very crowded nose, so an alternate solution was explored. With a single electrode on each side of the nose together forming a single channel, the placement became more manageable. This came at no cost to signal quality, on the contrary it proved less noisy. As a side effect, design of the custom nose bridge later in the development was also simplified. The rest of this chapter assumes the final electrode placement.

Both configurations included one additional electrode, serving as ground. This electrode should be placed on an area with the least amount of muscle activity. During early testing, the ground electrode was placed anywhere convenient, like the elbow. In the final prototype, the ground electrode was manually placed on the mastoid part of the temporal bone, just behind the ear. This area was chosen because of its accessibility to a possible further developed prototype.

3.3 Software

This section describes the software of the prototype. The prototype consists of four software applications described in Table 2 and in the following sections.

Software	Description
The Classifier	The algorithm responsible for classifying the nasal EMG-signals
The UI	The application the user interacts with
The Connector	The application connecting the OpenBCI Ganglion, the Classifier and UI together
The Logger	The web-server application responsible for receiving messages and logging them to a database

Table 2: Description of the four software applications

3.3.1 Data Pre-Processing

Before being fed to the classifier, the electrode data was pre-processed to reduce noise and amplify the desired frequencies. This intends to extract the nasal flare signal from the data and with this improve the accuracy of the classifier. 200 data-points at a time were collected from the Ganglion, equivalent to one second of data. The 200Hz sample rate of the Ganglion also implies the maximum frequency obtainable by the device is 100Hz, found by reversing the Nyquist-Shannon sampling theorem (Shannon 1949). Following this, the data was sent through a series of algorithms. First, a high-pass filter with an 80Hz cutoff was applied. This removes the frequencies lower than 80Hz, including EEG-signals and other artefacts, like eye-blinking. This leaves frequencies in the $80-100$Hz range. It also removes the need of a band-stop filter at 50Hz, which would otherwise be needed to remove electrical interference from the power grid. A second order Butterworth filter was selected, providing a balance between the Chebyshev and Bessel filters and sufficient steepness at the frequency cutoff. Following the high-pass, a noise reduction operation was performed. The purpose of denoising is to smooth the remaining signals, removing outliers. The Brainflow-library provides a range of wavelet transforms to choose from. In a test comparing the different wavelets to each other some performed worse, but most wavelets performed very similarly. Of these, sym7 (Wasilewski 2021) was arbitrarily chosen.

3.3.2 Classifier

The classifier is a neural network, and is the central software component of the prototype. It is the component responsible for classifying input in the form of nasal EMG data to an action in the UI. The most important metric of the classifier is the accuracy; it has to be able to correctly infer the user's actions in order to provide the user a good experience. To achieve this, finding a structure of layers able to model the problem accurately. Additionally, other hyperparameters such as learning rate, batch size and number of epochs have to be optimised. This is an iterative process, which in this study lead to two major phases. The first approach involved a traditional MLP. Over time however, the network evolved into a CNN. These phases are both described in detail in the following sections.

3.3.2.1 First Version

Initially the classifier was developed with an MLP-like structure, involving four fully connected hidden layers. The network consisted of the 400 node input layer followed by 200 node, 100 node, 50 node and 20 node hidden layers, before culminating in a four node output layer. The four nodes of the output layer each correspond to either one of the three gestures presented in Section 3.4, or the no gesture-class. The node with the highest activation was selected as the correct class for the input data. The network used the Rectified Linear Unit (ReLU) activation function, both for the hidden layers and the output layer. This is the standard activation function of typical modern neural networks. This function takes in a number and returns the number if it is greater than 0, and 0 otherwise. This deals with the vanishing gradient problem other popular activation functions suffers from, in which the gradient is dramatically diminished as it propagates back through the multi-layered network (Goodfellow et al. 2016).

This implementation of the classifier suffered from severe overfitting. This is due to the small data-set available to the complex model. Some measures were implemented to combat this issue. First, the amount of hidden layers were reduced to two, each with 100 nodes. A less complex network does not have the same capacity to overfit. Then both batch normalisation and dropout were implemented. Batch normalisation is an operation performed between the other layers of the network. It involves normalising the output of the previous layer before feeding it into the next layer. It also has a regularisation effect, reducing generalisation error. Dropout also introduces regularisation by ignoring the output of some percentage of hidden nodes. This forces each node to learn to detect a feature on its own, preventing complex co-adaptations in which the nodes become dependent on several other nodes (Hinton et al. 2012).

A small section of the final structure of the MLP can be seen in Figure 2, at this point containing a 200 node input layer, two 100 node hidden layers and a 4 node output layer. Two batch normalisation layers were included following each hidden layer. Dropout was performed on 10% of of the nodes in the final hidden layer. While significantly better than the initial network, this version still suffered from overfitting. Figure 3 and Figure 4 show accuracy and loss respectively on both the training and validation sets. Accuracy reaching near 100% on the test set while plateauing at less than 70% on the validation set, together with loss on the validation set suddenly ballooning after 100 epochs is a clear sign of overfitting. Thus, a more drastic approach was required.

