Eye gaze estimation and gesture recognition using mobile EEG
An application for sleep communication

Victoria Amo Olea

submitted on September 30, 2019
in partial fulfillment of the requirements for the degree of
Master of Science (M.Sc.)

First supervisor: Johannes Leugering
Second supervisor: Prof. Dr. Gordon Pipa

Institute of Cognitive Science
Universität Osnabrück

Abstract

Eye gaze estimation and gesture recognition systems offer a multitude of possibilities for human-computer interfaces. Many systems of this kind have been developed with medical purposes to help patients with paralysis to communicate via eye-typing. However, for research in lucid dreaming, these systems are inadequate since they are designed to work in combination with a grid presented on a computer screen, with open eyes, and in a sitting position. In this paper, we develop a real-time eye-gaze estimation and gesture recognition system using Electrooculography (EOG). The system is optimized to support eye typing interfaces in a sleeping setting. We measure EOG with the Traumschreiber, a multi-purpose mobile Electroencephalogram (EEG) system. Our software recognizes nine different eye-typing gestures and three command-gestures. We trained a Convolitional Neural Network (CNN) with data from one participant and tested it in a real-time setting with five different participants. The model reached an accuracy of 88% in a real-time setting by utilizing an adaptive threshold for detecting the beginning of a saccade. The results show that the developed eye-gaze estimation and gesture recognition system is adequate for eye typing in real-time while laying down with closed eyes. With this implementation, we contribute to the development of eye typing devices as well as to a tool that enables more fundamental research in lucid dreaming.

***Keywords*—** EOG-gaze estimation, EOG gesture recognition, eye-typing, sleep communication, lucid dreaming research.

1. Introduction

Sleep communication is a phenomenon that can take place within a distinct dreaming state known as lucid dreaming. In this state, the dreamer is aware that the experienced reality is a dream and can control eye movements at will (P. La Berge et al., 1981). This makes it possible for researchers to communicate with sleeping participants by decoding eye movement signals recorded with Electrooculography (EOG).

In this study, we develop a software for real-time recognition and classification of eye movements performed in a sleeping setting. Our purpose is to support the development of eye-typing interfaces for research in lucid dreaming. Accordingly, we define eye-movements and gestures to serve sleep communication purposes and optimize our prediction methods to work with closed eyes, in a laying down position, and with an imaginary 3x3 grid.

For our analysis, we record EOG signals with the Traumscheiber; a mobile electroencephalogram (EEG) developed by Appel and Leugering - at the department for Neuroinformatics at the University of Osnabrück - to facilitate experiments in the research field of sleep and dream. An advantage of the Traumschreiber is the possibility of processing data in real-time, allowing for online predictions with machine learning methods (Appel, 2018) . In this study, we use Convolutional Neural Networks (CNNs) to identify and classify eye-gestures for eye-typing.

This paper is divided into four sections. In the introduction (Section

1), we present background knowledge central to the understanding of our project and its motivation. Subsequently, we review related literature and present the state of the art for human-computer interfaces for eye-gaze typing and sleep communication, as well as techniques for eye-gaze estimation using EOG. After, in the contribution subsection, we shortly describe the main functionalities of our software. In the Methods section (Section 2), we explain the software, our machine learning approach for classifying eye-gazes and the method for identifying saccades in real-time for making predictions. Additionally, we describe the data collection procedure for training and testing the model. In the Results (Section 3), we present the accuracy of the CNN model tested online. In Section 4, we discuss the results. In Section 5, we address future work and the software's potential applications. We finish with a conclusion in Section 6.

1.1. Background

1.1.1. Electrooculography

Electrooculography is a method for recording eye movements by measuring the steady electric potential sustained between the front part of the eye, the cornea, and the back part of the eye, the retina. As observed for the first time by Emil du Bois-Reymond (1848), the cornea is positively charged in relation to the retina (Bois-Reymond, 1884). Since eye movements place in parallel the positive cornea closer to one side and the negative retina closer to the opposite side, they result in different electrical charges on different skin regions surrounding the eyes. Changes in the skin potential can be recorded with electrodes.

It is important to notice that the retino-corneal potential can be detected with closed eyes, even though the effect decreases as compared to when the eyes are open (Arden and Kelsey, 1962). In this study, we optimize predictions for sleep communication by training the machine learning model with data samples recorded in a closed eye setting as well as in an open eye setting. The goal is to provide higher variability in the data to make the model more robust. Variability mostly results from the difficulty of directing eye-movements with closed eyes and from the lack of a stable reference (such as a grid projected on a screen) to guide saccades precisely. In section 2.2.3 we discuss the difference between open and closed eyes settings in detail and report the exact amount of samples gathered from each condition for training purposes.

1.1.2. The Traumschreiber Project

The German word "Traumschreiber", composed by the words "Traum" and "Schreiber" can be translated as "dream writer" or "dream recorder". We also refer to it as the sleeping mask. Appel and Leugering initiated the Traumschreiber project during their doctoral studies at the department for Neuroinformatics at the University of Osnabrück, Germany. The goal of their project is to develop a low-cost mobile Electroencephalogram (EEG) system to facilitate research in the field of sleep and dream. Accordingly, the accuracy of the recordings allows for identifying sleep EEG features such as sleep spindles or slow waves and eye movements with Electrooculography (EOG). Moreover, it has proved useful for the recording of event-related potentials (Schalkamp, 2018; Reimann, 2018). For a detailed description of the project see (Johannes Leugering, 2017).

A central motivation of the Traumschreiber Project is the difficulty to carry out lucid dreaming experiments in a sleep laboratory setup. Such difficulties arise because of the little number of experienced lucid dreamers, from which only a few can have lucid dreams at will. Since participants only stay a couple of nights in the sleep laboratory, the chances of recording a lucid dream are low. Ideally, participants would have a mobile EEG at home to try the experiment multiple times, in a habitual sleep setting, and during a suitable period. The sleeping mask solves these problems and allows for crowd-based experiments, by letting multiple participants to perform experiments in parallel by using the Traumschreiber at home (Appel, 2018).

In the following, we describe the sleeping mask and its parts. The Traumschreiber is portable, small and light-weighted, designed to be worn while sleeping. It is composed of a plate containing connections for 10 electrodes that give place to eight channels and one reference channel. Channels can be used for EEG, EOG, Electromyogram (EMG) or Electrocardiogram (ECG). The battery and the connections for the electrodes are placed on the upper part of the plate. On its side, there is a port that serves to attach a mini USB speaker for presenting auditory cues or to charge the battery. Additionally, a small LED light system for visual stimulation is integrated beneath the plate. The sleeping mask transfers the recorded data in real-time via Bluetooth to other devices (e.g. computer, smartphone or Raspberry Pi) for further processing and analysis (Appel, 2018). Figure 1 shows the components of the sleeping mask.

1.1.3. Lucid dreaming

Lucid dreaming takes place when the dreamer becomes aware that he or she is dreaming. Aristotle wrote, "when one is asleep, there is something in consciousness which tells us that what presents itself is but a dream." (Aristotle, 1986). Other definitions are "dreaming while being conscious that one is dreaming" (P. La Berge et al., 1981) or "the experience of achieving conscious awareness of dreaming while still asleep" (Voss et al., 2009).

While in a lucid dream, the dreamer experiences wake-like coherence and clarity of mind, all embedded in a dream reality (Van Eeden, 1913). Studies on the neural correlates of lucid dreaming have found that the activity in this state, particularly in the prefrontal cortex, resembles more the waking state than the non-lucid dreaming state (Voss et al., 2009). It is a state characterized by the presence of metacognitive processes that are absent in normal dreaming (Windt and Metzinger, 2007; Filevich et al., 2015). In other words, by knowing that a dream is a dream, the lucid dreamer experiences the same dream reality at a sensory level but with an added dimension of awareness concerning the nature of the experience (i.e as dreamed instead of as real). Moreover, visual imaginary encountered while in REM sleep relates more to wake-like perception than to imagination (LaBerge et al., 2018). Lucid dreamers' reports highlight the capacity to freely direct attention and to have volitional control over actions and sometimes over the overall dream (LaBerge, 1985). A clear example of volitional control while lucid dreaming is sleep communication where participants communicate via eye movements with the external world (P. La Berge et al., 1981). Even though lucid dreamers are rare (Saunders et al., 2016), experiments testing methods for inducing the lucid state have shown that inexperienced persons can have lucid dreams (Appel et al., 2017; Stumbrys et al., 2012); lucid dreaming is considered a skill that can be trained (LaBerge, 1980). For example, a habitual meditation practice has shown to improve the capacity to lucid dream (Gackenbach and Alexander, 1986; Hunt, 1991). Practicing meditation to trigger lucid dreams is useful since meditation trains metacognition on thought processes.

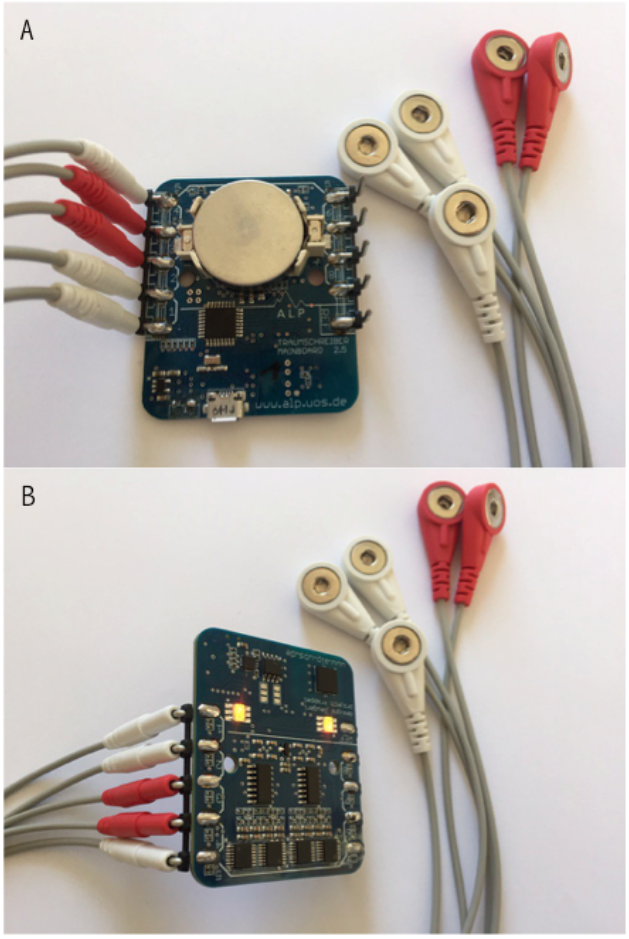


Figure 1. The Traumschreiber A)The Upper part contains the battery, connections for the electrodes, and a port for charging the battery or connecting speakers B) Beneath the plate the LED light system for visual stimulation is located.

Thereby, reflective awareness trained by recognizing thoughts as thoughts pervades to the dream state helping to recognize dreams as dreams.

