POLITECNICO DI TORINO

MASTER's Degree in BIOMEDICAL ENGINEERING

MASTER's Degree Thesis

Development of an assistive device based on multisymbolic pupillary communication for patients in a state of complete paralysis

Supervisors

Prof. SILVESTRO ROATTA

Candidate

MATTEO BOEMIO

JULY 2025

Abstract

In patients affected by Complete Locked-In Syndrome (CLIS), all forms of voluntary communication are impaired due to total motor paralysis, despite preserved consciousness and cognitive abilities. In this context, conventional communication methods prove ineffective, highlighting the need for assistive devices that rely on residual physiological signals that can still be voluntarily modulated. The pupil accommodative response (PAR), which regulates pupil diameter based on the distance of the visual focal point, is typically preserved even in individuals with CLIS. This makes it a promising communication channel, as it can be voluntarily triggered by focusing gaze on visual targets positioned at different depths. The aim of this thesis is the design and development of a non-invasive, compact, and low-cost interaction system based on real-time pupil area monitoring and the detection of constriction events associated with PAR. The device is built around a *Raspberry Pi Zero* equipped with an infrared camera and lighting system, and integrates a segmentation algorithm capable of extracting pupil area in real time from the video stream. Unlike previous approaches that exploit PAR as a binary signal, the proposed system leverages its continuous nature by implementing a four-level coding paradigm based on the combination of two focal distances and two possible durations of the pupillary constriction. This strategy enables the transmission of a larger amount of information, significantly enhancing the communicative potential of the system. The device is supported by a computer responsible for processing and hosting a modular graphical interface, through which the user can access various applications. Among these, a text-based communicator has been developed, enabling word composition by modulation of the PAR. The system was validated through a series of experiments involving healthy subjects, virtually in the absence of any external movement, including convergence/divergence movements of the relevant eye. The results demonstrated effective pupil segmentation, with *Dice coefficient* exceeding 92%, and an excellent detection of pupillary constriction events, achieving both *precision* and *recall* values above 95%. Event classification *accuracy* exceeded 80%, indicating a good capacity to discriminate the 4 different patterns of response, despite the high intra and inter individual variability of the signal. Nevertheless, to ensure a more robust interaction, higher accuracy would be desirable and could possibly be achieved by future developments involving artificial intelligence-based techniques. Overall, the device demonstrated promising capabilities in supporting communication without requiring user training, suggesting potential applicability in both clinical and home settings.

Table of Contents

List of Tables

List of Figures

Acronyms

AAC	Augmentative and Alternative Communication
ACK	Acknowledgment
ALS	Amyotrophic Lateral Sclerosis
BCIs	Brain Computer Interfaces
CLIS	Complete Locked-In Syndrome
CMOS	Complementary Metal-Oxide Semiconductor
CNNs	Convolutional Neural Networks
CPU	Computer Processing Unit
CSI	Camera Serial Interface
E-tran	Eye transfer
ECoG	Electrocorticography
EEG	Electroencephalogram
ElSe	Ellipse Selection
EMG	Electromyography
ERD	Event-Related Desynchronization
ERPs	Event-Related Potentials
ERS	Event-Related Synchronization
EW	Edinger–Westphal
fALS	familial Amyotrophic Lateral Sclerosis
FCNs	Fully-Convolutional Networks
fNIRS	functional Near-Infrared Spectroscopy
FPS	frames per second
FTD	Frontotemporal Dementia
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HbO	Oxyhemoglobin

HDMI	High-Definition Multimedia Interface
HT	Hough Transform
IoU	Intersection over Union
IP	Internet Protocol
IR	Infrared
LEDs	Light Emitting Diodes
Li-Po	Lithium-Polymer
LIS	Locked-In Syndrome
MAE	Masked Autoencoders
ML	Machine-Learning
MNDs	Motor Neuron Diseases
MRCPs	Movement-Related Cortical Potentials
NIR	Near-Infrared
NIRS	Near-Infrared Spectroscopy
OS	Operating System
OSA	Overt Spatial Attention
OTG	On-The-Go
PAR	Pupillary Accommodative Response
RANSAC	RANdom SAmple Consensus
RGB	Red Green Blue
ROI	Region-Of-Interest
sALS	sporadic Amyotrophic Lateral Sclerosis
SAM	Segment Anything Model
SDI	Sociodemographic Index
SET	Sinusoidal Eye Tracker
SSID	Service Set Identifier
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
VEPs	Visual Evoked Potentials

Chapter 1

Introduction

Amyotrophic Lateral Sclerosis (ALS) is a progressive neurodegenerative disease that selectively affects motor neurons in the brain and spinal cord, leading to a gradual loss of voluntary muscle control resulting in the inability to walk, breathe and talk. With a median survival time ranging from three to five years after symptom onset [1], ALS ultimately results in severe physical disability and, in some cases, culminates in a state known as Completely Locked-In Syndrome (CLIS). In CLIS, patients lose nearly all voluntary muscle function, making them unable to perform even the simplest motor tasks, including those required for traditional modes of communication [2].

ALS patients typically rely on a variety of communication methods that leverage their remaining motor abilities, such as eye-tracking systems, adaptive keyboards, and speech synthesizers [3]. These technologies convert minimal voluntary movements (like eye motions or facial gestures) into effective communication signals. However, in the case of CLIS patients, the total loss of voluntary muscle control makes these conventional methods impractical or even impossible to use. Although brain–computer interfaces (BCIs) offer a promising alternative by translating neural signals directly to interpret patient intent and thereby restore a degree of communication, these systems often face significant limitations. Signal variability, calibration complexity, and reduced reliability in real-world conditions continue to hinder their practical deployment [4]. Alternative approaches, such as BCIs based on near-infrared spectroscopy (NIRS), have shown potential by detecting hemodynamic responses associated with brain activity, enabling basic forms of communication in CLIS patients [5]. Nonetheless, these systems tend to be technically complex, resource-intensive, and not always feasible for widespread clinical use. Consequently, addressing the communication needs of CLIS patients is not only a complex technical challenge but also an ethical imperative, as the inability to communicate exacerbates isolation and limits participation in medical and personal decisions. To overcome the limitations of existing technologies, there is a growing interest in alternative, low-cost, and accessible solutions. One promising yet, relatively, underexplored avenue involves leveraging preserved autonomic functions, particularly pupillary responses, as a channel for intentional communication. In this context, the present thesis introduces a novel communication device based on the pupillary accommodative response (PAR), an involuntary response that remains largely intact in ALS and CLIS patients. By harnessing this reflex, the system enables users to generate interpretable signals without any voluntary muscle activity. The PAR, integral to the visual accommodation process, involves involuntary pupil size adjustments in response to voluntary focusing on objects at varying distances [6]. Specifically, when shifting focus from distant to near objects, the pupils constrict. Conversely, pupil dilatates during focus shifts from near to distant objects. Notably, autonomic functions, including pupillary responses, are generally preserved in ALS patients. Studies have indicated that both sympathetic and parasympathetic pathways governing pupil dynamics remain largely unaffected by the neurodegenerative processes characteristic of ALS [7]. This preservation suggests that pupillary reflexes can serve as reliable channels for communication in individuals with CLIS. The feasibility of such paradigms has been demonstrated in some studies. For instance, Stoll et al. [8] utilized task-evoked pupillary dilation, elicited through cognitively demanding and self-stressing tasks such as mental arithmetic. While this method demonstrated the potential of pupillary responses for communication, it is inherently limited by its reliance on the user's cognitive effort, which can be variable, fatiguing, and difficult to sustain over time. In contrast, the PAR, as employed in the study by Villalobos et al. [9], provides a faster, more deterministic, and more reproducible physiological response. However, despite its advantages, the PAR-based approach is still constrained by its binary nature and the need for precise timing relative to stimulus presentation windows, which limits the amount of information that can be transmitted. In response to this constraint, a preliminary study was conducted to investigate the relationship between pupil size and focal distance, with the aim of assessing the feasibility of extending this approach into a multi-symbolic BCI system. Using a professional grade pupillometric device (*Pupil Core* by *Pupil Labs*), pupil size was recorded during accommodation tasks involving visual targets placed at varying distances under controlled protocols. The findings revealed a pronounced nonlinearity in the pupillary accommodation response and limited repeatability. Notably, both the amplitude and detectability of the PAR increased significantly as the visual target moved closer to the eye. These observations, consistent with prior research [10], provided the foundation for exploring an expanded use of the response beyond binary paradigms. Building on this insight, the present thesis leverages the depth-dependent and reflexive nature of the PAR to design a novel communication system capable of distinguishing multiple accommodative levels. By doing so, the system seeks to expand the communication bandwidth available to individuals in advanced stages of ALS, offering a path toward richer and more flexible user interfaces. Although external factors such as ambient lighting and emotional arousal can influence pupil dynamics, accommodative constrictions are typically distinguishable in both time and amplitude under controlled conditions. Successfully implementing such a multi-symbolic communication paradigm requires a hardware solution that is not only capable of tracking pupil size reliably, but also accessible and affordable enough for widespread use. While the *Pupil Core* system provides high sampling rates and robust eye-tracking capabilities, its high cost limits its suitability for widespread. To address this, a low-cost, fully wireless, and battery-powered wearable pupillometric system was developed using a *Raspberry Pi Zero* and an *OV5647* infrared camera mounted on an eyeglasses frame. This custom setup significantly lowers costs while offering enhanced image quality and an adequate sampling frequency to reliably detect accommodative responses, making it a practical and scalable solution for both clinical and home-based use. In parallel with hardware development, dedicated software was created to manage image processing and accurately detect PAR events. A user-friendly graphical interface (GUI) was also designed to facilitate intuitive interaction with the system. This interface enables users not only to operate the communication module, but also to access a selection of predefined applications, forming the basis of a modular and extensible assistive platform. Potential future extensions include games, home automation controls, and tools for calendar management or messaging. Particular attention was given to usability, ensuring that the software is accessible and operable even by non-technical caregivers.

Building on this foundation, the following chapters detail the context, development, and evaluation of the proposed system. The remainder of this thesis is structured as follows. Chapter 2 provides a comprehensive background covering the clinical and physiological aspects of ALS, existing communication systems including brain-computer interfaces, and the visual focusing mechanisms underlying the PAR as well as relevant pupil segmentation algorithms. Chapter 3 details the materials and methods employed in this work, describing the custom pupillometry hardware, software components, image acquisition and processing pipelines, and the design of the GUI along with its applications. Experimental protocols and performance evaluation methodologies are also outlined. Chapter 4 presents the results obtained from device performance testing, pupil segmentation accuracy, and PAR event detection efficacy. In Chapter 5, these results are discussed in depth, comparing them with previous studies, analyzing system limitations, and proposing directions for future improvements. Finally, Chapter 6 summarizes the key findings and concludes the thesis, highlighting the potential impact of the developed device.

Chapter 2

Background

Understanding the clinical, technological, and physiological context is essential to frame the motivations and objectives of this thesis. This chapter provides a comprehensive overview of ALS, including its epidemiology, causes, pathophysiological mechanisms, and the communication impairments that frequently arise as the disease progresses. Given the critical need for alternative communication strategies in patients affected by severe motor disabilities, an overview of both traditional and emerging assistive technologies, such as BCIs, is presented, along with a discussion of their current limitations. The chapter then shifts focus to the human visual system, with particular attention to the accommodation mechanism and the PAR, a physiological response that has gained interest as a potential communication channel for patients in advanced stages of ALS. Previous studies utilizing the PAR have been reviewed to highlight existing approaches and identify areas requiring further development. Finally, the chapter concludes with an analysis of the state of the art in pupil segmentation algorithms, which are critical for accurately detecting and tracking pupillary responses in real-world applications.This multidisciplinary background sets the foundation for the development of a novel communication system based on the PAR, as explored in the subsequent sections of the thesis.

2.1 ALS Overview

ALS, also referred to as Lou Gehrig's disease, is the most common form of a broader group of disorders known as motor neuron diseases (MNDs), is a progressive and invariably fatal neurodegenerative condition that selectively affects motor neurons, namely nerve cells located in the brain and spinal cord responsible for the voluntary control of muscles involved in movement, speech, swallowing, and breathing. The gradual degeneration and loss of these neurons result in progressive muscle weakness, paralysis, and ultimately, respiratory failure. Beyond its devastating clinical manifestations, ALS poses a substantial emotional, social, and economic burden on patients, their families, and the healthcare system.

This section provides a comprehensive overview of ALS, beginning with the current understanding of its potential causes, including genetic and environmental factors. It then explores the pathophysiological mechanisms underlying the disease and the typical clinical manifestations, with particular attention to the progression toward conditions such as Locked-In Syndrome and Complete Locked-In Syndrome, which severely impair communication. While ALS is not the sole condition that can result in extensive paralysis, a focused analysis is warranted given that it represents one of the principal causes of patients progressing to LIS and CLIS. Finally, the chapter discusses epidemiological data on the incidence and prevalence of ALS, highlighting global trends and future projections. By contextualizing the growing burden of the disease, this overview underscores the urgent need for assistive communication technologies, especially as increasing numbers of patients experience profound communication challenges.

2.1.1 ALS Causes

The etiology of ALS is complex and remains only partially elucidated, reflecting the contribution of both genetic and environmental factors. ALS manifests in two principal forms: sporadic ALS (sALS), accounting for approximately 90–95% of cases, arises in individuals without a known family history of the disease; conversely, familial ALS (fALS) represents about 5–10% of cases and is directly associated with inherited genetic mutations [11]. The existence of both sporadic and familial forms highlights the multifactorial nature of ALS and suggests that a combination of intrinsic genetic susceptibility and extrinsic environmental exposures underpins its development.

Genetic factors

ALS cases classified as familial result from inherited genetic mutations. To date, more than 40 genes have been implicated in the pathogenesis of ALS, highlighting the considerable genetic heterogeneity of the disease [12]. Among these, mutations in *C9orf72*, *SOD1*, *TARDBP*, and *FUS* are the most frequently identified. The hexanucleotide repeat expansion in the *C9orf72* gene represents the most common genetic abnormality and is notably associated not only with ALS but also with frontotemporal dementia (FTD), suggesting a shared pathogenic mechanism between the two conditions [13]. Mutations in the *SOD1* gene were the first to be discovered and are responsible for approximately 19% of familial ALS cases and about 4% of sporadic cases, underlining their pivotal role in the early understanding of ALS genetics [14]. Alterations in *TARDBP* and *FUS*, two genes involved in the encoding of proteins participating in mRNA transport and motor neuron development, have further emphasized the critical importance of RNA processing defects in ALS pathophysiology [15]. Moreover, evidence suggests that rare genetic variants significantly contribute to disease susceptibility, supporting the notion that ALS lies on a continuum from purely genetic to multifactorial origins [16]. Notably, even in cases classified as sporadic ALS, genetic factors seem to play a substantial role [17]. This growing body of evidence suggests that the traditional dichotomy between familial and sporadic ALS may be overly simplistic. This distinction is often confounded by ascertainment bias and the absence of a standardized definition of fALS. Additionally, incomplete penetrance and the lack of a gold standard for its assessment further complicate decisions regarding the appropriateness of genetic testing in individuals without a clear family history, as well as the interpretation of positive findings [18]. Beyond genetic factors, however, environmental and lifestyle influences have also been increasingly recognized as important contributors to ALS development, further supporting the multifactorial nature of the disease.

Environmental and lifestyle factors

Environmental exposures are believed to contribute significantly to ALS pathogenesis, particularly in sporadic cases. Several risk factors have been proposed [19] [20], although definitive causal relationships are challenging to establish. These include:

- **Toxin exposure**: contact with heavy metals (such as lead or mercury), agricultural chemicals, and other neurotoxins have been linked to a higher risk of developing ALS.

- **Military service**: several studies report an increased incidence of ALS among military veterans, potentially due to a combination of environmental exposures, physical trauma, and intense physical exertion.

- **Physical activity**: while moderate exercise is generally beneficial, some evidence suggests that elite athletes may have a higher incidence of ALS, although the mechanisms remain unclear.

- **Smoking**: tobacco use has been consistently associated with an increased risk in developing ALS.

- **Dietary habits**: diets high in saturated fats or low in antioxidants can contribute to oxidative stress, mitochondrial dysfunction, and disrupted lipid metabolism, all of which are implicated in the pathogenesis of ALS.

Aging and other risk factors

Age is widely recognized as the most significant risk factor for the development of ALS, with the incidence of the disease peaking between the age of 60 and 70 [21] [22]. Several biological processes associated with aging are believed to contribute to motor neuron vulnerability, including a progressive decline in cellular repair mechanisms, increased oxidative stress, mitochondrial dysfunction, and the accumulation of genetic and epigenetic alterations [23]. Over time, these processes can impair the ability of neurons to maintain homeostasis and respond to injury, ultimately facilitating neurodegeneration. In addition to aging, a range of other potential risk factors for ALS has been proposed, although their roles remain under investigation. Chronic systemic inflammation and aberrant autoimmune responses have been hypothesized to contribute to disease onset and progression by creating a hostile environment for motor neurons [24]. Viral infections have also been suggested as possible environmental triggers; certain neurotropic viruses could anticipate the onset or accelerate the neuronal degeneration through persistent infection or by inducing immune-mediated mechanisms [25]. However, while these associations are biologically plausible, current evidence is inconclusive and often derived from small cohort studies or preclinical models. Further large-scale, longitudinal studies are needed to validate these findings and to better define the interplay between aging, environmental exposures, and individual susceptibility in ALS pathogenesis.

2.1.2 Physiopathology and symptomatology

ALS is characterized by the progressive degeneration of both upper motor neurons, located in the motor cortex, and lower motor neurons, located in the brainstem and spinal cord. The mechanisms leading to motor neuron death are multifactorial and self-reinforcing. This dual neuronal involvement leads to a distinctive clinical picture combining features of spasticity and muscle weakness. From a pathophysiological standpoint, multiple mechanisms contribute to neuronal death in ALS. These include excitotoxicity mediated by excessive glutamate signaling, oxidative stress due to an imbalance between reactive oxygen species and antioxidant defenses, abnormal protein aggregation within motor neurons, and chronic neuroinflammation [26]. These pathogenic mechanisms interact synergistically, creating a vicious cycle that culminates in irreversible neuronal loss and denervation of skeletal muscles. As motor neurons degenerate, their axons retract from the neuromuscular junctions, leading to the progressive denervation of skeletal muscle fibers. This loss of neural input results in muscle fiber atrophy, initially affecting fast-twitch fibers, and triggers collateral sprouting attempts by surviving motor neurons, which become increasingly insufficient as the disease advances [27].

Clinical presentation

The clinical manifestations of ALS are heterogeneous and largely depend on the distribution and predominance of upper versus lower motor neuron degeneration leading to a spectrum of disease manifestation where two principal variants are recognized based on initial symptomatology:

- **Bulbar-onset ALS**: affects the motor neurons of the cranial nerves, leading to early symptoms such as dysarthria, dysphagia, tongue atrophy, and fasciculations. This form tends to progress rapidly and is often associated with earlier respiratory compromise and a poorer overall prognosis.

- **Spinal-onset ALS**: typically presents with asymmetric weakness in the limbs, starting distally, accompanied by muscle wasting, cramps, and fasciculations. Disease progression leads to eventual involvement of proximal muscles, contralateral limbs, bulbar muscles, and respiratory musculature.

In addition to the classic spinal and bulbar onset forms, studies have identified fewer common variants of ALS onset, including mixed bulbar-spinal presentations, thoracic onset, initial respiratory involvement, early cognitive or behavioral changes, and presentations resembling dementia. However, these atypical forms are relatively rare [28]. Regardless of the initial presentation, as the disease advances, patients develop widespread muscle atrophy, spasticity, hyperreflexia, dysarthria, dysphagia, and progressive respiratory insufficiency due to diaphragmatic weakness. Notably, however, this widespread degeneration typically spares the oculomotor system. The nuclei controlling eye movements and their associated motor neurons exhibit a remarkable resistance to the pathogenic mechanisms that affect other motor neuron populations. Several hypotheses have been proposed to account for this selective vulnerability, including enhanced calcium-buffering capacity [29] and the distinct expression of neuroprotective factors within oculomotor neurons [30] [31]. Consequently, patients usually retain voluntary control over eye movements even in the advanced stages of ALS, making these movements an important channel for communication in severely impaired individuals. Nevertheless, in rare and particularly advanced cases, neurodegeneration may extend to the oculomotor system as well, eventually leading to impaired eye movement control and complicating both clinical management and patient communication.

Progression to LIS and CLIS

As ALS progresses, patients can enter a condition resembling the Locked-In Syndrome (LIS), characterized by near-total paralysis of voluntary muscles except for partial retention of ocular mobility, typically limited to vertical eye movements or blinking [32]. In this state, patients remain fully conscious and cognitively intact, and eye movements may serve as their sole means of communication through assistive technologies or caregiver-mediated systems. In the most severe and terminal phase of ALS, further neurodegeneration may affect the oculomotor nuclei or lead to functional exhaustion of the remaining oculomotor neurons. This results in the transition to Complete Locked-In Syndrome, a condition in which all voluntary muscle activity, including ocular movements, is lost. Although the patient remains fully aware and cognitively functional, the total absence of motor output renders conventional communication impossible. In such cases, establishing even minimal interaction with the external world requires the implementation of advanced neurotechnological systems, such as BCIs, which are still under development and not yet widely available in clinical practice. The progression from LIS to CLIS exemplifies the devastating trajectory of ALS and highlights the importance of preserving any remaining motor functions for as long as possible, not only for maintaining autonomy but also for enabling communication and preserving dignity in end-stage disease.

2.1.3 Global incidence

Epidemiological studies have provided significant insights into the global prevalence and incidence of ALS. A comprehensive systematic review and meta-analysis reported an overall crude worldwide prevalence of 4.42 cases per 100,000 population and an incidence rate of 1.59 cases per 100,000 person-years [33]. However, these figures represent global averages, and considerable variations exist across different regions. For instance, the incidence of ALS ranges from 0.26 per 100,000 person-years in Ecuador to 23.46 per 100,000 person-years in Japan [34]. These disparities suggest significant environmental or genetic factors that may vary by geographic location.

Analyzing trends over time reveals that the incidence of ALS has been rising globally since 1957, however, this upward trend diminishes after adjusting for age, suggesting that demographic shifts significantly influence these patterns [33]. According to the Global Burden of Disease Study, between 1990 and 2016 there was an increase in the prevalence and mortality of MNDs, while the age-standardized incidence remained stable in most regions of the world, except in those with a high Sociodemographic Index (SDI), where it increased. This pattern suggests that the rise in prevalence may partly be attributed to improved patient survival or enhanced diagnostic capabilities [35]. One study projected a 69% increase in the worldwide annual incidence of ALS within the next 25 years (based on 2015 data), primarily attributed to the aging global population [36]. This projection aligns with conservative estimates for the U.S., anticipating an increase of over 10% in ALS cases between 2022 and 2030 [37].

These demographic and epidemiological projections underscore the growing global burden of ALS and highlight the increasing demand for healthcare resources and support systems. As the number of ALS cases increases globally, a corresponding rise in patients progressing to severe stages of motor impairment, such as LIS or CLIS, can be logically anticipated. This trend inevitably leads to a growing demand for assistive communication technologies tailored to individuals with severely impaired voluntary motor control.

2.2 Communication systems

Effective communication is a fundamental human need, yet it becomes increasingly challenging for individuals affected by ALS. As speech and limb movements deteriorate, alternative means of communication become essential to preserve autonomy, express needs, and maintain social connections. This chapter provides an overview of the main assistive communication systems available to ALS patients, focusing on both traditional technologies and BCI solutions. Traditional systems, such as low-tech solutions, eye-tracking devices, adaptive keyboards, and speech synthesizers, are examined for their operational principles and real-world applicability. In parallel, BCI-based approaches, which bypass muscular activity by decoding neural signals directly, are explored as promising alternatives, particularly for patients in advanced disease stages.

2.2.1 Traditional communication systems

Traditional communication systems, which do not rely on BCIs, have historically served as the foundation of assistive communication strategies for patients with ALS. These methods fall under the broader category of Augmentative and Alternative Communication (AAC), designed to leverage any residual voluntary motor abilities, typically eye movements or subtle facial gestures, to enable meaningful interaction with caregivers and the external environment. Technologically advanced AAC solutions, such as eye-tracking systems and adaptive keyboards, are frequently combined with speech synthesizers to generate real-time voice output from user input. These synthesizers convert selected letters, words, or symbols into artificial speech, with options for voice customization and the storage of commonly used phrases to accelerate communication. While these AAC systems provide essential communication channels, particularly when neural interfaces are not feasible, their performance often depends on factors like proper calibration, consistent user input, and optimal environmental conditions, such as sufficient lighting or stable posture. Furthermore, their utility can decline as the disease progresses and voluntary motor control diminishes. Nonetheless, due to their accessibility, reliability, and integration into commercially available platforms, traditional AAC systems continue to play a crucial role in maintaining communication for individuals with ALS.

Low-tech solutions

Among the most fundamental tools are *communication boards* (Figure 2.1) and *eye transfer* (E-tran) boards (Figure 2.2), which typically consist of grids populated with letters, numbers, symbols, and frequently used phrases. The user communicates by directing their gaze toward specific areas of the board, while a trained caregiver observes and interprets these gaze patterns to identify the intended message. In some cases, the patient may confirm or deny selections through predefined gestures such as blinking, smiling, or raising an eyebrow. A commonly used variation of this method is partner-assisted scanning, in which a communication partner sequentially presents items, either aloud or by pointing to them on a board, and the patient signals acceptance of the desired option through a simple response, such as a blink or slight head movement. Although this method can be time-consuming and requires the constant presence of a trained communication partner, it remains a highly reliable and accessible technique. These low-tech solutions, while limited in speed and complexity, have the advantage of being inexpensive, portable, and easy to implement in various settings. They are often used as a first line of communication support or as a backup when high-tech systems are unavailable or impractical due to environmental constraints or disease progression.

Figure 2.1: Communication board

Figure 2.2: E-tran board

Eye-tracking systems

With the advent of modern technology, more advanced solutions have emerged, including eye-tracking systems. These systems detect gaze direction and allow users to control a graphical interface, select letters or words, and generate speech output. Eye-tracking is non-invasive and highly intuitive, making it one of the most widely adopted solutions in AAC for ALS.

Commercial eye-tracking systems typically employ near-infrared light sources and high-resolution cameras to monitor reflections on the cornea and pupil. Using advanced computer vision algorithms, these systems calculate the point of gaze in real time with high spatial and temporal accuracy. Calibration routines are often required to associate specific gaze directions with positions on the screen, and modern systems can automatically adjust to head movement or lighting changes to maintain precision. Some devices also support various input methods beyond eye gaze, such as touch or switch inputs, which can be adapted to the user's remaining motor abilities. These adaptable controls are designed to accommodate patients with varying levels of motor function, allowing for more personalized interaction. The accompanying software typically includes features like predictive text, customizable communication boards, and integration with environmental control systems, all of which enhance the device's overall usability and functionality.

Despite their widespread adoption and intuitive operation, eye-tracking systems present several limitations, particularly in the context of progressive neurodegenerative conditions like ALS. Accurate tracking requires a consistent ocular control that can be compromised by ptosis, fatigue or limitations in perform ocular movements due to degeneration [38] [39]. Calibration procedures, essential for maintaining system precision, may need to be repeated frequently and can be challenging for some users with central visual field loss, additionally not all eye types are easily trackable due to uses of glasses or contact lenses, also pupil color seems to have an impact on eye-tracking performances [40]. Furthermore, these systems are highly dependent on the user's ability to voluntarily move their eyes. As ALS advances, some patients enter in CLIS state in which all voluntary motor functions, including ocular movements, are lost. In such cases, eye-tracking technologies become entirely ineffective, leaving patients without any conventional means of interaction. Additionally, the high cost of commercial eye-tracking systems and the need for specialized technical support may pose barriers to accessibility, particularly in under-resourced settings.