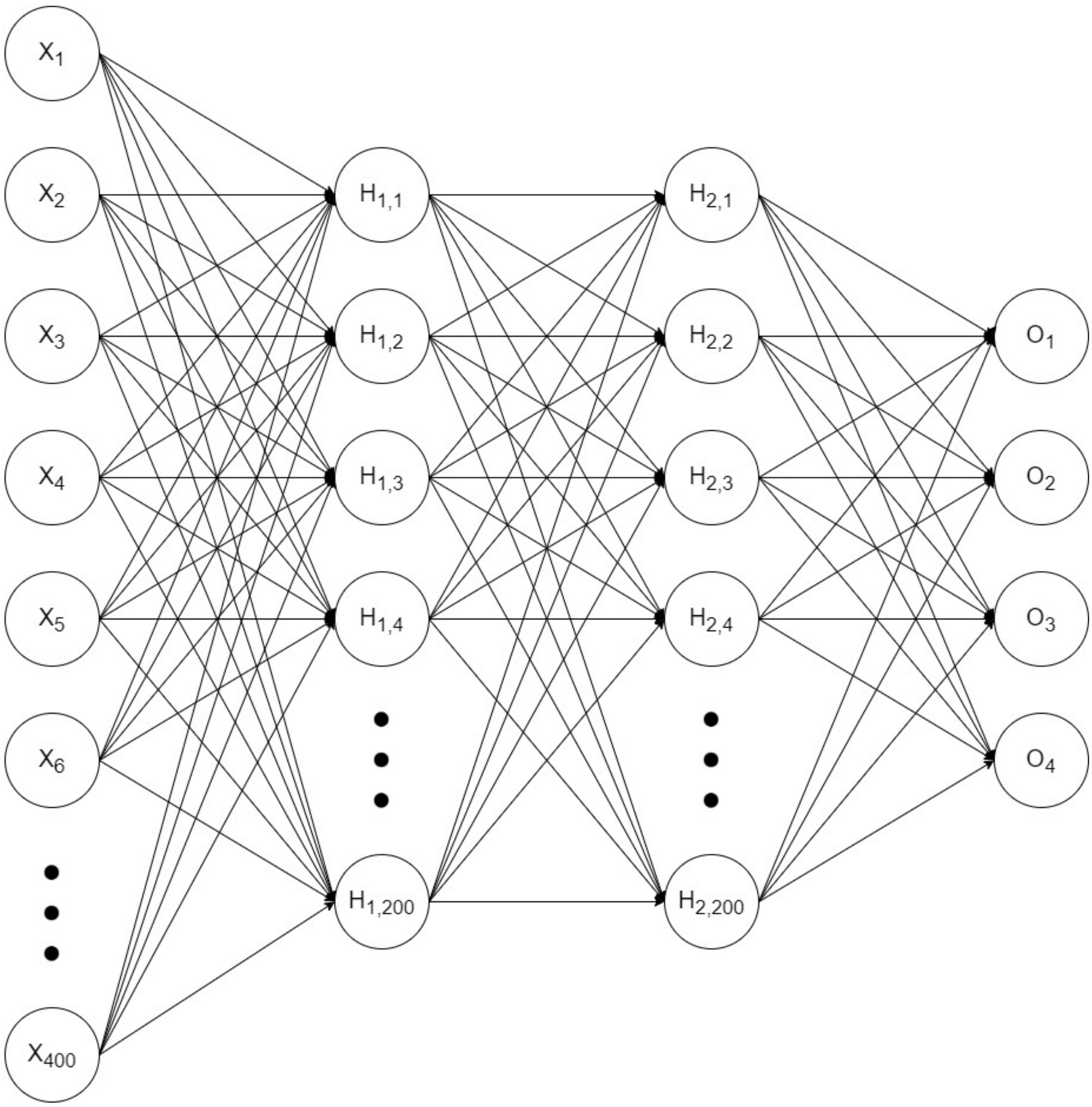

Figure 2: The final multilayer perceptron-architecture

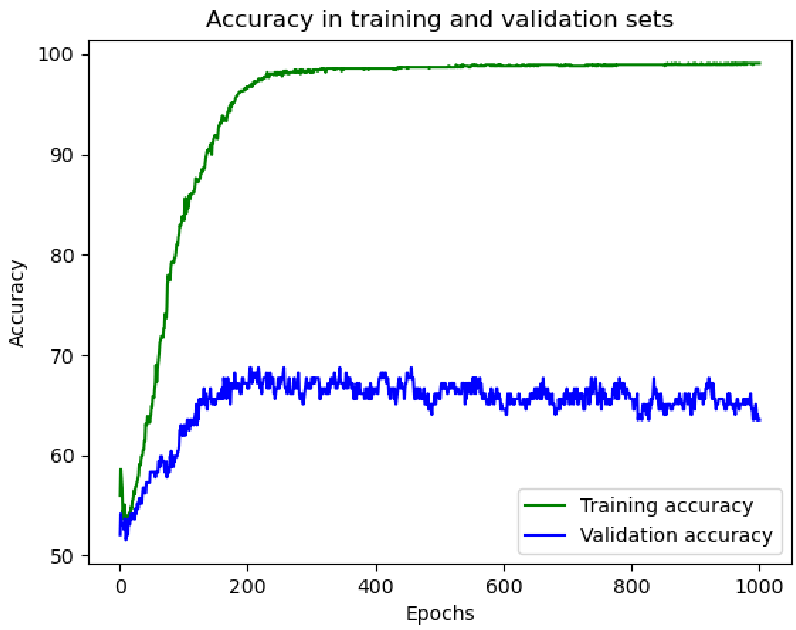

Figure 3: Accuracy in overfitting model

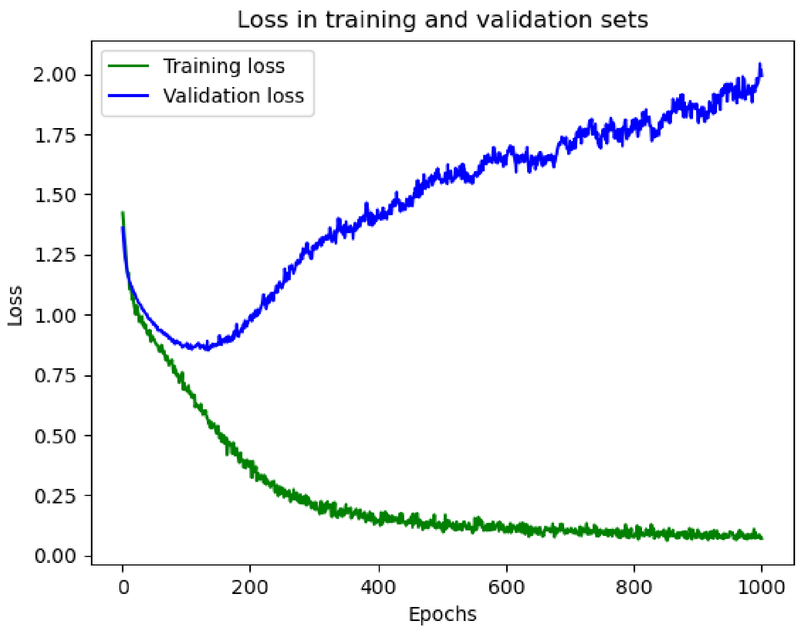

Figure 4: Loss in overfitting model

3.3.2.2 The Restructure

The CNN appeared as a viable solution due to its ability to typically generalise better, as mentioned in Section 2.1.5.3. As an initial attempt, two convolutional layers were added to the network using a kernel size of five, a stride of one and zero-padding of four. Both used the ReLU activation function and were followed by a max-pool layer with a kernel size of three and stride of three and a dropout layer dropping 30% of the outputs. Without padding, a kernel size of five with a stride of one on an input of length 200 would result in the outputs having a length of 196, using Equation

1 from the official PyTorch documentation (*Conv1d — Pytorch 1.9.0 documentation* 2021). When passed on to the max-pool layer, this would result in not every element being covered by the sliding pooling window. Therefore, zero-padding is added to the edges of the input before the convolution operation is performed. This alters the output shape, ensuring no information is lost. In this case, a padding of four would actually result in an output shape of 200, which is still affected by the aforementioned issue. This can be easily remedied by changing the padding amount. Following the convolutional layers and the associated operations were two fully connected layers, in this case with 480 input nodes, two hidden layers of 64 and 32 nodes respectively, and the same 4 node output layer as previously.

$$L_{out} = \lfloor \frac{L_{in} + 2 * padding - dilation * (kernel\_size - 1) - 1}{stride} + 1 \rfloor \quad (1)$$

This network already performed better than the final MLP-version, but through several iterations of small changes the network continued to improve. The architecture of the final version can be seen in Figure 5 and Figure 6. In addition to the altered structure, data augmentation was introduced to artificially increase the data set. Data augmentation involves repeating samples from the data set, each time with slight variations. In this case, each sample was repeated 20 times before rolling each element in the sample a random number of places to the left or right. Elements rolling beyond the boundaries were reintroduced at the other end.

This version of the network reached a peak accuracy on the validation set of 95%. Figure 7 and Figure 8 show how the accuracy and loss on the training and validation sets evolved over a 1000 epoch training session. When compared Figure 3 and Figure 4, we notice a significant improvement in accuracy. The validation loss still suggests some overfitting is happening, but to a much lower degree than previously. By stopping training after 100 epochs, this effect is less noticeable. It is also realistic to think the network could be further improved using more advanced techniques such as automated hyperparameter optimisation, as discussed by Feurer and Hutter (2019).