Lucid dreaming is not new to the history of humankind, ancient cultures have reported it, written about it and trained it. To name a few, lucid dreaming is central to Dream Yoga in the Hinduist (Easwaran, 2007; Satyasangananda, 2003) and Buddhist (Lingpa et al., 2013; Mullin, 2014) tradition and is also present in Greek Philosophy (Aristotle, 1986). However, it was first scientifically demonstrated in the late 20th-century (Hearne., 1978; P. La Berge et al., 1981). Since then, researchers have become increasingly interested in lucid dreaming, its characteristics, and potential concerning multiple areas such as creativity, psychotherapy, philosophy, and sports (Appel et al., 2018). Besides that, lucid dreams offer a unique phenomenological tool for studying dreams, mind and consciousness (LaBerge, 1985; Wallace, 2012).

1.1.4. Deep Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a machine learning method best known for Computer Vision classification tasks, where they have proved extremely useful for image recognition (e.g. see (Krizhevsky et al., 2012)), outperforming all other machine learning methods. However, they are not limited to this field. CNNs serve for processing data with grid-like topology, also including time series (Goodfellow et al., 2016). Therefore it can be used for classifying biological signals. For example, CNNs have been implemented for classifying Electromyogram (EMG), Electroencephalogram (EEG), Electrocardiogram (ECG), and EOG data (for an overview see (Faust et al., 2018)). Concerning EOG, CNNs have been employed for drowsiness detection (Zhu et al., 2014), mental workload classification (Zhang et al., 2017) and sleep stage classification (Mousavi et al., 2019). However, studies applying CNNs to eye-gaze estimation using EOG are absent in the literature. There are studies working with CNN for eye-gaze estimation, yet, they use image data of eyes instead of the EOG signal (e.g. see (Zhang et al., 2018; Dimililer et al., 2018)).

In this study, we use CNNs for eye-gesture classification in order to estimate gaze directions. This method provides many advantages for making real time predictions. First, due to the fact that CNNs learn the parameters from raw data integrating feature extraction and classification of a candidate sample sequence in one architecture. The second reason why CNNs prove useful is their power to detect patterns independent of their position which is optimal since EOG patterns can take different positions along the x-axis. In section 2.5.2, we present the exact design and architecture of the CNN.

1.2. Related Work

In this section, we present the state of the art for eye-gaze interfaces, starting from general-purpose interfaces and proceeding with those that are lucid dreaming specific. We also review the literature of EOG eye-gaze estimation methods. Within this context, our software is at the interface between lucid dreaming research and human-computer interaction. It is a technology developed for sleep communication. However, at a general level, it is related to numerous eye-gaze communication interfaces developed for medical purposes. Moreover, on a technical level, it is concerned with methods for processing and analyzing the EOG signal.

Since the rise of computers in the 1980s, researchers have developed numerous Human-Computer interfaces (HCI) using eye-movement analysis and eye-tracking techniques, a method for sequentially estimating the direction of gazes in time. On the one hand, recognizing and analyzing eye movements has proved useful for understanding human cognitive processes and identifying eye-related anomalies (Rayner, 1998). Nowadays, eye-movement detection technologies serve, among others, for monitoring sleep stages (Kuo et al., 2014; Paisarnsrisomsuk, 2018), diagnosing central nervous system diseases (C Chen et al., 2000) and subnormal eye conditions (Güven and S, 2006). On the other hand, eye-tracking is well known for usability-evaluation studies (Poole and Ball, 2006), as well as for marketing and consumer research (Pieters and Wedel, 2017). However, it has also had a large impact on developing communication-oriented HCI for medical use (Majaranta and Räihä, 2002). For example, eye-tracking interfaces allow patients suffering from muscle paralysis (i.e Amyotrophic lateral sclerosis) to communicate via eye-commands (e.g. (Kim et al., 2018; Septanto et al., 2009; Heo et al., 2017; Yagi, 2010)) and eye-typing (Majaranta and Räihä, 2002), as well as to con-

trol wheelchairs (Barea et al., 2000; 2003; Kim et al., 2006a) and robotic arms (Ukken et al., 2015).

Regarding sleep communication, EOG is the established method for measuring eye movements during Rapid Eye Movement (REM) sleep. Green (1968) first presented the idea of communicating with a person while in a lucid dream. Keith Hearn (1978) applied this idea by developing the Dream Machine, which allowed participants to communicate with scientists from a lucid dream via eye movements (Hearne., 1978). Laberge (1980), used this technique to confirm volitional communication from part of the participants during verified REM sleep (P. La Berge et al., 1981). This method is still the state of the art method for sleep communication. Before starting an experiment, researchers and participants agree on the meaning of eye gestures. Thereby, some gestures such as the sign for lucidity (left-right-left-right) have become standard between studies. Other studies have used code language, such as Morse Code, to allow the dreamer to write complex messages (LaBerge, 2000). In most of the cases, researchers have to recognize the EOG pattern themselves by looking at the EOG signals in order to make sense of the message written by dreamers. Later technologies developed for lucid dreaming focus on triggering the lucid state via light or wave stimulation. LaBerge (1995) created the first lucid dreaming induction device (LaBerge and Levitan, 1995). Today, there is a large number of interfaces available on the market that can detect REM sleep and play audio and visual cues to the dreamer. However, to the best of our knowledge, there is no eye-typing interface implemented for sleep communication.

Concerning EOG data analysis, a large number of studies are devoted to the analysis of eye movements (i.e. fixation, saccades, blinks, and smooth pursuit), (e.g. see (Vidal et al., 2011; Salvucci and Goldberg, 2000; Behrens et al., 2010)). Fewer studies focus on developing methods for eye-gaze estimation and gesture recognition (see section 2.1) using EOG. In the following, we review some techniques employed by interfaces differentiating between four or more eye movements in an online setting, either for eye-gaze estimation, eye-writing or eye-typing: EOG channels threshold analysis (Aungsakun et al., 2011; Tsai et al., 2008b), Finite Automata (Trikha et al., 2007; Postelnicu et al., 2012), Fuzzy logic classification (Tsai et al., 2008a; Vandhana, 2014), Linear Discriminant Analysis (Wissel and Palaniappan, 2011), Hidden Markov Models (Kim et al., 2006b; Fang and Shinozaki, 2018), Clustering methods (Usakli et al., 2010; Bárcia, 2010; Desai, 2013), Support Vector Machines (Bulling et al., 2010), Neural Networks (Barea et al., 2000; 2003; Sherbeny and Badawy, 2006; Ramkumar and Hema, 2013; CR and MP, 2014), and Deep Neural Networks (Fang and Shinozaki, 2018) (for a detailed summary see (Ramkumar et al., 2018)).

It is important to notice that none of these studies has employed CNNs for eye-gaze classification (see section 1.1.4) nor developed specialized methods to differentiate and classify eye-gazes within a sleep setting (i.e. laying down with closed eyes). Furthermore, software for sleep stage classification does not distinguish between the direction of volitional eye movements but recognizes the occurrence of eye movements as a whole.

From above, we can see that eye-gaze devices for real-time eye-movement analysis and eye-tracking are not new, nor is eye-gaze communication for research in lucid dreaming. However, to the best of our knowledge, there is no eye-typing interface implemented for sleep communication. Medical research has brought about numerous eye-typing interfaces. Yet, until the date, lucid dreaming researchers carry out experiments with limited or very complex means of communication, making it difficult for participants to freely write with their eyes or control the input they are receiving via eye commands (e.g to select and start an experiment within the dream for crowd-based experiments). We can attribute this gap to the difficulty of implementing existing eye-typing interfaces to sleep communication. First, because most eye-typing interfaces employ camera-based eye-tracking devices instead of EOG, which is unsuitable for making predictions with closed eyes. Second, because most eye-typing devices are designed to measure eye movements in relation to a virtual keyboard displayed on a computer screen (Majaranta and Räihä, 2002), a method that is not feasible while lucid dreaming.

1.3. Contribution

Our contribution is an EOG eye-gaze estimation and gesture recognition software optimized for sleep communication. The software sets the ground for implementing eye-typing interfaces for lucid dreaming. The purpose is to allow lucid dreamers to control the mask, the input they are receiving, and to write freely with eye-gazes from within the dream. For this, we defined nine basic eye-gestures for typing in an imaginary 3x3 grid, composed of eight boxes surrounding a center box. Additionally, we define three complex eye-gestures for giving direct commands.

2. Methods

This section is divided into five subsections. In the first, we explain the design of our system for eye-gaze estimation and gesture recognition. The second part describes the data recording setup and procedure. In the third subsection, we present the methods used for data preprocessing and filtering. Subsequently, in the fourth part, we present the Convolutional Neural Network's architecture and explain the training process in detail. Finally, we explain our approach for recognizing saccades and selecting data windows (i.e epochs of EOG data points) to make predictions with the model in real-time.

2.1. Software description

We define eye-gestures in relation to a 3x3 grid and differentiate between two categories; Eye-typing gestures and command gestures. Eye-typing gestures are basic patterns composed of two or three saccades. They serve for selecting specific fields within the 3x3 grid. Command gestures, instead, do not have a fixed meaning attached and can be used for different purposes, such as turning a functionality on or off. In the following, we explain the typing grid, eye-typing gestures and command gestures in detail.

We would like to emphasize that we define the 9-keys typing grid by having in mind that predictive text technologies (i.e algorithmus that predicts possible words given incomplete information) make eye-typing with this configuration efficient. Predictive text has already been implemented for computer-based eye-typing with 9 keys and has proven useful (e.g. see (Zhang et al., 2018; Hansen et al., 2001)). Therefore, we recommend using our eye-gaze recognition in combination with a predictive text algorithm. To develop the software for the predictive text feature (e.g. T9) or a working version of the typing grid is out of the scope of this work, however, we explain the design of eye-gestures in relation to such configuration.

2.1.1. EYE TYPING GESTURES

Eye-typing gestures serve to estimate the position to which a saccade is directed. We distinguish between nine positions; *center*, *up*, *down*, *left*, *right*, *up-left*, *up-right*, *down-left* and *down-right*. Figure 2 shows the 3x3 grid with the positions and an example for distributing letters. This matrix, composed of eight boxes surrounding a center box, is designed to support a typing keyboard with eight typing-keys surrounding a central fixation-key for resting the eyes. To select a specific key, the user starts by fixating on the center, continues by gazing at a particular position and ends by returning to the center. Eye-gestures are always preceded and followed by a fixation period in the center position.

Box 1 abc Up-left	Box 2 def Up	Box 3 ghi Up-right
Box 4 jkl Left	Box 0 Center	Box 5 mno Right
Box 6 pqrs Down-left	Box 7 tuv Down	Box 8 wxyz Down-right

Figure 2. Imaginary 3x3 typing grid. Box 1 to 8 represent typing keys and contain a set of letters. The center position, box 0, is for the user to rest between gazes.