Adaptive keyboards

Adaptive keyboards represent another important category of assistive communication tools, particularly suited for individuals with residual motor function. These devices are specifically designed to accommodate limited or highly localized movement capabilities, enabling users to input text or control interfaces through customized and accessible means. One widely used solution is the on-screen virtual keyboard, which displays a full keyboard layout on a screen. Users can select letters or commands through scanning input, a technique where rows or individual keys are highlighted sequentially, and the user activates a switch (via a blink, finger tap, cheek movement, etc.) when the desired option is selected. This method minimizes the need for extensive motor control while still allowing full text generation. In addition to virtual options, there are specialized physical keyboards designed with enlarged keys, reduced key sets, or customizable overlays that simplify interaction. These systems support various access methods, including sip-and-puff devices, head pointers, and adaptive switches. They are often used in combination with single or dual-switch setups, where one or two input signals, triggered by subtle voluntary movements such as eyebrow raises or jaw clenches, are detected by electromyography (EMG) system [41]. These setups can be tailored to the user's specific motor capabilities and can significantly improve communication speed compared to more basic methods [42], mostly if associated with prediction algorithms integrating also eye tracking features and customizable phrase banks that speed up text entry. Moreover, many platforms can integrate with environmental control systems, enabling users not only to communicate but also to interact with their surroundings (e.g., turning on lights, operating a TV, or controlling a wheelchair) through the same interface.

Adaptive keyboards thus represent a versatile and effective solution for ALS patients who retain minimal voluntary movements, offering both independence and a means to maintain social interaction as the disease progresses. However, these systems are not without limitations. Their effectiveness is highly dependent on the presence and stability of at least one reliable voluntary movement, which may deteriorate as the disease advances. Additionally, switch-based and scanning interfaces can be slow and cognitively demanding, requiring sustained attention and precise timing to select desired inputs. Additional factors, such as user posture and fatigue, can further impact usability. Moreover, initial setup and customization often require the involvement of trained professionals and caregivers, which may not always be readily available in all care settings [42]. Despite these challenges, adaptive keyboards remain a critical component of the assistive technology landscape for ALS, especially when integrated with predictive text systems and multimodal interfaces.

2.2.2 Brain-Computer-interface systems

Brain-computer interface systems are emerging as powerful technologies designed to restore communication and interaction for individuals with severe motor disabilities, such as those suffering from ALS. By establishing a direct link between the brain and external devices, BCIs allow patients to control computers, robotic systems, or communication devices through neural activity alone. These systems have the potential to significantly improve the quality of life for individuals who have lost the ability to speak or move, offering a lifeline for communication and control in daily activities.

This chapter will explore the functioning, advantages, and limitations of both invasive and non-invasive BCI technologies, focusing on their application in enabling communication for paralyzed patients.

Invasive systems

Invasive BCIs represents a frontier in assistive neurotechnology for communication with patients affected by ALS, particularly in advanced stages of the disease when voluntary muscle control is entirely lost. These systems typically rely on intracortical microelectrode arrays or electrocorticography (ECoG) to record neural signals directly from the motor or speech cortex. One notable example is the use of *Utah* arrays, which penetrate the cortical surface to capture action potentials from individual neurons. This high spatial and temporal resolution enables the decoding of fine motor intentions or speech-related neural activity. In a recent development at *UC Davis Health*, researchers have implemented a speech neuroprosthesis that maps neural activity from the ventral premotor cortex to phoneme probabilities using deep learning models, achieving real-time synthesis of intelligible speech with up to 97% accuracy [43]. Invasive BCIs allow communication even in LIS, using neuroelectrical activity from motor cortex or prefrontal region, respectively triggered by motor task imagination and mental calculations [44].

Despite their high performance, invasive BCIs present critical limitations: surgical implantation poses risks such as infection, hemorrhage, and long-term biocompatibility issues like gliosis and electrode encapsulation, which may compromise signal quality degrading device stability and potentially requiring re-implantation. Moreover, these systems typically need a training phase to calibrate decoding algorithms (e.g., recurrent neural networks or Kalman filters), which require time and recalibration. Furthermore, ethical concerns also arise regarding informed consent in vulnerable patients, as well as psychological effects and the protection of sensitive neural data. These factors necessitate careful evaluation when considering invasive BCIs for clinical applications [45].

Non-invasive systems

Non-invasive BCIs constitute a promising approach for enabling communication in patients with ALS, particularly in advanced stages where conventional motor output is severely compromised. Among these, electroencephalography (EEG) systems are the most widely used due to their high temporal resolution, portability, and relatively low cost. EEG-based BCIs commonly leverage Event-Related Potentials (ERPs), including:

- **Visual Evoked Potentials** (VEPs): VEPs are triggered by the conscious recognition of visual targets presented through flickering stimuli with distinct temporal characteristics (e.g., frequency, duration, coding, phase, or motion-onset). Each stimulus has a unique temporal signature that can be detected in the scalp EEG.

- **Overt Spatial Attention** (OSA): OSA paradigms rely on parietal cortex activity associated with directing spatial attention. Similar to motor imagery, OSA does not require external stimuli and typically induces localized alpha-band activity.

- **P300**: The P300 is an endogenous ERP arising in the parietal cortex during an oddball paradigm, where users respond to infrequent target stimuli amid frequent non-targets. Commonly used in speller applications, P300-based BCIs often present a flashing matrix of rows and columns, eliciting a P300 response when the attended item is highlighted.

- **Movement-Related Cortical Potentials** (MRCPs): MRCPs are broad-band, time-domain potentials that reflect slow cortical changes associated with both cued and continuous motor intentions. They are characterized by low-frequency (delta band) activity that is time- and phase-locked to the onset of voluntary movements.

- **Event-Related (De)Synchronization** (ERD/ERS): ERD/ERS describes a decrease (desynchronization) or increase (synchronization) in EEG band power relative to a baseline. Typically evoked by motor tasks, they manifest as contralateral ERD and ipsilateral ERS in the alpha/mu (8–13 Hz) and/or beta (14–30 Hz) bands. Time-frequency representations, such as ERD maps, are used to analyze the spatial, temporal, and spectral features of these phenomena [46].

Complementary to EEG, functional Near-Infrared Spectroscopy (fNIRS) offers a hemodynamic-based modality that measures changes in cortical oxygenation linked to cognitive tasks. fNIRS-based BCIs commonly employ mental arithmetic or spatial navigation tasks to elicit localized increases in oxyhemoglobin (HbO) concentration, detectable through optodes placed on the prefrontal cortex. These signals are processed using statistical or machine learning methods to classify intentional binary responses. While fNIRS provides better spatial resolution and is less sensitive to electrical noise than EEG, its lower temporal resolution limits its applicability to slower communication rates. Hybrid systems that combine EEG and fNIRS have been proposed to leverage the complementary strengths of each modality [47].

Despite their non-invasive nature, these systems face several limitations: EEG signals are highly susceptible to artifacts from eye blinks, muscle activity, and environmental interference, requiring careful signal preprocessing and controlled acquisition environments [48]. fNIRS systems, on the other hand, involve higher equipment costs and are not yet widely available in clinical or home-care settings. Moreover, both modalities often require user training, calibration sessions, and supervision by trained personnel, which can limit accessibility and scalability. The communication speed achieved is typically low, with bitrates often insufficient for fluent conversation, especially in patients with reduced attention spans or cognitive fatigue.

2.3 Visual focusing mechanisms and the PAR in ALS

The human visual system is one of the most intricate and highly developed sensory systems, enabling the perception of shapes, colors, movements, and depth. Among its many functions, the ability to adjust focus according to viewing distance is fundamental to our interaction with the environment. This dynamic adjustment, known as accommodation, is not only a mechanical process involving the lens and ocular muscles but also part of a broader, neurologically coordinated response that includes pupillary constriction and eye convergence. This chapter explores the mechanisms underlying visual accommodation with a particular emphasis on the PAR. Understanding the neural pathways that control this response offers insight into how the eye maintains clarity across different focal distances. Moreover, we examine why this reflex remains functional in certain neurodegenerative conditions, most notably ALS, and discuss the clinical and technological implications of this preservation. In particular, the resilience of PAR in ALS patients presents a valuable opportunity for developing non-invasive communication interfaces based on pupillary responses.

2.3.1 The near vision complex and neural pathways

The human eye focuses incoming light onto the retina through a coordinated system of refractive components. The cornea, with its fixed curvature, accounts for the majority of the eye's total refractive power. In contrast, the crystalline lens provides dynamic focusing capability, enabling the eye to adjust to objects at varying distances. The lens is suspended by the suspensory ligament which is anchored to the ciliary muscle. When the ciliary muscle contracts, it reduces the tension on the suspensory ligament, allowing the lens to adopt a more spherical shape. This increase in curvature enhances the lens's refractive power, effectively shortening its focal length, to accommodate near vision (Figure 2.3). Thus, while the cornea performs the primary focusing role, the lens fine-tunes the focus through the process of accommodation, with changes in its curvature actively regulated by ciliary muscle activity to maintain sharp retinal imaging.

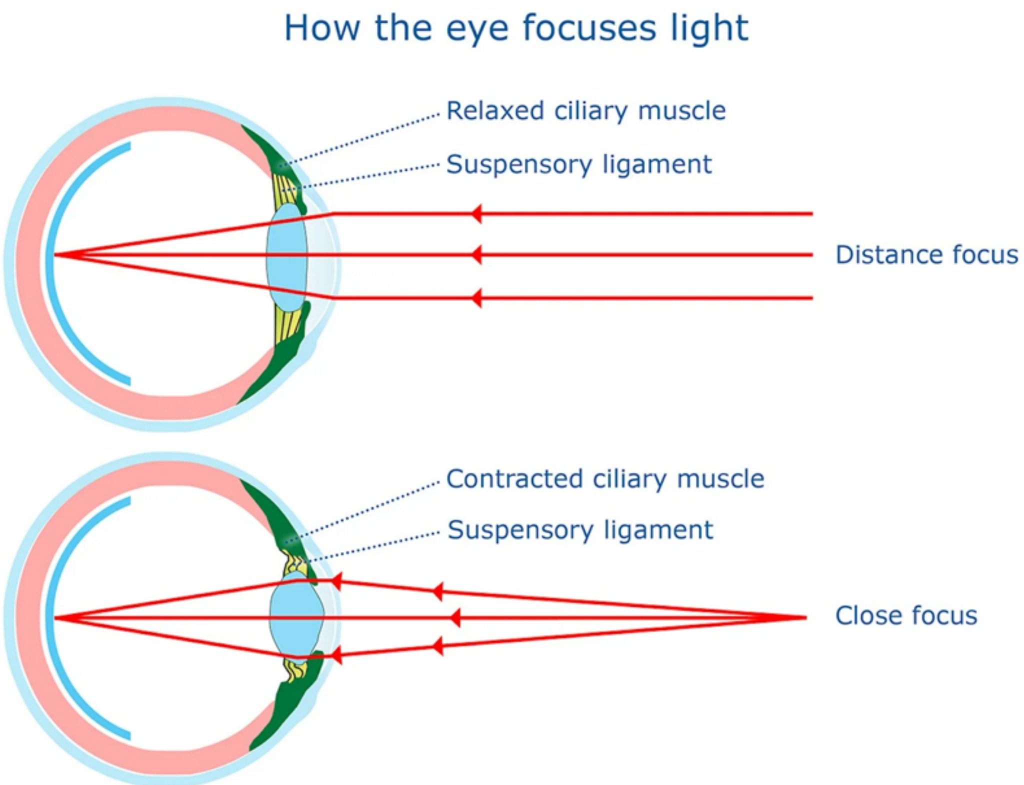


Figure 2.3: How the eye focuses light

Accommodation of the lens is part of a broader physiological response called the near vision complex or accommodation reflex, which involves three coordinated actions:

- **Convergence of the eyes**: both eyes rotate medially through the action of the medial rectus muscles, ensuring that their visual axes intersect at the near target. This alignment is essential to maintain binocular single vision and to prevent diplopia (double vision).

- **Accommodation of the lens**: activation of the ciliary muscles causes a release of tension on the zonular fibers (suspensory ligaments), allowing the lens to become more convex.

- **Pupillary constriction** (miosis): the sphincter pupillae muscles in the iris contract, reducing the pupil diameter. This limits the entry of peripheral light rays, reducing spherical aberrations and increases the depth of field (i.e. the range where objects are on focus). As a result, the image appears sharper, particularly on the central retina (fovea), which is responsible for high-acuity vision.

These three actions (convergence, accommodation, and miosis) occur simultaneously as part of a reflexive triad, allowing the eye to efficiently adapt to near vision [49].

Pupillary constriction during a far-to-near accommodation task or pupillary dilatation elicited from shifting gaze oppositely represents the PAR. Neural control of the PAR involves both afferent visual pathways and efferent parasympathetic output. Afferent signals begin in the retina and travel through the optic nerves, lateral geniculate body, to the visual cortex. Higher visual association areas detect retinal defocus and send signals to midbrain centers for near vision (like superior colliculus and pretectal areas). From there, impulses descend to the oculomotor complex. The Edinger–Westphal (EW) nucleus (component of cranial nerve III) is the key efferent relay. Fibers from EW travel via the oculomotor nerve to the ciliary ganglion; postganglionic fibers then innervate two targets. One branch innervates the ciliary muscle to adjust the lens; the other innervates the iris sphincter to constrict the pupil (Figure 2.4). At the same time, somatic efferent fibers from the oculomotor nucleus drive the medial recti for convergence [49].

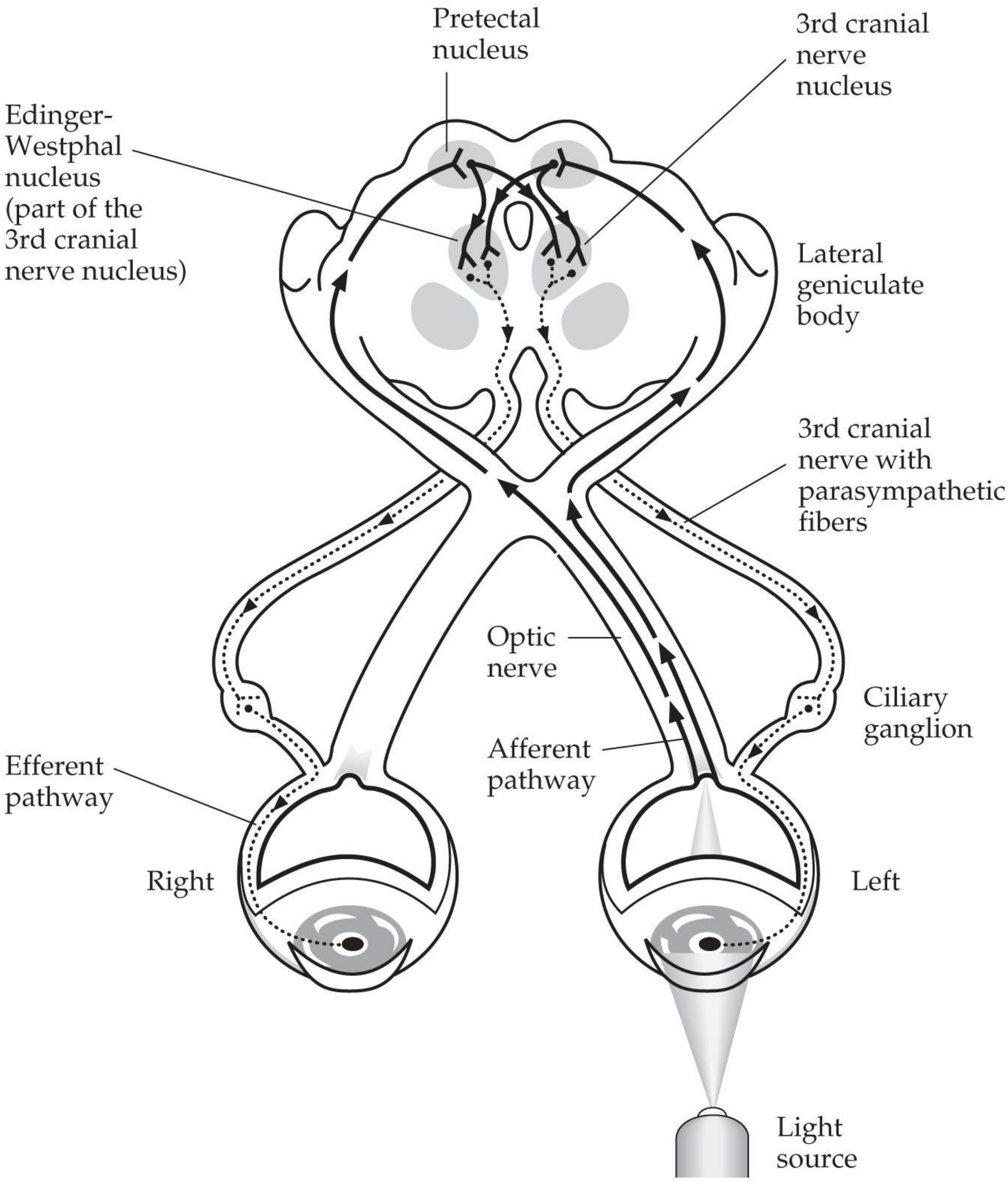


Figure 2.4: Diagram of the accommodation reflex neural pathways

2.3.2 Preservation and application of the PAR in ALS

In ALS skeletal muscle function progressively declines while Autonomic Nervous System (ANS) functions are largely preserved. Among these preserved autonomic functions is the PAR, which is mediated by the parasympathetic branch of the ANS and does not rely on voluntary muscle control. Since the PAR originates from the retina and is processed through brainstem parasympathetic pathways, its circuitry remains intact even in the most advanced stages of ALS. This preservation enables patients to constrict their pupils reflexively when shifting focus from distant to near targets, a phenomenon confirmed by both clinical observations and experimental studies. Empirical data demonstrate that this preserved reflex can be leveraged for communication in patients who have lost motor and speech abilities. For example, in the *e-Pupil* study, a completely paralyzed ALS patient successfully used focus-induced pupillary constriction to select "Yes" or "No" responses [50]. Such findings are supported by additional research showing that pupillary responses can be decoded, even in cases of severely impaired eye movement, to enable binary communication. While rare instances of autonomic dysfunction in fALS cases have been documented [51], they remain exceptions; the vast majority of patients retain normal pupillary size and reactivity, reflecting the functional integrity of the PAR.

These insights have significant implications for AAC technologies. Traditional gaze-based AAC systems, which rely on corneal reflections or video tracking to determine eye position, become ineffective when ocular motility is severely compromised or unstable, conditions frequently observed in late-stage ALS or in the CLIS. In contrast, pupillometry-based interfaces harness the preserved PAR by detecting pupil constrictions triggered by voluntary shifts in visual focus. Systems such as the low-cost *e-Pupil* have demonstrated that users can make binary selections, interacting with scanning interfaces, simply by looking at targets positioned at different depths, without the need for coordinated eye movements or extensive training.

The PAR is a continuous response, meaning that its magnitude varies in proportion to the proximity of the object being fixated. Specifically, closer targets elicit stronger pupillary constriction. This relationship can be approximated by a logarithmic function of viewing distance. Notably, for a given shift in target position, the resulting change in pupil diameter is highly dependent on distance: when the target is far, the pupil undergoes only minimal variation, whereas the same displacement at a closer range induces significantly larger changes in pupil size. This graded behavior offers the possibility of going beyond binary communication, allowing for multi-level input detection. However, the limited repeatability and physiological variability of the accommodative pupillary constriction introduces challenges in robustly distinguishing among multiple focus levels. While binary detection can be achieved with high reliability, accurately resolving three or more distinct levels requires more sophisticated signal processing and calibration and may still be subject to error due to inter- and intra-subject variability.

Despite these challenges, even the addition of a single intermediate focus level (i.e., moving from a binary to a ternary system) can significantly enhance the information transfer rate. From an information theory perspective, a binary system conveys 1 bit per selection, whereas a ternary system (with three distinguishable levels) conveys approximately $\log_2(3) \approx 1.58$ bits per selection, a 58% increase. Moreover, if the system can detect sequences or combinations of inputs across multiple time windows, the information throughput grows multiplicatively. For example, two successive ternary inputs can theoretically encode $3^2 = 9$ unique commands, equivalent to $\log_2(9) \approx 3.17$ bits. Thus, even modest increases in the number of reliably distinguishable levels can yield substantial gains in expressiveness and efficiency.

2.4 Pupil segmentation algorithms

Pupil segmentation is the crucial task of delineating the pupil region from an eye image in a pupillometry system. Several classes of algorithms have been developed, from simple intensity-based methods to advanced machine-learning (ML) approaches. Each technique makes different assumptions and has distinct advantages and limitations. Notably, there is no universally accepted gold standard for pupil segmentation; instead, clinically validated commercial devices, such as the *Neuroptics® VIP-300* or the *PLR-3000* pupillometer, are often treated as reference systems due to their proven accuracy and reliability in clinical settings. In the following subsections, the main categories of pupil segmentation algorithms are reviewed, from classical vision methods and feature-based approaches to modern deep-learning techniques, comparing their performance and applicability in real-time pupillometry. Advantages and disadvantages of each approach are highlighted, along with relevant references to the recent literature. This insight is necessary since an accurate pupil segmentation is fundamental for the development of the proposed device, as the system relies on detecting multiple response levels based on changes in pupil size. Therefore, precise segmentation is essential to ensure reliable detection of even subtle variations in pupil area.

2.4.1 Classical image processing methods

Classical image processing methods represent the foundational approaches to pupil segmentation and have historically played a central role in early pupillometry systems. These techniques rely on deterministic algorithms that exploit low-level visual features such as intensity, gradients, and edges to isolate the pupil region. Unlike deep learning-based approaches, they do not require prior training on annotated datasets and are generally lightweight in terms of computational resources, making them attractive for real-time and embedded applications.

Despite their simplicity, classical methods often perform well under controlled lighting and high-contrast conditions, where the pupil appears to be a dark, roughly circular region surrounded by brighter iris and sclera. However, they tend to degrade significantly in the presence of noise, variable illumination, occlusions (e.g. eyelids or eyelashes), or non-ideal pupil shapes.

Intensity-based thresholding

Intensity thresholding is one of the most straightforward techniques for pupil segmentation and consists in binarizing eye image based on pixel intensity values. In systems employing Near-Infrared (NIR) illumination, typical of "*dark-pupil*" imaging, the pupil appears significantly darker than the surrounding iris and sclera. Under these conditions, global thresholding methods such as Otsu's algorithm can generate an accurate binary mask of the pupil with minimal computational effort [52]. To address scenarios with gradual lighting variation across the image, adaptive thresholding techniques, which compute local thresholds for different regions, are often used as an alternative.

The principal appeal of thresholding lies in its simplicity and computational efficiency, which make it particularly suitable for real-time applications or resource-constrained platforms such as embedded systems. When operating under optimal conditions, namely with uniform NIR illumination and minimal occlusions, this approach can produce reliable and sharply defined pupil contours without the need for complex preprocessing [53].

Nevertheless, thresholding methods exhibit significant limitations in practical, unconstrained scenarios. They are highly sensitive to variations in illumination, the presence of specular reflections, shadows, and dark-colored irises. In such cases, the assumption that the pupil is the darkest region in the image may no longer be held, leading to frequent mis-segmentation where eyelashes, eyelids, or background areas are incorrectly identified as pupils [52]. Adaptive thresholding may partially mitigate these issues but often fails in the presence of glints, contact lenses, or significant non-uniformity in lighting. Consequently, the output masks are prone to irregularities and false detections, typically requiring additional filtering or post-processing to be usable in downstream analyses.

Edge-Detection

Edge-based segmentation methods aim to detect the boundaries of the pupil by identifying regions of rapid intensity change in the image. Algorithms such as Canny or Sobel operators are commonly employed for this purpose [54]. Typically, the image is first smoothed, often via Gaussian filtering, to suppress noise, and then processed to locate high-gradient transitions corresponding to the pupil-iris boundary. Unlike thresholding, which relies on absolute intensity values, edge detectors are sensitive to local contrast, enabling the identification of pupil contours even when the pupil region is not uniformly dark. This gradient-based approach is particularly useful in cases where the pupil does not present a clear intensity distinction from the surrounding iris, such as under non-uniform illumination. For this reason, edge detection is frequently used as a preprocessing step in more advanced segmentation pipelines, including the Hough Transform and active contour models, where the edge map guides the search for the pupil boundary. In addition, these methods are computationally inexpensive, thus applicable in real-time contexts.

However, edge-based methods are inherently sensitive to noise, reflections, and fine structures such as eyelashes or iris texture. The resulting edge maps often contain fragmented or spurious contours that do not correspond to anatomical features of interest. Therefore, substantial post-processing is usually required to isolate the pupil outline, typically involving morphological operations, contour filtering, or region linking [55]. Additionally, the performance of edge detectors depends critically on the choice of algorithm parameters, such as gradient thresholds and the standard deviation of the Gaussian kernel, which must be carefully tuned to balance sensitivity and robustness.

2.4.2 Feature-based and hybrid approaches

Pupil segmentation has long relied on classical image processing techniques that exploit features such as intensity gradients, shape regularity, and spatial consistency. Before the advent of deep- and machine-learning, a wide range of feature-based and hybrid algorithms were developed to detect and segment the pupil with reasonable accuracy and speed, even under challenging conditions such as reflections, occlusions, and non-uniform illumination. These methods often combine edge detection, geometric modeling, and clustering to infer the pupil region based on domain-specific assumptions.

In this section, several techniques are reviewed, ranging from Hough Transform and active contours to more specialized methods like Starburst, ellipse selection, and fuzzy clustering. While these approaches may not always match the performance of modern deep networks, they remain highly relevant due to their simplicity, efficiency, and suitability for real-time or resource-constrained applications.

Hough Transform

The Hough Transform (HT) is a popular method for detecting parametric shapes, most notably circles, within an image. In the context of pupil segmentation, the HT is primarily used to detect the circular boundary of the pupil by transforming the image space into a parameter space, where the shape of interest (i.e. a circle) is represented by a set of parameters such as radius and center position. The transformation allows the identification of potential circular regions that correspond to the pupil, even if the pupil is partially occluded. In practice, the HT operates by first detecting edges and then mapping these edge points into a parameter space where each edge point contributes to the accumulation of votes for possible circle parameters. The resulting accumulation map, often referred to as the Hough space, highlights the most probable locations and sizes of the circle. By selecting the peak in the Hough space, the center and radius of the pupil can be determined.

One of the advantages of HT is its robustness to incomplete or noisy edges. Even when parts of the pupil boundary are missing or corrupted by reflections, the HT can still detect the circle by finding a consistent accumulation of votes from the available edge points. This makes it more tolerant to gaps or noise in the image compared to raw edge detection [56].

However, the Hough Transform is computationally expensive, particularly when searching over a large parameter space, and may become impractical in real-time systems or resource-constrained devices. The performance also depends heavily on accurate edge detection; poor edge maps can lead to false positives or incorrect circle detections. Moreover, the method is limited in its ability to handle non-circular pupil shapes, which can arise due to off-axis images or other factors such as optical distortions [53].