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
Net                                      --                        --
├─Conv1d: 1-1                            [128, 16, 200]            96
├─BatchNorm1d: 1-2                       [128, 16, 200]            32
├─ReLU: 1-3                              [128, 16, 200]            --
├─Dropout: 1-4                           [128, 16, 200]            --
├─Conv1d: 1-5                            [128, 16, 204]            1,296
├─BatchNorm1d: 1-6                       [128, 16, 204]            32
├─ReLU: 1-7                              [128, 16, 204]            --
├─MaxPool1d: 1-8                         [128, 16, 68]             --
├─Dropout: 1-9                           [128, 16, 68]             --
├─Conv1d: 1-10                           [128, 64, 68]             5,184
├─BatchNorm1d: 1-11                      [128, 64, 68]             128
├─ReLU: 1-12                             [128, 64, 68]             --
├─Dropout: 1-13                          [128, 64, 68]             --
├─Conv1d: 1-14                           [128, 64, 72]             20,544
├─BatchNorm1d: 1-15                      [128, 64, 72]             128
├─ReLU: 1-16                             [128, 64, 72]             --
├─MaxPool1d: 1-17                        [128, 64, 24]             --
├─Dropout: 1-18                          [128, 64, 24]             --
├─Linear: 1-19                           [128, 64]                 98,368
├─BatchNorm1d: 1-20                      [128, 64]                 128
├─ReLU: 1-21                             [128, 64]                 --
├─Dropout: 1-22                          [128, 64]                 --
├─Linear: 1-23                           [128, 4]                  260
==========================================================================================
```

Figure 5: The final convolutional neural network architecture

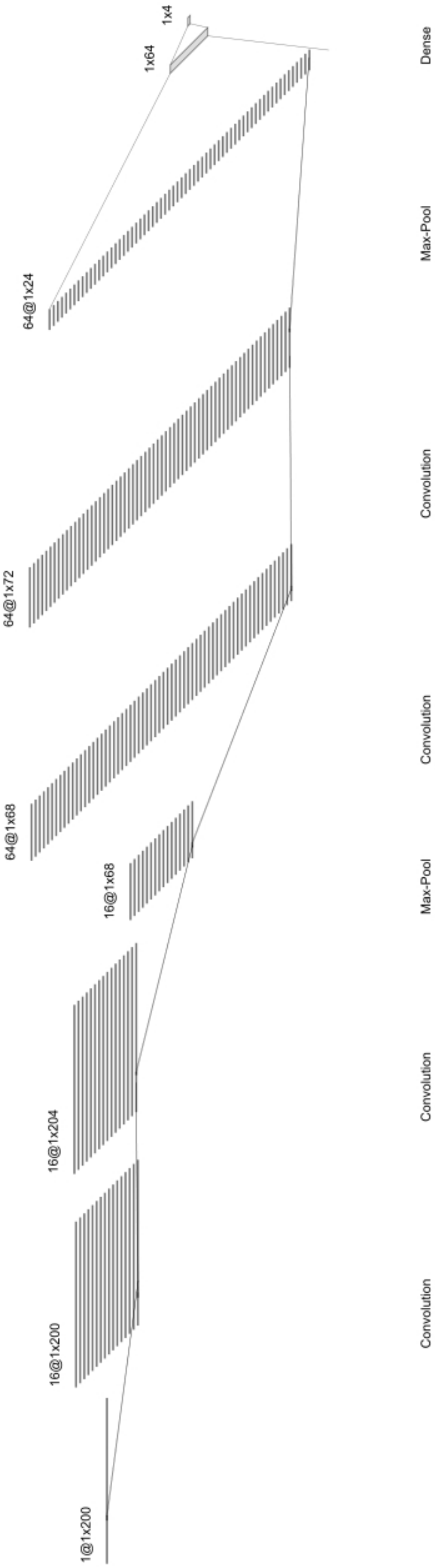

Figure 6: The final convolutional neural network architecture visualised

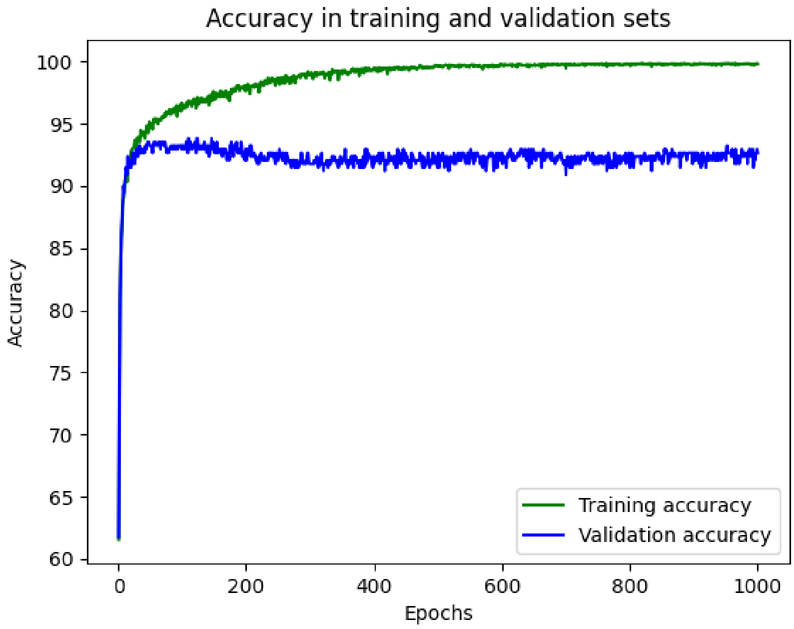

Figure 7: Accuracy in final model

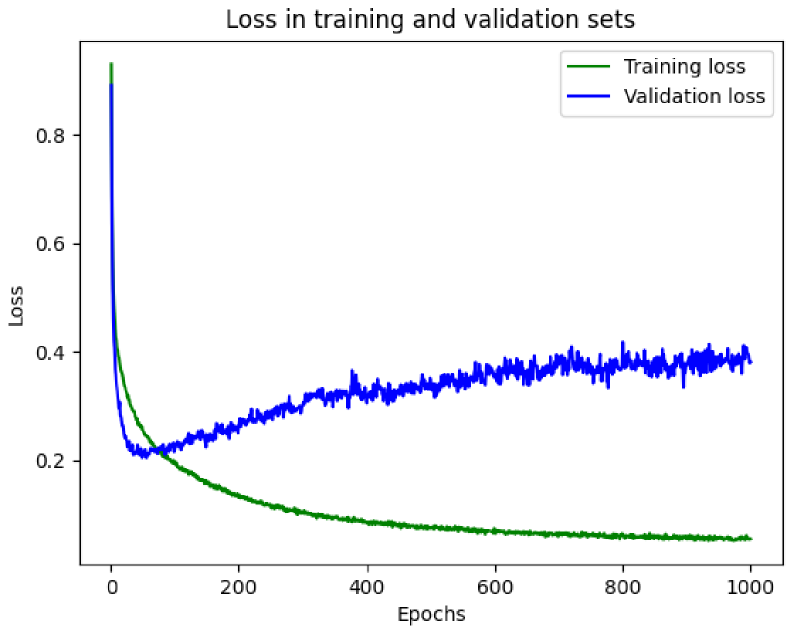

Figure 8: Loss in final model

3.3.3 User Interface

The UI the participant interacts with, running on the Vuzix Blade. The Vuzix Blade runs a customised version of the Android Operating System (OS). Vuzix provides a library making development similar to any other Android application. For the most part, the UI-application consists of a nested menu in which the user can scroll between the items, select an item to go to the next level of the menu or to click a button, or return to the previous level. The application also includes a customised way of connecting to a Bluetooth device, as well as a way to handle commands received over Bluetooth. This is used to connect to the Connector-application described in Section 3.3.4. Finally, the application provides a way to send a POST request message over HTTP. This is useful as it enables logging through the Logger web-server described in Section 3.3.5.

When opened, the application presents a main menu and waits for a Bluetooth connection. Once connected, it waits a Bluetooth command. The command received decides which task is started. The application contains a total of three different tasks. The tasks are described in Table 3. A timer is started together with the task. The application tracks the actions performed by the user so that once the correct steps are completed, the results can be sent to the logger. The application returns to the main menu and waits for the next command.

Task 1	Scroll to the end of the current active UI menu using the right action
Task 2	Navigate through the nested menu in the UI to find and press a button using the right action and the enter action
Task 3	Navigate through the nested menu in the UI to find and press two buttons in succession using the right action, the enter action and the back action

Table 3: Description of the experiment tasks

3.3.4 Connector

The Connector is the link between the OpenBCI Ganglion, the Classifier and the Vuzix Blade, binding it all together. Initially, this was meant to be presented in the form of a mobile Android-application. The idea was to have the prototype in its entirety, including both hardware and software, be mobile. This would enable testing the most realistic use case in which a user wears the prototype while going about their day-to-day life. However, the idea had to be scrapped late in development because of compatibility issues between the Brainflow library and Android. A complex algorithm was written in its place, interfacing directly with the Ganglion and parsing the data received. This seemed to work, but at the time of testing real-world results were poor, cascading to poor classification performance.

Instead, the Connector ended up as a hastily implemented command-line application running on a laptop. When started, this application automatically searches for and connects to the Vuzix Blade and the OpenBCI Ganglion. If successful, it waits for input telling it the user is ready to start. Following this it starts a data gathering session in which the user is asked to perform the nasal gestures. The purpose of this sessions is twofold. It allows the user to get familiar with the prototype, while also providing the researcher more nasal data which could end up being used to improve the Classifier.

Once this sessions is completed, the application again waits for input, this time telling it which task to start. It sends a command to the UI corresponding to the task selected. Depending on whether the task requires EMG enabled, it decides if it should start requesting data from the Ganglion. If yes, a loop running once every second is started, wherein the data is collected, pre-processed and classified using the Classifier. Depending on the prediction, a corresponding command is sent to the UI, telling it which action to perform.

3.3.5 Logger

The Logger is a web-server application which accepts HTTP-requests from the UI and stores the contents in a MongoDB database. As such, it can be considered a middleman between the front-end and the database. The Logger is built using Node.js. Using Express, a framework running on top of Node.js designed for building web applications and HTTP-APIs, a web-server is created. It then listens for requests to specific addresses. When a requests is sent to an address, the Logger handles it by parsing the body, or content, of the request. A JSON-object, as expected by MongoDB, is created based on the contents. This object is the sent through an established connection to a

MongoDB cloud database, where it is stored. A response is sent to the requester, depending on whether the request was successfully handled.

The Logger is deployed to Heroku, allowing a single access point to the database for all devices. This makes logging simple, as there is no need to be connected to the same network as the Logger.

3.4 Gestures

The system implements a set of nasal gestures providing different actions in the Vuzix Blade UI. In total, three actions were considered and implemented. When deciding on which actions to implement, feasibility was considered the most important. Actions which the user cannot reliably perform would provide a very poor user experience. In addition, the HCI concept of consistency was considered, in which the actions ideally should feel intuitive to the user with their previous experiences kept in mind.

Single nasal flare

The single nasal flare consist of a single rapid contraction of the nasalis muscle, lasting approximately 150 milliseconds. In the UI, this action is mapped to the scroll action, equivalent to a forward swipe on the touchpad. Figure 9 shows the EMG-signal of ten single nasal flares after being pre-processed.

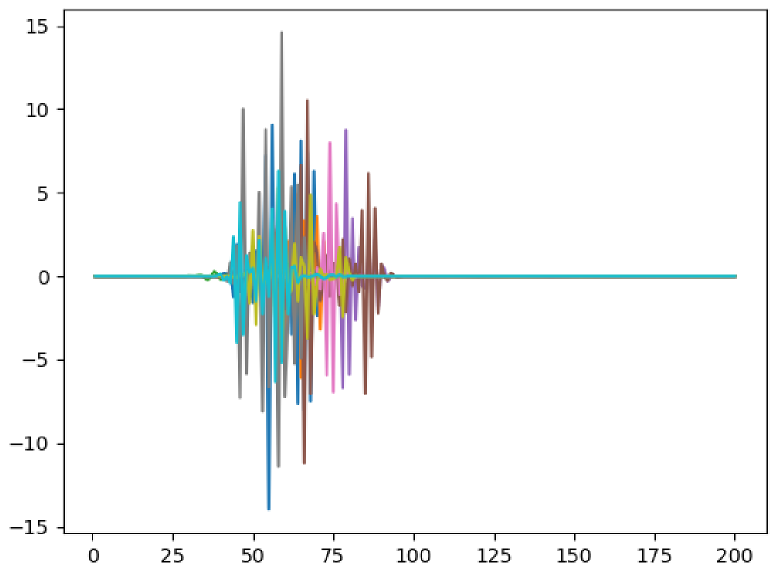

Figure 9: Single nasal flare

Double nasal flare

The double nasal flare consists of two contractions of the nasalis muscle in rapid succession, with a gap of approximately 250 milliseconds. In the UI, this action is mapped to the select action, equivalent to a one-finger tap on the touchpad. Figure 10 shows the EMG-signal of ten double nasal flares after being pre-processed.

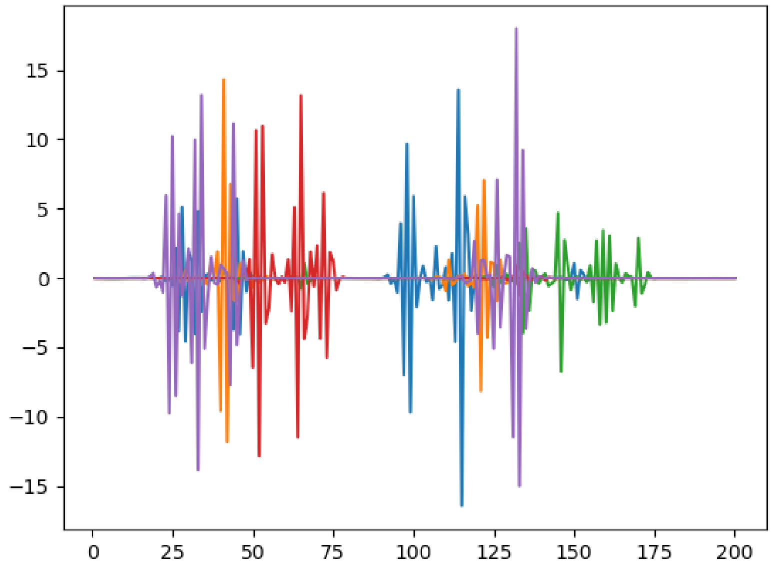

Figure 10: Double nasal flare

Long nasal flare

The long nasal flare consists of a long contraction of the nasal muscle, in which the user keeps the muscle flexed for approximately 600 milliseconds. In the UI, this action is mapped to the back action, equivalent to a two-finger tap on the touchpad. Anatomically this is the most difficult action, which is the reason it is mapped to the likely least used action. Figure 11 shows the EMG-signal of ten long nasal flares after being pre-processed.

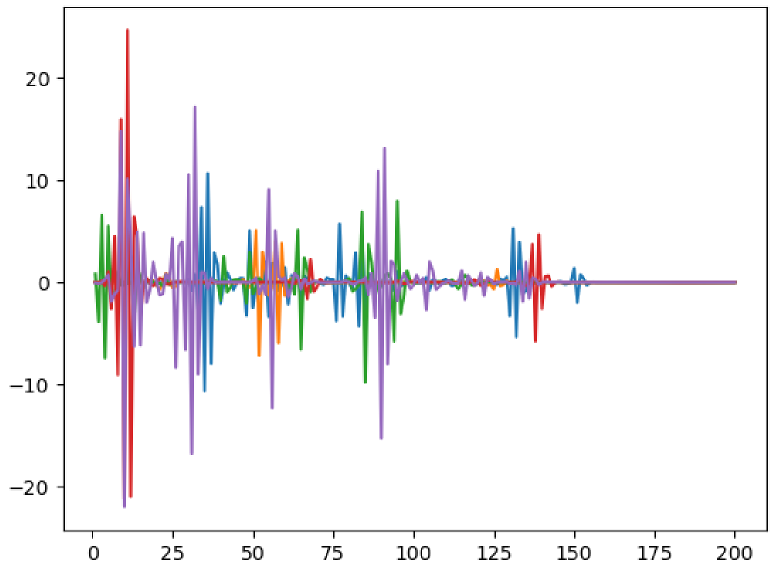

Figure 11: Long nasal flare

The gestures can be mapped to different actions in the glasses based on context. For instance, in a gallery application, the single flare could be mapped to browsing images, the double nasal flare could be mapped to selecting an image and the long nasal flare to return. If instead the user open a camera application, the single flare could be mapped to scroll through the settings, the double nasal flare could be mapped to starting a recording or capturing an image and the double nasal flare could be mapped to exiting the application.

4 User Study