While recording data with closed eyes, we realized that eye-gazes along the diagonals are inconsistent between trials. Participants tend to look close to neighboring keys and not in a straight line to the diagonals. As a result, patterns belonging to different keys look very similar. To avoid this problem, we define a unique gaze-path to reach each key. Unique paths help to create a distinct EOG pattern for each position. Figure 3 illustrates the eye-gesture path for reaching each key. Figure 13, in the appendix, displays each gesture together with an exemplary resulting EOG pattern.

2.1.2. EYE TYPING COMMANDS

Command-gestures allow the dreamer to communicate without needing to type a word or a message. They can take different meanings depending on the interface. For example, gestures can be useful to turn the typing system on or off, to activate a functionality or to change between screens if the interface has more than one (e.g. main menu, letters grid, numbers grid, grid with fixed answers and so on). Figure 4 summarizes and illustrates command-gestures. The corresponding EOG pattern for each command-gesture can be found in the appendix (Figure 13). In this study, we include three basic command gestures; *Lucid-sign*, *upper rectangle* and *lower rectangle*; however, new gestures could be added to meet the needs of specific typing interfaces.

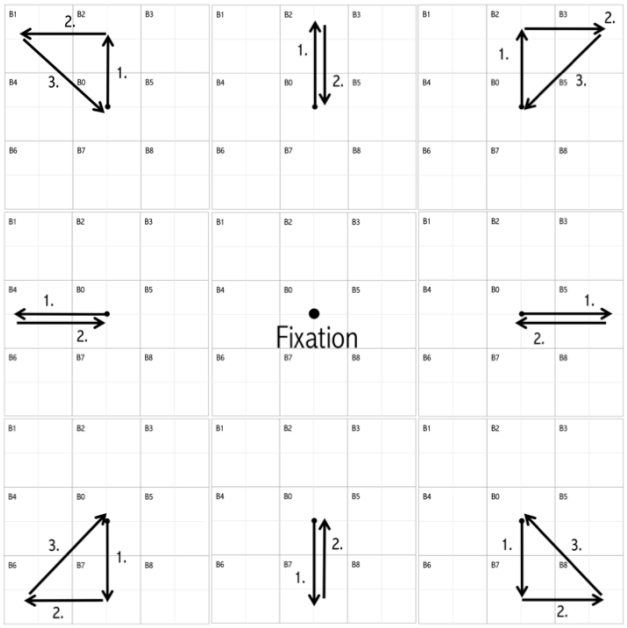


Figure 3. Typing grid. Eye gestures for typing 1) *Up-left*: up-left-center 2) *Up*: up-center 3) *Up-right*: up-right-center 4) *Left*: left-center 5) *Right*: right-center 6) *Down-left* : down-left-center 7) *Down*: down-center 8) *Down-right*: down-right-center. Additionally, fixation in the center position serves as a resting point between gestures.

2.2. EOG setup and electrodes configuration

2.2.1. MOBILE EEG SETUP

For recording data, we set up the Traumschreiber as follows. For the signal amplifier, we set the gain to 4, which captures saccadic movements while keeping the noise in the data at a low level. Low gains are adequate for capturing rough changes in the signal such as saccades or muscle tone. For measuring finner signals changes such as ERPs, the gain needs to be set higher. To measure the EOG signal, we use hydrogel disposable electrodes.

2.2.2. ELECTRODES CONFIGURATION

To measure vertical and horizontal eye movements, we record data from four channels (denoted as Ch0, Ch1, Ch2, and REF) resulting from the combination of four electrodes (E1, E2, E3, E4) and one reference electrode (R). Figure 5 illustrates the configuration of the electrodes. The signal of a channel is calculated by subtracting the value of one electrode from another (i.e. Ch0=E1-E2, Ch1=E3-E2, Ch2=E3-E4 and REF=R-E4). Additionally, we re-reference the data by subtracting the value of the REF channel from Ch0, Ch1 and Ch2. To measure horizontal eye movements, we placed one electrode near the lateral canthi of each eye. The resulting channel, HEOG or Ch0, presents negative voltage values for left

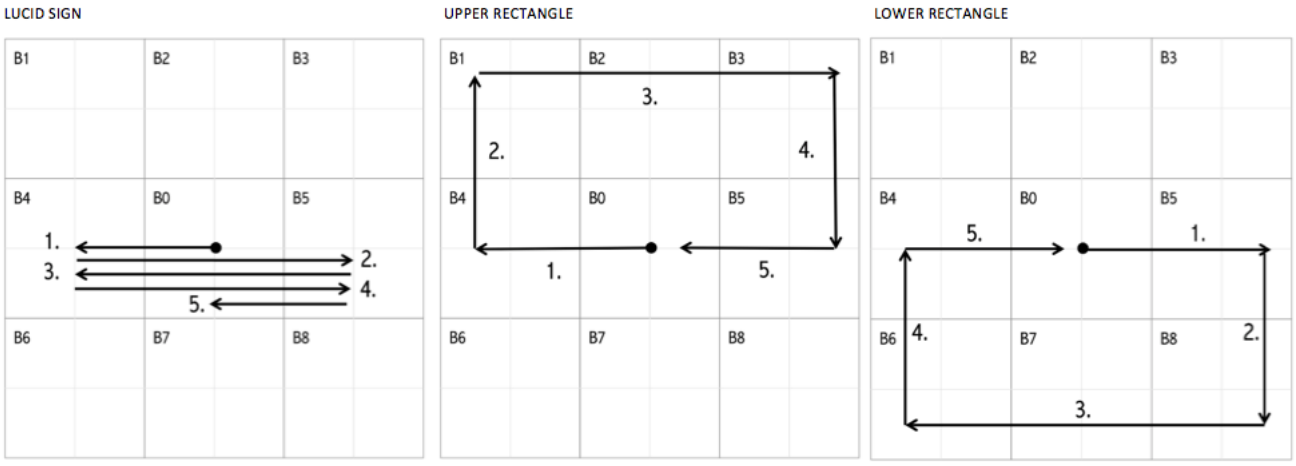


Figure 4. Command gestures are composed of the following sequences 1) *Lucid sign* by left-right*-left*-right*-center 2)*upper rectangle* by left-up-right*-down-center and 3) *lower rectangle* by right-down-left*-up-center. The sign "*" marks gazes crossing the entire grid.

gazes and positive voltage values for right gazes. The second channel, denoted as Ch1, is a combination of electrodes belonging to the vertical and horizontal channels. To measure vertical eye movements, we placed one electrode above and one below the left eye. Channel VEOG or Ch2, turns negative by down gazes and positive by up gazes. The last electrode is placed right in the middle of the forehead and yields the reference channel. The values recorded by the reference channel, which measures the base electrical potential, is subtracted from all other channels to reduce noise in the data.

2.3. Data recording

As a standard procedure before starting a data recording session, either for training or testing the machine learning model, participants read and signed a consent form concerning their participation in the study and the use of their data. After signing the form, participants washed their faces with soap and cleaned the areas for placing the electrodes with alcohol pads.

In the following, we explain why we record training data with both, open and closed eyes. Subsequently, we describe the datasets used for training and testing the machine learning model in detail.

2.3.1. Open versus closed eyes samples

Data recorded with open eyes, by following a visual queue presented in a screen, is more precise than data recorded with closed eyes by following auditory cues. Whereas EOG patterns are consistent in the open eye condition, they show much more variance in the closed eye condition. Figure 6 shows exemplary EOG patterns for the gestures *left* and *right*, for the closed or open eyes condition. For a similar display of all remaining gestures (comparing the EOG signal for the open a closed eyes setting), see Figure 13 in the appendix. When training the model we realized that predictions for gestures performed with closed eyes are improved when the model was trained with samples from both the open as well as the closed eyes condition. We attribute this to the clarity of the patterns as expressed in open eyes data, which makes it easy to recognize, map and learn the most important features of each saccade. Therefore, we take a "transition" approach to train the model in which we include data from trials with open eyes as well as data from trials in a closed eyes condition. This approach helps us create more variance in the training data and for the neural network to learn the shared characteristics between them.

2.3.2. Data recording for training the model

We trained the model with data belonging to one participant in two conditions; sitting with open eyes and laying down with closed eyes. In the first condition, the participant sat in a chair and followed visual cues presented on a computer screen. The participant sits at a distance of 60 cm to the screen, measured from the eyes. Additionally, the screen height is adjusted such that the resting gaze of the participant points to the center of the grid. For the second condition, the participant lay down on a bed, placed the head on a pillow and rested the gaze in a central position with closed eyes. For making gazes we asked the participant to follow the instructions presented by auditory cues using an imaginary 3x3 grid. We recorded 6 runs with open eyes, each of which contains a complete sequence of gestures (i.e typing-gestures and command gestures) in the order: *up-left*, *up*, *up-right*, *left*, *right*, *down-left*, *down*, *down-right*, *lucid sign*, *upper rectangle*, *lower rectangle*. For the closed eyes condition, we recorded each typing-gesture in isolation. The intention is not to confuse the participant by constantly changing the gesture. Thereby, a run is composed of 40 iterations of a specific gesture. For command-gestures, we recorded three runs, each containing 10 iterations of each. Within a run, after each iteration, participants fixate in the center position for three seconds and then rest for five seconds before a new gesture starts. Table 1 summarizes the number of iterations recorded for each gesture for each condition. For training the CNN model we augment this dataset (see section 2.5.1).

2.3.3. Data recording for testing the model

We measure the accuracy of the CNN model in combination with the method for selecting windows (section 2.5.2) in a real-time setting and with data belonging to 5 participants, 3 runs each, gathering a total of 15 runs for testing the model. A run contains

Condition	Instructions	Position	Grid	Typing gestures	Command gestures	Fixations	Total #
				#gestures x #Iterations x #runs	#gestures x #Iterations x #runs	#gestures x #Iterations x #runs	
Open eyes	Visual cues	Sitting	On screen	8 x 1 x 6 = 48	3 x 1 x 6 = 19	(7 x 1 x 6) +(3 x 1 x 6)= 60	186
Closed eyes	Auditory cues	Laying down	Imaginary	8 x 40 x 1 = 320	3 x 10 x 3 = 90	(39 x 1 x 8) +(3 x 1 x 3)= 321	731

Table 1. Total number of samples collected for training the CNN model. We recorded a different number of samples for the open eyes and closed eyes condition. The majority of the samples are recorded with closed eyes.