Active Contours

Active contour models, also known as "*snakes*", are a class of techniques used for boundary detection that aim to iteratively deform a curve to fit the contours of an object within an image. These methods are particularly useful in applications where the object boundary is not clearly defined by simple intensity or gradient-based methods, such as in pupil segmentation under non-ideal conditions. The active contour model starts with an initial curve placed near the object of interest and evolves this curve to minimize an energy function that captures both the image features (such as edges) and the internal properties of the curve (such as smoothness). The energy function typically consists of two main components: an image-based term, which forces the contour to align with strong gradients or edges, and a geometric term, which penalizes irregularities in the contour shape to maintain smoothness and avoid overfitting to noisy or irrelevant features. These competing forces guide the evolution of the contour, with the goal of achieving an optimal boundary that corresponds to the pupil's shape. Variants of the method, such as geodesic active contours or level-set models, extend this framework to better handle topological changes (e.g., splits or mergers of the contour) [57].

One of the key advantages of active contours is their flexibility in adapting to irregular object shapes and their ability to accurately capture boundaries even when the object is occluded or has non-uniform intensity. This makes active contours particularly suitable for pupil segmentation in cases where the pupil is not perfectly circular. Additionally, active contours are relatively robust to noise and can incorporate prior knowledge (such as shape constraints) to improve performance.

However, the active contour method has several limitations. First, it requires good initial contour placement, which can be challenging in dynamic or poorly lit environments. The evolution of the contour can also be computationally expensive, especially in real-time applications, as it involves solving partial differential equations iteratively. Moreover, the choice of energy function is also a critical point [58]. Furthermore, active contours can struggle with large deformations if the initial contour is too far from the true boundary, or if the image lacks clear edge features to guide the process. In such cases, the method may fail to converge correctly or may become trapped in local minima.

Starburst algorithm

Hybrid algorithms combine image features with model fitting to improve segmentation accuracy and robustness. One well-known example of such an approach is the Starburst algorithm [59], commonly used in eye-tracking applications. Starburst begins with an initial estimate of the pupil center (often the image center) and casts multiple rays in all directions. Along each ray, it searches for significant intensity gradients that indicate potential boundary points. Once these edge points are detected, they are typically filtered through techniques like RANdom SAmple Consensus (RANSAC) ellipse fitting to refine the pupil contour, and the center is updated. This iterative process continues until the algorithm converges on an optimal pupil model.

The efficiency of the Starburst algorithm lies in its approach of performing one-dimensional searches along rays, which makes it computationally faster than more exhaustive methods like voting in HT. By combining local feature detection (such as intensity changes) with a global model (typically an ellipse), Starburst achieves good performance even in the presence of moderate noise or occlusion. The iterative refinement of the center further enhances its ability to adjust the pupil's location, making it well-suited for eye-tracking applications. This speed, along with its robustness to noise, has led to its integration into several open-source toolkits, highlighting its effectiveness in real-world scenarios.

However, Starburst's performance is highly dependent on the resolution and quality of the image. It relies on the presence of sharp, uniform edges along the rays, which can be problematic in situations with heavy eyelid or eyelash occlusion or very low contrast images. In such cases, the rays may fail to accurately detect the pupil boundary. Additionally, Starburst can struggle if the pupil moves suddenly between frames, as the initial center estimate becomes invalid [60]. Moreover, the algorithm requires careful tuning of parameters such as the number of rays, thresholding for edge detection, and stopping criteria. Although modified versions of Starburst have been proposed to address some of these limitations, the method remains less reliable in challenging lighting conditions or when the pupil is partially outside the image.

Ellipse selection

Another hybrid method is Ellipse Selection (ElSe), proposed by Fuhl et al. [61]. ElSe is a fast and efficient pupil detection method designed for resource-constrained applications such as embedded systems and automotive environments. It begins with a lightweight edge filtering process to emphasize the pupil region, followed by the evaluation of candidate ellipses on the resulting edge map to identify the pupil. According to authors, ElSe was created with a focus on conserving computational resources while maintaining reliability under non-ideal conditions, including variable lighting and reflections from eyewear. Its main strengths are speed and robustness, enabling effective performance in real-world scenarios. In testing, ElSe achieved a 14.53% improvement in detection rates over the best-performing algorithms on a large annotated dataset, underscoring its practical value.

However, like HT, ElSe assumes that the pupil is approximately elliptical in shape, which can cause performance degradation in cases where the pupil is highly distorted, such as off-axis or irregularly lit situations. Its effectiveness is also highly dependent on the quality of the edge map; weak or discontinuous edges can lead to failures in selecting correct ellipse candidates. Additionally, while ElSe is optimized for speed and designed to be lightweight, this may come at the cost of accuracy in challenging images where more detailed analysis is needed.

Other feature-based methods, such as the Sinusoidal Eye Tracker (SET) [62] and graph-based methods [63], also employ similar principles. Some methods enhance detection using polar coordinate projections [64] or corneal reflection cues (the "*bright-pupil*" effect) [65]. While these hybrid methods often outperform simpler thresholding techniques, they generally require careful integration of various heuristics such as assumptions about shape, intensity gradients, or anatomical constraints. In essence, feature-based methods trade off generality for speed: by assuming a simplified pupil model, they achieve fast processing but may struggle with unusual cases or extreme conditions.

Region and Clustering Methods

Region-based segmentation methods, such as region growing, watershed, or clustering, treat pupil detection as a problem of grouping pixels with similar intensity values. Techniques like K-means clustering or fuzzy C-means have been widely used, where pixels are categorized into different classes, typically foreground and background, based on both intensity and spatial proximity. For instance, Bai et al. [66] proposed a fuzzy clustering approach that iteratively refines the pupil threshold. They found that clustering outperforms global thresholding, particularly in low-contrast or noisy images, as it can better capture pupil features.

Fuzzy clustering, by allowing partial membership, can handle intensity inhomogeneities within the pupil, making it highly adaptable to complex image characteristics. Unlike strict thresholding, it is better equipped to tolerate noise and irregularities, as it accommodates minor misclassifications or intrusions without leading to complete failure. This ability to adapt to complex intensity distributions is one of the key strengths of region-based methods. Moreover, such methods can incorporate spatial smoothness or neighborhood constraints to enhance segmentation consistency, which helps avoid isolated misclassifications. These techniques have demonstrated higher sensitivity and specificity compared to thresholding, especially when dealing with challenging images.

However, region-based methods come with some trade-offs. They tend to be computationally heavy and iterative, with techniques like K-means and fuzzy C-means requiring multiple iterations to converge, which may not be suitable for very high frame rates. Additionally, these methods require the setting of various parameters, such as the number of clusters and fuzzy weights, and without proper regularization, they may suffer from over-segmentation. Region growing algorithms may also face issues where weak edges separate the pupil from the background.

2.4.3 Deep learning methods

Convolutional neural networks (CNNs) and other architectures have become popular for pupil segmentation, leveraging large labeled datasets to output pupil masks (segmentation). Common choices include U-Net variants, fully-convolutional networks (FCNs), and encoder–decoder architectures. These models excel at learning spatial features and can handle complex variations in images. For instance, Chen et al. [67] applied an FCN to pupil localization, achieving high accuracy, while U-Net has been adapted for pupil and iris segmentation, further demonstrating the power of deep learning in this area. With sufficient training data, deep networks can effectively deal with occlusions, lighting artifacts, and varying eye characteristics, outperforming traditional methods, thus systems like DeepVOG [68], using an FCN, has shown strong performance in various datasets. More recently, transformer-based models like the Segment Anything Model (SAM) have been fine-tuned for pupil segmentation, signaling a shift toward large pre-trained models. In particular, the second version SAM2, has demonstrated strong performance on diverse eye-tracking datasets—achieving mean Intersection over Union (IoU) scores of over 90% on both synthetic and real-world data, such as the *NVGaze* and *OpenEDS* datasets. Without requiring any fine-tuning, SAM2 matches the performance of domain-specific models, while drastically reducing annotation effort. Thanks to its ability to propagate a single point prompt across an entire video, SAM2 enables fast and scalable dataset annotation: for example, only one click was needed to annotate over 12,000 images in the *OpenEDS* dataset. This capability positions SAM2 not only as a high-performing segmentation tool but also as a powerful enabler for accelerating dataset creation in eye-tracking research [69].

The main advantage of deep learning methods is their high accuracy and robustness, as they can automatically learn complex features (e.g., reflections, eye color variations) that traditional algorithms struggle with. Once trained, CNNs can segment the pupil in a single forward pass, which, when run on a Graphics Processing Unit (GPU), can be achieved in real time. Additionally, deep models are flexible, allowing new training data to be incorporated, and recent innovations have included shape priors, such as an ellipse-fit loss, to improve geometric accuracy.

However, deep learning approaches have significant drawbacks. They require large annotated datasets and substantial computational resources for training, which can be challenging for resource-constrained environments or embedded systems. Even during inference, complex networks introduce a latency that may not be acceptable for online applications. Furthermore, deep models are prone to overfitting to the training distribution, and models trained for specific conditions may require retraining or fine-tuning for new conditions. Additionally, these models offer lower interpretability, with unpredictable failure modes when test images differ from the training set.

2.4.4 Comparative summary

To summarize, Table 2.1 compares the key characteristics of the main pupil segmentation approaches.

Table 2.1: Comparison of pupil segmentation methods: key ideas, advantages, and disadvantages.

Method	**Key Idea**	**Advantages**	**Disadvantages**
Thresholding	Binarize by intensity.	• Very fast and simple • Low computational cost	• Fails in variable light, low contrast, or noise • Eyelid/glint artifacts easily cause misclassification
Edge Detection	Detect strong gradients to find pupil boundary	• Precise localization of sharp edges • Low processing time	• Sensitive to noise and weak edges • Fragmented contours require additional processing
Hough Transform	Vote in parameter space for circular shapes	• Robust to broken or partial edges and noise • Directly yields center/radius	• Computationally intensive • Assumes ideal circular/elliptical shape • Slower on large images
Active Contours	Iteratively deform a curve by energy minimization	• Flexible shape adaptation • Captures boundary even with weak edges • Provides smooth, closed contour	• Requires good initialization and parameters • Iterative solving is slow and may miss the correct boundary
Hybrid Feature-Based (Starburst, ElSe)	Detect features and fit geometric model	• Starburst is valued for speed • ElSe is robust in real-world conditions	• Dependent on initial seed/parameters • Starburst can fail on rapid pupil motion • ElSe struggles with severely distorted shapes
Region and Clustering	Cluster pixels by intensity or grow region	• Can handle non-uniform intensities	• Iterative and slow • Requires setting cluster parameters • May over-segment or leak into background
Deep Learning	Learn segmentation from annotated data	• Highest accuracy and robustness • Can learn complex features	• Requires a large training set • GPU is necessary for real-time • Heavy models may not run on low-power devices • Generalization must be ensured for new conditions

Chapter 3

Materials and methods

This chapter outlines the materials and methods used for the design, implementation, and validation of the proposed system. It begins by presenting the architecture of the pupillometry platform and the general setup, including the hardware components and the custom modifications made. The procedures for image acquisition and data transmission are described, followed by the development of the algorithm for pupil segmentation and the detection of pupillary responses related to accommodative effort. The software interface and its role in user interaction are also illustrated, along with the functionalities integrated to support the intended communication tasks. The chapter concludes with a description of the experimental protocols and metrics adopted to assess system performance and functionality in realistic use scenarios.

3.1 Pupillometry system

The pupillometry system developed for this thesis is a wearable, custom-built solution specifically designed to meet the needs of high-resolution pupil monitoring in a compact form and low-cost factor. At its core, the system is based on a *Raspberry Pi Zero* microcontroller, which serves as the acquisition and transmission unit, in combination with a camera module and a wireless communication protocol. Unlike commercial eye-tracking devices, which are often characterized by high frame rates and a broad set of features, this system has been tailored to focus exclusively on pupillometry. In this context, high temporal resolution is not strictly required. Instead, spatial resolution and image clarity are prioritized in order to detect variations in pupil size, such as those associated with the PAR. This choice makes it possible to significantly reduce both the hardware complexity and overall cost of the device, without compromising the scientific goals of the study and the applicability in the designed context. In addition, developing a custom system allows full control over hardware and software components, enabling tailored optimization for specific experimental needs and facilitating testing with novel algorithms. The overall architecture of the system follows a linear, modular workflow represented in the subsequent schematic (Figure 3.1). Each stage will be described in detail in the following subsections of this chapter.

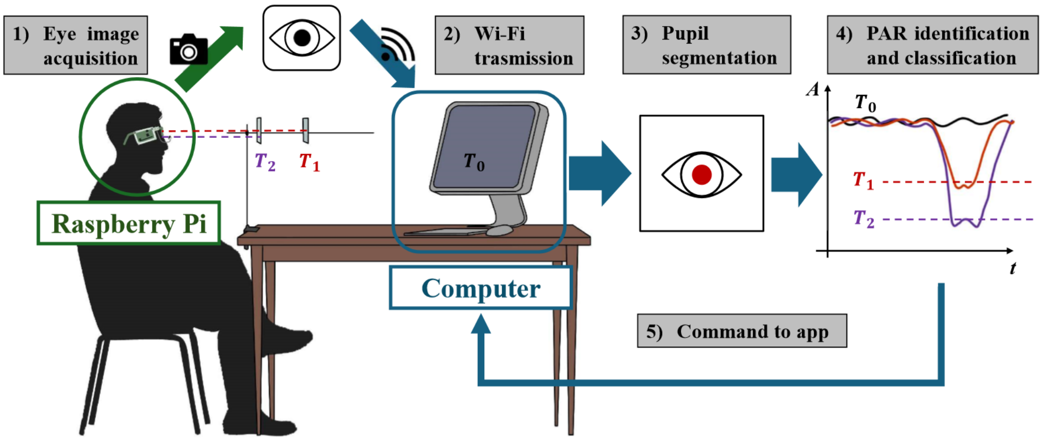


Figure 3.1: General setup and architecture of the pupillometry system

3.1.1 Device hardware components

The device essentially consists of a modified eyeglass frame, from which the lenses have been removed and the structure reshaped to preserve the nose pads while eliminating the lower part of the lens housing. This modification allows for an unobstructed view of the eye for image acquisition. A *Raspberry Pi zero* is mounted on the left temple of the frame using a custom 3D-printed support. On the opposite temple, a battery module is secured, serving as the power supply for the controller and peripherals. The camera is housed in a 3D-printed capsule, whose spatial position can be adjusted via a flexible arm to frame the left eye correctly. Two infrared (IR) LEDs are embedded into the same capsule and can be activated when needed through the microcontroller. The *Raspberry* operating system and application software are stored on a MicroSD card inserted into the designated slot on the microcontroller. A schematic representation of the device's hardware components is viewable in the following figure (Figure 3.2) while pictures of the resulting prototype are in the appendix (Figures A.1, A.2). The total cost of the proposed device was kept deliberately low to ensure accessibility and facilitate potential large-scale deployment. In total, the approximate cost of the entire system is around €40–€50, making it a highly affordable solution compared to commercial alternatives, especially in assistive technology contexts.

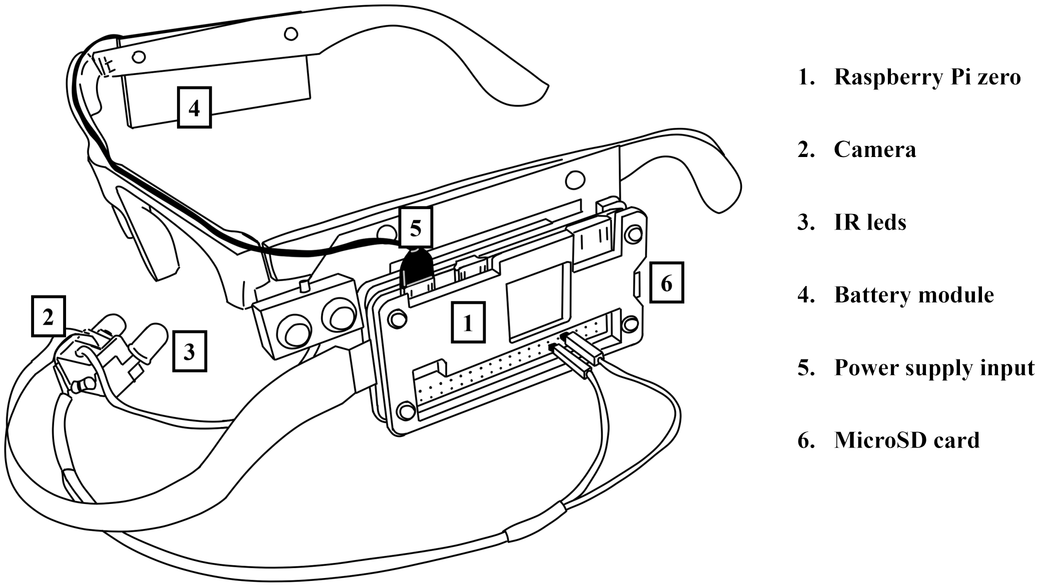


Figure 3.2: Device's hardware components schematic

Raspberry Pi zero

The *Raspberry Pi zero* is a compact and cost-effective single-board computer, well-suited for embedded applications where space, power consumption, and affordability are critical constraints. It features a 1 GHz single-core ARM11 processor and 512 MB of RAM, providing sufficient computational power for lightweight real-time image processing tasks. Despite its small footprint (65 mm $\times$ 30 mm), it includes essential interfaces such as two micro-USB ports for data and power, and a Camera Serial Interface (CSI), which enables direct connection with compatible camera modules.

In this project, the *Raspberry Pi zero* serves as the acquisition unit for the pupillometry system. Its ability to run a full Linux operating system (OS), combined with GPIO pins for hardware interfacing and camera support via the CSI port, makes it an ideal choice for integrating image acquisition, processing, transmission, and control tasks within a compact wearable setup. Furthermore, its low energy consumption is particularly advantageous for battery-powered mobile systems, ensuring extended operational time without compromising performance.

However, the *Raspberry Pi Zero* also presents some limitations. Its relatively low computational power and the shared RAM between the CPU and GPU can significantly affect processing speed. This can result in slower performance compared to more powerful embedded platforms. Nevertheless, these alternatives typically come at a higher cost and larger form factor.

Camera and IR LEDs

The image acquisition component of the system is based on the *OV5647* infrared camera sensor, a 5-megapixel CMOS image sensor widely used in embedded vision applications for its compact form factor and compatibility with the Raspberry Pi platforms via the CSI port. The *OV5647* is capable of capturing images at resolutions up to 2592×1944 pixels and supports video output at various frame rates (up to 30 frames per second), making it suitable for real-time pupil detection.

To ensure high-contrast imaging of the eye, particularly under suboptimal lighting conditions, the camera is supported by two IR LEDs positioned to evenly illuminate the ocular surface. These LEDs emit light in the infrared spectrum, which is invisible to the human eye and therefore does not interfere with natural vision or cause discomfort. The infrared illumination enhances the contrast between the pupil and the surrounding iris and sclera, facilitating pupil detection. The two LEDs are connected in series along with a 330 Ω resistor, which is included to prevent excessive current flow and protect the LEDs from potential damage. This circuit is powered directly by the *Raspberry Pi zero*: the positive terminal is connected to physical pin 12 (GPIO 18), which is controlled via software to enable or disable the illumination as needed, while the negative terminal is connected to physical pin 9 (Ground) (Figure 3.3). This configuration allows for programmatic control of the infrared lighting, enabling activation only when required for image acquisition, thus optimizing power consumption and thermal management in the system.

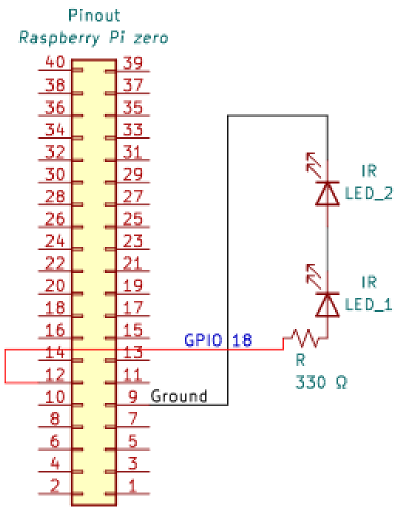


Figure 3.3: Schematic of IR LEDs connection

Battery module and power supply

The device is powered by a rechargeable lithium-polymer (Li-Po) battery with a nominal voltage of 3.7 V and a capacity of 2000 mAh, chosen for its compactness, light weight, and adequate energy density for wearable applications. To ensure safe operation and satisfy voltage constraints, a *TP4056*-based power management module is used, it provides also information about battery charging state through two small, incorporated LEDs. This module integrates essential protection features, including overcharge, over-discharge, short-circuit protection, and provides a micro-USB port for battery recharging. The efficiency during discharge is generally high as the internal losses from the protection circuitry are minimal. However, since the *TP4056* is a linear charger, energy losses during the charging phase can be more significant, especially at higher currents. Overall, the module introduces only a small power overhead during operation, it was therefore excluded from the calculation of power consumption. The battery is connected to the *TP4056* module, which regulates voltage raising it up to 5 V, as necessary for the functioning of *Raspberry Pi zero* and its peripherals. An inline toggle switch is included in the circuit, allowing the caregiver to manually power the system on or off. Importantly, the *TP4056* module allows the system to remain operational while the battery is charging. Additionally, the *Raspberry Pi zero* features a secondary micro-USB port, which can be used to power the device directly from a computer or from a standard USB wall adapter, offering flexible powering options for both portable and stationary use.

In the absence of precise and official documentation regarding the power consumption of each component in the system, only approximate estimations can be made based on available technical datasheets and community-reported measurements. The *Raspberry Pi zero* has a variable current draw depending on its workload, ranging from around 100 mA in idle conditions to 300 mA under stress, according to both the Raspberry Pi Foundation and independent tests. The *OV5647* camera module, consumes approximately 200 mA when fully active, as derived from the product documentation. Additionally, a pair of standard 5 mm infrared LEDs typically draw about 20 mA each, and USB Wi-Fi dongles used for wireless transmission may consume up to 150 mA depending on the data load. Taking together, these components result in a worst case power consumption scenario estimated at 690 mA. However, due to the lack of standardized and official measurements across all configurations, these figures must be regarded as conservative approximations rather than definitive values. Under these conditions, the estimated operational time, in the worst-case scenario, provided by the 2000 mAh battery is computed in the following equation:

$$\text{Battery life} = \frac{\text{Battery capacity}}{\text{Device consumption}} = \frac{2000\ \text{mAh}}{690\ \text{mA}} \approx 2.9\ \text{hours} \approx 2\ \text{h}\ 55\ \text{m} \tag{3.1}$$

this represents the estimated device's consumption and serves only as an indication to evaluate the suitability of the hardware components in relation to the application. Real power consumption will be significantly lower than the value reported, mainly because rarely all components work at maximum load simultaneously. Moreover, as described in the following subsection, modifications were made to the system configuration to enhance device performance. These changes likely lead to increased power consumption; however, a theoretical estimation is not sufficiently accurate due to factors such as dynamic frequency scaling and individual chip variability. For this reason, an experimental verification of the actual power consumption has been performed and will be reported in the Results chapter.

3.1.2 Raspberry Pi OS and modifications

For the implementation of the system, the *Raspberry Pi OS Lite* (Bullseye) was chosen as the operating system. This version of the OS is a minimal Debian-based distribution developed specifically for the Raspberry platforms. Unlike the full version, the *Lite* edition does not include a graphical user interface, which makes it particularly well-suited for headless, resource-efficient applications such as the one described in this thesis. The operating system was installed on a microSD card using the official *Raspberry Pi Imager* tool. During the installation process, the advanced settings of the *Imager* were used to preconfigure several essential parameters. In particular, the wireless network credentials were entered to allow the *Raspberry Pi zero* to automatically connect to the Wi-Fi network on first boot. Additionally, the Secure SHell (SSH) service was enabled to allow remote access via terminal using *PuTTY* software. This was a crucial step to manage the device entirely from a PC, without requiring a display, keyboard, or mouse connected to the Raspberry.

In order to enhance the performance of the *Raspberry Pi zero* and ensure compatibility with camera-based applications, several modifications were made to the system configuration file. These changes are aimed at improving both computational speed and graphical responsiveness while maintaining system stability. Firstly, GPU memory allocation was increased to 256 MB, a crucial adjustment for image processing tasks and real-time video streaming, especially when using the option to enable camera support via the legacy firmware interface. Additionally, the core and GPU frequencies were manually raised to 400 MHz to provide better throughput for graphical operations. The ARM11 frequency parameter was explicitly set to 1000 MHz enabling also the option that allows the CPU to scale dynamically to its maximum supported frequency based on thermal and power constraints. Furthermore, USB communication was enabled by activating the option `"dtoverlay"` in the configuration file. This allows the *Raspberry Pi zero* to behave as a USB gadget when connected to a host computer. These adjustments were specifically tailored to optimize the system for this application and to improve the usability of the device. Collectively, these custom system settings are compared with default settings in the following table (Table 3.1).

Table 3.1: Comparison between default and custom system settings on *Raspberry Pi zero.*

Component	**Parameter**	**Default Value**	**Custom Value**	**Description**
CPU	`arm_freq`	700 MHz	1000 MHz	ARM CPU clock frequency determines overall processing speed.
GPU	`core_freq`	250 MHz	400 MHz	Clock frequency of the GPU core handling general-purpose processing tasks.
GPU	`gpu_freq`	250 MHz	400 MHz	Clock frequency of the GPU graphics engine.
GPU	`gpu_mem`	64 MB	256 MB	RAM allocated to the GPU, required for video capture and graphical tasks.
Camera	`start_x`	0 (disabled)	1 (enabled)	Enables legacy camera interface and required GPU features for video handling.
USB	`dtoverlay`	Not present	`dwc2` (Enabled)	Enables USB OTG support for device mode communication.

Additionally, a custom device label (via static USB descriptor modification) was assigned, ensuring that the device is always recognized as "*RPIZERO*" by connected systems, facilitating consistent identification across multiple sessions and enabling automatic detection even when multiple peripheral USB are connected to the computer. Moreover, a memory partition of 20 MB was created in the MicroSD in order to store text files containing Wi-Fi credentials and computers IP addresses with the aim of making it easier to connect the device to different Wi-Fi networks. Furthermore, Bluetooth and HDMI output were disabled to reduce power consumption as other various unnecessary services to minimize background processes, ensuring the *Raspberry Pi zero* operates with optimal energy efficiency.

3.1.3 Computer hardware and Python libraries

In the proposed system architecture, the *Raspberry Pi zero* is responsible for image acquisition, while all processing, user interaction, and graphical rendering are delegated to an external computer. This design choice allows for greater computational flexibility and simplifies the development of a user-friendly interface.

This section describes the specifications of the computer used during development and testing, along with the Python libraries employed to implement the software components.

Computer hardware components and OS

The computer used is the *HP Pavilion Gaming Desktop TG01* series, introduced in 2020, and is designed as a mid-range gaming and multimedia desktop. This configuration cost around €850 at the release. The main hardware components, along with OS are specified in Table 3.2.

Python libraries

The Python environment used for this project was based on version *Python 3.9.13* and included several scientific and GUI libraries. A full list of packages and versions is available in the following table (Table 3.3) to permit reproducibility and compatibility between libraries.

Table 3.2: Computer specifications.

Component	**Specification**
Model	HP Pavilion Gaming Desktop TG01
Processor	Intel Core i5-10400F CPU @ 2.90 GHz
GPU	NVIDIA GeForce GTX 1660 SUPER (6 GB)
RAM	16 GB DDR4
Storage	1 TB SSD
Operating System (OS)	Windows 11 Home (22H2)
Display resolution	2560 $\times$ 1440 (2K)
Network Interface Card (NIC)	Realtek Gaming GbE Family Controller

Table 3.3: Python libraries and versions.