After implementing the system, a user study was performed to evaluate its real world performance. This section will present the study procedure. Section 4.1 introduces the participants. Section 4.2 lists the measures collected during the experiments. Finally, Section 4.3 presents the methodology of the experiments.

4.1 Participants

Six healthy participants, five men and one woman in the age group $22-25$, were recruited for this study. A seventh participant was unable to flare their nostrils, leading to their results being discarded. Three participants were unable to complete all tasks. Five participants were students, one participants was working as a software developer. All participants were right handed, and none had corrected vision. They were also asked to list their familiarity with HMDs, resulting in a mean of 6.17 and a standard deviation of 2.54.

4.2 Measures

A set of measures to be collected during the experiment was devised to enable an evaluation of the system. Both objective and subjective measures were employed.

4.2.1 Objective Measures

The objective measures answers how well the system performs.

Task Completion Rate

Effectiveness is measured by assigning all tasks a percentage score signifying the completion rate. The completion rate is calculated using Equation 2.

$$\frac{\text{Number of completed tasks}}{\text{Number of attempted tasks}} * 100\% \tag{2}$$

Task Completion Time

Efficiency is measured by recording time from start to completion of each task. Sergeev (2010) suggests an equation for calculating efficiency. Let N be the total number of tasks and R be the total number of participants. n_ij is 1 if participant j was able to complete task i, 0 otherwise. t_ij is the amount of time participant j spent on task i. If the participant was unable to finish the task, the amount of time spent before being stopped is recorded. Efficiency is then calculated using Equation 3.

$$\frac{\sum_{j=1}^{R}\sum_{i=1}^{N}\frac{n_ij}{t_ij}}{NR} \tag{3}$$

In addition to the task completion time, an additional 500 milliseconds were added to each task involving Condition A, emulating the average time participants spent lifting their hand from their lap to the touchpad. This is a delay Condition B gets around, making this a more fair comparison.

Number of Gestures for Task Completion

Number of gestures performed during the task is recorded. Comparing this to the number of gestures in the known optimal solution, one can deduce approximately how many error occurred during the trial.

4.2.2 Subjective Measures

The subjective measures answers how the user felt when interacting with the system.

System Usability Scale

The System Usability Scale (SUS) is a highly robust and versatile tool where a user gives their subjective usability score of a product (Bangor et al. 2008). It consists of ten statements the user is asked to give a score on a 5-point Likert scale, depending how much they agree or disagree with the statement. The form is administered immediately following the user's interacting with the product, with the intention of having the user rate the usability intuitively without thinking too much.

When the form has been filled, an aggregated usability score can be calculated. The scores are all converted to a new number which are summed and multiplied by 2.5 to convert the scale from a $0-40$ scale to a $0-100$ scale. It is important to note however that this is not a traditional percentage score. The average SUS-score has been shown to be 68 Sauro 2011. Anything below this should be considered below average, while anything above it should be considered above average. Anything below 51 falls in the bottom 15% of products, and should be considered a failure. A score of 80.3 puts the product in the top 10%.

Task Load Index

The NASA-Task Load Index (TLX) is rating scale consisting of six subjective rating subscales, developed by Hart and Staveland (1988). The user is asked to rate each of the subscales on a scale of $1-21$ based on perceived contribution to workload. The six scales are described in Table 4. An overall rating of perceived workload of a task can be derived based on the scores of these scales. Originally a weighted average was found by letting the user rate their perceived importance of each of the scales. However, an unweighted version named raw NASA-TLX has been validated as a viable alternative (Said et al. 2020). Using this analysis method, a perceived workload is calculated by multiplying each scale rating by five before subtracting five and finding their mean. This gives a score on a scale of $0-100$, with 100 being the highest perceived workload.

Scale	Explanation
Mental demand	How much mental activity was required
Physical demand	How much physical activity was required
Temporal demand	How much time pressure was felt
Performance	How successful was the user
Effort	How much effort was required to reach the level of performance
Frustration	How frustrated was the participant

Table 4: Explanation of the TLX-scales

User Experience Questionnaire

The User Experience Questionnaire (UEQ) is a questionnaire designed to measure the user's subjective user experience with a product. It consists 26 items the user is asked to rate on a 7-point Likert scale. Put together, these items gives six scores, rating the attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty of the product (Laugwitz et al. 2008). What each scale measures is explained in Table 5.

Scale	Explanation
Attractiveness	The overall impression of the product
Perspicuity	How easy it is to get familiar with the product
Efficiency	How quickly the product lets the user solve tasks
Dependability	How secure and predictable the product is
Stimulation	How fun and motivating the product is
Novelty	How creative or interesting the product is

Table 5: Explanation of the UEQ-scales

This study utilises a modified version of the UEQ, selectively removing items from the questionnaire until 15 remained. This was done to reduce the time spent answering forms during the experiments. The modified UEQ can be found in Figure 12.

annoying	○	○	○	○	○	○	○	enjoyable
not understandable	○	○	○	○	○	○	○	understandable
easy to learn	○	○	○	○	○	○	○	difficult to learn
boring	○	○	○	○	○	○	○	exciting
unpredictable	○	○	○	○	○	○	○	predictable
fast	○	○	○	○	○	○	○	slow
inventive	○	○	○	○	○	○	○	conventional
good	○	○	○	○	○	○	○	bad
complicated	○	○	○	○	○	○	○	easy
usual	○	○	○	○	○	○	○	leading edge
motivating	○	○	○	○	○	○	○	demotivating
meets expectations	○	○	○	○	○	○	○	does not meet expectations
inefficient	○	○	○	○	○	○	○	efficient
clear	○	○	○	○	○	○	○	confusing
impractical	○	○	○	○	○	○	○	practical

Figure 12: Modified UEQ

4.3 Methodology

The user testing consisted of an A/B test in which the EMG input device, condition B, was compared to the established input device for the Vuzix Blade; the side-mounted touchpad, condition A. The test involved the user attempting three tasks five times each for each condition, for a total of 30 tasks. The tasks are denoted using the condition involved, as well as the task number from 1 to 3. For example, Task B2 denotes Task 2 performed with condition B, the EMG input device. The tasks are described in Section 3.3.3. During the experiment, the participants were asked to complete the tasks as quickly and accurately as possible. To deal with participants unable to complete a task, they were interrupted after attempting a single run of a task for three minutes, and moved on to the next task.

Two measures were implemented in order to reduce bias towards either condition. First, a practice run was included for each task for each condition to give the participant some familiarity with the system before starting the task. Secondly, counterbalancing was applied, in which the first condition being tested alternated with every new participant. The experiments were run in a lab-like environment to minimise the influence of external factors.

4.3.1 Experimental Procedure

The experiments were scheduled on a per person basis, allowing the participants to pick a time slot that fit their schedule. The participants were also rewarded with a gift card, so they would not consider participating a favour. The experiments were held in the UX-Lab on Gløshaugen campus, ensuring a neutral environment with access to and room for all the necessary equipment.

Before the participant arrived, a table was set up with two chairs, one for the participant and one for the observer. A web camera mounted on a tripod was stationed at the floor, facing the chair of the participant. The camera was connected to laptop placed on the table in front of the chair of the observer. This laptop served as a host for the connecting link software. The Vuzix Blade and OpenBCI Ganglion were turned on and placed on the table in front of the chair of the participant.

When the participant arrived they were greeted and asked to sit in their designated chair. They were explicitly informed that the experiment was about evaluation the prototype — not them. After a consent form was signed, the web camera started recording. The participant was then asked to fill a demographics form. They were then asked to wear the prototype. The observer applied conductive gel to the electrodes before helping the participant put the prototype on. Following this, a procedure was started in which the participant got to practice the nasal gestures while collecting data. They were asked to sit as still as possible for one minute, collecting a baseline EMG signal. They were then asked to perform each of the three gestures 20 times each consecutively, every 1.5 seconds on a timer, with a break between each different gesture.