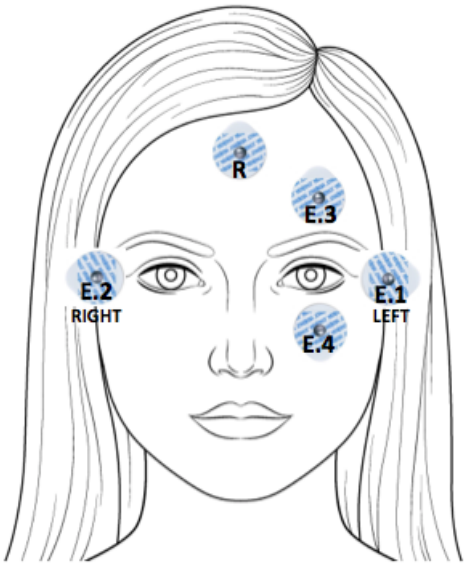


Figure 5. Electrodes for measuring horizontal eye movements are placed close to the lateral cathi of each eye (Ch0 or HEOG = E1 - E2). Vertical eye movements are recorded by a pair of electrodes positioned above and beneath the left eye (Ch2 or VEOG = E3 - E4). The reference channel measures alongside in the middle of the forehead (REF = R - E4).

all gestures in the order: *up-left*, *up*, *up-right*, *left*, *right*, *down-left*, *down*, *down-right*, *lucid sign*, *upper rectangle*, *lower rectangle*. We recorded data with participants in a laying down position with closed eyes and presented instructions via auditory cues.

2.4. Data pre-processing

Before analyzing the EOG data, we use standard filtering methods to obtain a clear signal, which models saccadic movements and is free of noise produced by other sources such as the power line or electrical devices. For filtering as well as for normalization, we always use the last 3000 data points, which precede and contain the sample to be filtered. First, we use a median filter with kernel size = (3,1) to avoid possible outliers in the signal. Subsequently, we use a Butterworth filter (Critical frequency = 9, Nyquist frequency = 0.034, type = digital low-pass-filter), which flattens the signal as much as possible outside the passband. We use a low-pass filter to eliminate high-frequency noise, thereby we obtain a smoother signal reflecting the trend of saccadic movements and ignoring small fluctuations in the data. To finish, we rescale the epoch implementing equation 1. Here, we subtract the mean of the complete sample $\bar{x}$ from each data point x_i and divide it by the

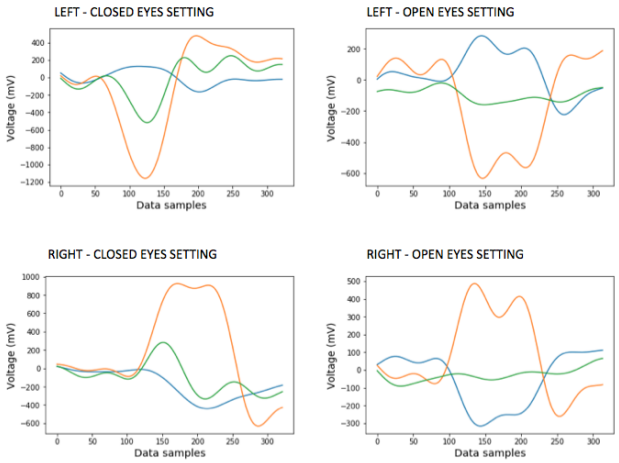


Figure 6. Open versus closed eyes EOG patterns. Changes in the signal for HEOG (blue line) are more pronounced in the open eyes than the closed eyes condition. Furthermore, for the closed eyes condition signals tends to peak instead of forming a plateau.

sample range (V_{max} - V_{min}). Scaling the data helps to make it more consistent between subjects and runs.

$$x_i = \frac{x_i - \bar{x}}{V_{max} - V_{min}} \tag{1}$$

After filtering, the resulting data set contains filtered and centered voltage measurements. Figure 7 shows the pattern *up-right* before and after filtering.

2.5. Convolutional Neural Network

CNNs differentiate from traditional artificial neural networks because they utilize convolutions instead of a full connectivity matrix between layers. The parameters in convolution layers are learned for a set of smaller filters that are moved systematically over the input data structure (Goodfellow et al., 2016). The performance of CNNs is attributed to their ability to represent highly nonlinear functions by a composition of nested nonlinear activation functions and linear transformations (Goodfellow et al., 2016). Furthermore, CNNs seem to learn features that are translation invariant. This is important for image processing tasks, where objects can appear in any location within the image, but also for the problem of recognizing eye gaze, as patterns of saccade movements can occur at different time points within a recording.

Here, we utilize several standard building blocks of CNNs, which

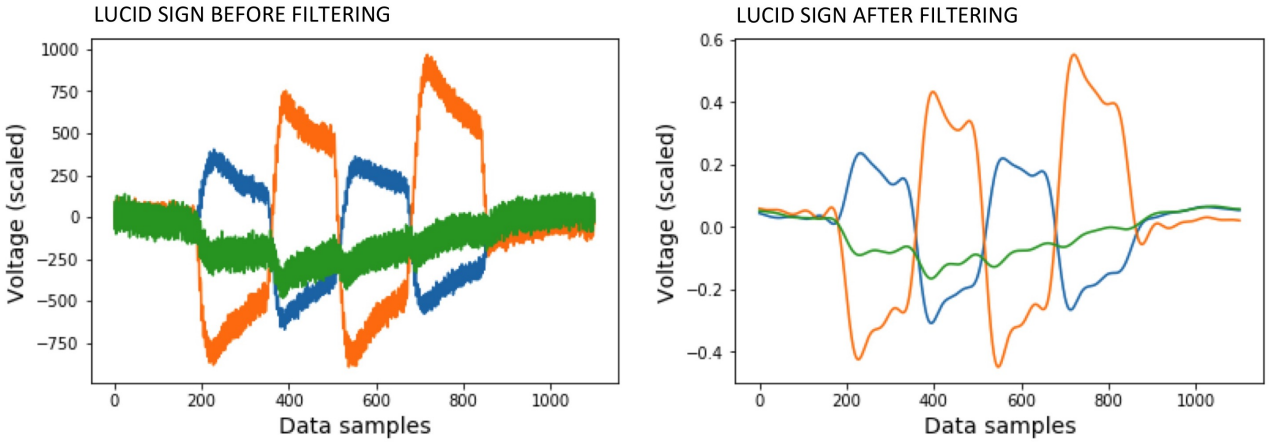


Figure 7. The left plot shows the recorded EOG traces of a subject performing the lucid sign gesture. The right plot shows the same data after filtering and scaling have been applied.

take the activation of a previous layer as input and produce activations on their own to be passed to the following layer. We use linear convolutions that act locally on parts of the input, followed by a nonlinear activation function, namely a rectified linear unit (ReLU). ReLU is applied to the output of the linear convolutions. Furthermore, we use nonlinear pooling or aggregation layers that summarize local activations and yield more robustness in activation levels against perturbations within the input. We also make use of batch normalization (Ioffe and Szegedy, 2015) to correct for covariance shift within the training data and to speed up the learning as well as a Dropout layer (Srivastava et al., 2014) to prevent the network from overfitting the training data.

Moreover, we use data augmentation in order to provide input data that is distributed more evenly across aspects of the input against which the model should be invariant, i.e. the time of the onset of saccadic movement within the recording. In the following section, we describe the process to generate augmented data in detail.

2.5.1. Augmenting data for Deep Learning

Data augmentation is a popular method for generating more training samples for CNNs. It consists of generating multiple training instances from one sample by applying small transformations. The purpose is to produce a bigger set of data with realistic variability from a limited set of data.

One characteristic of real-time data processing is that the segmentation of saccades is not clear; therefore, the model has to be trained to recognize patterns of saccades located at different positions within a data window. Furthermore, the model has to recognize fixation periods that can partly contain the start or the end of a saccade. To produce training samples that are representative of these characteristics, we apply a moving window of 700 points over each sample. The window shifts over a defined range, which includes periods previous and following to the occurrence of the saccade. As different gestures have different lengths, we define a specific shift range for each type of saccade. Thereby we can capture each gesture at different positions without creating samples with incomplete patterns. Specific ranges are crucial to be consistent in the labeling process because basic gestures (e.g. *Left*: left-center) constitute more complex gestures (e.g. *Up-left*: up-left-center), hence incomplete patterns can result in samples containing the same pattern but different labels (e.g. If the *Up-left* gesture only contains the last two saccades). We choose the ranges by visually inspecting the data. The defined ranges are adequate for a window size of 700 points and fixation periods of a minimum of 3 seconds between gestures. Additionally, gestures are expected to take place within 3.5 seconds. We train the model with a total of 12 labels. Table 2 summarizes the ranges defined for each gesture, the labels and the number of samples after augmentation.

2.5.2. Model description

The input to the model are patches of 700 data points for each of the 3 channels with voltage, velocity and acceleration information, yielding a tensor of size (3, 700, 3) for each labeled data sample. The labels are according to the defined classes described in the previous section.

The model itself consists of a series of 5 convolutional layers followed by one fully connected layer. In each of the 5 convolutional layers, 20 filters are learned. The size of the filters along the sampling axis decreases with increasing layer number - In layer 1, a filter spans 13 points along the sampling axis, with 7, 5, 3 and 3 points along the sampling axis in the following layers, respectively. The filter size along the EOG-channel axis of the input tensor remains fixed at a value of 3. The padding for all convolutional layers is set to 'same', such that the output size of the layer after application of the filters stays the same along the original sampling axis as well as the EOG-channel axis of the data. The feature axis (i.e. the axis carrying voltage, velocity, and acceleration in the original data) is substituted by the result of the application of the 20 learned filters of the convolution layers such that the size of the output tensor of the first layer is (3, 700, 20).

After each convolutional layer, batch normalization and max-pooling are performed. Importantly, max pooling is only performed across 3 points along the sampling axis such that the filter size for the max-pooling filters after all convolutional layers are (1,3). The strides of the max-pooling layer are set to (1,2) such that the size of the output is halved after every application of a

Label	Shift range	Closed eyes samples	Open eyes samples	Total # of samples
	(start, end, shift)	(shifts x iteration x runs)	(shifts x iteration x runs)	
Fixation	(-60, 50, 10)	12 x (10 x 40) x 1 = 4800	12 x (10 x 1) x 6 = 720	5520
Up-left	(-150, 0, 10)	16 x 40 x 1 = 640	16 x 1 x 6 = 96	736
Up	(-350, 0, 10)	36 x 40 x 1 = 1440	36 x 1 x 6 = 216	1656
Up-right	(-150, 0, 10)	16 x 40 x 1 = 640	16 x 1 x 6 = 96	736
Left	(-250, 0, 10)	26 x 40 x 1 = 1040	26 x 1 x 6 = 156	1196
Right	(-250, 0, 10)	26 x 40 x 1 = 1040	26 x 1 x 6 = 156	1196
Down-left	(-150, 0, 10)	16 x 40 x 1 = 640	16 x 1 x 6 = 96	736
Down	(-350, 0, 10)	36 x 40 x 1 = 1440	36 x 1 x 6 = 216	1656
Down-right	(-150, 0, 10)	16 x 40 x 1 = 640	16 x 1 x 6 = 96	736
Lucid sign	(-30, 30, 10)	7 x 10 x 3 = 210	7 x 1 x 6 = 42	252
Upper rectangle	(-30, 30, 10)	7 x 10 x 3 = 210	7 x 1 x 6 = 42	252
Lower rectangle	(-30, 30, 10)	7 x 10 x 3 = 210	7 x 1 x 6 = 42	252

Table 2. Overview of the data samples that constitute the dataset, including the data augmentation. Each row refers to a gesture. The total number of samples per gesture is the sum of the augmented data samples from closed eye as well as open eye condition. The number of samples per gesture varies, depending on the shift range used to augment the data.

max-pooling layer.