Library	Installed Version
cv2	4.10.0.84
numpy	1.26.4
pandas	2.2.2
matplotlib	3.9.0
scikit-learn	1.5.0
scipy	1.13.0
pyqtgraph	0.13.7
PyQt5	5.15.11
pyttsx3	2.98
tkinter	built-in (no version needed)
re	built-in (no version needed)
os	built-in (no version needed)
sys	built-in (no version needed)
time	built-in (no version needed)
subprocess	built-in (no version needed)
socket	built-in (no version needed)
threading	built-in (no version needed)
queue	built-in (no version needed)
glob	built-in (no version needed)

3.1.4 Images acquisition and transmission protocols

In the context of a low-power embedded system, reliable and efficient video transmission plays a crucial role. The implementation described herein enables the real-time streaming of video frames from a Raspberry camera module to a remote computer using the Transmission Control Protocol/Internet Protocol (TCP/IP) only after the connection via User Datagram Protocol (UDP) has been verified. Notably, both communication protocols need the IP address of the host (PC), this information is stored in the memory partition dedicated and is read at the beginning during device startup. After checking the correct functioning of IR LEDs, images are captured in the YUV format, cropped to ensure computational efficiency, and sent to the computer for elaboration. The following scheme (Figure 3.4) represents the overall functioning of the transmission protocols. Details are discussed in the following subsections.

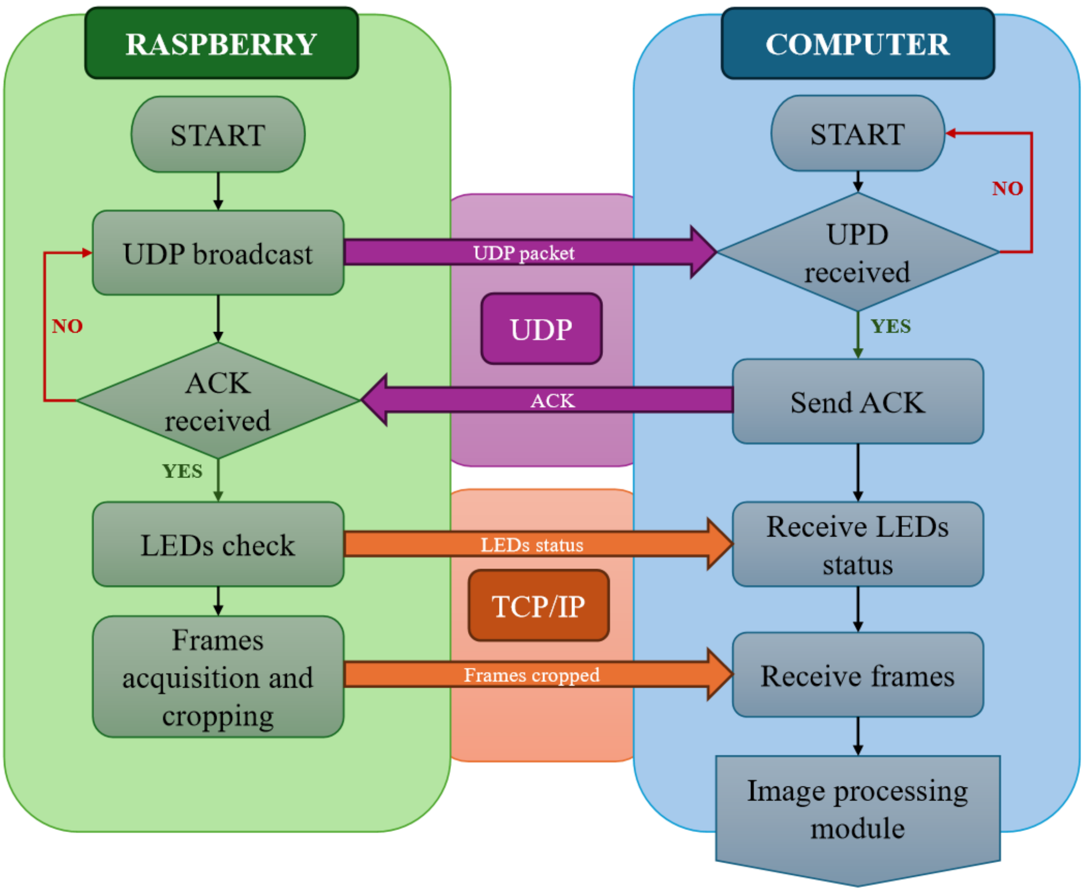


Figure 3.4: Schematic of transmission protocols.

UDP-based device discovery and activation handshake

Before initiating the real-time video streaming over TCP/IP, the system must ensure that the receiving computer is ready to accept and process the video stream. This role is fulfilled by a lightweight discovery and handshake protocol implemented via UDP. A secondary script is responsible for this phase, which precedes and triggers the main streaming logic. Unlike TCP/IP, UDP is a connectionless protocol that allows fast and efficient sending of small packets without the overhead of connection setup or guaranteed delivery. In this application, UDP is used for broadcasting the presence of the device on the network and waiting for an acknowledgment (ACK) from the host PC. The *Raspberry Pi zero* continuously sends the message "*Raspberry Presente*" to the PC at a predefined IP address and port. This packet acts as a heartbeat or discovery message, announcing the microcontroller's availability and readiness to stream video. The receiving PC, upon being ready, replies with a confirmation message (ACK) on a different UDP port. Key characteristics of this approach include:

- **Low latency and overhead**: Since UDP does not establish a persistent connection, communication is immediate and well-suited for lightweight signaling.
- **Periodic retry with timeout**: To avoid indefinite blocking in case the PC is not yet ready or unavailable, a timeout mechanism is implemented. If no response is received, *Raspberry Pi zero* try again after a 5-second pause.
- **Robust initiation logic**: Once the ACK is received, the microcontroller launches the main camera acquisition script using a subprocess call. This separation of concerns ensures that the camera is only activated after successful coordination with the PC.

this modular two-step communication model improves both robustness and resource management. The camera and LEDs are only activated when strictly necessary, minimizing power consumption.

TCP/IP protocol for data transmission

The (TCP/IP) suite is the fundamental communication protocol used across networks, ensuring reliable and ordered delivery of data. In this implementation, *Raspberry Pi zero* acts as a TCP client, establishing a connection with a preconfigured remote PC server. TCP offers several advantages for video data transmission in this context:

- **Reliability**: TCP ensures that all data packets are delivered and reassembled in order.

- **Flow control and congestion handling**: TCP adjusts the transmission rate based on the network conditions, reducing the risk of packet loss and jitter.

- **Ease of implementation**: TCP abstracts away lower-level error handling and retransmission mechanisms, simplifying the development process.

To transmit frames captured from the Raspberry's camera, a simple application-layer protocol is implemented over TCP. Each image frame is encoded in JPEG format and preceded by a 4-byte header, which encodes the frame length in bytes. This header allows the receiver to know exactly how many bytes constitute the current frame, avoiding data misalignment and ensuring that frame boundaries are respected.

LEDs control

Before video transmission begins, the system performs a self-check of the LEDs illumination, verifying its functionality by comparing the average luminance before and after turning the LEDs on, in particular the system confirms LEDs functioning if the average luminance on 10 frames after the lighting is at least 50% greater than before. If the test confirms LED functionality, the system transmits over TCP a status message (“`LED_OK`”) to the PC. Otherwise, an error code (“`LED_FAIL`”) is sent, and further action can be taken. This ensures that lighting conditions are well-controlled during acquisition, but in brightly lit environments the saturation of the camera sensor could lead to false LEDs malfunctioning warning. This control is essential, as LEDs illumination plays a critical role in the system's operation, and a malfunction could lead to a complete failure. For this reason, LED control is performed prior to the transmission phase, allowing for immediate feedback in case of a malfunction and enabling appropriate corrective actions.

YUV and frames format

The images are captured in YUV format, specifically the *YUV420* planar format supported by the Raspberry's *PiCamera* module. This format separates the luminance (Y) from the chrominance (U and V) components of the image. The system utilizes only the Y channel, directly extracted from the camera, for the following reasons:

- **Brightness-based processing**: Pupil detection relies on brightness contrast. The luminance channel contains all the brightness information, making it well-suited for segmentation tasks such as thresholding or contour detection performed afterwards.

- **Performance**: Capturing and processing only the Y channel reduces the computational load significantly. On devices like the *Raspberry Pi zero*, this efficiency is crucial to achieving real-time performance.

The YUV format is the preferred choice for this application over simple grayscale images where 3 RGB channels are combined by software utilizing important computational resources. Moreover, the extracted Y channel does not lose detail capturing the same level of information as grayscale images. Frames were recorded at a resolution of 640×480 pixels with a target rate of 30 frames per second (FPS). However, the actual frame rate is influenced by system performance, particularly the Wi-Fi transmission bandwidth and the computational load of the acquisition and encoding process. After acquisition, each frame undergoes a cropping operation that removes 80 pixels from both the left and right sides, and 50 pixels from the top and bottom borders. This processing step assumes that the eye remains approximately centered within the frame. It allows the use of a standard and supported camera resolution (640×480 pixels) while reducing the amount of data transmitted, thereby optimizing system performance. Cropping the frames reduces the data size by R, as explained by Equation 3.2:

$$R = \frac{N_o - N_c}{N_o} \cdot 100\% = \frac{\Delta B}{N_o} \cdot 100\% \tag{3.2}$$

where R is the data reduction in percentage, computed as the relative reduction between the original number of pixels N_o and the cropped one N_c, and ΔB is the difference in bytes, since the pixels/bytes ratio is 1:1 for single-channel images. Applying the specific values of this application we obtain the quantities expressed in the following Equation 3.3:

$$R = \frac{307{,}200 - 182{,}400}{307{,}200} \cdot 100\% = \frac{124{,}800 \text{ bytes}}{307{,}200} \cdot 100\% \approx 40.6\% \tag{3.3}$$

thus, the reduction translates to a saving of around 122 KB per frame, in other terms a percentage of saving approximately equal to 40.6%, which consistently reduces the load on data transmission and image processing. Nevertheless, the time required to perform the cropping operation on the *Raspberry Pi zero* must be considered, as the system's limited processing capability may impact on the overall performance. On *Raspberry Pi zero*, this operation is estimated to require 0.35 milliseconds per frame (t_{cropping}). Assuming a Wi-Fi transmission speed as the maximum available of 65 Mbps (i.e. 8.125 MB/s), the transmission time saved by cropping frames calculated as:

$$\Delta t_{\text{trasmission}} = \frac{\Delta B}{v_t} = \frac{124{,}800 \text{ bytes}}{8{,}125{,}000 \text{ bytes/s}} \approx 15.4 \text{ ms} \tag{3.4}$$

where v_t is the Wi-Fi transmission velocity. Considering the cropping operation, the net time gain per frame becomes:

$$\Delta t_{\text{net}} = \Delta t_{\text{trasmission}} - t_{\text{cropping}} = 15.4 \text{ ms} - 0.35 \text{ ms} \approx 15 \text{ ms} \tag{3.5}$$

this analysis demonstrates that the time required to perform the cropping operation is significantly outweighed by the time saved during data transmission. Moreover, as evident from Equation 3.4, a lower transmission speed (v_t) results in a greater transmission time difference $\Delta t_{\text{trasmission}}$, while the cropping time (t_{cropping}) remains constant. Consequently, the relative impact of frame cropping on processing time further diminishes as the transmission speed decreases, thereby reinforcing the justification for adopting this approach. These considerations are relevant, given that v_t is influenced by various factors and may substantially decrease under real-world operating conditions. Comparable reasoning can be extended to segmentation processing; however, since this is performed on a host computer with substantially higher computational capacity, the associated cost is negligible and was not considered in this analysis.

3.1.5 Pupil segmentation algorithm

This section details the pupil segmentation pipeline; a computationally efficient and robust method developed for identifying and characterizing the pupil within video frames. The approach integrates several techniques, including spatial match filtering, region-of-interest (ROI) extraction, histogram analysis, thresholding, edge detection, and contour filtering. The workflow of this segmentation algorithm is visually represented in the subsequent scheme (Figure 3.5), and each step will be thoroughly explored in the following sections.

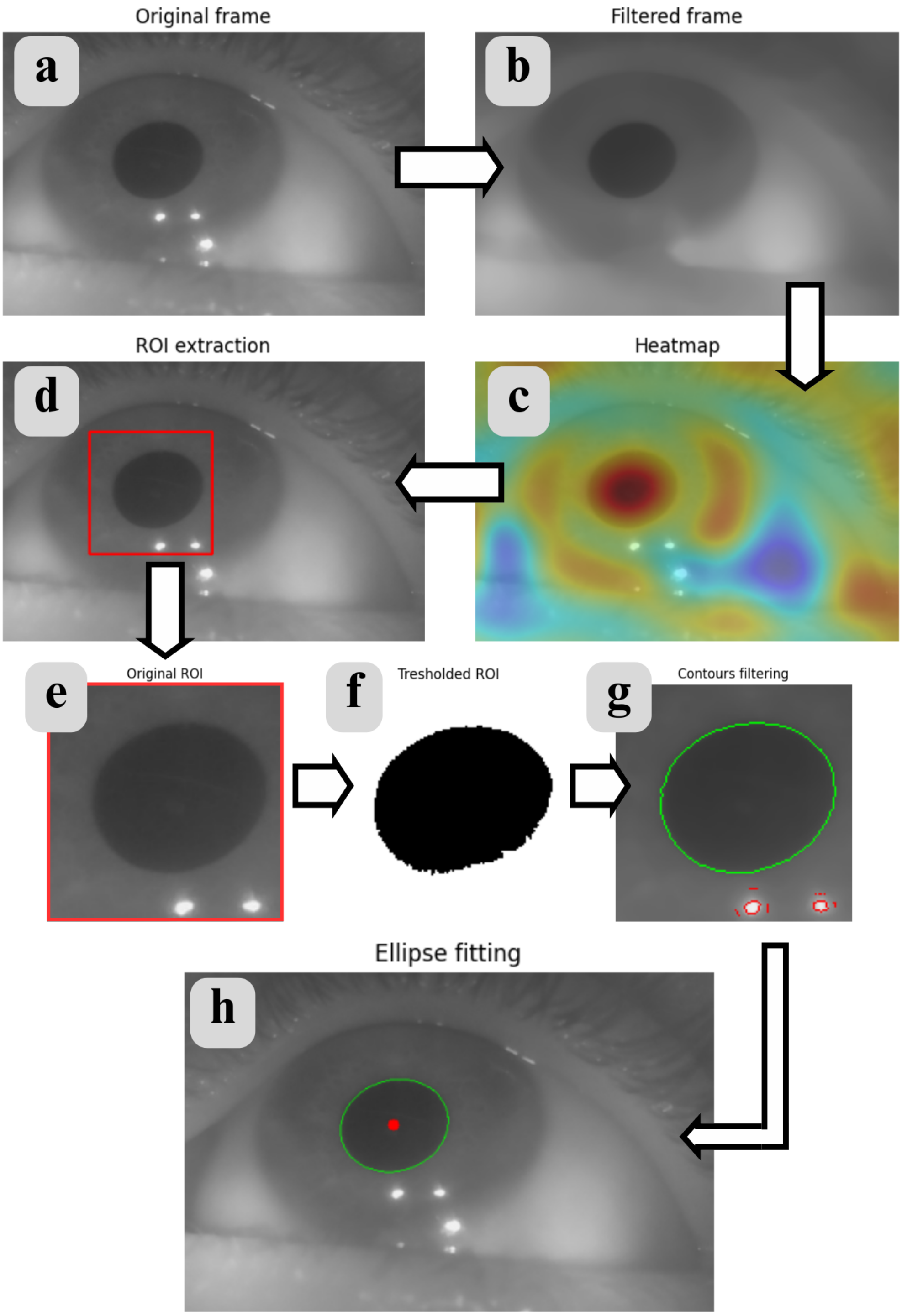
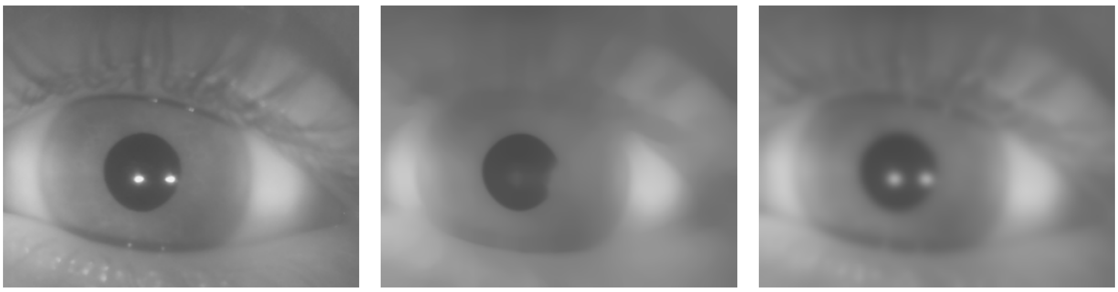


Figure 3.5: Segmentation algorithm workflow, where: (a) original, (b) median filtered, (c) heatmap, (d) ROI extraction, (e) original ROI, (f) thresholded ROI, (g) contours filtering, (h) ellipse fitting.

Median filtering

The processing begins with the application of a median filter to the original grayscale image in order to reduce noise while preserving important structural details, such as edges. In this implementation, a kernel of size 33×33 is employed. For each pixel, the intensity values within the corresponding neighborhood are sorted, and the median is assigned to the central pixel. This non-linear filtering technique is highly effective in eliminating impulse noise and isolated intensity spikes, commonly referred to as "*salt-and-pepper*" noise, without introducing significant blurring. Unlike linear filters (e.g. Gaussian) that tend to smooth across boundaries and dilute edge information, the median filter maintains sharp transitions between regions because it does not average intensity values but instead selects a representative value from the local neighborhood. In this context, the median filtering step is employed exclusively as a preprocessing technique to attenuate unwanted reflections that could interfere with the subsequent stage of ROI detection. Reflections located at the center or along the boundary of the pupil are particularly problematic, as they can distort the response of the kernel-based ROI localization method. In contrast, reflections confined entirely to the iris or sclera tend not to affect the performance of the algorithm. As illustrated in the following image (Figure 3.6b), after applying the median filter, the edges of the pupil may appear partially distorted, especially in the presence of strong central reflections. However, this does not compromise the final result, since the actual segmentation step is performed on the original ROI extracted from the unfiltered frame. Another confounding factor in ROI detection is the presence of eyelashes. These structures may be misinterpreted as part of the pupil or its boundaries leading to incorrect ROI positioning, also in such cases, the median filter is effective. As further shown in Figure 3.6c, Gaussian filtering fails to adequately suppress both reflection artifacts and eyelashes, confirming the superior suitability of the median filter for this application.



(a) (b) (c)

Figure 3.6: Effect of median filtering (b) and Gaussian filtering (c) on the original frame (a).

Heatmap calculation and ROI extraction

The filtered image is convolved with a circular kernel specifically designed to emphasize dark, circular regions resembling the pupil. The kernel is implemented as a two-dimensional square matrix of size 133×133 pixels, where a circular region centered within the kernel and with a radius equal to one fourth of the kernel size is assigned negative weights, while the outer area is assigned positive weights (Figure 3.7.). The entire kernel is then normalized to have zero mean.

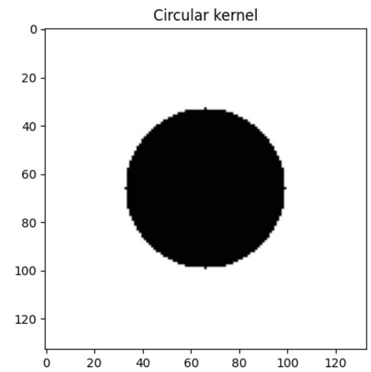


Figure 3.7: Representation of the circular kernel used for convolution: black area (negative weights) and white area (positive weights).

The convolution operation involves sliding the kernel across the median filtered image with a stride of one pixel both horizontally and vertically. At each position, the kernel is element-wise multiplied with a corresponding patch of the same size, and the resulting products are summed to produce a single scalar value. This value becomes the intensity of the corresponding pixel in the output heatmap. Mathematically, the convolution at position (i, j) can be expressed as:

$$H(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} I(i+u, j+v) \cdot K(u,v) \tag{3.6}$$

where:

- $H(i, j)$ is the value of the heatmap at position (i, j),
- I is the median filtered image,
- K is the convolution kernel of size n,
- $k = \frac{(n-1)}{2}$.

In this context, the kernel is designed with a negative inner region and a positive outer region. When this kernel overlaps with a pupil, the dark center of the pupil aligns with the negative weights, and the surrounding brighter areas align with the positive weights. This results in a strong response in the heatmap due to the contrast in intensity and the structure of the kernel. The pixel location corresponding to the global maximum in the heatmap is then identified, as it most likely represents the center of the pupil. A square ROI centered at this maximum is then extracted from the original image, with a side length equal to the kernel size. This ROI, now centered around the likely pupil position, is passed to subsequent stages for segmentation. A visual representation of this process can be seen in the following schematic (Figure 3.8.).

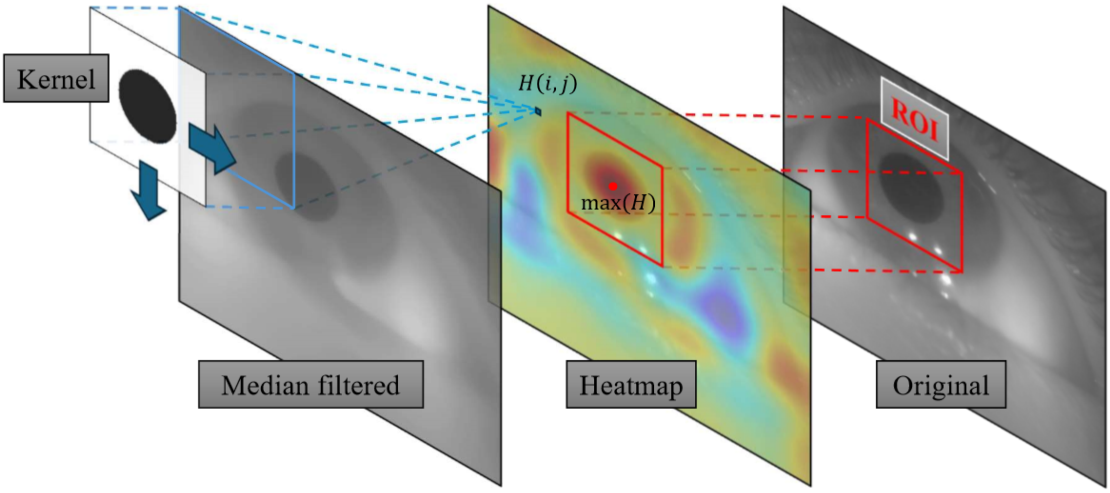


Figure 3.8: Schematic representation of convolution for heatmap generation and ROI extraction.

Although this approach yields satisfactory results in most cases, it relies on the implicit assumption that the pupil's size is known and relatively constant, which is not always true in real-world conditions, moreover during accommodation tasks. A more robust strategy would involve performing multiple convolutions using kernels of varying sizes, enabling the detection of pupils across a broader range of diameters. However, since the purpose of this step is not the precise segmentation of the pupil but rather the localization of a candidate ROI, this additional complexity is unnecessary. A single convolution with a fixed-size kernel provides a good trade-off between accuracy and computational efficiency. In fact, the convolution step represents one of the most computationally intensive parts of the algorithm. For this reason, GPU-based acceleration was also evaluated. However, the time required to transfer data between CPU and GPU offsets the benefits gained from faster convolution operations, resulting in no net performance improvement.

As an alternative to the convolution-based approach, a faster method was tested to estimate the pupil center by analyzing pixel intensity distributions. Specifically, this strategy involves thresholding the grayscale image using a luminance value derived from the image histogram to identify dark regions. Then, the number of pixels below the threshold is counted along each row and each column, producing one-dimensional projection profiles. The row with the highest count corresponds to the estimated vertical coordinate of the pupil center, while the column with the highest count indicates the horizontal coordinate. This method is computationally efficient, as it relies on simple arithmetic operations rather than costly convolutions. However, since it considers only pixel intensity and not shape, it is more susceptible to errors under non-uniform lighting conditions or in images with large dark areas unrelated to the pupil. Therefore, the convolution implementation with a fixed-size kernel was adopted as the most effective and robust solution.

ROI thresholding and contours filtering

After identifying the ROI around the estimated pupil center, a segmentation step is performed. This process involves thresholding the ROI to identify dark and bright areas, and filtering the contours based on this classification. First, a local histogram of the grayscale ROI is computed (Figure 3.9). Since the pupil appears as the darkest region, a peak detection algorithm is applied to the inverted histogram to estimate an appropriate threshold value. The first detected peak is interpreted as the intensity valley between the pupil and surrounding structures; the threshold is then set slightly above this value (by 10 gray levels) to better isolate the dark region. This threshold is used to binarize the ROI: pixels below the threshold are set to 0 (black), and others to 255 (white). This binarized ROI is used solely to classify regions of the image based on their darkness levels, enabling the distinction between contours that are likely to belong to the pupil and those that originate from other structures such as eyelids or reflections. This step is essential because using simple binarization alone to extract the pupil would often lead to incorrect segmentations, especially in cases where a reflection overlaps the center or the boundary of the pupil as it is viewable in the following figure (Figure 3.10).