In the next step, the tasks were explained to the participant. Condition A or Condition B were selected as the first condition, as per the counterbalancing. Assume Condition A was selected. The participant was asked to do an initial trial run of each of the three tasks. They were then asked to perform Task A1 five times in a row. The statistics of each attempt were automatically logged using software. The task was considered successfully completed only if all five attempts were successful. When done, a TLX form was administered. This process was the repeated for Task A2 and Task A3. After Task A3, an SUS form and and a UEQ form were administered. After switching to Condition B, the process was repeated.

Semi-structured interviews were conducted to conclude the experiments. A semi-structured interview is designed to be an open conversation in which the interviewer has prepared a list of questions used to guide the conversation. The interviewee is encouraged to express their opinions. This allows the interviewer to deviate from the questions, skipping or jumping between questions. The questions prepared can be found in Table 6.

Question
What did you enjoy the most?
What did you dislike?
Was there anything particularly frustrating to you?
What was the easiest part of the experiment?
What was the hardest?
What was most impressive to you?
Which input device was most convenient to you?
Which input device was easiest to use?
What would you have done differently?

Table 6: Semi-structured interview questions

5 Results

This chapter presents the results of the user study.

5.1 Task Completion Rate

Table 7 shows the completion rate for the different tasks. All participants were able to complete the tasks related to Condition A. They were also able to complete Task B1. This was expected, since there are no buttons to accidentally click in Task B1. As long as the participant is able to produce a nasal EMG-signal, the menu will eventually scroll to the end. Not everyone was able to complete Task B2 and B3, with only two out of six participants able to fully complete Task B3.

Task	Number of Completions	Number of Attempts	Completion Rate
A1	6	6	100%
A2	6	6	100%
A3	6	6	100%
B1	6	6	100%
B2	3	6	50%
B3	2	6	33.33%

Table 7: Task completion rate

5.2 Time Spent on Task

Table 8 shows the average time spent on each task, as well as calculated efficiency of each task. The calculated efficiency tells how many times the goal is completed per second on average, keeping in mind that in some cases some participants were unable to complete them at all. The results show that participants spent on average 135% more time on Task B1 compared to Task A1, 1873% more time on Task B2 compared to Task A2 and 722% more time on Task B3 compared to Task A3.

Task	Time Spent	Efficiency
A1	3.69	0.278
A2	5.55	0.189
A3	15.58	0.070
B1	8.66	0.118
B2	109.50	0.014
B3	128.03	0.004

Table 8: Task efficiency

5.3 Number of Gestures per Task

Table 9 shows the average number of gestures a participant performed while working on a task. The lowest number of gestures possible for Task 1, Task 2 and Task 3 for either condition are 7, 6 and 13 respectively. The results show that the number of errors were low for Task 1 for both conditions. For Task 2 and Task 3 however, the amount of errors increased significantly for Condition B, while staying relatively low for Condition A.

Task	Number of Gestures
A1	7.07
A2	6.53
A3	14.83
B1	7.47
B2	65.90
B3	74.88

Table 9: Number of gestures per task

5.4 System Usability Scale

Condition A received an average SUS-score of 87.08 with a standard deviation of 4.19, whereas Condition B received an average score of 40.42 with a standard deviation of 11.22. Converting this to a graded scale, Condition A places well withing the top 10% of products and receives an A. Condition B on the other hand, receives an F after placing near the bottom.

5.5 Task Load Index

The results in Table 10 show that the participants found the tasks involving Condition B significantly harder than their Condition A counterparts. When compared the tasks individually, participants found Task B1 143% more demanding than Task A1, Task B2 233% more demanding than Task A2 and Task B3 133% more demanding than Task A3.

	A1	**A2**	**A3**	**B1**	**B2**	**B3**
Mental Demand	2.17	3.83	6.17	4.33	6.50	7.50
Physical Demand	1.83	2.17	3.83	6.33	10.50	12.33
Temporal Demand	6.17	4.83	4.67	6.50	10.00	10.17
Performance	1.67	3.83	6.50	6.33	13.50	16.00
Effort	1.33	2.83	6.33	7.50	14.50	15.50
Frustration	1.83	2.67	5.5	5.5	12.17	15.50
Total	15.00	20.17	33.00	36.50	67.17	77

Table 10: TLX raw numbers

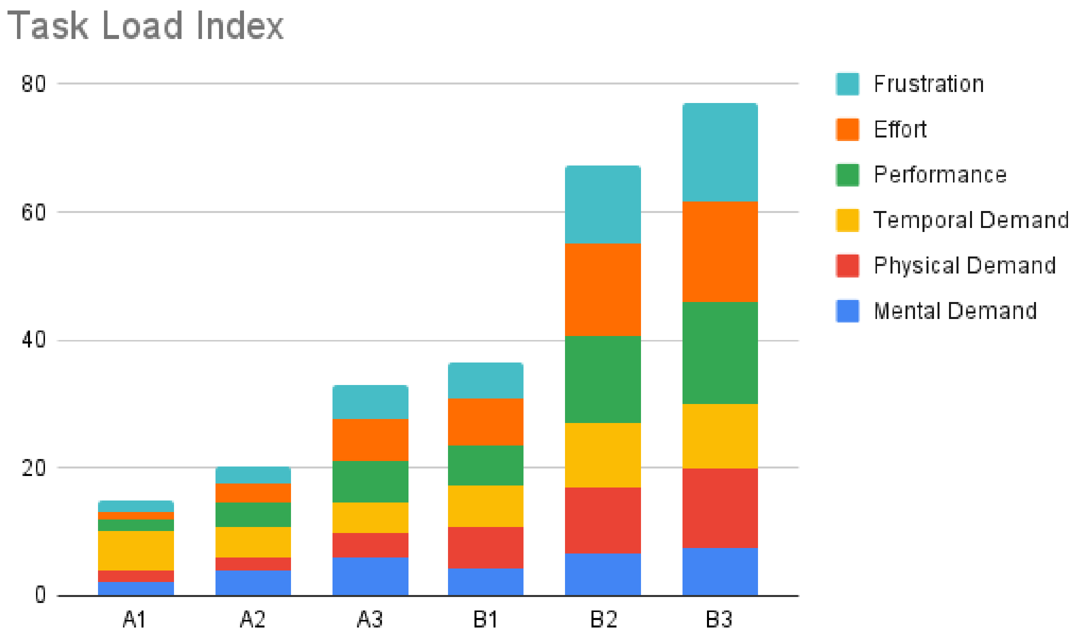

Figure 13: TLX chart

5.6 User Experience Questionnaire

Table 11 shows the average value for each scale measured by the UEQ, as well as the standard deviation and the confidence value at a 95% threshold. The confidence interval is defined as $Avg \pm Conf$. Figure 14 provides a visualisation of the mean values for each scale. The results show that Condition A scores significantly higher on attractiveness, perspicuity, efficiency and dependability, slightly higher on simulation, and significantly lower on novelty.

	Condition A			**Condition B**		
Scale	Avg	STD	Conf	Mean	STD	Conf
Attractiveness	1.67	0.88	0.70	-1.33	0.98	0.79
Perspicuity	2.46	0.58	0.46	0.63	1.50	1.20
Efficiency	2.50	0.55	0.44	-1.22	1.24	0.99
Dependability	2.00	0.71	0.57	-1.25	0.42	0.33
Simulation	1.08	0.97	0.78	-0.17	1.17	0.94
Novelty	-0.17	1.08	0.86	2.42	0.49	0.39

Table 11: UEQ results

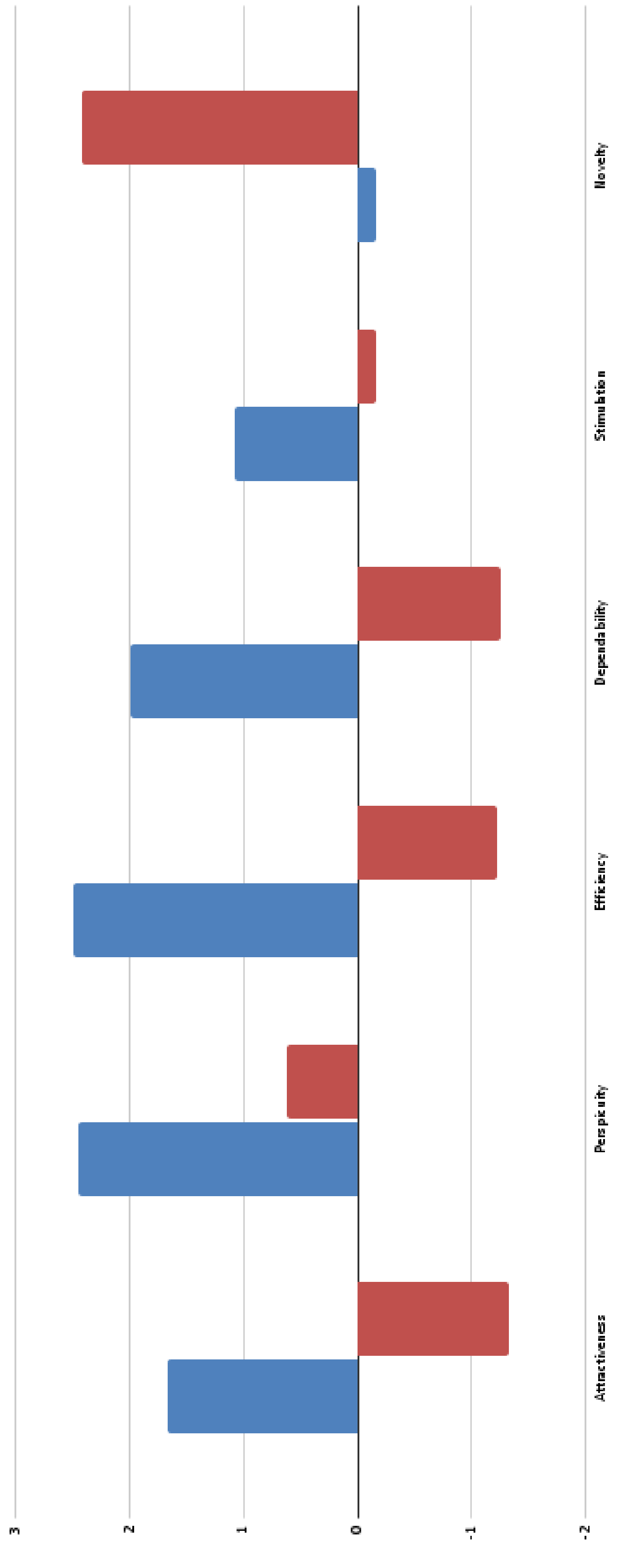

Figure 14: UEQ results

5.7 Semi-Structured Interviews

The short semi-structured interviews provided some useful feedback after concluding the experiments. All participants agreed that the touchpad was more convenient and easier to use. Most participants still expressed that the most enjoyable part of the experiment was seeing and interacting with the new technology of the nasal EMG input device, and enjoyed it when it worked well. Participants were generally frustrated with the double and long flare gestures, expressing that they were unable to consistently use them to perform the desired action. One participant expressed some discomfort wearing the device, making them more physically tired over timed, leading to worse performance. Two participants agreed that it could have helped if they focused on spending more time being accurate as opposed to being quick.

The transcribed interviews can be read in Appendix C.

6 Conclusion

The goal of this study was to discover whether nasal EMG was a viable input modality for HMDs. To answer this, a hardware prototype was developed. This prototype consisted of two electrodes connected to the nasalis muscle using a custom nose bridge and one electrode connected to the mastoid part of the temporal bone. The nasal EMG-signals were classified as one of four gestures using a purpose-built CNN. Depending on the result, a command was sent to an HMD corresponding to an action in the UI. The study included a literature review, in which existing solutions providing input modality to wearable devices were explored. This phase helped identify how nasal EMG as an input modality could contribute to advancing the field and as such guided early development. Identified contributions include providing a more subtle approach which could possibly increasing social acceptability, as well as providing an input modality more compliant with the universal design principle by making it more accessible to people suffering from a mobility disability.

A user study was conducted in order to answer the three research questions posed in this study. The user study set out to compare the prototype to an established way of interacting with smart glasses; the side-mounted touchpad. Six participants able to flare their nostrils were recruited and asked to perform three different tasks with both input devices in a lab environment. Both objective and subjective measures were used to evaluate the input devices. Finally, a semi-structured interview was conducted to let the participants further elaborate their opinions. One main conclusions was drawn from the results. Nasal EMG is not viable as an input modality at this point. Several participants struggled when using the prototype, spending significantly more time on each tasks and in some cases not being able to complete a task at all. It was outclassed by the touchpad in every single objective measure, and all subjective measures but novelty. In total, it justly received a very poor usability score.

However, the prototype stood out as a very interesting product, with several participants expressing enjoyment using it in spite of poor performance. One gestures in particular shows more promise than the others. All the participants were able to consistently perform a single nasal flare. From this one can conclude that the idea does show some promise, and further development could feasibly achieve better results.