Just before the output layer and after the convolutional layers follows a single fully connected layer with 20 neurons. All layers, despite the output layer, use rectified linear units as activation functions. The output layer uses softmax as activation function. Categorical cross-entropy is optimized by the Adam optimizer. Training is performed for 20 epochs with a batch size of 100. The described architecture is displayed in Table 3.

As our input data is effectively time-series data, the use of 1D convolutions to learn filters along the time axis would be a straightforward choice. However, we choose spatial 2D convolutions instead of 1D convolutions in order to learn filters that represent the interdependencies between the EOG channels. In all convolutional layers, we learn filters with a height of 3, which represents the 3 different EOG channels. Due to the padding with 0 values around the input data, the result after application of the convolutional filters of height 3 remains the same. For the topmost row, channels 1 and 2 are combined in addition to the "padding channel" that consists only of zeros. For the middle row, all three EOG channels are combined. The bottom-most row is again the result of two EOG channels, namely channel 2 and channel 3, in addition to a padding channel including only zeros.

A further aspect of the chosen model architecture is the decreasing widths along the sampling axis of the 2D convolutional filters (13, 7, 3, 3, 3). Though the width decreases in the convolutional filters, due to the pooling along the sampling axis, the net receptive field of the filters increases with respect to the input of the network.

Layer	Input Shape	Operation	Params
1	3 x 700 x 3	20 x Conv2D (3 x 13)	2360
	3 x 700 x 20	BatchNormalization	80
	3 x 700 x 20	MaxPooling2D (1 X 3)	0
2	3 x 350 x 20	20 x Conv2D (3 x 7)	8420
	3 x 350 x 20	BatchNormalization	80
	3 x 350 x 20	MaxPooling2D (1 X 3)	0
3	3 x 175 x 20	20 x Conv2D (3 x 5)	6020
	3 x 175 x 20	BatchNormalization	80
	3 x 175 x 20	MaxPooling2D (1 X 3)	0
4	3 x 88 x 20	20 x Conv2D (3 x 3)	3620
	3 x 88 x 20	BatchNormalization	80
	3 x 88 x 20	MaxPooling2D (1 X 3)	0
	3 x 44 x 20	Dropout (0.5)	0
	3 x 44 x 20	Flatten	0
5	2640	Dense1	52820
	20	BatchNormalization	80
6	20	Dense2	252
	12	Softmax	0

Table 3. The architecture of the model. We use four convolutional layers and two fully connected layers. The total number of tuneable parameters of the model is 73892. For a detailed description of the model architecture please see the text.

2.6. Real-time prediction

2.6.1. Data exchange

To work with the model and predict eye movements in real-time, we use Python scripts and exchange data between scripts with the help of the Robot Operation System (ROS) publishers and subscribers. First, the Traumschreiber sends information via Bluetooth to the computer at an average frequency of 220 Hz. Data

points are collected with a callback system that starts processing data points at the moment of being received. After re-referencing channels (see section 2.2.2), the values are published together with a timestamp (nanoseconds accuracy) from the moment they were received. For further preprocessing and analysis of the data, we subscribe to the above-mentioned publisher from another script.

2.6.2. Selecting window for prediction

In real-time predictions, users can make eye movements at any time. As we analyze data in discrete windows, it can happen that a saccade takes place at the intersection of two windows, resulting in incomplete patterns and incorrect predictions for both. Our goal is to provide the model with data windows containing complete saccades for making predictions. For this purpose, we move a window of fixed size along the data to ensure we capture the onset of a saccade and select a data window that starts from this position. In the following, we explain the method for saccade identification and window selection in detail.

In order to recognize the occurrence of a saccade, we extract and analyze a 700 points data window each time 50 new points are available. To recognize saccades, we implement an adaptive threshold that is a modified version of the threshold presented by Behrens et al. (2016) /citeadaptivthreshold. In this study, the authors define a threshold for differentiating saccades from fixation by using acceleration values. The threshold is calculated as the mean of the standard deviation from the preceding 200 acceleration points, multiplied by a factor N =3.4. The resulting value constitutes the upper and lower threshold (in negative). Saccades occur when the acceleration curve crosses the upper or lower threshold. If the acceleration curve stays within the margins of the thresholds, it signalizes fixation. We calculate the threshold as the mean of the standard deviation of the acceleration. However, we update the threshold for each epoch by using the last 3000 acceleration points and set N = 2.3. If the signal crosses the threshold, we extract a 700 points window starting 50 data points previous to the saccade's onset. Figure 8 shows an exemplary acceleration data window for the command-gestures together with the adaptive upper and lower threshold.

3. Results

In this section, we report the accuracy of the CNN model in combination with the system for identifying saccades (see section 2.5.2) in a real-time setting. The dataset used for testing is explained in section 2.3.3.

We tested the accuracy of the CNN model with data from 5 participants obtaining an average accuracy over subjects and gesture types of 88%. Figure 9 shows the accuracy reached by each participant for each run. Here, we can observe that the system performs very good for some participants, but not for others. For example, for participants one and two, the predictions are 100% and 95% accurate respectively, while the prediction for participant four is lagging behind with 76% accuracy.

The percentages presented in the plot above are calculated by considering each gesture once (11 in total) plus the periods of fixations in between them (10 fixations). Since fixations are predominant and have very high accuracy, it is important to look at the prediction accuracy for each gesture in detail. The confusion in Figure 10 summarizes the errors made by the system for all participants. Here, we can see that some gestures are consistently confound with another gesture. For example, *down-right* is always confound with *right* (53% of all instances) and *up-left* with *left* (33% of the instances)

Figure 11 shows the scores for precision, recall and f1 score for each of the classes separately.

4. Discussion

In this section, we discuss the results and the main causes underlying incorrect predictions.

When comparing the accuracy belonging to different participants, we can see that the model does not perform equally well across participants. By visually inspecting the data and by plotting the incorrectly classified samples, we realized that the accuracy of the model is highly dependent on two factors: the performance of the saccade detection system and the size and clarity of the saccades performed by the participants.

The accuracy of the model depends on the pattern contained in the data window that is selected for prediction, which in turn depends on the saccade detection system. If the selected window does not contain a gesture pattern in its entirety, the model prediction is bound to be incorrect. This is often observed in the data of participants who perform small size saccades, for which the multiplicator N = 2.3 (see section 2.5.2) results to be too high. If the threshold is set too high (which can happen even though the threshold is adaptive), the system can fail to correctly detect the onset of a saccade or the saccade all together. Incorrect onset detection results in prediction samples containing incomplete patterns, while failing to recognize a saccade leads to omitting that prediction. Setting the multiplicator N to a smaller number does not solve the problem because then, fixation periods and smaller involuntary eye movements can be confounded with saccades.

Regarding the clarity of saccades, some participants tend to do large and clear eye movements that draw the expected pattern in the EOG signal (see fig 13. in the appendix). However, other participants do smaller and/or erratic saccades that result in EOG patterns that are more difficult to detect and distinguish. Figure 12 contrasts the gestures *up-left* and *left* from run number one, belonging to participant 1 (100% accuracy) and participant 5 (80% accuracy). Here it becomes apparent that the "up" component of the gestures belonging to participant 1 is more pronounced as compared to the same component from participant 5. For the latter participant, the up saccade is small such that the initial part of diagonal gestures is unclear. Since the model was trained with distinct saccades, patterns are confound for participant 5 but not for participant 1.

In Figure 10, we can see that most misclassifications happened between basic gestures embedded within complex gestures. For example between *left* and *up-left* or *down-right* and *right*. The two problems presented above lead to this error. For example, on the one hand, if the window only contains the last part of the pattern up-left, the prediction will be left, which is false only because the system failed to recognize the onset of the saccade. On the other hand, if the onset is recognized but is not pronounced sufficiently, *up-left* is predicted as left because it resembles the left pattern more than the up-left sample in the training data. This explains why we can distinguish between two groups of participants, one for which the system performs optimally, obtaining excellent results, and another group for which the system does not perform well. It also explains why the prediction accuracy is consistent within participants. Furthermore, within participants, in many cases, the model makes the same errors repeatedly. For example, for

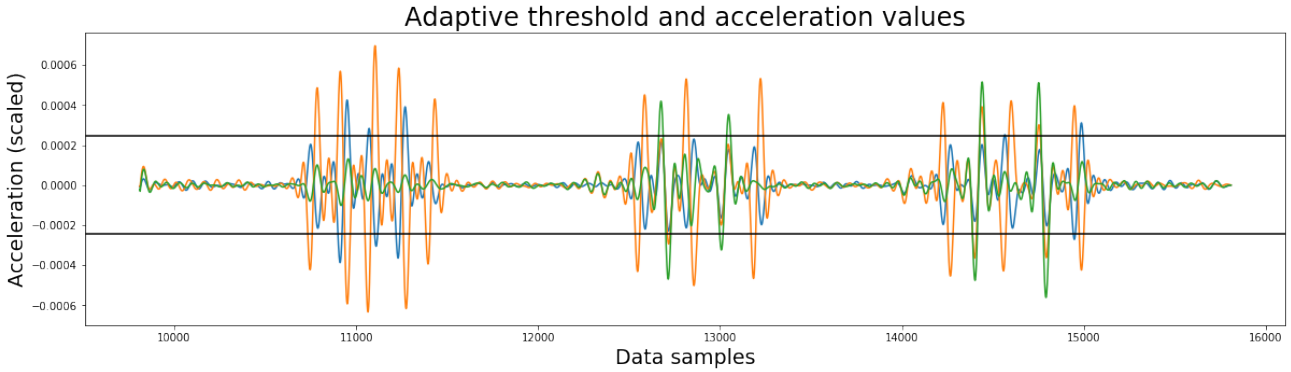


Figure 8. Adaptive threshold. In the plots we can see the tree command gestures (*lucid sign*, *upper rectangle* and *lower rectangle*) and how saccades are distinguished from fixation by the adaptive threshold (black lines).

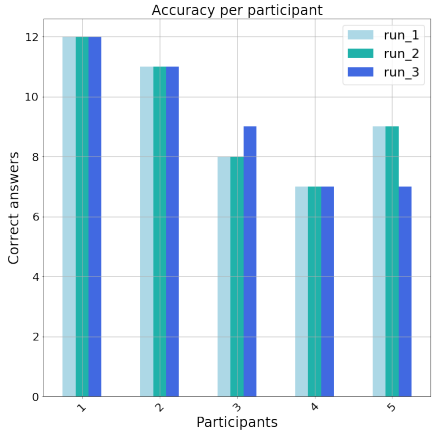


Figure 9. Accuracy per participant: The proportion of correctly predicted labels out of the samples of a run are displayed. Three bars for each participant represent the performances of each of the three runs. One run consists of 11 gestures and 10 fixation periods in between.

participant 4, *up-left* and *down-left* were confound with *left* twice.