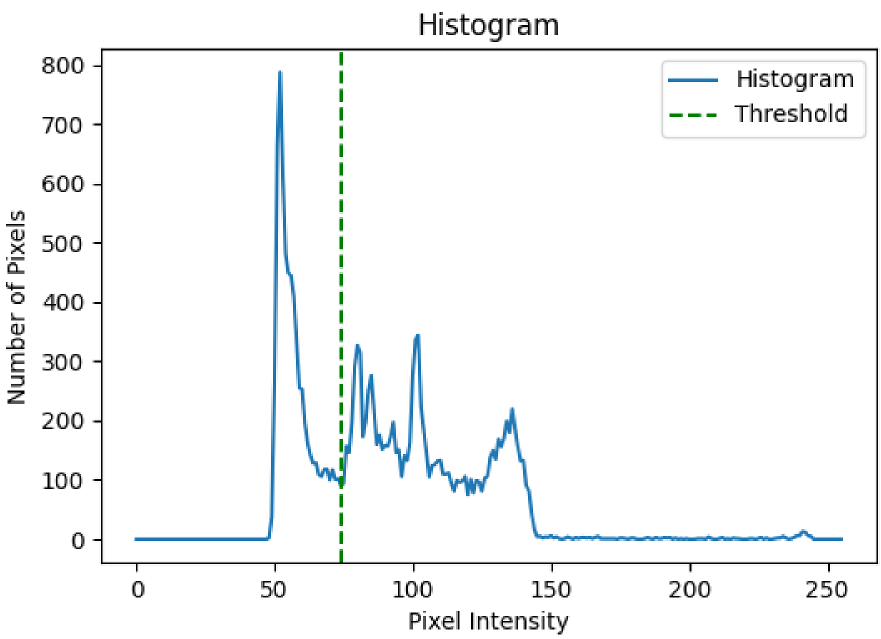
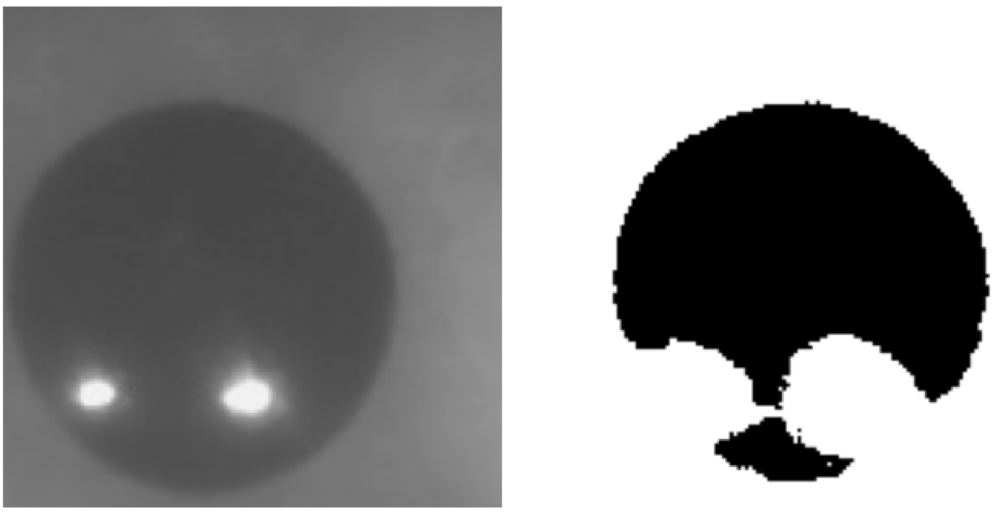


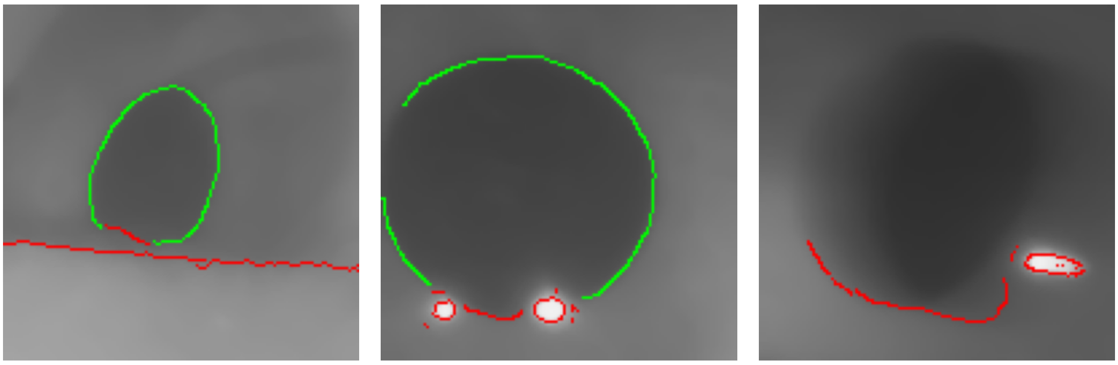
Figure 3.9: Example of an intensity histogram for an eye image, with the selected threshold indicated by a vertical green line.



(a) (b)

Figure 3.10: Effect of thresholding on the ROI: (a) original region of interest with a reflection overlapping the pupil; (b) corresponding binarized image after thresholding.

To perform the segmentation, the Canny edge detection algorithm is applied to the original version of the ROI in order to extract prominent edges. The contours corresponding to these edges are then identified and individually evaluated. Each contour is assessed based on its spatial overlap with the binary mask obtained from thresholding. Specifically, a contour is retained as a candidate for the pupil boundary only if it includes at least a subset of 5 pixels that fall within the dark regions identified by the thresholded mask. This filtering step effectively discards contours associated with brighter areas, such as eyelids or reflections, thereby reducing the likelihood of false positives and improving the robustness of the segmentation process. Moreover, the contour filtering step proves effective even in blurred images caused by eye movements. In such cases, the edges of the pupil become indistinct or completely undetectable, but the Canny operator may still detect spurious edges. Since these false contours generally do not overlap with the dark regions identified by thresholding, they are naturally excluded by the filtering process. This behavior enhances the robustness of the segmentation pipeline under real-world, non-ideal imaging conditions. Examples of contour filtering under the discussed conditions are shown in Figure 3.11, where green contours represent accepted candidates, while red contours indicate those that have been rejected.



(a) (b) (c)

Figure 3.11: Effect of contour filtering: (a) eyelid occlusion; (b) reflection; (c) blurred pupil due to movement. Green lines represents accepted contours while red lines denote rejected ones

Ellipse fitting and eccentricity controls

Among the valid contours, the one with the largest area is selected for further analysis. This choice is based on the assumption that, under normal imaging conditions, the pupil appears as the darkest and most prominent circular or elliptical region within the eye image. Smaller contours are more likely to result from partial shadows, reflections, or noise, and therefore are less reliable indicators of the actual pupil boundary. Once the largest contour is identified, an ellipse is fitted to it using a least-squares approximation method. This step is crucial because the pupil, although not perfectly circular in every frame, typically exhibits an approximately elliptical shape due to perspective distortion and anatomical variability. Fitting an ellipse allows for a robust and geometrically meaningful representation of the pupil, which can be easily used for subsequent analysis, such as area calculation. Additionally, this approach smooths out small imperfections in the detected contour, making the segmentation more stable across frames.

To ensure geometric plausibility, the ellipse is further evaluated based on its eccentricity. The eccentricity is computed as:

$$\epsilon = \sqrt{1 - \frac{b^2}{a^2}} \tag{3.7}$$

where a and b are the semi-major and semi-minor axes of the ellipse, respectively. Only ellipses with eccentricity below a predefined threshold, equal to 0.85, are accepted, as this corresponds to near-circular shapes typical of the pupil. At the end of the segmentation process, the area of the pupil is estimated by computing the area of the fitted ellipse using the standard formula:

$$A = \pi \cdot a \cdot b \tag{3.8}$$

in addition to the area, the center of the fitted ellipse is also extracted. This point represents the estimated centroid of the pupil in the image plane and provides valuable information for tracking the spatial position of the pupil over time. Monitoring the position of the pupil center across frames is particularly important during the testing phase of the system. It allows for the detection of excessive eye movements that may compromise the validity of the measurements. In fact, the target application of the system is to assist individuals in CLIS, a condition in which voluntary eye movements are severely limited or absent. Therefore, trials in which the pupil undergoes substantial displacements are not representative of the intended use case and can be excluded from the analysis.

At the end of the segmentation process, if a valid ellipse has been detected, it is transformed back from the coordinate system of the ROI to the coordinate system of the original full-frame image. This transformation is necessary because the ellipse fitting is performed on the cropped ROI, and its position must be realigned with the original image for visualization purposes. Once reprojected into the full image, the fitted ellipse is superimposed on the original frame. The contour of the ellipse is drawn in green, while its center is marked with a red dot. This graphical overlay serves as immediate visual feedback to assess the correctness of the segmentation and to intuitively verify whether the detected region aligns with the pupil. An example of a resulting segmented frame, including the overlaid ellipse and center marker, is shown in the following image (Figure 3.12). This visual confirmation is especially useful during the development and testing phases, as it allows rapid identification of segmentation errors or anomalies. Furthermore, during the operational phase, it can also assist in the adjustment of the device's position. By visualizing the location of the pupil within the frame, the operator can ensure that the eye remains properly centered in the field of view of the camera.

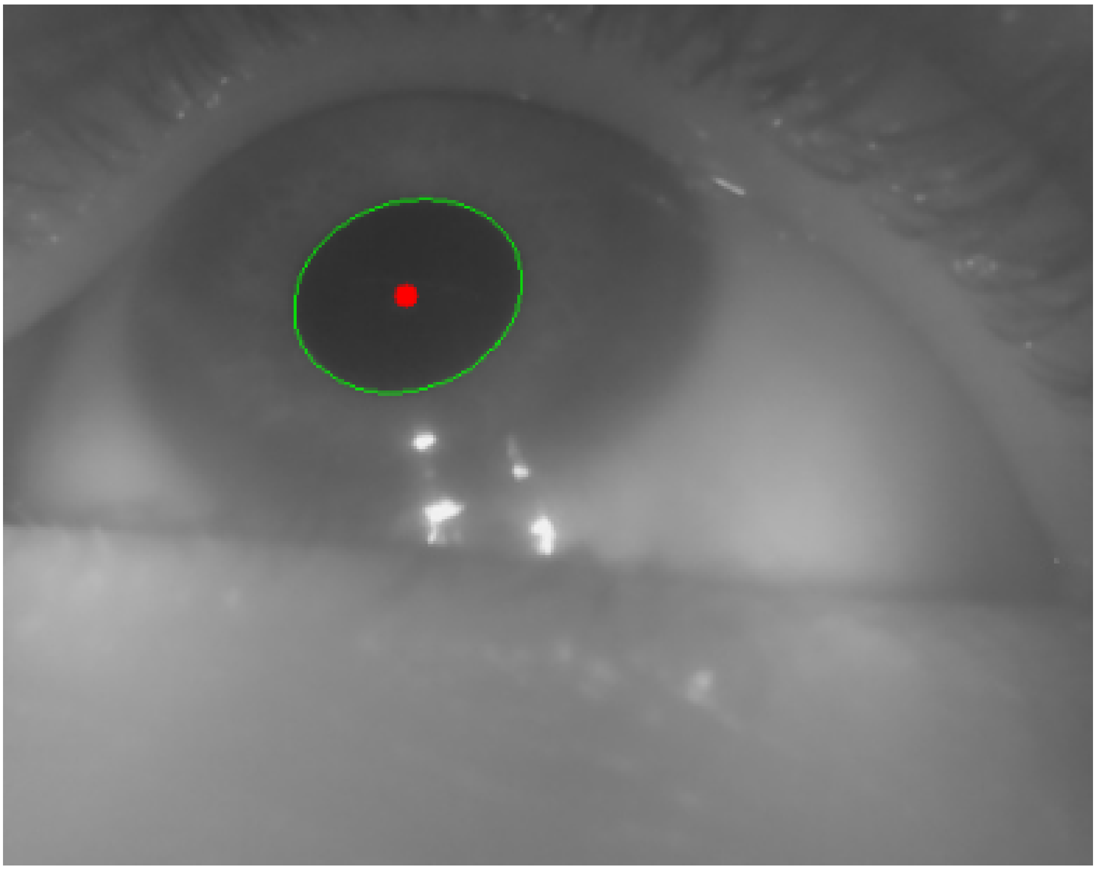


Figure 3.12: Example of final result of pupil segmentation algorithm.

3.1.6 PAR events identification module

This module is designed to detect and classify pupillary constriction events in real time, as part of the communication interface based on PAR. The system operates in two main phases: a decoder configuration phase and an operational phase. During the configuration phase, the user is asked to gaze at the screen to allow the system to compute a stable pupil area, which serves as a reference (baseline) for identifying future constrictions. After the baseline is established, the user is prompted to perform two constrictions: the first associated with option 1 (far target related to a moderate constriction), and the second with option 2 (near target linked to a pronounced constriction). These responses are used to calibrate two adaptive levels necessary for distinguishing between the two command types. Once configuration is successfully completed, the system enters the operational phase, where it continuously monitors pupil size. The algorithm detects and classifies constriction events in real time, analyzing features such as duration and amplitude, enabling control of an external application through the execution of specific commands. A schematic representation of the module's overall workflow is provided below (Figure 3.13). The following subsections of this chapter will describe in detail the implementation of the main components of the module.

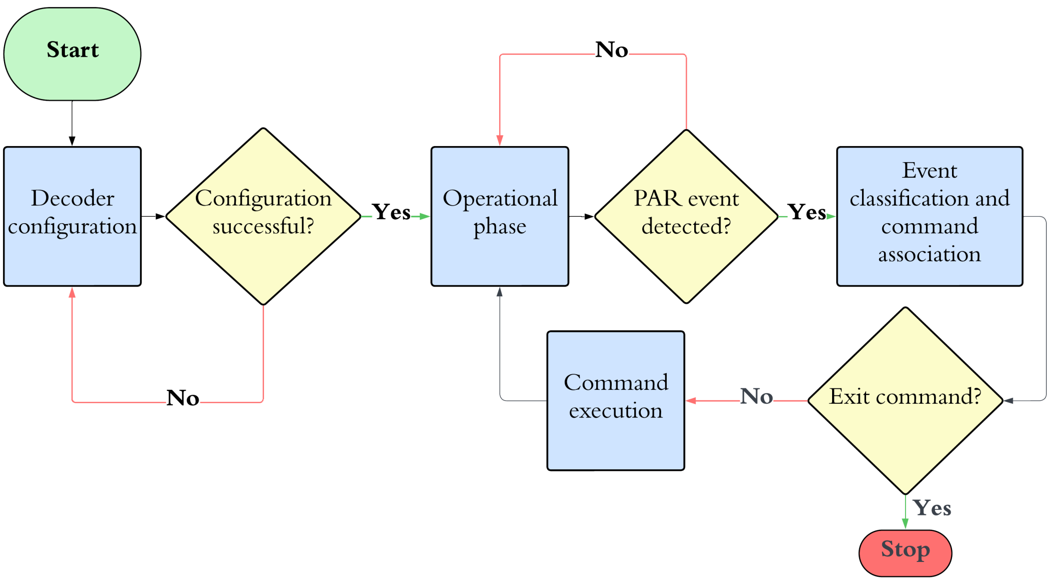


Figure 3.13: Flowchart of PAR identification module.

Decoder configuration phase

The configuration phase is a crucial initial step that ensures the system can reliably interpret future pupillary constrictions as intentional commands. It consists of three stages: baseline acquisition, type 1 constriction, and type 2 constriction. The system begins by recording the pupil area continuously for 30 seconds while the user maintains a steady gaze at the computer screen that is associated with no command. This period is used to compute a baseline signal, obtained by calculating the average of the pupil area over time. The baseline serves as a dynamic reference, allowing the system to account for the user's physiological variability and environmental conditions. To ensure the stability of the baseline during configuration phase, the system evaluates the coefficient of variation, defined as:

$$CV = \frac{\sigma}{\mu} \tag{3.9}$$

where σ is the standard deviation and μ is the mean of the pupil area signal during the baseline window. If the computed CV exceeds a predefined threshold equal to 30%, the configuration is considered unsuccessful, and the baseline acquisition is repeated. Throughout this process, the user receives audio feedback via a text-to-speech synthesizer, which announces the start and end of the configuration, as well as whether it was successful or needs to be repeated. Following a successful baseline acquisition, the system proceeds with the collection of two reference pupillary constriction events, corresponding to type 1 and 2. The user is first instructed, via an audio command, to focus on target 1 (far target) and then on target 2 (near target). Focusing on target 1 induces a moderate pupillary constriction due to the PAR, as the eye shifts focus from the screen to a nearer point. The user is asked to maintain focus on target 1 until a subsequent audio cue instructs them to return gaze to target 0 (i.e., the screen, located beyond target 1). This change in focus induces a pupillary dilation, causing the pupil area to return approximately to the previously acquired baseline. The same procedure is then repeated for target 2, which is placed even closer to the user than target 1. Focusing on target 2 induces a stronger PAR, resulting in a more pronounced pupillary constriction. As before, the user maintains gaze on the target until prompted to refocus on the screen. After both constriction events have been acquired, the system proceeds to extract the constriction levels associated with each target. These levels are estimated as the mean pupil area over a small window (1 second) centered around the time in which the audio cue for target 0 is issued. At this point, the pupillary constriction is assumed to have been completed, and the pupil is about to begin re-dilating. This ensures that the computed level corresponds approximately to the minimum pupil area reached during the constriction task, provided that no significant user errors occurred. Mathematically, let n_i denote the sample index corresponding to the audio cue to look back at the screen, following fixation on target i, with

$i \in 1;2$. The level A_i for Target i is computed as the average over a symmetrical window of size $2k+1$ samples centered at n_i:

$$A_i = \frac{1}{2k+1} \sum_{j=-k}^{k} P[n_i + j] \tag{3.10}$$

where $P[n]$ is the pupil area signal at sample n, and k is the half-window size in samples. Once both levels A_1 and A_2 are computed, the system checks whether they are sufficiently separated to allow reliable classification. This is done by verifying the following condition:

$$|A_2 - A_1| > \alpha \cdot \sigma_{\text{baseline}} \tag{3.11}$$

where σ_{baseline} is the standard deviation of the baseline pupil area signal, and α is a parameter that determines the minimum acceptable separation between levels (equal to 0.5). If the condition is satisfied, the configuration is considered successful and the system switches to the operational phase. Otherwise, the entire configuration process, including baseline acquisition, is repeated. An example of pupil area signal during configuration phase is provided in the following figure (Figure 3.14).

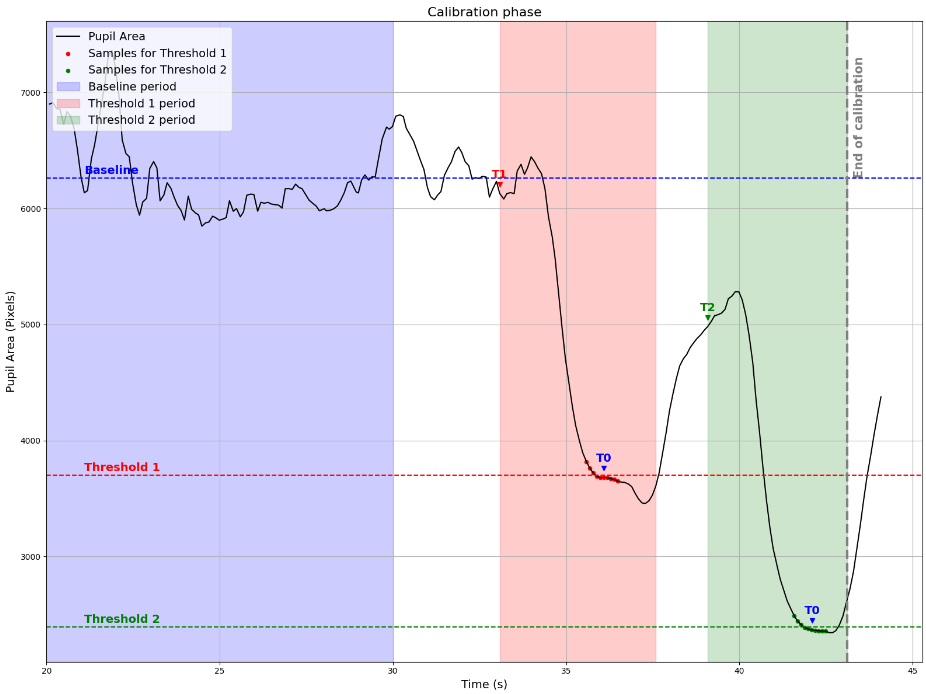


Figure 3.14: Final segment of the configuration phase. The horizontal lines represent the baseline (blue) and the levels target 1 (red), target (green). The shaded areas indicate the baseline acquisition zone (blue), and the level calculation zones.

Operational Phase

Once the configuration is successfully completed, levels A_1, A_2, and the baseline have been defined, the system enters the operational phase, during which it continuously monitors the pupillary signal in real time to detect pupillary constrictions and associate them with specific commands. Firstly, a constriction threshold is computed as:

$$Th_c = baseline - 0.5 \cdot (baseline - A_1) \tag{3.12}$$

this value defines the minimum drop in pupil area required to consider a constriction event and is defined as the middle area value between the baseline and the first level A_1.

All levels, the baseline, and consequently the threshold, are constantly updated to follow the trend of the pupil signal, by vertically shifting their values according to the average of the last 2.5 seconds, computed exclusively on those samples that lie above the constriction threshold. This strategy ensures that the system adapts to slow drifts or non-stationary behavior of the signal, while maintaining stability by updating only during non-constriction phases. Moreover, with a similar mechanism, levels and baseline values are updated when the pupil area falls below the second level (A_2) for at least 0.5 seconds. This mechanism ensures that more pronounced constrictions, resulting in smaller pupil areas than those estimated during the decoder configuration phase, do not interfere with the system's logic, preventing situations where the area fails to rise back above the constriction threshold and potentially halts the overall functioning of the process.

The system continuously checks whether the most recent pupil area samples fall below the constriction threshold. Once a drop is detected the system computes the first positive zero-crossing in the derivative of the pupil area (i.e., where the derivative changes from negative to positive), marking the point at which the pupil begins to re-dilate or remains approximately constant, this is considered the end of the pupillary constriction and is used to estimate the amplitude of the PAR event. To do so, the system selects a small window of 0.5 seconds starting from the identified end of the constriction and computes the average pupil area within this window. This value is then compared to the pupil area at the end point itself, and the lower of the two is taken as the minimum value reached during the constriction as shown in the following equation:

$$P_{min} = min(P[end], \mu_{window}) \tag{3.13}$$

where $P[end]$ is the area value corresponding to the end of constriction, and μ_{window} is the mean area value across the window. This dual approach is necessary due to the typical morphology of the PAR, which is characterized by a rapid and pronounced decrease in pupil area, followed by a stabilization phase where the signal oscillates around a minimum level. Simply taking the value at the point of derivative inversion does not always guarantee that the true minimum has been reached, as the pupil area may continue to decrease slightly after that point. Conversely, relying solely on the average value computed in the window may also be misleading, especially in cases where the constriction includes brief, sharp minima followed by high-amplitude oscillations. In such scenarios, the mean tends to overestimate the true minimum. By considering both the instantaneous value and the short-term average, and conservatively selecting the lower of the two, the system achieves a more robust and accurate estimation of the minimum pupil area reached during constrictions. This, in turn, improves the reliability of the event classification process. The estimated minimum $P_{\min}$ is then compared with the dynamic amplitude levels A_1 and A_2. The classification rules are:

$$A_1 \leq P_{\min} < Th_c \implies E = 1$$

$$P_{\min} < A_1 \implies E = \begin{cases} 1, & \text{if } |P_{\min} - A_1| < |P_{\min} - A_2| \\ 2, & \text{otherwise} \end{cases}$$

where E denote the classification type of the event. Once an event is classified, it is stored along with its time, and type. The following figure (Figure 3.15) shows an example of a pupil area signal with the zones relating to the classification of constrictions highlighted in different colors. The area highlighted in blue represents the dilation zone, where no constriction is detected, but the values of baseline and thresholds are dynamically updated. The red and green areas correspond respectively to the regions where events of type 1 and type 2 are detected, respectively. Additionally, vertical lines represent the time when constrictions are detected by the algorithm.

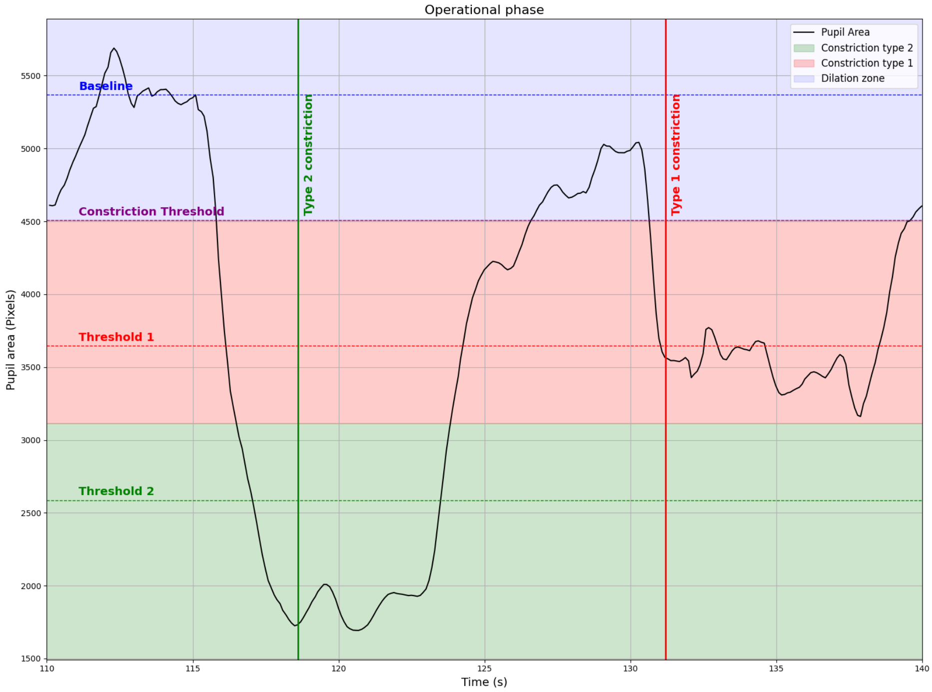


Figure 3.15: Segment of the signal during the operational phase. Vertical lines indicate detected events: type 1 (red) and type 2 (green). The horizontal bands mark the regions where constrictions are classified (red and green), and the dilation zone (blue).

In addition, the events are appended to a pending event queue, used to track whether the user returns to look at the screen after each constriction. This pending event queue is essential for determining the duration of a complete constriction event, which is defined as the time interval from the start of the constriction util the moment when the pupil area returns and remains consistently above the constriction threshold for at least 1 second. This condition ensures robustness against transient fluctuations or noise, confirming a true return to the baseline dilation state. This definition captures both the sustained phase of the constriction (the time during which the pupil remains constricted) and the recovery phase (the time it takes to return to baseline). Since type 2 constrictions are associated with a stronger PAR, resulting in a more pronounced decrease in area value, the return to the constriction threshold tends to be slower. This leads to a systematic overestimation of the type 2 event duration. To compensate for this effect, a correction factor of 0.25 seconds is subtracted from the measured duration, only for type 2 constrictions. The measured duration is then stored and associated with the corresponding event type E, allowing the system to differentiate between short and long constrictions for each of the two types. This temporal encoding significantly expands the communication bandwidth, enabling more expressive control. It is important to note that, since the system waits for the pupil area to return above the constriction threshold before measuring the event duration, the actual duration of the event is registered with a delay compared to the classification of its type, which occurs shortly after the derivative inversion. This temporal separation ensures that the system remains responsive, immediately classifying the event type as soon as a constriction is detected. As a result, short constrictions trigger commands almost instantaneously. In contrast, longer constrictions are recognized at a later stage, once the pupillary area recovery has been observed. In these cases, any previously triggered action related to the short command is cancelled and replaced with the long command associated with the same constriction type. This design does not affect usability, since executing a prolonged constriction requires the user to look at one of the targets, momentarily shifting their gaze away from the main screen, where visual feedback and GUI actions occur. Consequently, the early short command is typically not even perceived by the user, who is focused on maintaining the gesture to elicit the intended long command. The cancellation thus happens transparently, without introducing perceptible inconsistencies in user interaction.

3.2 GUI and applications

This section presents the graphical user interfaces developed for the configuration and operation of the device, along with the implemented applications. The interface plays a key role in the interaction between the user (or caregiver) and the system, offering a seamless way to configure Wi-Fi connectivity and manage the various operational phases. In addition to describing the visual layout and functional components of each interface, this section also details the set of commands used to control the GUI and govern the behavior of the applications, ensuring intuitive and efficient user interaction.

3.2.1 USB vs WIFI connection

The device can be connected to a PC via two modes: USB or Wi-Fi. The USB connection is intended solely for network configuration and enables subsequent wireless connectivity. In contrast, the Wi-Fi mode, accessible only after successful configuration, is used for the device's primary assistive interface. Both modalities are described in the following subsections.

USB mode

When the device is connected to a computer via USB, it is automatically recognized as a USB mass storage device. Upon detection, the host computer opens a graphical interface for network configuration (Figure 3.16).

Figure 3.16: GUI for network configuration.

This GUI allows the caregiver to input the necessary Wi-Fi credentials required for the device to connect to the local network. The interface includes the following features:

- **Credential entry**: the user can manually insert the SSID and password of the desired Wi-Fi network.
- **Auto-detection of current network**: as a convenience, the program can automatically retrieve the SSID of the Wi-Fi network currently in use by the host computer, thereby reducing the configuration time and the risk of input errors.
- **IP address retrieval**: in addition to network credentials, the program automatically detects the IP address of the host computer. This information is crucial to enable communication between the device and the PC once the device connects to the network
- **Password visibility toggle**: for security, the GUI allows the user to choose whether to display the password in plain text or hide it for privacy.