The undeniably small sample size of participants in the user test is a limitation to this study and makes it hard to draw many definitive conclusions. A future study with increased sample size could be useful to further explore the idea of nasal EMG as an input modality. Section 6.2 gives suggestions on what should be improved for such a future study.

6.1 Answering the Research Questions

Despite the difficulties mentioned, this section tries to explicitly answer the research questions defined in Section 1.2 based on the findings.

RQ1: *Is nasal EMG an effective method for interacting with HMDs?*

In its current form, nasal EMG is overall not an effective method for interacting with HMDs. The inconsistencies are too severe to provide a pleasant user experience, and more conventional solutions prove much more effective.

RQ2: *How effective is nasal EMG for interacting with HMDs?*

Several participants were able to complete all the tasks using nasal EMG, highlighting the potential of the idea. However, it is currently far less efficient than more conventional solutions.

RQ3: *How effective are the individual nasal gestures for interacting with HMDs?*

The single nasal flare shows potential as an effective nasal gesture. Both the double nasal flare and the long nasal flare have issues which must be solved before they can be considered effective.

6.2 Future Work

The number one priority of further development should be to improve usability. In its current state, the prototype is not practically usable at all. To achieve this, some suggestions are presented. First and foremost, a solution should be found in which the OpenBCI Ganglion is polled more often. In the final version of the prototype, the Ganglion was polled once every second. This was done to ensure a big enough window to separate single nasal flares and double nasal flares. Some delay is required to give the user enough time to perform the second flare, but this could be implemented for instance in the form of a small delay from when the first nasal flare is detected. This at least gets around the issue of static one second windows, which were often confusing to users.

Moreover, the gestures could be refined. Through more research on the nasalis muscle, one might find some gestures which are more feasible to perform.

Adding to this, the Classifier itself could be further improved. Through additional hyperparameter optimisation, one might find a configuration which gives better results, and maybe eliminate overfitting altogether. Another feasible way of improving the classifier is to collect a bigger data set. The Classifier was in this study trained with a data set consisting of 1700 labelled samples distributed approximately evenly over the four classes.

To create a commercial product, the hardware of the prototype should also be refined. The nasal bridge could be improved to include a mechanism for removing the electrodes for easier cleaning and maintenance. Some mechanism should be added to avoid having to manually attach the ground electrode.

Finally, a bigger user test should be conducted, with at the very least three times as many participants. This would provide significantly more data for an analysis, making it far easier to draw conclusions.

Bibliography

Albawi, Saad, Tareq Abed Mohammed and Saad Al-Zawi (2017). 'Understanding of a convolutional neural network'. In: *2017 International Conference on Engineering and Technology (ICET)*. Ieee, pp. 1–6.

Bangor, Aaron, Philip T Kortum and James T Miller (2008). 'An empirical evaluation of the system usability scale'. In: *Intl. Journal of Human–Computer Interaction* 24.6, pp. 574–594.

Basmajian, John V and Carlo J De Luca (1985). *Muscles alive. Their functions revealed by electromyography*. Baltimore: Williams & Wilkins.

Birkfellner, Wolfgang et al. (2002). 'A head-mounted operating binocular for augmented reality visualization in medicine-design and initial evaluation'. In: *IEEE Transactions on Medical Imaging* 21.8, pp. 991–997.

Bruintjes, TD et al. (1996). 'Electromyography of the human nasal muscles'. In: *European Archives of Oto-rhino-laryngology* 253.8, pp. 464–469.

Castellini, Claudio and Patrick Van Der Smagt (2009). 'Surface EMG in advanced hand prosthetics'. In: *Biological cybernetics* 100.1, pp. 35–47.

Conv1d — Pytorch 1.9.0 documentation (2021). URL: https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html (visited on 21st June 2021).

Dobbelstein, David, Philipp Hock and Enrico Rukzio (2015). 'Belt: An unobtrusive touch input device for head-worn displays'. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2135–2138.

Ducrohet, Xavier, Tor Norbye and Katherine Chou (2013). *Android Studio: An IDE built for Android*. URL: https://android-developers.googleblog.com/2013/05/android-studio-ide-built-for-android.html.

Feddersen, WE (1962). *Simulation Evaluation of a Head Mounted Orientation Display*. Tech. rep. BELL HELICOPTER TEXTRON INC FORT WORTH TX.

Feurer, Matthias and Frank Hutter (2019). 'Hyperparameter optimization'. In: *Automated Machine Learning*. Springer, Cham, pp. 3–33.

Gong, Jun and Peter Tarasewich (2005). 'Alphabetically constrained keypad designs for text entry on mobile devices'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 211–220.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. The MIT Press. ISBN: 0262035618.

Ham, Jooyeun et al. (2014). 'Smart wristband: Touch-and-motion–tracking wearable 3D input device for smart glasses'. In: *International Conference on Distributed, Ambient, and Pervasive Interactions*. Springer, pp. 109–118.

Hart, Sandra G and Lowell E Staveland (1988). 'Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research'. In: *Advances in psychology*. Vol. 52. Elsevier, pp. 139–183.

Haung, Albert (2019). *PyBluez*. URL: https://pybluez.readthedocs.io/en/latest/ (visited on 1st June 2021).

Hinton, Geoffrey E. et al. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv: 1207.0580 [cs.NE].

Jain, A.K., Jianchang Mao and K.M. Mohiuddin (1996). 'Artificial neural networks: a tutorial'. In: *Computer* 29.3, pp. 31–44. DOI: 10.1109/2.485891.

Laugwitz, Bettina, Theo Held and Martin Schrepp (2008). 'Construction and Evaluation of a User Experience Questionnaire'. In: *HCI and Usability for Education and Work*. Ed. by Andreas Holzinger. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 63–76. ISBN: 978-3-540-89350-9.

Lee, Juyoung et al. (2018). 'Automated Data Gathering and Training Tool for Personalized "Itchy Nose"'. In: *Proceedings of the 9th Augmented Human International Conference*, pp. 1–3.

Letourneau, A and RK Daniel (July 1988). 'The superficial musculoaponeurotic system of the nose'. In: *Plastic and reconstructive surgery* 82.1, pp. 48–57. ISSN: 0032-1052. URL: http://europepmc.org/abstract/MED/3380925.

Liao, Shih-Chieh, Fong-Gong Wu and Shu-Hsuanb Feng (2019). 'Playing games with your mouth: Improving gaming experience with EMG supportive input device'. In: *International Association of Societies of Design Research Conference*. Vol. 2019.

Müller, Florian et al. (2019). 'Mind the tap: Assessing foot-taps for interacting with head-mounted displays'. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

Nakao, Takuro et al. (2018). 'Make-a-face: a hands-free, non-intrusive device for tongue/mouth/cheek input using EMG'. In: *ACM SIGGRAPH 2018 Posters*, pp. 1–2.

O'Shea, Keiron and Ryan Nash (2015). 'An introduction to convolutional neural networks'. In: *arXiv preprint arXiv:1511.08458*.

Parfenov, Andrey (2021). *BrainFlow*. URL: https://brainflow.org/ (visited on 1st June 2021).

Paszke, Adam et al. (2019). 'Pytorch: An imperative style, high-performance deep learning library'. In: *arXiv preprint arXiv:1912.01703*.

Peshock, Anna, Julia Duvall and Lucy E Dunne (2014). 'Argot: A wearable one-handed keyboard glove'. In: *Proceedings of the 2014 ACM international symposium on wearable computers: adjunct program*, pp. 87–92.

Said, Sadiq et al. (Sept. 2020). 'Using the Raw NASA Task Load Index Questionnaire to Assess Perceived Workload in Patient Monitoring Tasks: A Meta-Analysis Using a Mixed Model'. In: DOI: 10.2196/19472.

Sauro, Jeff (2011). *Measuring Usability with the System Usability Scale (SUS) – MeasuringU*. URL: https://measuringu.com/sus/.

Sergeev, Anton (2010). *UI Designer - ISO-9241 Efficiency metrics - Theory of usability*. URL: http://ui-designer.net/usability/efficiency.htm.

Shannon, C.E. (Jan. 1949). 'Communication in the Presence of Noise'. In: *Proceedings of the IRE* 37.1, pp. 10–21. DOI: 10.1109/jrproc.1949.232969. URL: https://doi.org/10.1109/jrproc.1949.232969.

Sugawara, Yasushi (2020). 'Clinical Anatomy'. In: *A Practical Approach to Asian Rhinoplasty: How to Design A Fine Nasal Shape*. Tokyo: Springer Japan, pp. 1–17. ISBN: 978-4-431-56885-8. DOI: 10.1007/978-4-431-56885-8_1. URL: https://doi.org/10.1007/978-4-431-56885-8_1.

Surhone, Lambert M., Mariam T. Tennoe and Susan F. Henssonow (2010). *Node.Js*. Beau Bassin, MUS: Betascript Publishing. ISBN: 6133180196.

Taniguchi, Kazuhiro et al. (2018). 'Earable TEMPO: A novel, hands-free input device that uses the movement of the tongue measured with a wearable ear sensor'. In: *Sensors* 18.3, p. 733.

Vourvopoulos, Athanasios, Evangelos Niforatos and Michail Giannakos (2019). 'EEGlass: An EEG-eyeware prototype for ubiquitous brain-computer interaction'. In: *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 647–652.

Wasilewski, Filip (2021). *Symlets 7 wavelet (sym7) properties, filters and functions - Wavelet Properties Browser*. URL: http://wavelets.pybytes.com/wavelet/sym7/.