For training the model, the participant practiced doing eye movements with closed eyes several times, which helped her to obtain a distinct pattern for each sign in this condition. This was not the case for the rest of the participants who had two runs for practicing (even though participants 1 and 2 reached high accuracy without such training). One way to obtain better results would be to prolong the time participants are allowed to train before recording data for testing. Additionally, it would be important to train the model with more variability regarding the size and clarity of the saccades and calculate the multiplicator N for each subject depending on the average size of saccades.

5. Further work

In this section, we present a possible direction to further develop the eye-gaze estimation and gesture recognition system, as well as ideas for usage of the system.

The first step, before implementing the typing grid, is to train the model with data from more participants, for it to generalize move evenly between users. The second step is to add relevant functionalities such as the detection of artifacts and a feedback mechanism to corroborate the right predictions or report false ones. Regarding artifacts, the CNN model could be trained to recognize noise coming from muscle tone (e.g teeth grinding) or changes in position. Regarding the feedback mechanism, including a short gesture could be useful for giving feedback to the system after a prediction. For instance, looking with both eyes to the nose to confirm that a predicted word is correct. In this case, the mask could play auditory cues to the user to notify each prediction. Another possibility would be to recognize the Error-related negativity (ERN) Event-related potential (ERP). ERN captures the brain response (largest at frontal and central electrode sites) when an error is recognized, either made by oneself or by someone else (Scheffers and Coles, 2000; Kreilinger et al., 2012). ERN has already been implemented for Brain-computer-interfaces and has proven useful as a feedback mechanism (Chavarriaga et al., 2014). By applying this method, the mask could extract feedback directly from the brain after informing each prediction.

After integrating the above-mentioned functionalities to the eye-gaze estimation system, the next step would be to implement the typing grid and to integrate it with a computer to control sounds and lights. Subsequently, it would be relevant to test the system with participants while lucid dreaming. Additionally, it would be advisable to develop a software for participants to learn and practice how to type with the imaginary grid within a dream (e.g. using virtual reality).

We think that this system for writing and for controlling the input to the dream has huge potential for both training and research, especially for lucid dreams, consciousness studies, research in cognitive capacities (i.e memory, perception, time perception), and learning. In the following, we share some of our ideas concerning these areas.

Crowd-based experiments: The typing grid would allow par-

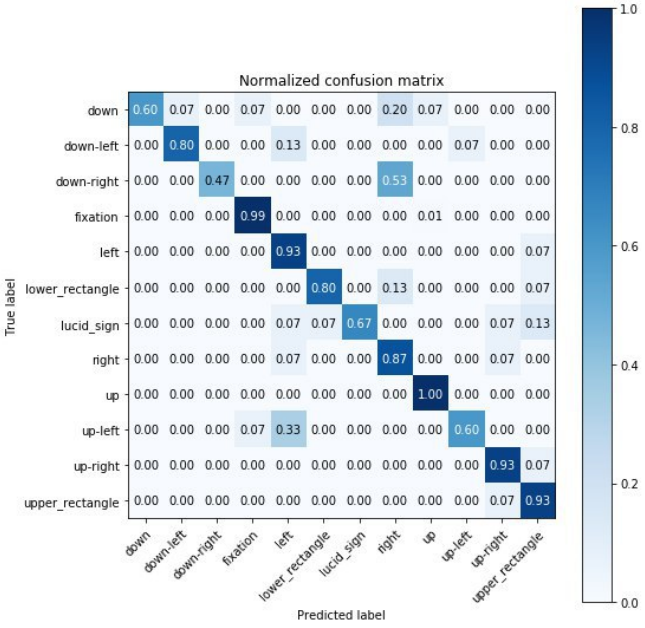


Figure 10. Normalized confusion matrix: The proportions for all pairs of true class versus predicted class are displayed. Some pairs of classes are more often confused than others. For an analysis of this matter, please see the text.

	precision	recall	f1-score	support
down	1.00	0.60	0.75	15
down-left	0.92	0.80	0.86	15
down-right	1.00	0.47	0.64	15
fixation	0.99	0.99	0.99	150
left	0.61	0.93	0.74	15
lower_rectangle	0.92	0.80	0.86	15
lucid_sign	1.00	0.67	0.80	15
right	0.50	0.87	0.63	15
up	0.88	1.00	0.94	15
up-left	0.90	0.60	0.72	15
up-right	0.82	0.93	0.87	15
upper_rectangle	0.74	0.93	0.82	15
accuracy			0.88	315
macro avg	0.86	0.80	0.80	315
weighted avg	0.91	0.88	0.88	315

Figure 11. Precision, Recall, F1 score and support (i.e. number of samples) for each individual class as well as the averages. The values refer to the test data.

ticipants to select and start experiments from within the dream. Thereby they could take part in multiple experiments while having the sleeping mask, by simply selecting a different experiment each time.

Customized grids for experiments: By using the eye-gaze software, one can create typing grids as needed and also superimpose grids with different functionalities. For example, from the main grid one could select between eight other grids. This could contain any content such as letters, numbers, predefined answers, emotions, degrees or levels (e.g lucidity, the intensity of emotions, the materiality of the perceived dream, and so on) as well as shortcuts for audiovisual content, experiments, and training sessions.

Data recording and labeling: Recording and labeling data in real-time while lucid dreaming would be possible by typing the label before performing an activity in the dream and then performing in the time interval between two tones played by the mask. For example, one could record data from flying by activating this functionality and typing the label "Flying". Then the mask could play a tone for the dreamer to start performing the activity and another to stop. This method would be useful to see if some dream activities can be recognized by the mask and also for having exact data for comparing activities performed while being awake with those performed while lucid dreaming.

Train skills: The mask could be used to start a guided training session for sports, meditation, psychotherapy or learning. Furthermore, training sessions could be used to investigate whether the lucid state is conducive to such activities and, if yes, how it behaves when compared to the waking state. The system can help to explore questions like: Can we learn languages better while lucid dreaming than when awake? Is lucid dreaming conducive for meditation? Can we learn new motor abilities while dreaming? Can we create or break habits while lucid dreaming?

Consciousness studies: The mask could help the dreamer to explore how different inputs such as audio or light affect the dreamscape (e.g sounds of nature or animals). Additionally, it could allow participants to notify levels of lucidity while dreaming given different conditions (e.g hours of the night, meditational practices and other techniques). The mask could save this information in a journal for further analysis. Besides this, an interesting question is whether it is possible to remain lucid while transitioning from a lucid dream to non-REM sleep. For example, to investigate this the mask could give deep sleep wave stimulation to a subject within the lucid dream, in order for him or her to train extending lucidity to other phases of sleep.

6. Conclusion

In this paper, we developed a real-time eye-gaze estimation system and defined unique gestures to perform eye typing in a 3x3 grid. We specialized our system to recognize eye movements performed in a sleeping setting and defined eye gestures accordingly. The system recognizes nine different eye-typing gestures and three command-gestures. We trained a CNN model with data from one participant and tested it in a real-time setting with five different participants. The model reached an accuracy of 88% in a real-time setting by utilizing an adaptive threshold for detecting the beginning of a saccade. The results show that the developed eye-gaze estimation and gesture recognition system is adequate for eye typing in real-time while laying down with closed eyes. With this implementation, we contribute to the development of eye typing devices as well as to more fundamental research in lucid dreaming.

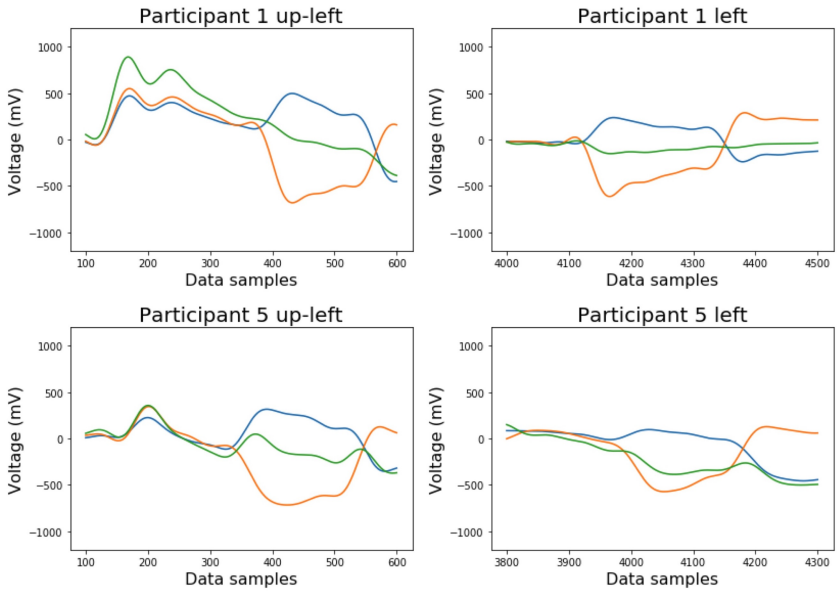


Figure 12. Clarity of EOG pattern: comparison between participant 1 and 5 for the gestures *up-left* with *left*.

References

K Appel, S Füllhase, S Kern, A Kleinschmidt, A Laukemper, K Lüth, L Steinmetz, and L Vogelsang. How to induce lucid dreams in beginners in a sleep laboratory setting with high success rate. 03 2017. doi: 10.13140/RG.2.2.31505.48488.

Kristoffer Appel. *The Traumschreiber System: Enabling Crowd-based, Machine Learning-driven, Complex, Polysomnographic Sleep and Dream Experiments*. PhD thesis, Universität Osnabrück, 2018.

Kristoffer Appel, Gordon Pipa, and Martin Dresler. Investigating consciousness in the sleep laboratory–an interdisciplinary perspective on lucid dreaming. *Interdisciplinary Science Reviews*, 43(2):192–207, 2018.

GoB Arden and JH Kelsey. Changes produced by light in the standing potential of the human eye. *The Journal of physiology*, 161(2):189–204, 1962.

Aristotle. *On Dreams*. Harvard University Press, 1986.

Siriwadee Aungsakun, Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Robust eye movement recognition using eog signal for human-computer interface. In *International Conference on Software Engineering and Computer Systems*, pages 714–723. Springer, 2011.

João Cordovil Bárcia. Human electrooculography interface. *Lisbon Technical University*, 2010.

Rafael Barea, Luciano Boquete, Manuel Mazo, María Guillén, and Luis Bergasa. E.o.g. guidance of a wheelchair using neural networks. volume 4, pages 4668–4671, 01 2000. doi: 10.1109/ICPR.2000.903006.