Once the credentials are entered and the user clicks the "*Salva credenziali*" button, the information (SSID, password, and IP address) are stored in a plain-text configuration file located in a dedicated partition of the device's microSD card. This partition is accessible both via the USB interface and by physically removing the microSD card and reading it from any standard card reader. This design ensures maximum compatibility and flexibility, allowing configuration even in the absence of the GUI application. It is important to note that the configuration interface is operated entirely via mouse and keyboard inputs. This design reflects the assumption that configuration is typically carried out by a caregiver or operator, as opposed to the operational phase, which relies on the patient's control through the PAR mechanism.

On Windows and Linux platforms, all functions are fully supported. On macOS, however, due to system limitations and lack of certain API access, automatic network detection and IP acquisition are currently not implemented. Nevertheless, users can still manually modify the configuration file by accessing the memory partition, as described earlier.

The Wi-Fi configuration procedure must be performed once during the initial setup or whenever the device needs to connect to a different wireless network. It can also serve as a troubleshooting mechanism in case the router's DNS dynamically changes the device's IP address. However, given that the target environment is typically a domestic setting, where the router and IP configuration are relatively stable, such changes are expected to be infrequent. Nonetheless, the flexibility of the system allows users to reconfigure the device with minimal effort. Once the configuration file has been successfully written, the device stores the credentials for future use. On the next boot, the system automatically initiates the connection protocol described in Section 3.1.4. this is implemented using the `wpa_supplicant` service on the Raspberry Pi, ensuring connection to the specified network. Upon successful connection, the system starts the operational phase, detailed in the following chapter, during which the user interacts with the main interface through pupil-based commands.

Wi-Fi mode

Once the device successfully establishes a connection, a user interface is displayed showing the live video stream acquired by the camera. The real-time pupil segmentation is overlaid on the video feed, allowing the operator to verify the correct positioning of the camera and ensure that the eye is properly framed. The same interface also includes an animation that visually indicates the operational status of the IR LEDs. When functioning correctly, an icon is briefly displayed to confirm proper operation, then disappears automatically after a short time. If a malfunction is detected, a warning icon is shown and remains visible until the issue is resolved.

The indicator will automatically disappear following a system reboot, during which a new functional check is performed. Alternatively, the caregiver can manually disable the warning via a mouse input once the issue has been addressed. In the top-left corner of the interface, a four-level indicator displays the quality of the wireless connection reflecting the number of frames received by the computer within a given time window. As such, it serves as a general performance index, providing insight into the overall speed of image acquisition and transmission. If the frame rate drops below a predefined threshold (8 frames per second), a warning message is shown. It also presents a toggle switch placed on the upper right side of the window allowing the caregiver to manually check the video stream. At the bottom of the interface, a real-time graph displays the segmented pupil area signal. The x-axis represents the sample index, while the y-axis indicates the segmented pupil area in pixels. An image of this interface is represented in the figure below (Figure 3.17):

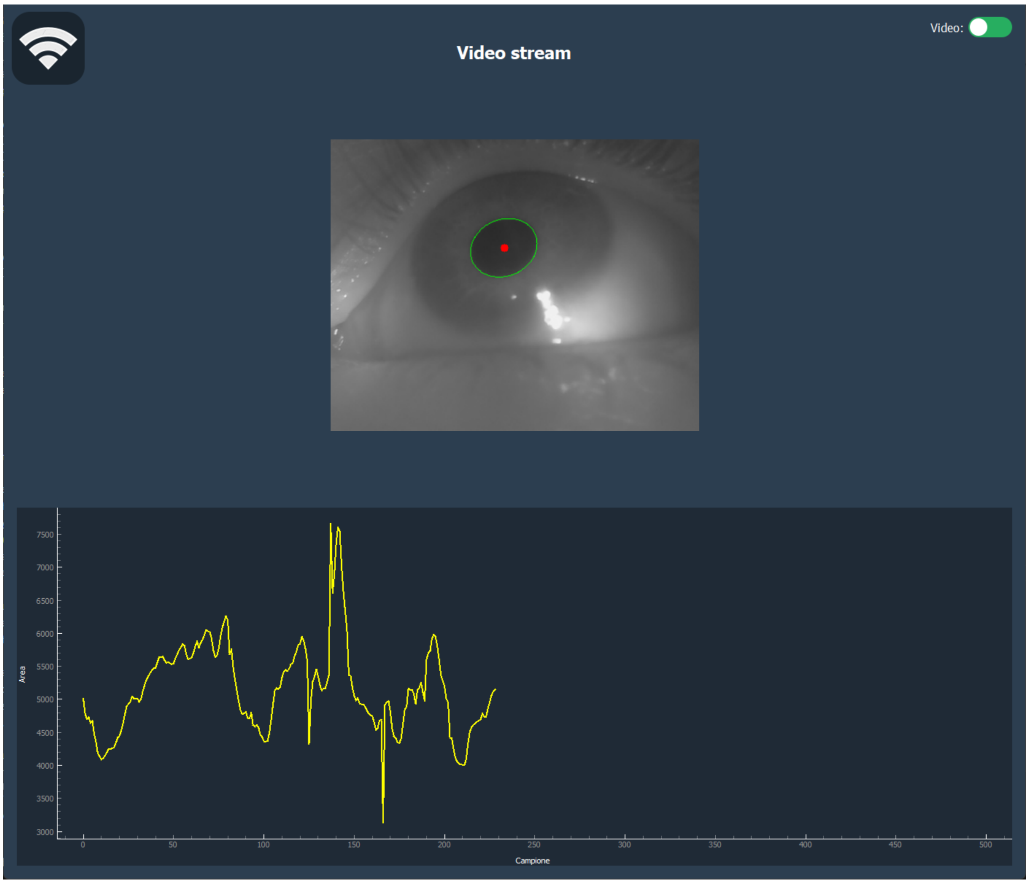


Figure 3.17: GUI during initial configuration.

During this stage, the system prompts the user via audio commands to perform a configuration procedure, as described in Section 3.1.6. Once configuration is completed, the system displays the main menu screen automatically (Figure 3.18). This menu is graphically represented as a circular wheel, with each segment corresponding to a selectable option. The GUI offers access to various applications, including a virtual keyboard for communication ("SPEAKER"), and an exit button. The interface is entirely controlled through the voluntary modulation of the PAR. Specifically, type 1 command is used to scroll through the options (highlighted with a darker color), while type 2 command is used to select the highlighted item. In this primary interface, only signal levels, which are more reliable than duration-based interpretations, are utilized, as only two distinct commands are required. This structure forms the basis for the interaction system described in the following subsection, which details the available application.

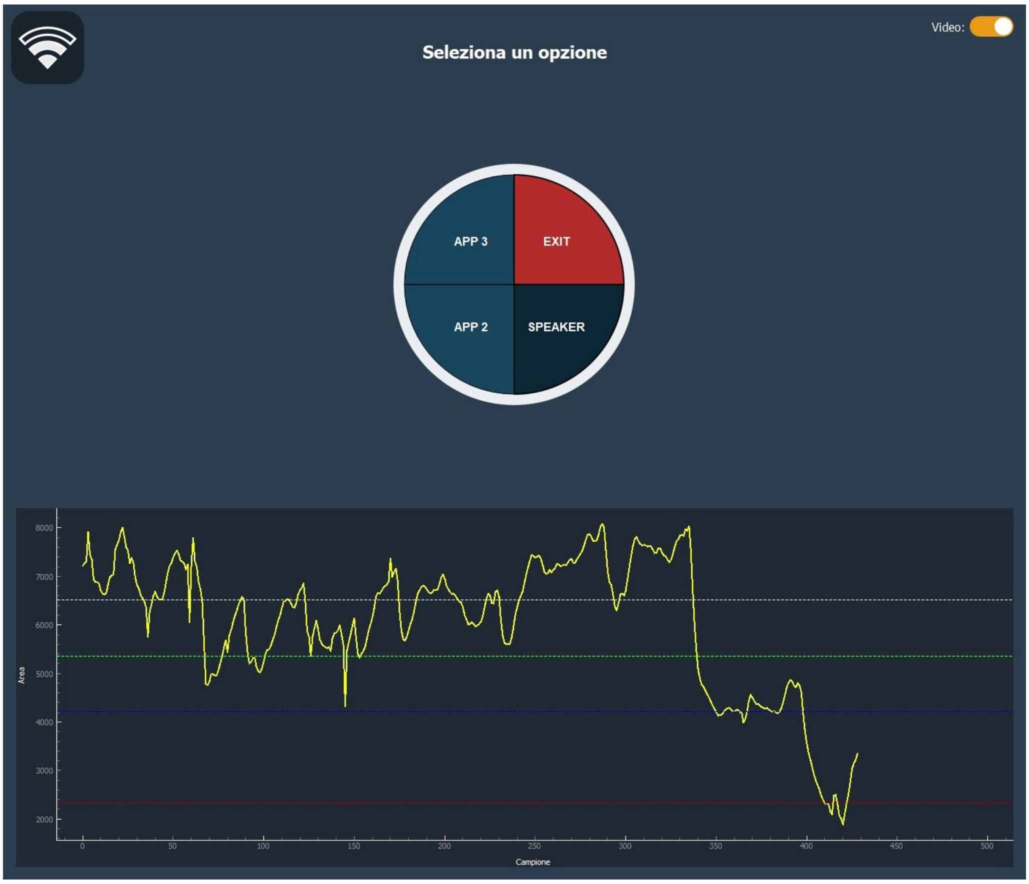


Figure 3.18: Main menu GUI.

3.2.2 Possible applications: "SPEAKER"

In the context of this work, only one application was developed: a communication aid named "SPEAKER", which appears as a selectable segment on the menu wheel. The user interface of the SPEAKER application is shown in Figure 3.19.

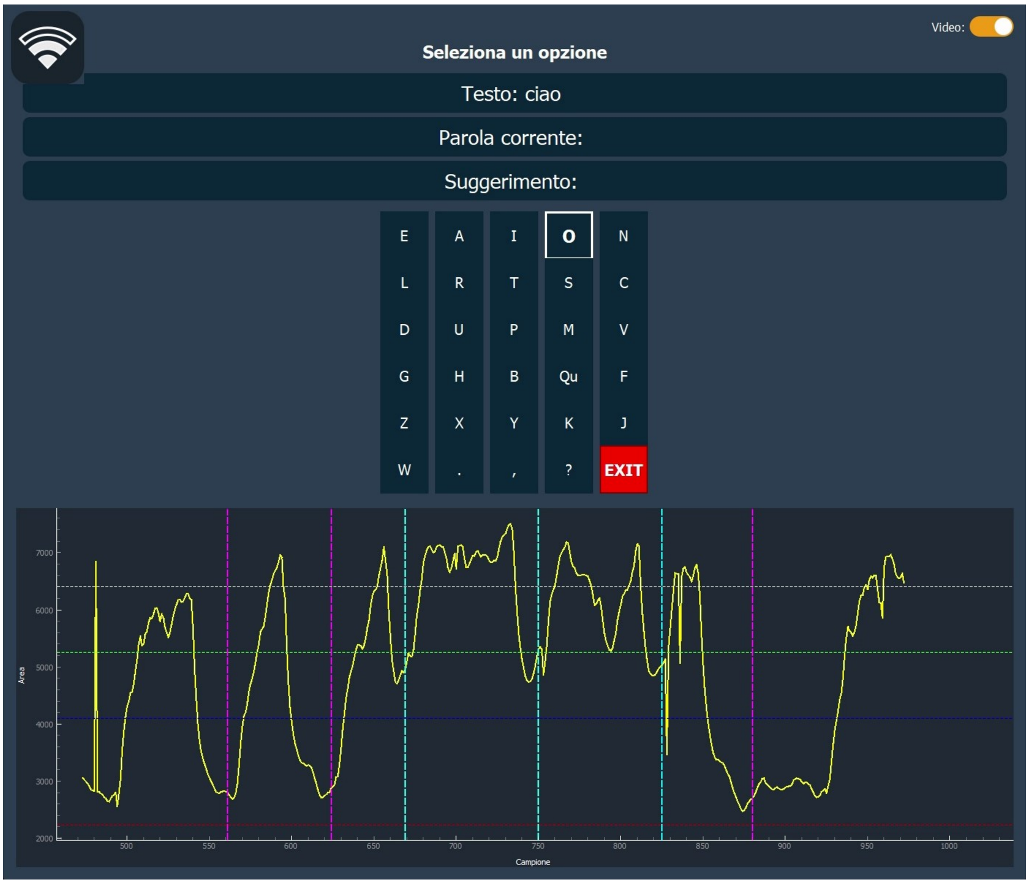


Figure 3.19: SPEAKER application GUI: vertical lines represents detected events (cyan for type 1 events and magenta for type 2).

The on-screen keyboard is designed to include the most frequently used letters in the Italian language, arranged in a layout that optimizes the speed of text entry. The selection mechanism follows a scanning-based approach: columns are automatically highlighted in sequence with a fixed interval of 1 second, while a short type 2 input is used to move to row selection. The desired letter can then be selected using a short type 1 input. A long type 1 input functions as backspace, deleting the most recently entered character, while a long type 2 command is used to confirm a suggested word or correction offered by the system. This command mapping was informed by the experimental results on PAR detection accuracy discussed later in the Results section. More frequent actions were assigned to the PAR events that the algorithm recognized with higher reliability, while less frequent or secondary tasks were deliberately associated with inputs that showed lower detection accuracy. Once the typed word is confirmed, the system vocalizes it using a text-to-speech synthesizer through an audio output. To provide word suggestions and spelling corrections in real time, the application integrates an Italian dictionary and employs the `difflib.get_close_matches()` function from Python's standard library. This function compares the string being typed with words in the dictionary and returns the closest match based on a similarity score. Specifically, the parameter `cutoff=0.75` ensures that only candidates with a similarity ratio of at least 75% are considered. If no sufficiently close match is found, the function returns an empty string, meaning no suggestion is shown. At the bottom of the keyboard interface, an exit button is available. When selected, it closes the SPEAKER application and returns the user to the main menu. While only the communication application was developed in this work, the same interaction model can be extended to a wide variety of use cases. The system architecture supports up to four distinct commands, and by combining these with scanning mechanisms, such as the automatic highlight cycling demonstrated in the SPEAKER interface, it becomes possible to implement a broad range of accessible applications.

3.3 Experimental verification

In order to validate the functionality and reliability of the proposed device, a series of experimental tests were conducted. This section presents the methodologies adopted to assess the system's performance across multiple parameters, both at the hardware and software levels. Specifically, the evaluation focused on battery life and processing performances. Importantly, the effectiveness of the pupil segmentation algorithm and the system's ability to detect PAR events were examined. The following subsections describe the testing procedures, tools, and criteria used to verify each of these aspects, while the outcome of these verification are presented in the Results chapter.

3.3.1 Evaluation of device performances

These tests aimed to characterize the system in terms of energy efficiency and computational performance. All measurements were conducted using the final hardware configuration.

Power profile assessment

The power profile of the system was measured using a USB digital multimeter, which provides real-time readings of voltage, current, accumulated charge, and elapsed operating time. The multimeter was placed between the output of the *TP4056* battery management module and the *Raspberry Pi zero*, thus enabling direct monitoring of the current drawn by the device during operation. Two operational states were considered:

- **Active mode**: the system is fully functional, performing continuous image acquisition, cropping operation, and Wi-Fi transmission.

- **Standby mode**: the system is powered on and idle, sending UDP packages and waiting for a PC response to initiate the connection.

For both operating conditions, a complete discharge cycle was monitored to estimate the average current consumption and to evaluate the battery life per full charge. Additionally, a complete charging cycle was performed to evaluate the charging behavior and to estimate the time required for a full recharge. The charging process follows a typical constant-current/constant-voltage (CC/CV) profile: during the initial phase, the battery charges at a nearly constant current, and as the battery approaches full capacity, the charging current gradually decreases while the voltage remains constant. This behavior results in a significant increase in the time required to complete the final portion of the charge, with only a marginal gain in stored energy. To balance charging time and usable capacity, a trade-off point was identified from the battery's charging curve, allowing early termination of charging while still retaining most of the total capacity. Since the digital multimeter used in the experiments does not provide any digital output, displaying measurements exclusively on a seven-segment screen, an alternative method was adopted to reconstruct the charging profile. A smartphone was securely positioned to continuously record the multimeter display during the charging process. Subsequently, a custom algorithm was employed to extract numerical values from the recorded video by recognizing digits on the seven-segment display. These values were then associated with their corresponding timestamps to reconstruct the charging current over time. The trade-off point was identified as the earliest time at which the derivative of the charging efficiency curve consistently falls below a fixed negative threshold ($-0.005\ mAh/min^2$), indicating the onset of a significant decrease in charge efficiency.

Computational load analysis and FPS measurement

The performance of the system was assessed in terms of computational load and frame rate, with the goal of identifying potential bottlenecks and evaluating the performances in real-time operation. The computational load analysis was divided into two main components: the tasks executed on the *Raspberry Pi zero* and the segmentation pipeline running on the host PC. Both analyses processed the same 5 recordings, each lasting 1 minute, using a similar profiling methodology based on internal timestamp logging. On the *Raspberry Pi zero*, the average processing time (and standard deviation) was measured for each of the following operations: image acquisition from the camera, cropping, JPEG compression of the cropped image, and transmission via TCP socket. On the host PC, each received frame was processed through a multi-step pupil segmentation pipeline (detailed in Section 3.1.5). The execution time of each stage was logged to compute the average duration and standard deviation across the 5 recordings, allowing identification of the most time-consuming steps.

To assess the real-time performance of the system, an analysis was conducted to evaluate the effective FPS during normal operation. Specifically, the frame rate was calculated as the inverse of the time interval between the arrival of two consecutive frames on the host PC. This measurement inherently includes the entire processing chain, comprising pupil segmentation, PAR detection, potential command handling, and graphical updates. Given the variability of these operations, along with potential fluctuations in Wi-Fi transmission speed, the resulting frame rate was not constant but followed a distribution. To characterize this behavior, the mean and standard deviation of the frame rate were computed over a dataset of 20,000 frames. This analysis provides an estimate of the actual responsiveness of the system under typical usage conditions.

3.3.2 Evaluation of pupil segmentation performance

To evaluate the segmentation performance of the proposed algorithm, it was first attempted to use a publicly available dataset of eye images. However, preliminary tests showed that the algorithm performed poorly on this dataset, likely due to substantial differences in image resolution, camera perspective, and lighting conditions compared to the images acquired by the device developed in this work. On the other hand, the segmentation algorithm yielded satisfactory results when applied to frames directly acquired by the device itself.

Given the unsuitability of existing annotated datasets, a custom dataset was constructed. Twenty subjects were recorded in different conditions using the developed system, and a subset of frames was manually annotated by marking the visible pupil region. To expand the evaluation beyond the manually labeled subset, the Segment Anything Model v2 (SAM2), a recent artificial intelligence model for general-purpose image segmentation, was employed. This model was used to semi-automatically segment the pupil across all recorded frames, providing a broader ground truth for comparison. The segmentation quality was assessed using two standard metrics: the Dice coefficient ($Dice$) and the Intersection over Union (IoU), both widely used to quantify similarity between predicted and reference masks. Additionally, the number of missed frames, defined as frames where the tested algorithm failed to segment the pupil while SAM2 successfully did, was computed.

Testing Procedure

The experimental setup was specifically designed to evaluate segmentation performance under three different conditions:

- Static gaze.
- Pupillary constriction.
- Eye movements.

In the first phase (static gaze), the subject was instructed to fixate on a point displayed on the monitor. This point had been previously aligned with a physical target positioned halfway between the subject and the screen. The alignment was performed by asking the subject to close the right eye (not recorded) and to adjust their head position as the near target and the point on the screen appeared superimposed. This alignment procedure was adopted to minimize the vergence reflex of the measured eye, ensuring greater stability of the pupil position during constriction task. During the second phase (pupillary constriction), the subject received audio instructions to shift their focus from the screen to the near target level and back, to simulate the PAR and evaluate pupil segmentation performances under these conditions. In the third phase (eye movements), the subject was asked to move their eyes by fixating on the four corners of the screen in succession. Although this phase is not relevant for the final application, given that patients in CLIS cannot perform voluntary eye movements, it was included to evaluate the limitations of the segmentation algorithm when the pupil appears off-center or blurred due to movements. Additionally, off-axis pupil results in more elliptical shapes, which typically degrade the performance of segmentation algorithms optimized for near-circular, centered pupils.

Primary manual ground truth

To establish a reliable ground truth for the evaluation of the segmentation algorithm, a subset of frames was manually annotated. Specifically, 15 frames per subject were randomly selected, extracting 5 frames from each of the three phases. Each selected frame was magnified by a factor of 5 to improve annotation accuracy. The pupil contour was then manually traced using a graphics tablet and stylus minimizing noise using a Gaussian smoothing filter. This subset of annotated frames constitutes the primary ground truth dataset used to evaluate segmentation performance with high precision. The overall workflow used for primary ground truth creation is viewable in the following figure (Figure 3.20).

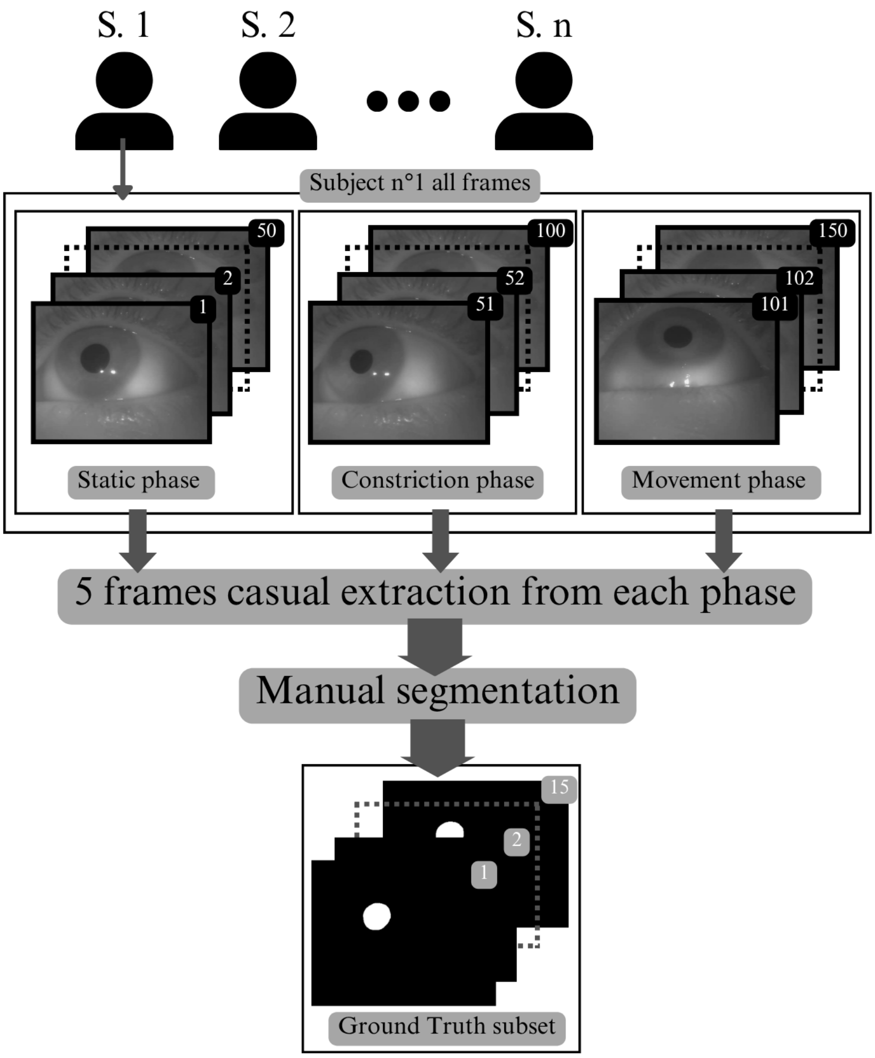


Figure 3.20: Primary ground truth creation workflow.

Secondary deep learning-based ground truth

To enable a broader and more comprehensive evaluation across the entire set of frames, an extended ground truth dataset was generated using SAM2, a deep learning model that has been previously adopted in literature for pupil segmentation [69]. The SAM2 model produces more accurate results when initialized with a point located inside the target object. To automate this process, the center of the ROI identified by the proposed algorithm was used as the input point for initialization. Prior to this, the reliability of the ROI localization was validated by checking whether the centroids of the manually annotated pupil areas consistently fell within the corresponding automatically detected ROIs. As this condition was always satisfied, the center of the ROI was considered a robust initialization point. Nevertheless, if the custom segmentation algorithm developed for this device failed to detect any pupil structure, it typically indicates a challenging frame, such as one where the eye is closed, the pupil obstructed, or the image is blurred. In such cases, manual intervention is requested: the user is prompted to either specify a new initialization point or confirm that the pupil is not visible, in which case an empty mask is assigned. For the construction of the dataset, empty masks were only assigned to frames in which the eye was fully closed. In all other scenarios segmentation was still attempted using the SAM2 model initialized from user-defined point. Subsequent quality checks were applied to discard incorrect segmentations. Manual point selection was also used when the segmented area showed an abrupt variation exceeding 50% between consecutive frames. This approach mitigates errors caused by reflections: for instance, the ROI center may fall on a light reflection caused by the IR LEDs, leading to the segmentation of a small circular shape that closely resembles the pupil in geometry but not in area (Figure 3.21).

(a) (b)

Figure 3.21: Example of incorrect segmentation by SAM2: (a) original frame with the ROI (red) and its center (green); (b) resulting segmentation mask.

It is crucial to detect such errors during the model inference stage, as subsequent post-processing operations, based on morphological and geometric criteria, are not capable of reliably identifying this specific type of artifact. Failing to address these cases at this stage would result in the loss of valid segmentation masks; by intervening during inference, it is possible to reposition the initialization point and obtain an accurate segmentation. The workflow for creating the raw secondary ground truth dataset is represented in the figure below (Figure 3.22).