Withana, Anusha et al. (2015). 'zSense: Enabling shallow depth gesture recognition for greater input expressivity on smart wearables'. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3661–3670.

Yasuhiro, Suzuki and Jing Lei (2018). 'Motion Capture Glove: An Intuitive Wearable Input Device'. In: *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*. IEEE, pp. 1–5.

Yoon, Sang Ho et al. (2015). 'TIMMi: Finger-worn textile input device with multimodal sensing in mobile interaction'. In: *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 269–272.

Zhang, Xian-Da (2020). 'Machine learning'. In: *A Matrix Algebra Approach to Artificial Intelligence*. Springer, pp. 223–440.

Appendix

A Result Data

Part	Gender	Age	Profession	Handedness	Vision	HMD	Task	Time	Actions
1	Male	22-25	Student	Right	No	6	A1	5.305	7
1	Male	22-25	Student	Right	No	6	A1	3.639	7
1	Male	22-25	Student	Right	No	6	A1	3.259	7
1	Male	22-25	Student	Right	No	6	A1	4.437	7
1	Male	22-25	Student	Right	No	6	A1	4.337	7
1	Male	22-25	Student	Right	No	6	A2	9.747	10
1	Male	22-25	Student	Right	No	6	A2	5.483	6
1	Male	22-25	Student	Right	No	6	A2	7.272	10
1	Male	22-25	Student	Right	No	6	A2	4.125	6
1	Male	22-25	Student	Right	No	6	A2	5.574	6
1	Male	22-25	Student	Right	No	6	A3	21.524	17
1	Male	22-25	Student	Right	No	6	A3	24.403	19
1	Male	22-25	Student	Right	No	6	A3	19.944	14
1	Male	22-25	Student	Right	No	6	A3	15.83	16
1	Male	22-25	Student	Right	No	6	A3	22.854	19
1	Male	22-25	Student	Right	No	6	B1	10.173	7
1	Male	22-25	Student	Right	No	6	B1	7.21	7
1	Male	22-25	Student	Right	No	6	B1	8.226	8
1	Male	22-25	Student	Right	No	6	B1	7.203	7
1	Male	22-25	Student	Right	No	6	B1	7.208	7
1	Male	22-25	Student	Right	No	6	B2	175.663	132
1	Male	22-25	Student	Right	No	6	B2	180	126
1	Male	22-25	Student	Right	No	6	B3	97.679	59
1	Male	22-25	Student	Right	No	6	B3	180	118
2	Female	22-25	Student	Right	No	3	A1	2.508	7
2	Female	22-25	Student	Right	No	3	A1	2.677	7
2	Female	22-25	Student	Right	No	3	A1	2.352	7
2	Female	22-25	Student	Right	No	3	A1	2.369	7
2	Female	22-25	Student	Right	No	3	A1	2.12	7
2	Female	22-25	Student	Right	No	3	A2	3.775	6
2	Female	22-25	Student	Right	No	3	A2	2.534	6
2	Female	22-25	Student	Right	No	3	A2	6.962	8
2	Female	22-25	Student	Right	No	3	A2	4.458	6
2	Female	22-25	Student	Right	No	3	A2	3.098	6
2	Female	22-25	Student	Right	No	3	A3	17.073	16
2	Female	22-25	Student	Right	No	3	A3	8.391	12
2	Female	22-25	Student	Right	No	3	A3	9.359	12
2	Female	22-25	Student	Right	No	3	A3	7.598	12
2	Female	22-25	Student	Right	No	3	A3	9.231	12
2	Female	22-25	Student	Right	No	3	B1	9.828	9
2	Female	22-25	Student	Right	No	3	B1	7.166	7
2	Female	22-25	Student	Right	No	3	B1	7.201	7
2	Female	22-25	Student	Right	No	3	B1	7.231	7
2	Female	22-25	Student	Right	No	3	B1	7.239	7
2	Female	22-25	Student	Right	No	3	B2	24.057	15
2	Female	22-25	Student	Right	No	3	B2	41.235	29
2	Female	22-25	Student	Right	No	3	B2	30.575	23
2	Female	22-25	Student	Right	No	3	B2	43.771	31
2	Female	22-25	Student	Right	No	3	B2	25.464	17
2	Female	22-25	Student	Right	No	3	B3	50.87	32
2	Female	22-25	Student	Right	No	3	B3	178.174	113
2	Female	22-25	Student	Right	No	3	B3	69.987	46
2	Female	22-25	Student	Right	No	3	B3	114.622	76
2	Female	22-25	Student	Right	No	3	B3	75.139	50

Part	Gender	Age	Profession	Handedness	Vision	HMD	Task	Time	Actions
3	Male	22-25	Student	Right	No	9	A1	4.916	8
3	Male	22-25	Student	Right	No	9	A1	3.646	7
3	Male	22-25	Student	Right	No	9	A1	2.689	7
3	Male	22-25	Student	Right	No	9	A1	2.867	7
3	Male	22-25	Student	Right	No	9	A1	2.787	7
3	Male	22-25	Student	Right	No	9	A2	8.654	8
3	Male	22-25	Student	Right	No	9	A2	8.051	10
3	Male	22-25	Student	Right	No	9	A2	5.689	6
3	Male	22-25	Student	Right	No	9	A2	4.315	6
3	Male	22-25	Student	Right	No	9	A2	3.277	6
3	Male	22-25	Student	Right	No	9	A3	14.986	20
3	Male	22-25	Student	Right	No	9	A3	16.537	18
3	Male	22-25	Student	Right	No	9	A3	12.176	16
3	Male	22-25	Student	Right	No	9	A3	11.431	15
3	Male	22-25	Student	Right	No	9	A3	8.075	14
3	Male	22-25	Student	Right	No	9	B1	7.944	7
3	Male	22-25	Student	Right	No	9	B1	7.214	7
3	Male	22-25	Student	Right	No	9	B1	7.224	7
3	Male	22-25	Student	Right	No	9	B1	7.224	7
3	Male	22-25	Student	Right	No	9	B1	10.249	9
3	Male	22-25	Student	Right	No	9	B2	35.187	22
3	Male	22-25	Student	Right	No	9	B2	65.175	37
3	Male	22-25	Student	Right	No	9	B2	52.729	34
3	Male	22-25	Student	Right	No	9	B2	104.617	64
3	Male	22-25	Student	Right	No	9	B2	24.447	12
3	Male	22-25	Student	Right	No	9	B3	55.698	33
3	Male	22-25	Student	Right	No	9	B3	102.803	57
3	Male	22-25	Student	Right	No	9	B3	47.497	25
3	Male	22-25	Student	Right	No	9	B3	159.445	88
3	Male	22-25	Student	Right	No	9	B3	35.636	20
4	Male	22-25	Student	Right	No	10	A1	3.133	7
4	Male	22-25	Student	Right	No	10	A1	3.372	7
4	Male	22-25	Student	Right	No	10	A1	2.906	7
4	Male	22-25	Student	Right	No	10	A1	2.524	7
4	Male	22-25	Student	Right	No	10	A1	2.537	7
4	Male	22-25	Student	Right	No	10	A2	5.696	6
4	Male	22-25	Student	Right	No	10	A2	13.77	6
4	Male	22-25	Student	Right	No	10	A2	3.859	6
4	Male	22-25	Student	Right	No	10	A2	3.824	6
4	Male	22-25	Student	Right	No	10	A2	3.546	6
4	Male	22-25	Student	Right	No	10	A3	19.118	14
4	Male	22-25	Student	Right	No	10	A3	28.487	16
4	Male	22-25	Student	Right	No	10	A3	18.85	16
4	Male	22-25	Student	Right	No	10	A3	27.858	19
4	Male	22-25	Student	Right	No	10	A3	13.689	16
4	Male	22-25	Student	Right	No	10	B1	13.902	7
4	Male	22-25	Student	Right	No	10	B1	8.241	7
4	Male	22-25	Student	Right	No	10	B1	7.212	7
4	Male	22-25	Student	Right	No	10	B1	7.182	7
4	Male	22-25	Student	Right	No	10	B1	10.243	10
4	Male	22-25	Student	Right	No	10	B2	180	112
4	Male	22-25	Student	Right	No	10	B3	180	133

Part	Gender	Age	Profession	Handedness	Vision	HMD	Task	Time	Actions
5	Male	22-25	Software dev	Right	No	5	A1	6.193	8
5	Male	22-25	Software dev	Right	No	5	A1	3.639	7
5	Male	22-25	Software dev	Right	No	5	A1	3.209	7
5	Male	22-25	Software dev	Right	No	5	A1	2.467	7
5	Male	22-25	Software dev	Right	No	5	A1	2.273	7
5	Male	22-25	Software dev	Right	No	5	A2	6.167	6
5	Male	22-25	Software dev	Right	No	5	A2	3.81	6
5	Male	22-25	Software dev	Right	No	5	A2	3.613	6
5	Male	22-25	Software dev	Right	No	5	A2	3.415	6
5	Male	22-25	Software dev	Right	No	5	A2	3.064	6
5	Male	22-25	Software dev	Right	No	5	A3	30.125	21
5	Male	22-25	Software dev	Right	No	5	A3	12.973	14
5	Male	22-25	Software dev	Right	No	5	A3	11.619	13
5	Male	22-25	Software dev	Right	No	5	A3	9.547	12
5	Male	22-25	Software dev	Right	No	5	A3	8.61	12
5	Male	22-25	Software dev	Right	No	5	B1	7.235	7
5	Male	22-25	Software dev	Right	No	5	B1	7.208	7
5	Male	22-25	Software dev	Right	No	5	B1	7.202	7
5	Male	22-25	Software dev	Right	No	5	B1	7.198	7
5	Male	22-25	Software dev	Right	No	5	B1	7.225	7
5	Male	22-25	Software dev	Right	No	5	B2	23.433	14
5	Male	22-25	Software dev	Right	No	5	B2	43.741	29
5	Male	22-25	Software dev	Right	No	5	B2	21.371	16
5	Male	22-25	Software dev	Right	No	5	B2	46.821	30
5	Male	22-25	Software dev	Right	No	5	B2	13.214	9
5	Male	22-25	Software dev	Right	No	5	B3	68.441	43
5	Male	22-25	Software dev	Right	No	5	B3	39.676	25
5	Male	22-25	Software dev	Right	No	5	B3	61.016	43
5	Male	22-25	Software dev	Right	No	5	B3	180	124
5	Male	22-25	Software dev	Right	No	5	B3	107.689	64
7	Male	22-25	Student	Right	No	4	A1	3.801	7
7	Male	22-25	Student	Right	No	4	A1	2.934	7
7	Male	22-25	Student	Right	No	4	A1	2.352	7
7	Male	22-25	Student	Right	No	4	A1	2.279	7
7	Male	22-25	Student	Right	No	4	A1	2.209	7
7	Male	22-25	Student	Right	No	4	A2	5.278	6
7	Male	22-25	Student	Right	No	4	A2	3.387	6
7	Male	22-25	Student	Right	No	4	A2	3.457	6
7	Male	22-25	Student	Right	No	4	A2	2.774	6
7	Male	22-25	Student	Right	No	4	A2	2.686	6
7	Male	22-25	Student	Right	No	4	A3	16.197	12
7	Male	22-25	Student	Right	No	4	A3	9.037	12
7	Male	22-25	Student	Right	No	4	A3	8.891	12
7	Male	22-25	Student	Right	No	4	A3	7.437	12
7	Male	22-25	Student	Right	No	4	A3	10.414	12
7	Male	22-25	Student	Right	No	4	B1	7.225	7
7	Male	22-25	Student	Right	No	4	B1	7.202	7
7	Male	22-25	Student	Right	No	4	B1	7.214	7
7	Male	22-25	Student	Right	No	4	B1	11.303	9
7	Male	22-25	Student	Right	No	4	B1	25.5	11
7	Male	22-25	Student	Right	No	4	B2	180	78
7	Male	22-25	Student	Right	No	4	B3	180	60

B Forms

Demographics

* Required

1. Age *

   Mark only one oval.

   - ◯ <18
   - ◯ 18-21
   - ◯ 22-25
   - ◯ 26-29
   - ◯ 30-39
   - ◯ 40-49
   - ◯ 50-59
   - ◯ 60-69
   - ◯ <69

2. Gender *

   Mark only one oval.

   - ◯ Male
   - ◯ Female
   - ◯ Prefer not to say
   - ◯ Other: ____________________

3. Occupation *

   ____

4. Left/right-handedness *

   Mark only one oval.

   - ◯ Left-handed
   - ◯ Right-handed

5. Do you wear glasses/lenses to correct your vision? *

Mark only one oval.

- Yes
- No

Familiarity with HMDs (Head Mounted Displays)

6. How familiar are you with HMDs? *

Mark only one oval.

	1	2	3	4	5	6	7	8	9	10	
Not familiar at all											Very familiar

7. Are you familiar with any of the following HMDs? *

Check all that apply.

- Oculus
- HTC Vive
- Other VR HMDs
- Google Glass
- Hololens
- Vuzix Blade
- Other AR HMDs (smartglasses)

PARTICIPANT NAME: DATE:

System Usability Scale

For each of the following statements, please mark one box that best describes your reactions to the input system today.

		Strongly disagree				Strongly agree
1.	I think that I would like to use the input system frequently.	1	2	3	4	5
2.	I found the input system unnecessarily complex.	1	2	3	4	5
3.	I thought the input system was easy to use.	1	2	3	4	5