Rafael Barea, Luciano Boquete, Manuel Mazo, and María Guillén. System for assisted mobility using eye movements based on electrooculography. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 10:209–18, 01 2003. doi: 10.1109/TNSRE.2002.806829.

F Behrens, Manfred Mackeben, and W Schröder-Preikschat. An improved algorithm for automatic detection of saccades in eye movement data and for calculating saccade parameters. *Behavior research methods*, 42:701–8, 08 2010. doi: 10.3758/BRM.42.3.701.

E. H. Du Bois-Reymond. *Untersuchungen über thierische Elektricitat*. Berlin: Reimer, 1884.

Andreas Bulling, Jamie A Ward, Hans Gellersen, and Gerhard Troster. Eye movement analysis for activity recognition using electrooculography. *IEEE transactions on pattern analysis and machine intelligence*, 33(4):741–753, 2010.

S C Chen, T T Tsai, and C H Luo. Portable and programmable clinical eog diagnostic system. *Journal of medical engineering & technology*, 24:154–62, 07 2000.

Ricardo Chavarriaga, Aleksander Sobolewski, and José del R Millán. Errare machinale est: the use of error-related potentials in brain-machine interfaces. *Frontiers in neuroscience*, 8:208, 2014.

Hema CR and Paulraj MP. Classification of eye movements using electrooculography and neural networks. *International Journal of Human Computer Interaction (IJHCI)*, 5(4):51, 2014.

Yash Shaileshkumar Desai. Natural eye movement & its application for paralyzed patients. *International Journal of Engineering Trends and Technology*, 4(4):679–686, 2013.

Kamil Dimililer, Yoney Kirsal Ever, Cansu Somturk, Fulden Ergun, Guner Urun, and Mehmet Kara. Effect of dct image compression on eye gaze direction detection. In *ITM Web of Conferences*, volume 16, page 01003. EDP Sciences, 2018.

Eknath Easwaran. *The Upanishads:(Classics of Indian Spirituality)*, volume 2. Nilgiri Press, 2007.

Fuming Fang and Takahiro Shinozaki. Electrooculography-based continuous eye-writing recognition system for efficient assistive communication systems. *PLOS ONE*, 13:e0192684, 02 2018. doi: 10.1371/journal.pone.0192684.

Oliver Faust, Yuki Hagiwara, Tan Jen Hong, Oh Shu Lih, and U Rajendra Acharya. Deep learning for healthcare applications based on physiological signals: A review. *Computer methods and programs in biomedicine*, 161:1–13, 2018.

Elisa Filevich, Martin Dresler, Timothy R Brick, and Simone Kühn. Metacognitive mechanisms underlying lucid dreaming. *Journal of Neuroscience*, 35(3):1082–1088, 2015.

R. Cranson Gackenbach, J. and C. Alexander. Lucid dreaming, witnessing, and the transcendental meditation: A developmental relationship. *Lucidity Letter*, 5:34–41, 1986.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Ayşegül Güven and Kara S. Classification of electro-oculogram signals using artificial neural network. *Expert Systems with Applications*, 31:199–205, 07 2006. doi: 10.1016/j.eswa.2005.09.017.

John Paulin Hansen, Dan Witzner Hansen, and Anders Sewerin Johansen. Bringing gaze-based interaction back to basics. In *HCI*, pages 325–329, 2001.

Keith M. T. Hearne. *Lucid dreams: an electro-physiological and psychological study*. PhD thesis, University of Liverpool (UK), 1978.

Jeong Heo, Heenam Yoon, and Kwang Park. A novel wearable forehead eog measurement system for human computer interfaces. *Sensors*, 17:1485, 06 2017. doi: 10.3390/s17071485.

H. T. Hunt. Lucid dreaming as a meditative state: Some evidence from long-term meditators in relation to the cognitive-psychological bases of transpersonal phenomena. *Dream Images: A Call to Mental Arms, edited by J. Gackenbach and A. A. Sheikh*, 5:265–285, 1991.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Kristoffer Appel Johannes Leugering. *The Traumschreiber Project*. 2017. `https://www.traumschreiber.uni-osnabrueck.de/`.

Do Yeon Kim, Chang-Hee Han, and Chang-Hwan Im. Development of an electrooculogram-based human-computer interface using involuntary eye movement by spatially rotating sound for communication of locked-in patients. *Scientific Reports*, 8, 12 2018. doi: 10.1038/s41598-018-27865-5.

Ki-Hong Kim, Hong Kee Kim, Jong-Sung Kim, Wookho Son, and Soo-Young Lee. A biosignal-based human interface controlling a power-wheelchair for people with motor disabilities. *Etri Journal - ETRI J*, 28:111–114, 02 2006a. doi: 10.4218/etrij.06.0205.0069.

Ki-Hong Kim, g Kee Kim, Jong-Sung Kim, Wookho Son, and Soo-Young Lee. A biosignal-based human interface controlling a power-wheelchair for people with motor disabilities. *ETRI journal*, 28(1):111–114, 2006b.

Alex Kreilinger, Christa Neuper, and Gernot R Müller-Putz. Error potential detection during continuous movement of an artificial arm controlled by brain–computer interface. *Medical & biological engineering & computing*, 50(3):223–230, 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Chih-En Kuo, S.-F Liang, Y.-C Li, Fu-Yin Cherng, Wen-Chieh Lin, P.-Y Chen, Y.-C Liu, and Fu-Zen Shaw. An eog-based sleep monitoring system and its application on on-line sleep-stage sensitive light control. *PhyCS 2014 - Proceedings of the International Conference on Physiological Computing Systems*, pages 20–30, 01 2014.

Stephen LaBerge. Lucid dreaming as a learnable skill: A case study. *Perceptual and Motor Skills*, 51(3_suppl2):1039–1042, 1980.

Stephen LaBerge. Lucid dreaming. 1985.

Stephen LaBerge. Lucid dreaming: Evidence and methodology. *Behavioral and Brain Sciences*, 23(6):962–964, 2000. doi: 10.1017/S0140525X00574020.

Stephen LaBerge and Lynne Levitan. Validity established of dream-light cues for eliciting lucid dreaming. *Dreaming*, 5:159–168, 09 1995. doi: 10.1037/h0094432.

Stephen LaBerge, Benjamin Baird, and Philip Zimbardo. Smooth tracking of visual targets distinguishes lucid rem sleep dreaming and waking perception from imagination. *Nature Communications*, 9, 08 2018. doi: 10.1038/s41467-018-05547-0.

Karma Lingpa et al. *The Tibetan Book of the Dead: Awakening Upon Dying*. North Atlantic Books, 2013.

Päivi Majaranta and Kari-Jouko Räihä. Twenty years of eye typing: Systems and design issues. volume 2002, pages 15–22, 01 2002. doi: 10.1145/507072.507076.

Z Mousavi, T Yousefi Rezaii, S Sheykhivand, A Farzamnia, and SN Razavi. Deep convolutional neural network for classification of sleep stages from single-channel eeg signals. *Journal of neuroscience methods*, page 108312, 2019.

Glenn H Mullin. *The Six Yogas of Naropa*. Shambhala Publications, 2014.

Stephen P. La Berge, Lynne Levitan, William Dement, and Vincent P. Zarcone. Lucid dream verified by volitional communication during rem sleep. *Perceptual and motor skills*, 52:727–32, 06 1981. doi: 10.2466/pms.1981.52.3.727.

Sarun Paisarnsrisomsuk. Deep sleep : Convolutional neural networks for predictive modeling of human sleep time-signals. 2018.

Rik Pieters and Michel Wedel. A review of eye-tracking research in marketing. In *Review of marketing research*, pages 143–167. Routledge, 2017.

A Poole and Linden Ball. *Eye tracking in human-computer interaction and usability research: Current status and future prospects*, pages 211–219. 01 2006.

Cristian-Cezar Postelnicu, Florin Girbacia, and Doru Talaba. Eog-based visual navigation interface development. *Expert Systems with Applications*, 39(12):10857–10866, 2012.

S Ramkumar and CR Hema. Recognition of eye movement electrooculogram signals using dynamic neural networks. *KJCS*, 6: 12–20, 2013.

S Ramkumar, K Sathesh Kumar, T Dhiliphan Rajkumar, M Ilayaraja, and K Shankar. A review-classification of electrooculogram based human computer interfaces. 2018.

Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124 3: 372–422, 1998.

Merle Reimann. Measuring vmmn, p300 and aeps with a mobile eeg system. unpublished thesis, 2018.

Dario Salvucci and Joseph Goldberg. Identifying fixations and saccades in eye-tracking protocols. pages 71–78, 01 2000. doi: 10.1145/355017.355028.

S Satyasangananda. *Sri Vijnana Bhairava Tantra: The ascent*. Yoga Publications Trust, 2003.

David Saunders, Chris Roe, Graham Smith, and Helen Clegg. Lucid dreaming incidence: A quality effects meta-analysis of 50 years of research. *Consciousness and Cognition*, 43:197–215, 07 2016. doi: 10.1016/j.concog.2016.06.002.

Ann-Kathrin Schalkamp. Analyzing event-related potentials in 8-channel eeg data using machine learning methods. unpublished thesis, 2018.

Marten K Scheffers and Michael GH Coles. Performance monitoring in a confusing world: error-related brain activity, judgments of response accuracy, and types of errors. *Journal of Experimental Psychology: Human Perception and Performance*, 26 (1):141, 2000.

Harry Septanto, Ary Prihatmanto, and Adi Indrayanto. A computer cursor controlled by eye movements and voluntary eye winks using a single channel eog. *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, ICEEI 2009*, 1, 08 2009. doi: 10.1109/ICEEI.2009.5254806.

AS Sherbeny and S Badawy. Eye computer interface (eci) and human machine interface applications to help handicapped persons. *TOJEEE*, 5:549–553, 2006.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tadas Stumbrys, Daniel Erlacher, Melanie Schädlich, and Michael Schredl. Induction of lucid dreams: A systematic review of evidence. *Consciousness and cognition*, 21:1456–75, 07 2012. doi: 10.1016/j.concog.2012.07.003.

M Trikha, A Bhandari, and T Gandhi. Automatic electrooculogram classification for microcontroller based interface design. In *2007 IEEE Systems and Information Engineering Design Symposium*, pages 1–6. IEEE, 2007.

J Tsai, C Lee, C Wu, J Wu, and K Kao. A feasibility study of an eye-writing system based on electro-oculography. *Journal of Medical and Biological Engineering*, 28(1):39, 2008a.

Jang-Zern Tsai, C.-K Lee, Chao-Min Wu, J.-J Wu, and K.-P Kao. A feasibility study of an eye-writing system based on electro-oculography. *Journal of Medical and Biological Engineering*, 28:39–46, 03 2008b.

Veena J Ukken, S Vaishnodevi, and S Mathankumar. Eog based prosthetic arm-hand control. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(5):3693–3698, 2015.