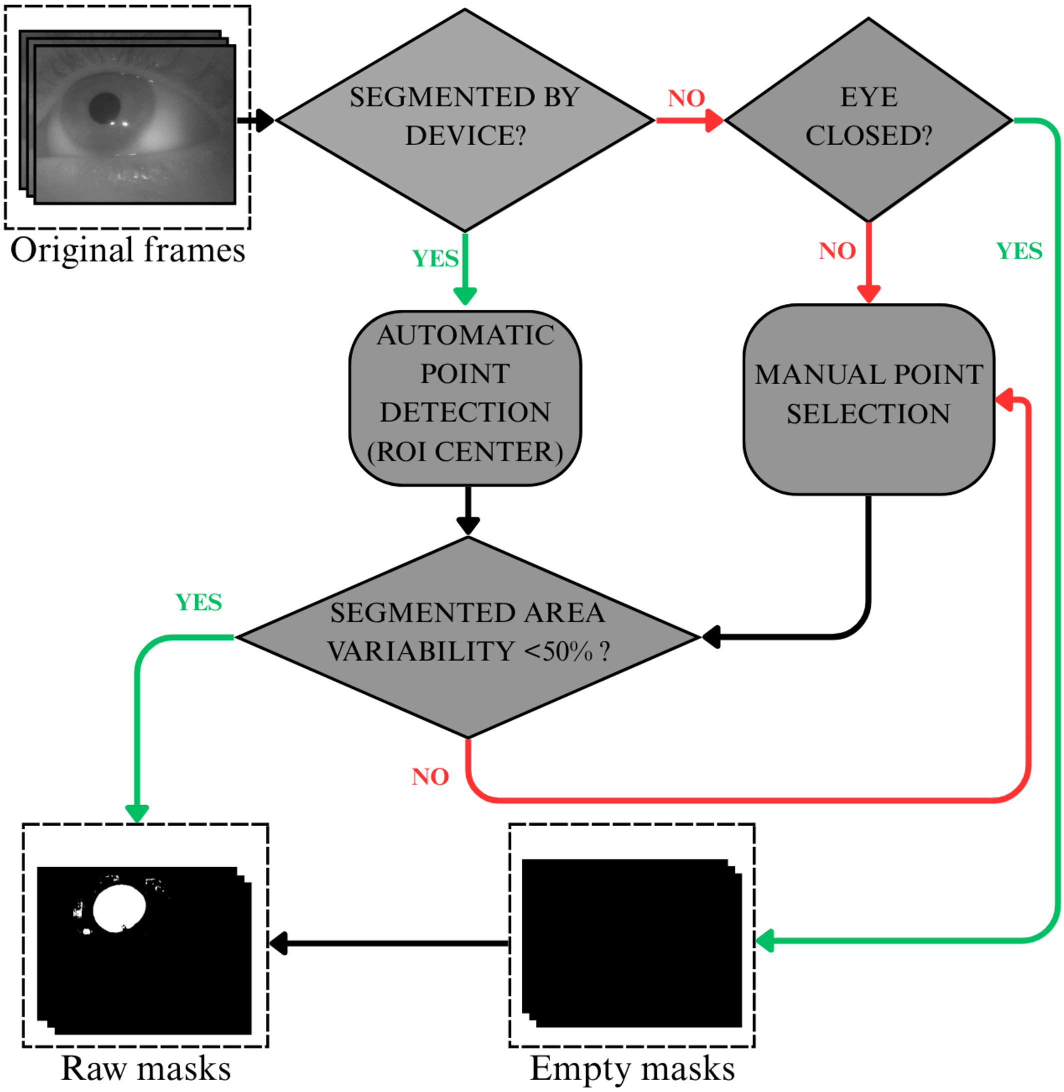


Figure 3.22: Secondary ground truth (SAM2) creation workflow.

Once segmentation is completed, all masks undergo post-processing using a *morphological opening* operation with a circular kernel of 75-pixel diameter to remove small artifacts and refine the contours. Subsequently, a semi-automated quality control step is applied based on geometric features of the segmented shapes. Specifically, *Circularity* and *Solidity* are computed for each mask to assess its plausibility. Circularity measures how closely the shape of an object resembles a perfect circle, and is defined as:

$$Circularity = \frac{4\pi \cdot Area}{Perimeter^2} \tag{3.14}$$

Values closer to 1 indicate a more circular shape, while lower values suggest elongation or irregular contours.

$$Solidity = \frac{Area}{Convexarea} \tag{3.15}$$

A *Solidity* close to 1 indicates that the shape is mostly convex with few concavities. Applying both metrics in combination is necessary because they capture complementary aspects of shape quality. While *Circularity* is sensitive to perimeter irregularities and elongated shapes, *Solidity* effectively detects internal concavities and disconnected regions. Using only one metric could lead to false positives or negatives; for example, a fragmented but roughly circular shape might pass a *Circularity* check but fail the *Solidity* criterion. These metrics are evaluated across all masks for each subject, and those with values lying beyond two standard deviations from the mean are flagged as outliers (examples are viewable in Figure 3.23.). These are then visually presented, superimposed with the original frame, to manually decide whether to accept or reject them. If rejected, the corresponding mask is replaced with an empty one, indicating that SAM2 pupil detection process was not successful.

(a) (b)

Figure 3.23: Examples of outliers discarded from dataset: (a) *Circularity* outlier; (b) *Solidity* outlier.

Segmentation accuracy metrics

To quantitatively assess the performance of the proposed pupil segmentation algorithm, three main metrics were employed: the Dice Similarity Coefficient, the Intersection over Union, and the percentage of missed frames. These metrics were computed by comparing the binary masks generated by the algorithm with the corresponding ground truth masks of both datasets (primary and secondary) for each phase of the test (static, constriction and movement). The Dice coefficient ($Dice$) measures the overlap between the predicted segmentation mask S and the ground truth mask GT, and is defined as:

$$Dice(S, GT) = \frac{2|S \cap GT|}{|S| + |GT|} \tag{3.16}$$

This metric ranges from 0 (no overlap) to 1 (perfect overlap). It is particularly useful in medical image analysis where class imbalance is common, as it places more emphasis on correctly identifying the relevant region. The Intersection over Union (IoU), also known as the Jaccard index, is another widely used metric for segmentation tasks and is defined as:

$$IoU(S, GT) = \frac{|S \cap GT|}{|S \cup GT} \tag{3.17}$$

Like the Dice coefficient, the IoU ranges from 0 to 1. While both metrics reflect the degree of overlap between the predicted and ground truth masks, IoU tends to penalize over- and under-segmentation more heavily, making it a stricter measure. Although accuracy is commonly used in classification tasks, it is not suitable for evaluating segmentation performance in this context. This is due to the extreme class imbalance inherent in pupil segmentation: the background occupies the vast majority of the image, while the segmented pixels make up only a small fraction.

In addition to $Dice$ and IoU, the percentage of missed frames was used to evaluate the robustness of the algorithm across the different experimental conditions. A frame was considered "lost" if the segmentation algorithm did not output any ellipse, meaning no pupil contour was detected or the detected ones are discarded due to controls. This metric was computed as follows:

$$Lostframes(\%) = \frac{N_{lost}}{N_{total}} \cdot 100 \tag{3.18}$$

where N_{lost} is the number of frames without a valid segmentation output, and N_{total} is the total number of frames in the corresponding test phase (excluding frames with empty ground truth mask). This additional metric helps capture failure cases, providing a more complete assessment of the algorithm's reliability in real-world usage.

3.3.3 Evaluation of PAR event detection performance

To validate the reliability of the developed system in detecting PAR events under realistic conditions, a dedicated experimental protocol was designed. The goal was to simulate typical operational scenarios while maintaining experimental control, allowing direct comparison between algorithm-detected events and ground truth.

Experimental setup

The experimental evaluation involved 15 healthy adult volunteers and lasted 3 minutes. A custom-built support system was employed to ensure stable positioning of the subject. The structure included a chin rest to keep the head still and at a fixed distance from both the screen and the physical visual targets. This setup allowed for consistent eye positioning and minimized involuntary head movements. The targets, positioned at different distances, are two transparent panels each presenting a visible point. The setup was designed to be adjustable, to ensure that the targets fell at appropriate distances to elicit measurable and distinct PAR events. The entire structure was secured to a table using a mechanical clamp, with a computer screen positioned behind the most distant target. A schematic representation of the structure is depicted in Figure 3.24.

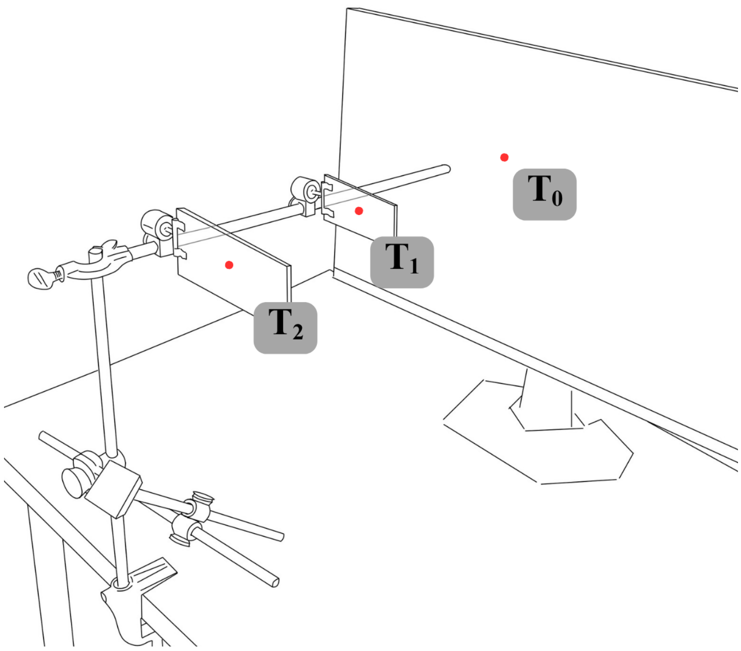
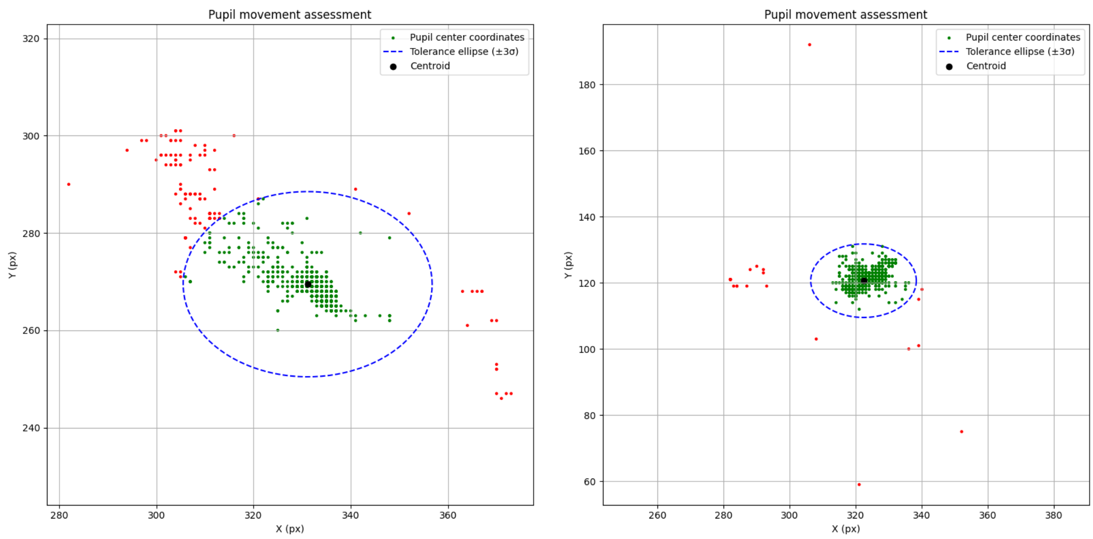


Figure 3.24: Experimental setup with targets where T0 is the screen, T1 the far target and T2 the near target.

Prior to the recording, the subject was asked to close the right eye (which was not being monitored) and align both targets so that they appeared colinear with a highlighted fixation point displayed on the screen. This alignment procedure ensured that only the right eye performed the vergence movement, leaving the left eye (which was recorded) virtually motionless, effectively replicating the ocular immobility expected in CLIS patients. To verify this condition, the stability of the left eye was quantitatively assessed by tracking the center coordinates of the ellipse fitted during pupil segmentation. Minimal variation in these coordinates over time confirmed the absence of significant eye movement. Specifically, during the baseline acquisition phase, in which the eye was maintained in a steady position, the centroid of the pupil center was calculated, along with the standard deviations on the horizontal and vertical direction. Based on these values, a tolerance ellipse was constructed using a threshold of ±3 standard deviations from the mean. In the subsequent operational phase, the pupil center displacement was monitored, and the number of frames in which the pupil center fell outside the tolerance ellipse was recorded. If this number exceeded 10% of the total frames in the operational phase, the measurement was considered invalid and discarded. A visual representation of this process is represented in the figures below (Figure 3.25).



(a) (b)

Figure 3.25: Example of pupil movement assessment. The blue line represents the tolerance ellipse; green dots indicate valid pupil positions, while red dots denote invalid ones. (a) Distribution of invalid pupil positions; (b) distribution of valid pupil positions.

Subsequently, the subject proceeded through the standard configuration phase, followed by the operational test phase. An audio guidance system was integrated to provide vocal instructions, specifying:

- **The target type**: (0 = screen, 1 = far target, 2 = near target).
- **The duration**: "long" or "short".

The participants were instructed to maintain sustained accommodation when prompted with a long-duration stimulus for at least 5 seconds. Target commands were delivered in a structured sequence, alternating between a stimulus (target 1 or 2) and the baseline (target 0), in order to ensure return to resting pupil diameter before issuing the next command. This mimicked the dynamics of actual system use and allowed multiple independent PAR events to be captured. The sequence of target types and durations was pseudo-randomized, with probabilities progressively adjusted based on prior data to ensure a balanced dataset across all classes. During the test, the system logged all automatically detected constriction events, including: the predicted event type (1 or 2), the predicted duration (short or long) and the timestamp of detection. These logged events were subsequently compared to the ground truth audio commands, which were generated by the PC and timestamped during the experiment. In the figure below (Figure 3.26) is viewable a portion of one of the signals recorded with the timestamp of both ground truth and detected events along with the calculated duration.

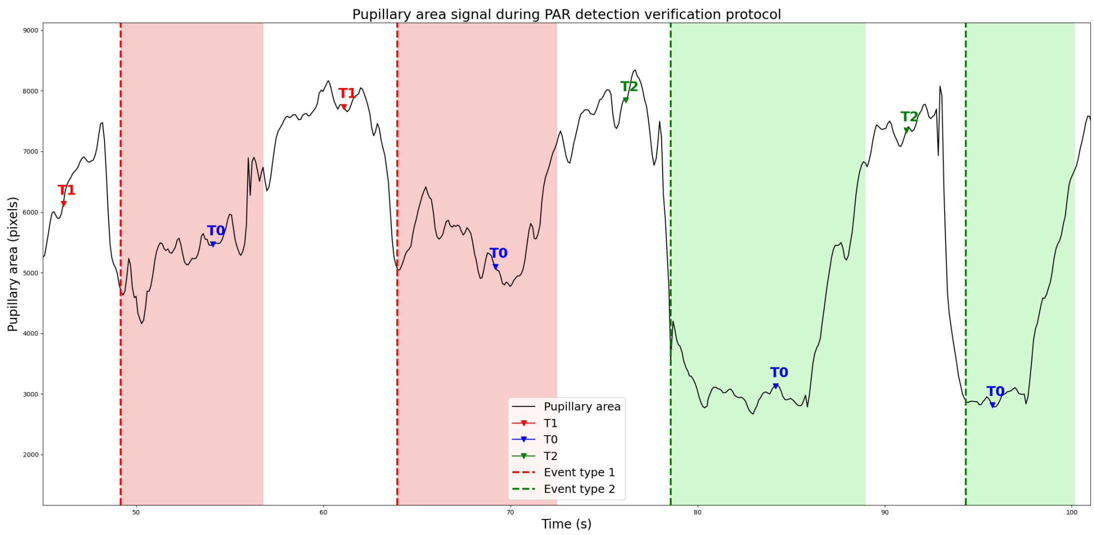


Figure 3.26: Portion of a signal: ground truth sound timestamps labeled as T0, T1, and T2; detected event times indicated by vertical lines, type 1 (red) and type 2 (green); and the event durations highlighted by shaded areas.

Performance metrics

To quantitatively evaluate the effectiveness of the proposed system in detecting PAR events, a set of standard performance metrics was computed. These metrics compare the algorithm's output against a predefined ground truth, derived from the sequence of audio commands that indicate both the timing and type of stimulus presented to the participant. An event detected by the system was classified as a *True Positive (TP)* if it matched the correct stimulus class and occurred within a predefined temporal window starting from the corresponding ground truth event and ending to the next call to 0 target (screen). Conversely, detections that occurred outside the valid time window, belonged to the incorrect class, or occurred in the absence of any stimulus were labeled as *False Positives (FP)*. Ground truth events that were not detected by the system were counted as *False Negatives (FN)*. In this context, the absence of a stimulus does not require explicit detection; hence, *True Negatives (TN)* are not well-defined. For this reason, binary accuracy (event vs not event) was excluded from calculations and supplemented with more informative metrics, including $Precision$, $Recall$, and $F1 - score$. In particular, $Precision$ quantifies the proportion of correctly detected events among all events reported by the system:

$$Precision = \frac{TP}{TP + FP} \tag{3.19}$$

A high $Precision$ score indicates a low rate of false detections, which is particularly important in assistive communication contexts where erroneous detections could lead to unintended commands.

$Recall$ measures the proportion of actual events that were correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{3.20}$$

High $recall$ ensures that most real PAR events are successfully captured, which is critical for maintaining system responsiveness.

$F1-score$, the harmonic mean of precision and recall, offers a balanced indicator of detection performance:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.21}$$

the score ranges from 0 to 1, with higher values reflecting better overall performance.

To provide a comprehensive assessment of classification performance, three distinct confusion matrices were constructed:

1. **Type classification matrix**: evaluates the system's ability to distinguish between stimulus types (i.e. far vs. near targets).
2. **Duration classification matrix**: assesses the classification between short and long stimulus durations.
3. **Joint type-duration classification matrix**: a four-class matrix that jointly considers both stimulus type and duration.

These matrices allowed for the quantification of the system's performance in classifying PAR, using accuracy as the evaluation metric.

In addition to classification performance, the events detection latency was also analyzed. Latency analysis is particularly important since interfaces like the communicator described in the possible applications section relies on timing, so the responsiveness of the system must be adequate. During the recordings the system saves the detection timestamp for each event. This timestamp was compared to the corresponding stimulus onset time, as defined by the audio command, to compute the raw latency for each detected event:

$$L_{RAW,i,j} = t_{detected,i,j} - t_{stimulus,i,j} \tag{3.22}$$

where i and j are respectively subjects and events. However, this raw latency measurement inherently includes delays introduced by the text-to-speech synthesizer, as well as the lag associated with the Bluetooth audio output device. Additionally, it accounts for the time required to "pronounce" the spoken word. The pronounced words correspond to one of four possible combinations, defined by the event type (1 or 2) and duration (short or long), each potentially affecting the total latency due to differences in word length and synthesis time. To isolate and remove this contribution from the initially computed raw latency, an experimental procedure was performed. A wired microphone was used to precisely detect the actual moment when the spoken stimulus ended, thus the instant when the subject receipts the cue. This allowed the estimation of the latency associated with each of the four combinations (L_{speech,c_j}), which was then subtracted from the corresponding raw latency values. The result is a refined latency ($L_{refined,i,j}$) that reflects both the subject's reaction time and the intrinsic delay of the detection system for each detected event:

$$L_{refined,i,j} = L_{RAW,i,j} - L_{speech,c_j} \tag{3.23}$$

Subsequently, the refined latency average value was calculated for each subject:

$$\bar{L}_{\text{refined},i} = \frac{1}{E_i} \sum_{j=1}^{E_i} L_{\text{refined},i,j} \tag{3.24}$$

Where E_i is the number of events for the subject i. However, under normal operating conditions, the subject initiates the response voluntarily, without reacting to an external stimulus. Therefore, to further isolate the latency attributable solely to the recognition system, the average human reaction time to auditory stimuli in a choice task, approximately 283 milliseconds [70], was subtracted to the average refined latency value for each subject.

Finally, the corrected latency is calculated as:

$$L_{C,i} = \bar{L}_{\text{refined},i} - t_{\text{reaction}} \tag{3.25}$$

Corrected latencies were analyzed across all participants, reporting the mean and standard deviation.

To conclude the grand average between subjects was calculated as:

$$\text{Grand average} = \frac{1}{N}\sum_{i=1}^{N} L_{C,i} \tag{3.26}$$

Where N is the number of subjects.

Chapter 4

Results

This chapter presents the analysis of the experimental results obtained throughout the validation of the device. The focus lies on evaluating several key performance aspects and examines the accuracy and reliability of the pupil segmentation algorithm as well as the device's capability to detect and classify PAR events.

4.1 Device performances

This section presents the results related to the overall performance of the device, focusing on operational parameters. Specifically, it analyzes battery profiles, computational load and frame rate during system operation.

4.1.1 Power profile

In the following subsection, the results of the experimental tests conducted on the battery during both charging and discharging phases are presented. Figures 4.1. illustrates the charge level over time with the identified trade-off point highlighted in red (approximately 3 hour and 30 minutes to nearly reach 88% of total capacity).

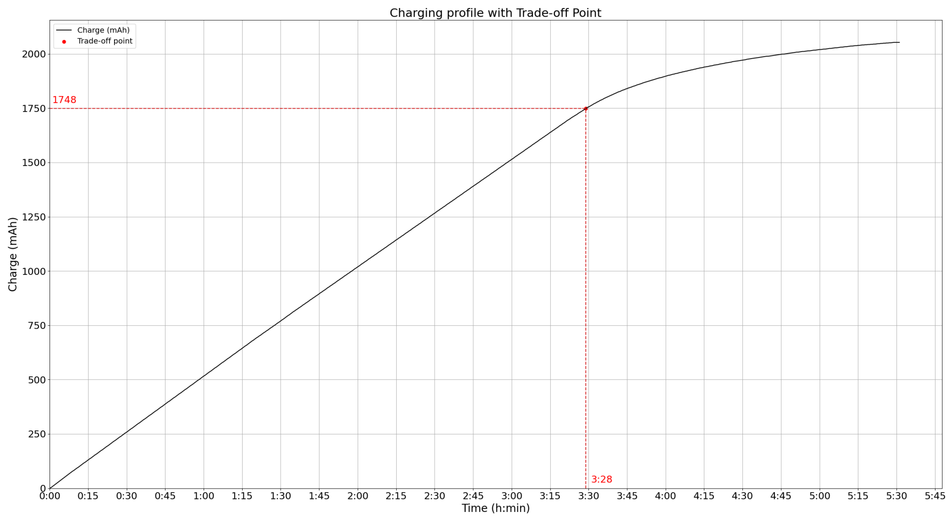


Figure 4.1: Charging power profile with identified trade-off point highlighted in red.

Battery discharge tests were conducted over a complete discharge cycle, starting from a fully charged state, analyzing both the active (operational) phase and the idle (standby) modality. The values obtained are reported in the following Table 4.1.

Table 4.1: Results of battery discharge tests in operational and standby conditions.

Modality	Average current drain (mA)	Delivered capacity (mAh)	Duration (hh:mm)
Operational	242	1250	5:10
Standby	60	1250	20:50

4.1.2 Computational load and FPS

Computational load

This subsection presents the analysis of the computational load on the *Raspberry Pi Zero*, the performance of segmentation on the PC, and the total number of FPS achieved by the system. In the following image (Figure 4.2), the processing times for each operation executed on the *Raspberry Pi Zero* are shown with green bars: acquisition (ACQ), cropping (CROP), JPEG encoding (ENCODE), and data transmission (SEND). For the PC (in sky blue), the analyzed steps include median filtering (MEDIAN), heatmap computation for ROI identification (HEATMAP), histogram calculation (HIST) and peak detection (PEAKS), thresholding (THRESHOLD), Canny filtering (CANNY), and finally contour analysis (CONTOURS).

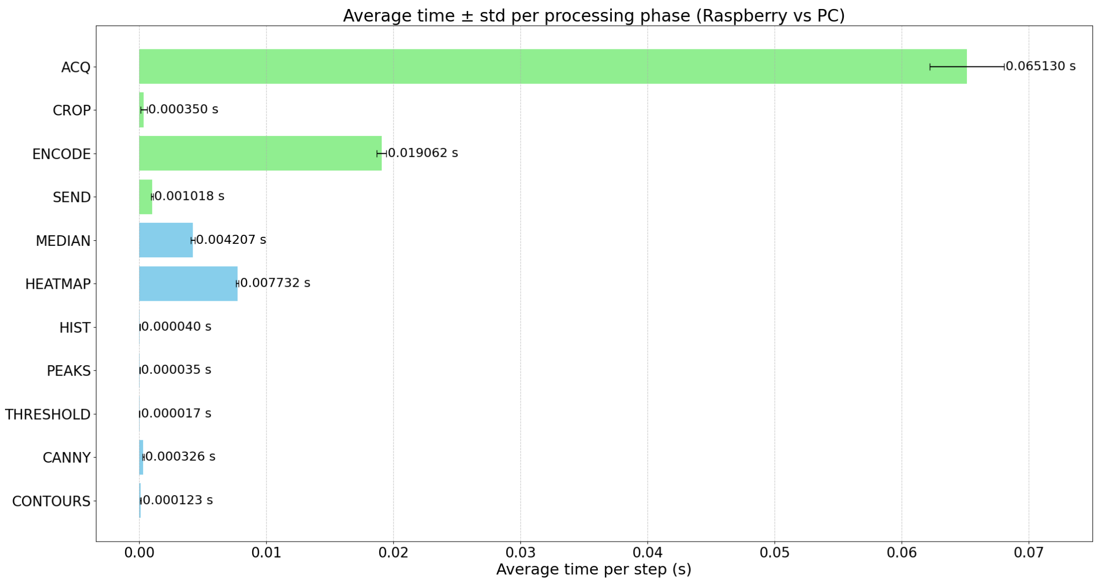


Figure 4.2: Computational load for each processing step performed on Raspberry (green) and during the segmentation pipeline on the PC (sky blue).

FPS

The total average time for acquisition and transmission (Raspberry side) is roughly 0.0856 seconds, setting the maximum theoretical framerate limit to approximately 11.7 FPS. Additionally, the segmentation pipeline costs around 0.0125 seconds in total, leading to a total time of 0.0981 second per frame, thus 10.2 FPS. The histogram of the frame rate distribution is shown below (Figure 4.3). The average framerate was found to be 10.04 $\pm$ 1.13 FPS.

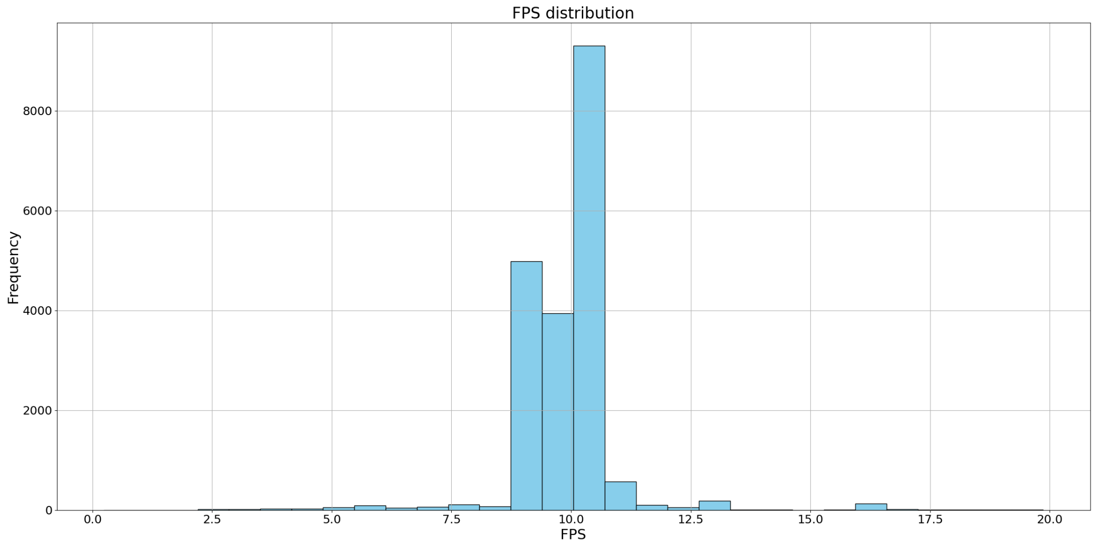


Figure 4.3: FPS distribution.