4.	I think that I would need the support of a technical person to be able to use the input system.	1	2	3	4	5
5.	I found the various functions in the input system were well integrated.	1	2	3	4	5
6.	I thought there was too much inconsistency in the input system.	1	2	3	4	5
7.	I would imagine that most people would learn to use the input system very quickly.	1	2	3	4	5
8.	I found the input system very cumbersome (awkward) to use.	1	2	3	4	5
9.	I felt very confident using the input system.	1	2	3	4	5
10.	I needed to learn a lot of things before I could get going with the input system.	1	2	3	4	5

Comments (optional):

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

annoying	○ ○ ○ ○ ○ ○ ○	enjoyable
not understandable	○ ○ ○ ○ ○ ○ ○	understandable
easy to learn	○ ○ ○ ○ ○ ○ ○	difficult to learn
boring	○ ○ ○ ○ ○ ○ ○	exciting
unpredictable	○ ○ ○ ○ ○ ○ ○	predictable
fast	○ ○ ○ ○ ○ ○ ○	slow
inventive	○ ○ ○ ○ ○ ○ ○	conventional
good	○ ○ ○ ○ ○ ○ ○	bad
complicated	○ ○ ○ ○ ○ ○ ○	easy
usual	○ ○ ○ ○ ○ ○ ○	leading edge
motivating	○ ○ ○ ○ ○ ○ ○	demotivating
meets expectations	○ ○ ○ ○ ○ ○ ○	does not meet expectations
inefficient	○ ○ ○ ○ ○ ○ ○	efficient
clear	○ ○ ○ ○ ○ ○ ○	confusing
impractical	○ ○ ○ ○ ○ ○ ○	practical

C Interviews

Participant 1

- What did you enjoy the most?
  - When I managed to correctly perform an enter action, as well as generally scrolling through the menu with my nose.
- What did you dislike?
  - When the device kept scrolling when I tried to press enter. I felt like I couldn't do it.
- Was there anything particularly frustrating to you?
  - In addition to the above, the device over time started pinching my nose, making me more and more physically tired, making it harder and harder to do the correct actions.
- What was the easiest part of the experiment?
  - The scrolling was quite simple, both with the touchpad and using EMG. The touchpad might have an advantage because of the ability to go left if you overshoot, but I didn't feel like using my nose was particularly harder.
- What was the hardest?
  - Performing the enter action with my nose.
- What was most impressive to you?
  - Generally it worked about as well as I expected. I hadn't expected [the EMG] to work as well as it did. Even though I could get enter to work, the rest of it felt like it responded nicely and quickly to my actions, which I liked.
- Which input device was most convenient to you?
  - Right now I have to say the touchpad. As long as I have both hands available, I feel like I have much more control over the device using the touchpad, especially with the ability to scroll left. I still did the wrong actions several times with the touchpad.
- Which input device was easiest to use?
  - (See above)
- What would you have done differently?
  - I don't know, I don't think I could have done anything differently.

Participant 2

- What did you enjoy the most?
    - I was more comfortable using the touchpad, but I really enjoyed the times I felt like I was able to successfully use my nose.
- What did you dislike?
    - It was pretty frustrating when I tried to do something with my nose, and something entirely different happened on screen.
- Was there anything particularly frustrating to you?
    - -
- What was the easiest part of the experiment?
    - Task A1.
- What was the hardest?
    - Task B3. Particularly the double nasal flare.
- What was most impressive to you?
    - I thought it was very cool when the nasal input worked. I believe it could be useful for many people, but probably not fully functioning people as it's unnecessarily cumbersome.
- Which input device was most convenient to you?
    - (See below)
- Which input device was easiest to use?
    - The touchpad.
- What would you have done differently?
    - Spent more time on the tasks involving the nose to maybe perform less errors.

Participant 3

- What did you enjoy the most?
  - The touchpad, without a doubt felt more responsive, leading to less errors and also less irritation because getting back to where I was was easier.
- What did you dislike?
  - The nasal input device. Specifically when you think you've clicked something and you're forced to wait for the delay to see if the action registered correctly or not. It makes me feel like I don't know if the program is unresponsive, or if I'd done something wrong.
- Was there anything particularly frustrating to you?
  - (See above)
- What was the easiest part of the experiment?
  - Scrolling through news articles (Task 1). This wasn't that hard with nasal input, but I did make some mistakes, making the touchpad.
- What was the hardest?
  - Either double nasal flare or long nasal flare. I think double nasal flare as I very often ended up overshooting the button since it registered as a single flare, forcing me to scroll all the way around.
- What was most impressive to you?
  - I thought the touchpad was really well implemented.
- Which input device was most convenient to you?
  - Touchpad.
- Which input device was easiest to use?
  - (See above)
- What would you have done differently?
  - I don't know, maybe spend more time to be more accurate.

Participant 4

- What did you enjoy the most?
  - I enjoyed seeing the new technology, I've never seen anything like it before.
- What did you dislike?
  - I found it hard to double nasal flare to select items from the menu. I also sometimes got disoriented in the menu and struggled to tell where I should navigate.
- Was there anything particularly frustrating to you?
  - Like I mentioned, when I got lost in the menu.
- What was the easiest part of the experiment?
  - Scrolling with the touchpad.
- What was the hardest?
  - Double nasal flares.
- What was most impressive to you?
  - The glasses themselves maybe.
- Which input device was most convenient to you?
  - (See below)
- Which input device was easiest to use?
  - The touchpad.
- What would you have done differently?
  - Not that I can think of.

Participant 5

- What did you enjoy the most?
  - I enjoyed trying the new technology.
- What did you dislike?
  - The screen was a bit blurry to me, so I had to spend more time telling what was happening on screen.
- Was there anything particularly frustrating to you?
  - The blurry screen, in addition to the nasal input device. After performing nasal flares for so long, it felt like I forgot how to flare my nostrils. It also felt like it jumped around without me controlling it at some points due to the delay. Not knowing why this was happening was pretty frustrating.
- What was the easiest part of the experiment?
  - Task 1 was quite simple with both input devices, scrolling through the menu. At this point, both devices acted like I expected them to.
- What was the hardest?
  - Sometimes I struggled using the double nasal flare to select items in the menu. Long nasal flare to return worked better, but it was a bit hard to know exactly how long to hold the flare.
- What was most impressive to you?
  - I was impressed that you were able to register nasal emg at all. I don't really know what I expected, I didn't understand how this would work. It also felt very low latency, with things happening on screen very quickly after I performed an action.
- Which input device was most convenient to you?
  - Definitely the touchpad.
- Which input device was easiest to use?
  - (See above)
- What would you have done differently?
  - It felt like I could have done the tasks with my eyes closed after doing them two or three times, though obviously not with the nasal input device as I didn't really know where in the menu I was.

Participant 7

- What did you enjoy the most?
  - I thought it was interesting trying new ways to navigate. Also it's nice knowing this could end up helping people unable to use their hands.
- What did you dislike?
  - Using the nasal input system was really frustrating. Often it felt like when I tried to double flare my nose, I did it exactly between the delay windows, and it ended up registering as two single flares. Because this happened so often, I tried to compensate by double flaring more rapidly, which just ended as a long flare.
- Was there anything particularly frustrating to you?
  - I'd say when I was returned to the previous level of the menu without intending to. Second to this, I was also frustrated when I missed a double click and had to go all the way around to get back, resulting in a lot of extra actions.
- What was the easiest part of the experiment?
  - The touchpad was definitely much easier to use. I thought every touchpad gesture was simple to use. Regarding nasal navigation, swipe worked quite well, but selecting items or returning to the previous level was very hard, and I was unable to do it consistently. Maintaining nasal control was also hard for me over time.
- What was the hardest?
  - (See above)
- What was most impressive to you?
  - I was impressed with how well swiping right with my nose worked.
- Which input device was most convenient to you?
  - (See above)
- Which input device was easiest to use?
  - (See above)
- What would you have done differently?
  - I'd wished there were some alternate ways of navigating with the nose. While swiping worked well, I think some things could be done to make the other gestures work better. For instance, changing the one seconds intervals could help, allowing me to spend more time fine-tuning my nasal flares rather than feeling like I have to make this cycle.