Ali Bülent Usakli, Serkan Gurkan, Fabio Aloise, Giovanni Vecchiato, and Fabio Babiloni. On the use of electrooculogram for efficient human computer interfaces. *Computational intelligence and neuroscience*, 2010:1, 2010.

Frederik Van Eeden. A study of dreams. In *Proceedings of the Society for Psychical Research*, volume 26, pages 431–461, 1913.

P Vandhana. A novel efficient human computer interface using an electrooculogram. *International Journal of Research in Engineering and Technology*, 3(4):799–803, 2014.

Melodie Vidal, Andreas Bulling, and Hans Gellersen. Analysing eog signal features for the discrimination of eye movements with wearable devices. *PETMEI'11 - Proceedings of the 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*, 09 2011. doi: 10.1145/2029956.2029962.

Ursula Voss, Romain Holzmann, Inka Tuin, and J Allan Hobson. Lucid dreaming: A state of consciousness with features of both waking and non-lucid dreaming. *Sleep*, 32:1191–200, 09 2009. doi: 10.1093/sleep/32.9.1191.

B. Alan Wallace. *Dreaming yourself awake : lucid dreaming and Tibetan dream yoga for insight and transformation*. Boston : Shambhala, 2012.

Jennifer M. Windt and Thomas K. Metzinger. The philosophy of dreaming and self-consciousness: What happens to the experiential subject during the dream state? 2007.

Tobias Wissel and Ramaswamy Palaniappan. Considerations on strategies to improve eog signal analysis. *International Journal of Artificial Life Research (IJALR)*, 2(3):6–21, 2011.

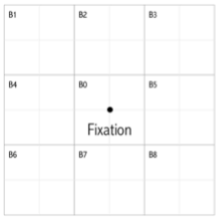
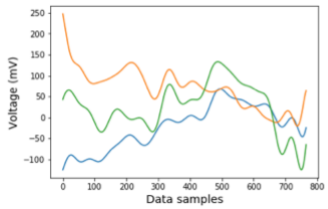
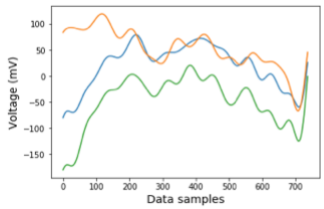
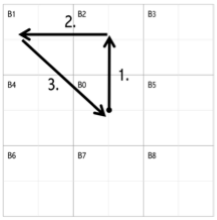
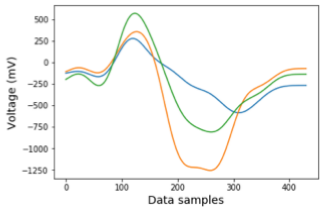
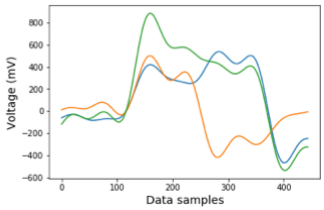
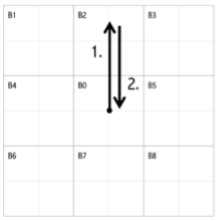
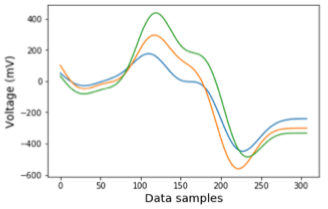
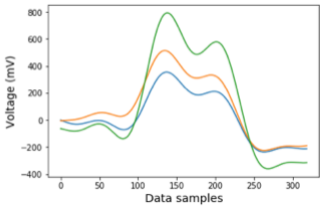
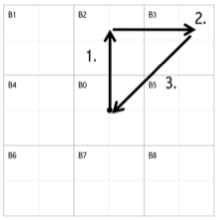
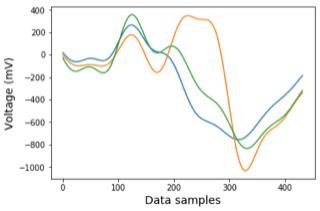
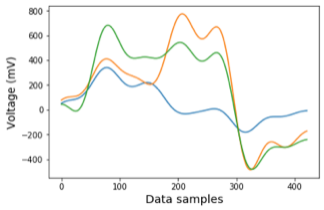
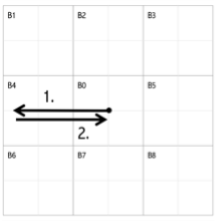
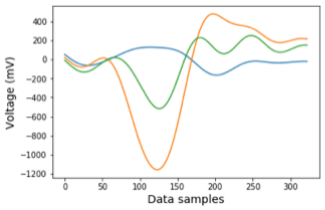
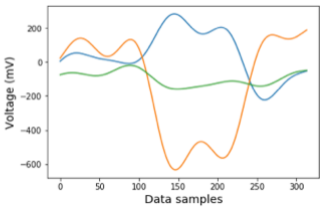
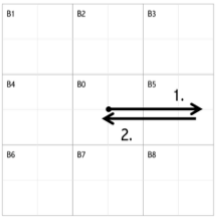
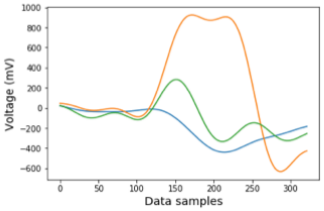
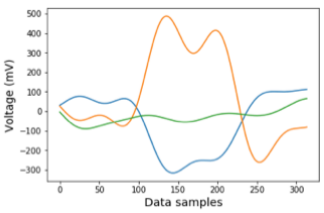
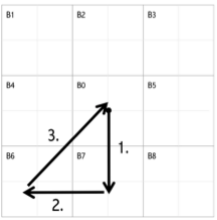
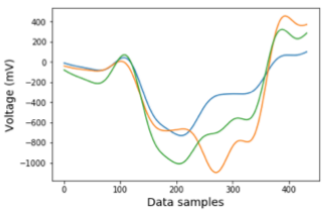
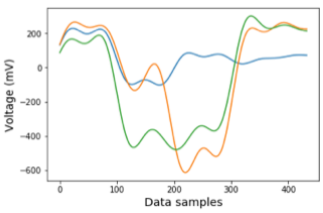
Tohru Yagi. Eye-gaze interfaces using electro-oculography (eog). *International Conference on Intelligent User Interfaces, Proceedings IUI*, 01 2010. doi: 10.1145/2002333.2002338.

Chi Zhang, Rui Yao, and Jinpeng Cai. Efficient eye typing with 9-direction gaze estimation. *Multimedia Tools and Applications*, 77(15):19679–19696, 2018.

Jianhua Zhang, Sunan Li, and Rubin Wang. Pattern recognition of momentary mental workload based on multi-channel electrophysiological data and ensemble convolutional neural networks. *Frontiers in neuroscience*, 11:310, 2017.

Xuemin Zhu, Wei-Long Zheng, Bao-Liang Lu, Xiaoping Chen, Shanguang Chen, and Chunhui Wang. Eog-based drowsiness detection using convolutional neural networks. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 128–134. IEEE, 2014.

7. Appendix

	GESTURE	CLOSED-EYES EOG	OPEN-EYES EOG
FIXATION			
UP-LEFT			
UP			
UP-RIGHT			
LEFT			
RIGHT			
DOWN-LEFT			

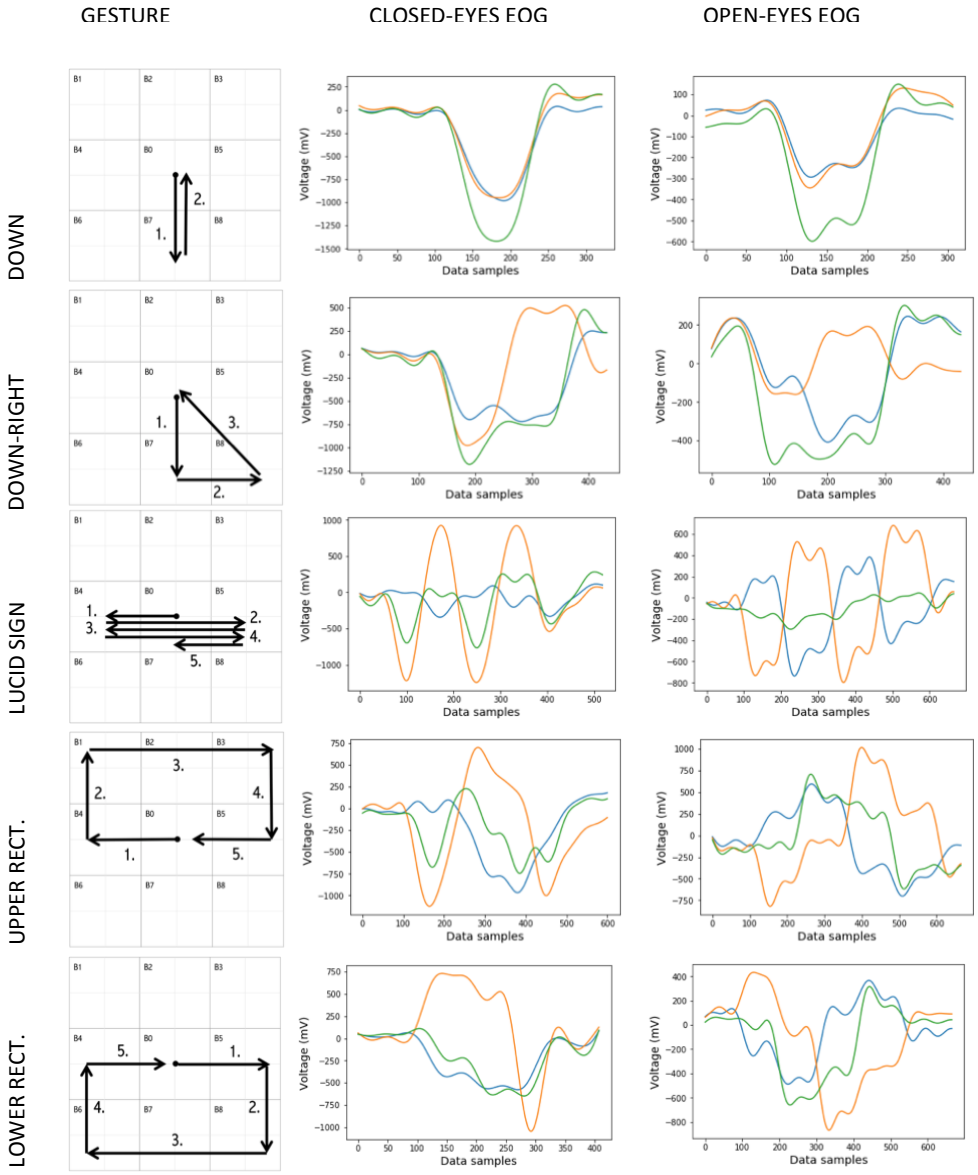


Figure 13: Path for each eye movement together with the resulting EOG signal in closed and open eyes condition

Declaration

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

Osnabrück, March 2019 - September 2019

_____
Victoria Amo Olea