4.2 Pupil segmentation performance

This section reports the quantitative evaluation of the segmentation algorithm (S) on data collected from 20 subjects. Performance was assessed against two ground truths:

- **GT1**: manual segmentation (300 frames)
- **GT2**: semi-automatic segmentation using SAM v2 (all frames i.e. 3000)

Segmentation accuracy was measured using *Dice* and *IoU*, separately for three experimental phases: static, constriction, and ocular movement. The agreement between GT1 and GT2 was also quantified. Finally, the percentage of lost frames was calculated for each phase.

4.2.1 Algorithm vs primary ground truth

Table 4.2 reports the *Dice* and *IoU* scores of S compared to the GT1, separately for each phase.

Table 4.2: Algorithm performances on primary ground truth dataset.

Phase	**Dice (%) ± std (%)**	**IoU (%) ± std (%)**
Static	92.17 ± 9.18	86.22 ± 8.81
Constriction	92.54 ± 5.19	86.43 ± 6.34
Movement	81.62 ± 29.69	75.94 ± 27.95
Mean over phases	**88.78 ± 14.69**	**82.86 ± 14.37**

4.2.2 Algorithm vs secondary ground truth

Table 4.3 reports the *Dice* and *IoU* scores of S compared to the GT2, separately for each phase.

Table 4.3: Algorithm performances on secondary ground truth dataset.

Phase	**Dice (%) ± std (%)**	**IoU (%) ± std (%)**
Static	91.73 ± 7.29	85.19 ± 7.07
Constriction	91.42 ± 6.24	84.55 ± 6.35
Movement	81.44 ± 28.52	75.10 ± 26.71
Mean over phases	**88.20 ± 14.02**	**81.61 ± 13.38**

4.2.3 Agreement between ground truths

To quantify the consistency between GT1 and GT2, Table 4.4 reports the *Dice* and *IoU* scores between the two ground truths, computed over the subset of frames where both were available.

Table 4.4: Agreement between the two ground truths (primary and secondary).

Phase	**Dice (%) ± std (%)**	**IoU (%) ± std (%)**
Static	95.92 ± 1.21	92.19 ± 2.22
Constriction	95.48 ± 1.42	91.39 ± 2.57
Movement	92.70 ± 16.20	88.77 ± 15.83
Mean over phases	**94.70 ± 6.28**	**90.78 ± 6.87**

4.2.4 Frame loss rate

The percentage of frames for which the segmentation failed was computed for each phase. Results are shown in Table 4.5.

Table 4.5: Percentage of lost frames for each phase.

Phase	**Loss frames (%)**
Static	0.40 %
Constriction	0.30 %
Movement	8.38 %

4.3 PAR identification performance

This section presents the classification performance of the proposed system in detecting PAR events. The analysis is structured according to three criteria: event type, event duration, and a combined classification. In addition, a detailed evaluation of corrected latencies is provided, following the methodology described in Section 3.3.3. The results aim to assess the accuracy, reliability, and temporal responsiveness of the system.

4.3.1 Event classification performances

The following figures represent the confusion matrix obtained by the experimental protocol. Specifically Figure 4.4. refers to the classification of event type, Figure 4.5 to the classification of duration and Figure 4.6 represent the combined classification of the event. Table 4.6 summarizes the classification accuracy for each confusion matrix.

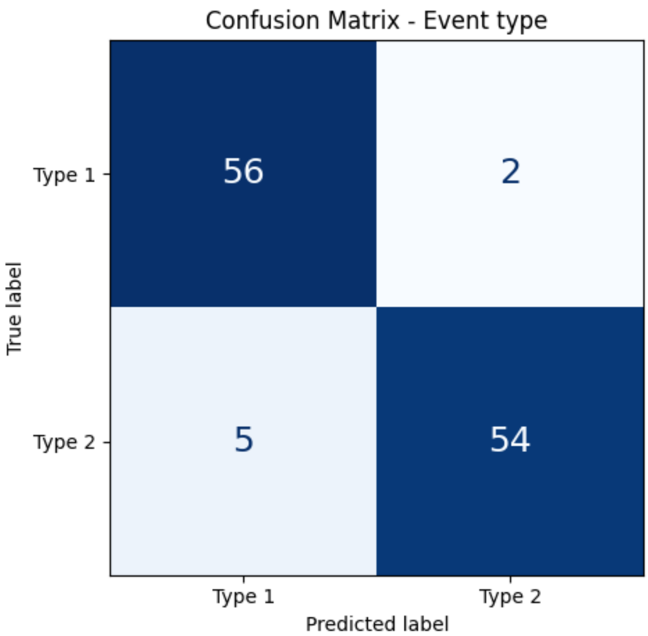


Figure 4.4: Confusion matrix of event type classification.

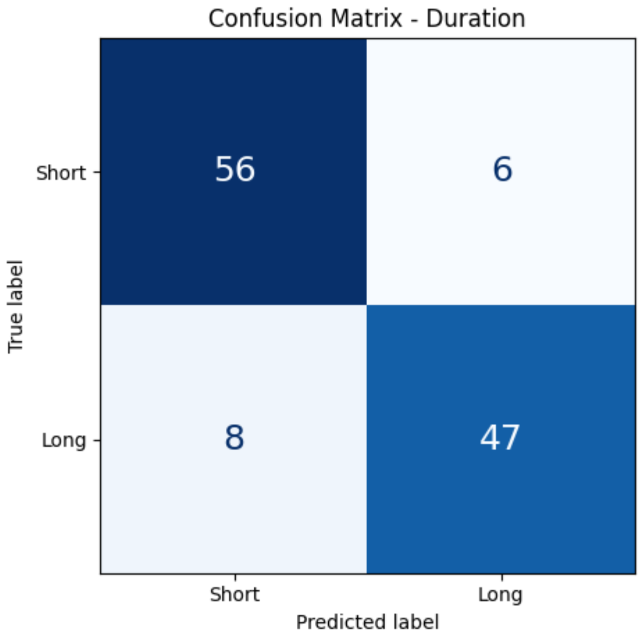


Figure 4.5: Confusion matrix of event duration classification.

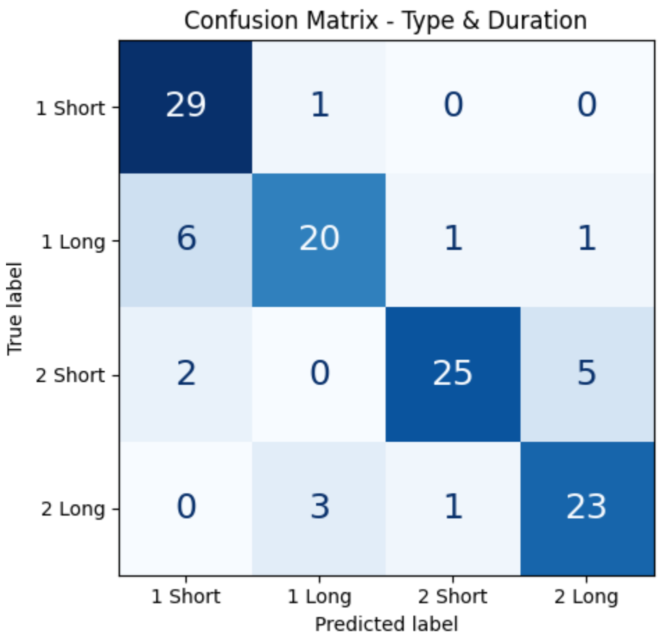


Figure 4.6: Confusion matrix of combined (type + duration) events classification.

Table 4.6: Summary of accuracy for each classification task.

Classification task	**Accuracy (%)**
Event type	94.02
Event duration	88.03
Event type and duration	82.91

Additionally, Table 4.7 reports the metrics for the binary classification task, which evaluates the algorithm's ability to correctly detect the presence of an event regardless of its type or duration. Moreover, In Table 4.8 the corrected latencies of event detection are displayed for each subject along with the grand average value. Table 4.9 contains the corrected latencies divided for event type.

Table 4.7: Binary classification task (Event vs not event) performance metrics.

Metrics	**Values (%)**
Precision	95.90
Recall	94.35
F1-score	95.12

Table 4.8: Average corrected latency values for each subject and grand average value.

Subjects	**Average corrected latency (s) ± std (s)**
1	2.12 ± 0.53
2	1.17 ± 0.76
3	1.51 ± 0.61
4	1.07 ± 0.63
5	0.84 ± 0.51
6	1.45 ± 0.42
7	1.53 ± 0.53
8	1.04 ± 0.37
9	1.60 ± 0.65
10	1.45 ± 0.32
11	2.12 ± 0.54
12	1.38 ± 0.34
13	1.49 ± 0.43
14	2.05 ± 0.89
15	1.59 ± 0.58
Grand average	**1.50 ± 0.55**

Table 4.9: Average corrected latency values for each event type.

Event type	**Average corrected latency (s) ± std (s)**
Type 1 (far target)	1.28 ± 0.58
Type 2 (near target)	1.70 ± 0.63

Chapter 5

Discussion

This chapter discusses the experimental results in relation to the goals of the study, considering their significance, alignment with existing literature, and potential implications for real-world applications. It highlights the system's strengths and limitations, both technical and physiological, and identifies areas where the approach could be improved. The discussion also explores future directions for hardware and software development proposing possible extensions of the system to broader assistive and interactive scenarios.

5.1 Interpretation of the results

During the experimental testing phase, the overall performance of the system across multiple dimensions was assessed, including power consumption, computational load, FPS, segmentation accuracy, and PAR event detection capabilities. In this section, the results obtained are interpreted.

5.1.1 Battery and power consumption

Contrary to the initial theoretical estimations described in Section 3.1.1., the actual current consumption measured during typical operation turned out to be significantly lower. This suggests that the energy requirements of the system were overestimated in the design phase, leading to the selection of a battery with greater capacity than strictly necessary.

The battery used for testing has a nominal capacity of 2000 mAh. However, direct measurements during controlled discharge cycle revealed that only 1250 mAh were effectively delivered. This discrepancy can be attributed to a combination of factors, including the behavior of the power control module, which cuts off the output to prevent over-discharge, thus leaving part of the battery's capacity unused.

Despite delivering only about 63% of the rated capacity, the battery still ensured sufficient runtime for practical use. Additionally, analysis of the battery charging cycle revealed that the charge-vs-time curve gradually flattens in the final phase. This behavior is consistent with the CC/CV charging profile implemented by the power module. Based on these observations, a trade-off point of approximately 3 hours and 30 minutes was identified, beyond which the gain in charge becomes marginal. This partial charging level was found to be sufficient to deliver around 4 hours and 45 minutes of continuous operation. While this charging-to-runtime ratio may be considered acceptable for a prototype, it would be less than ideal for a commercial or clinical-grade device, where shorter charging times would be expected. However, in the specific context of this application, such a long runtime is not strictly necessary, as it is unlikely that a patient in a CLIS condition would be able to engage with the system continuously for several hours. This further supports the conclusion that the battery is oversized relative to the actual usage requirements.

5.1.2 Pipeline analysis and FPS considerations

Experimental analysis revealed that the main performance bottleneck lies in the *Raspberry Pi Zero*, which handles image acquisition and transmission. The real frame rate was measured experimentally by analyzing the distribution shown in Figure 4.3, resulting in an average per-frame processing time of approximately 0.01 seconds, 86% of which corresponds to operations executed on the *Raspberry Pi Zero*. As shown in Figure 4.2, the majority of the total per-frame processing time is spent on the image capture operation. Although the camera module was configured for a frame rate of 30 FPS, the actual acquisition rate reached only about 15 FPS. This discrepancy is probably due to the way image data is transferred from the GPU to the CPU. The camera interface and image acquisition are handled by the GPU, but further processing steps (such as cropping and JPEG encoding) must be performed by the CPU, requiring the image data to be copied from GPU-managed memory to system RAM. This architecture introduces significant delays, ultimately reducing the effective frame rate. Moreover, although the JPEG encoding step introduces an additional delay of approximately 0.02 seconds per frame, it proves advantageous in this context. Transmitting raw image data would otherwise result in excessive bandwidth usage, potentially hindering real-time communication between Raspberry and the PC. On the PC side, the most time-consuming operations include median filtering and, more notably, the convolution used for the generation of heatmap for ROI detection. These results justified the attempt to develop an alternative, faster ROI detection method, as described in Section 3.1.5. Although this method was ultimately discarded due to its low accuracy on the current device.

Despite the limitations in hardware performances, the achieved frame rate is considered adequate for the intended application, since pupillary constriction events are relatively slow and do not require high temporal resolution. This characteristic was one of the motivations for developing a custom hardware solution, as most commercial pupillometry systems are optimized for high-speed eye tracking, leading to high cost and limited accessibility for deployment in domestic settings. Additionally, the relatively low frame rate provides an inherent advantage by filtering out eye blinks, which are typically fast. A blink occurring between two frames is likely to be skipped, avoiding the risk of misinterpreting a partial eyelid closure as a sudden drop in pupil size, a condition that could falsely trigger the detection of a PAR event.

5.1.3 Segmentation accuracy

To evaluate the accuracy of the proposed segmentation algorithm (S), a comparative analysis was performed against two distinct ground truths: GT1, obtained through manual annotation, and GT2, generated semi-automatically using the SAM v2 model. The evaluation was conducted across three distinct phases: static, constriction, and movement.

When compared with GT1, S achieved a high overall performance, with a mean *Dice* of nearly 89% and a mean IoU of 83%. Performance remained stable across static and constriction phases (*Dice* > 92%, IoU > 86%) but dropped significantly during the movement phase. A similar trend was observed in Table 4.3, comparing S with GT2, where *Dice* and IoU both undergo a decrease of approximately 10% from static and constriction phases to movement one. The agreement between GT1 and GT2 was considerably higher across all phases (mean *Dice*: 95%, IoU: 91%), confirming that both reference masks are largely consistent and can serve as reliable benchmarks. This strengthens the validity of the observed performance gap under motion conditions as genuinely due to algorithm limitations, rather than annotation inconsistencies. Complementary to the accuracy metrics, the analysis of lost frames, further underscores the challenges posed by motion. The percentage of lost frames was negligible during the static (0.40%) and constriction (0.30%) phases. However, during movement, lost frames increased substantially to more than 8%. This elevated loss rate in dynamic conditions is indicative of occasional failures in frames where motion-induced blur hinder accurate segmentation.

In conclusion, the proposed segmentation algorithm achieves performance comparable to that of the SAM2 method (GT2) in scenarios characterized by limited motion, despite operating in real time and requiring significantly lower computational resources. Its high reliability during both static and constriction phases, the most relevant conditions for the intended assistive context, confirms the suitability for the present application.

5.1.4 PAR events detection and classification

The performance of the proposed algorithm in detecting PAR events was assessed through a two-step classification task. The first step involved binary detection of the presence or absence of an event, while the second focused on multiclass classification, where the algorithm aimed to determine the event's type, duration, or both.

The binary classifier demonstrated excellent performance, achieving a $Precision$ of nearly 96%, *Recall* of 94%, and an $F1 - score$ of 95%. These results indicate a low false positive and false negative rate, suggesting that the system is reliable in detecting whether a PAR event occurred, regardless of its specific type or duration. This level of performance is particularly relevant in assistive contexts, where missing an event (false negative) or misreporting one (false positive) could compromise system utility or user trust.

In classifying the type of detected events, the system achieved an overall accuracy of 94%. The confusion matrix in Figure 4.4. indicates a slight bias toward type 1 events, which were more accurately recognized compared to type 2. This discrepancy may be attributed to the internal logic of the classification algorithm: the decision boundary for type 1 spans a broader range of pupillary sizes, whereas type 2 requires reaching a stricter threshold. As the experiment progresses, a gradual reduction in pupillary excursion was observed in some subjects, likely due to fatigue or habituation. This results in some type 2 constrictions falling short of the expected amplitude, causing them to be misclassified as type 1.

The recognition of event duration yielded a lower accuracy of 88%. This can be attributed to several contributing factors. First, the duration classifier employs a relatively coarse decision mechanism that proves insufficient when applied to a signal with high variability. Notably, in some cases, the pupil diameter may transiently re-cross the constriction threshold after an initial PAR event, momentarily exceeding it before returning below. This causes the algorithm to prematurely label the event as short (particularly in type 1 events), even though the participant performed a long constriction. Second, the experimental protocol allowed participants to voluntarily control the constriction duration, which naturally introduces timing inconsistencies and reduces the robustness of fixed temporal thresholds used to discriminate short from long events. Furthermore, some participants reported difficulty in maintaining a steady fixation on the visual target for the required duration to elicit a long event. Some noted that their vision became blurred during sustained fixation. This may be explained by the monotonous nature of the target, which remained static and non-luminous, offering little visual stimulation. Such an effect could undermine the subject's ability to consistently maintain a strong accommodative effort, thereby affecting both the amplitude and consequently the detected duration of the event. Additionally, type 2 events involve deeper constrictions, which are followed by a slower return to baseline, causing the algorithm to systematically overestimate the duration due to the extended recovery phase, a phenomenon that occurs despite the compensatory strategy described in Section 3.1.6.

The most complex task, joint classification of both event type and duration, yielded an accuracy of roughly 83%, which remains acceptable given the four-class difficulty and the proof-of-concept nature of the work. The confusion joint matrix in Figure 4.6. highlights several trends:

- The most accurate category was type 1 short, with very few misclassifications.

- Type 1 long events were occasionally misidentified as short, likely due to the previously described mechanisms.

- Type 2 short was mainly confused with type 2 long, reflecting again the tendency to overestimate duration in pronounced constrictions.

some other misclassifications suggest occasional ambiguities.

Additionally, the average corrected latency recorded across all participants was 1.50 s. This value was obtained by adjusting the raw latency based on the actual end time of the auditory instruction and by subtracting an expected reaction time derived from literature. As a result, the corrected latency reflects both the system delay in detecting a pupil constriction and the duration of the constriction itself. This latency is higher than initially expected, potentially limiting the responsiveness of the system, particularly in applications that rely on timing-based scanning interfaces, such as the one described in Section 3.2.2. However, a closer look at the signal trends, such as the one shown in Figure 3.26, reveals that the system correctly identifies the onset of PAR events at the end of the signal drop. Moreover, since latency values were already corrected as outlined in Section 3.3.3, the remaining delay can be reasonably attributed to two main factors. First, the latency introduced by the Wi-Fi communication, which was not explicitly measured but could contribute non-negligibly to the overall delay. Second, the pupillary accommodation reflex itself is not instantaneous, and its physiological latency inherently limits the speed of response. To this extent, different types of events exhibit varying durations; for instance, type 2 constrictions, due to their greater amplitude, require more time to reach completion, which in turn contributes to a longer effective latency when compared to the quicker type 1 constrictions. This difference is notable in Table 4.9, indicating that type 1 events, in addition to being more reliably classified, also exhibit shorter latencies. This makes them the preferred choice for implementing frequently used commands, where rapid response is critical.

5.2 Comparison with previous work

The use of PAR as a communication mechanism remains a relatively new area of research, with a still limited body of literature. Nonetheless, several prior studies have laid foundational groundwork. Notably, the work by Ponzio et al. [6] demonstrated the feasibility of leveraging voluntary pupil constrictions for communication, thereby providing a validated paradigm upon which the present system is based. Their experimental methodology, particularly the use of auditory cues to guide user responses, informed the design of the testing protocol adopted in this thesis.

What distinguishes the present system from earlier approaches is the introduction of a multi-symbolic communication framework, capable of classifying both the type and duration of the PAR within a unified structure. This advancement allows for the recognition of up to four distinct commands, representing a substantial increase in information throughput compared to conventional binary systems. As such, the current work constitutes a pilot study in multi-symbolic PAR-based interaction, making direct quantitative comparisons with previous devices inherently limited due to the increased complexity of the classification task. Earlier systems typically relied on time-constrained paradigms, requiring users to trigger pupil constrictions either within a fixed time window or for a specific duration, with each mode operating independently. Despite their relative simplicity, such systems have shown high robustness. For instance, the e-Pupil device [50] achieved up to 100% recognition accuracy within its subject pool for binary classification tasks. While these results are not directly comparable to those of the present system, they underscore the reliability of the PAR as a control signal and its promise for the development of assistive communication technologies.

A further key innovation introduced in this thesis is the use of the PAR not solely as a binary selector, but as an active navigation trigger within a dynamic user interface. The development of the SPEAKER interface along with the structure of the main menu GUI illustrates how PAR-based control can extend beyond basic yes/no commands to enable letter-by-letter output through structured selection processes. This greatly expands the communicative potential, allowing users to construct words and sentences, thereby supporting more expressive and functional interactions.

5.3 System limitations

While the proposed system demonstrates the feasibility of using the PAR as a communication channel, several limitations must be acknowledged before considering its deployment in real-world contexts. The following subsections explore these limitations in detail.

5.3.1 Robustness of pupil segmentation

A primary limitation concerns the reliability of the pupil segmentation algorithm. Although the use of IR LEDs improves robustness in controlled environments, performance may degrade significantly in the presence of intense ambient light, particularly sunlight. In such conditions, IR illumination may be overpowered, resulting in low contrast between the pupil and surrounding structures, which prevents accurate contour detection. Additional segmentation issues arise in cases where users wear eye makeup, which may introduce dark regions that interfere with ROI detection. Some contact lenses can also cause light reflections or alter the perceived border of the pupil, generating high-frequency noise in the pupil area signal and reducing events recognition stability.

5.3.2 Visual impairments and focus limitations

The effectiveness of the PAR also depends on the user ability to focus on near and far targets. Visual impairments, can hinder the ability to perform the accommodation task correctly, often resulting in a reduced amplitude of the pupillary response. While corrective solutions such as eyeglasses can improve focus, they may introduce unwanted reflections that compromise image segmentation. Contact lenses might be a more practical alternative, although, as mentioned, they too may negatively affect segmentation performance, probably depending on their design and optical properties. Moreover, sustaining prolonged constrictions, required for temporal encoding, often proves challenging when fixating on visually unengaging and opaque targets, as the user tends to lose focus over time. This limitation can reduce the reliability of duration-based command recognition and highlight the importance of carefully designing visual stimuli to support sustained accommodation.

5.3.3 Algorithm limitations

From a technical standpoint, the current algorithm for detecting the duration of the PAR event and classifying its type is intentionally kept simple, aligning with the proof-of-concept nature of this work. While this approach has proven functional for demonstrating the system's feasibility, its precision and robustness may not yet be sufficient for deployment in a finalized assistive product. In particular, the estimation of the PAR duration is affected by a known physiological characteristic reported in the literature: the re-dilation phase of the pupil often begins before the user fully shifts their gaze back from the near to the far target. This anticipatory behavior introduces intrinsic ambiguity in identifying the exact end of the accommodation response, complicating reliable duration assessment, particularly crucial in this system where temporal encoding is essential for command selection.

Another limitation concerns the initial decoder configuration required by the system, which must be performed at each startup and may need to be repeated during extended sessions. This requirement stems from the algorithm's sensitivity to individual physiological characteristics and environmental variability, such as lighting conditions. A more advanced solution would ideally eliminate the need for calibration altogether or at least reduce its frequency by fitting itself to the user's specific features in a dynamic and adaptive manner, while maintaining robustness against the primary sources of variability that currently necessitate re-calibration. To ensure the level of precision required for real-world use, the integration of more advanced signal analysis techniques or machine learning models appears essential. Advancing the algorithmic core in this direction is therefore critical for evolving the system from a proof-of-concept into a reliable assistive technology.

5.3.4 Quantitative benchmarking

To further validate the system's effectiveness, it would be good practice to conduct quantitative comparisons with commercial eye-tracking systems. Although the low-cost nature of the device is a strength, demonstrating comparable performance in capturing PAR events would substantiate the rationale behind this design choice. In addition, a formal characterization of the system's information rate, expressed in bits per minute, would allow direct comparison with other assistive communication devices in terms of throughput and efficiency, offering a more complete assessment of its potential impact on the panorama of AAC technologies.

5.4 Future work

This chapter outlines potential future developments of the system, both in terms of hardware and software improvements, as well as possible applications. In addition, it highlights the importance of conducting clinical testing on patients to gather direct feedback on the user interface and to assess the overall usability of the developed application in real-world conditions. These future steps are essential to validate the effectiveness of the system and to ensure that it meets the practical needs of its intended users.

5.4.1 Hardware and software improvements

The present prototype represents a functional starting point, but several improvements are required to transition toward a robust and clinically viable system. First and foremost, redesigning the hardware is necessary. The current use of *Raspberry Pi Zero* has proven to be the bottleneck in the pupillometry pipeline, significantly limiting the overall performance of the system. Additionally, the battery used has shown to be oversized with respect to the actual energy demands, suggesting that more compact and efficient power solutions could be adopted.

Given the intended use in clinical and domestic environments, the introduction of a wired communication interface should also be considered. While Wi-Fi connectivity offers flexibility, especially when the patient is moved or operates outside a fixed setting, a wired connection would eliminate delays and potential instability caused by network speed fluctuations during regular use. Wireless functionality could be preserved as an optional mode for mobile or non-standard contexts.

From a software perspective, future developments should aim at enhancing the segmentation robustness, both against noise and occlusions (e.g., corrective eyeglasses), as well as reducing computational load. An initial attempt has already been made to replace the convolutional stage, the most resource-intensive element of the segmentation pipeline, with a more efficient ROI detection strategy. However, this alternative approach led to poorer performance under variable or low-light conditions, highlighting the need for further optimization.

In terms of event recognition, more advanced algorithms could significantly improve performance across all key aspects: classification of PAR type, duration estimation, and detection latency. Preliminary work has already explored the use of lightweight online models, such as Masked Autoencoders (MAE) adapted to time series data, achieving excellent binary event detection with low latency. Nevertheless, these models have yet to be evaluated for their ability to capture constriction magnitude and to associate such information with specific PAR types and durations. Despite these current limitations, the high intra- and inter-subject variability and the lack of event repeatability strongly support shifting the research focus toward machine learning and deep learning approaches, which are better suited to handle the complexity and variability inherent in biological signals.

Lastly, the development of additional applications represents the natural progression of this work. Expanding the system's functionality to include assistive interfaces, such as home automation controls, could greatly enhance user independence. Moreover, the integration of interactive applications like video games may offer substantial psychological and emotional benefits, particularly in patients affected by neurodegenerative diseases such as ALS. In such cases, gaming, especially when involving online interaction, can serve as a meaningful form of mental escape for individuals physically constrained by their condition.

5.4.2 Clinical and usability tests

Another important aspect of the present work is the absence of clinical validation in the target population, namely patients in CLIS. Although the experimental detection of the PAR performed in this study is valid for assessing the technical performance of the device, particularly in terms of its ability to detect and classify PAR events with sufficient accuracy, it does not fully reflect the dynamics of real-world use. In the controlled experimental setting, the initiation of PAR events was externally guided through auditory cues, simplifying the task by removing the cognitive load associated with decision-making. In a practical application, however, the user must independently initiate the response, and may encounter additional complexities: for instance, hesitations, last-minute changes in the selected target, or corrective shifts in gaze during an already ongoing constriction. These scenarios, common in real communication contexts, were not explored in this study, and the system's robustness in such situations remains unknown. As a result, the voluntary and intentional nature of the interaction, which is central to assistive use, was only partially addressed. This limitation becomes even more critical when considering the use of the developed SPEAKER interface. This application introduces an additional layer of complexity by requiring not only the recognition of a PAR event but also its precise timing. Without usability testing on a subject pool, including individuals with motor impairments, the intuitive use and accessibility of the interface remain uncertain. Evaluating these aspects is essential to determine the true applicability of the system as a communication aid in real-world assistive scenarios.

Chapter 6

Conclusion

This thesis presented the design and development of a prototype pupillometric communication device, addressing both hardware and software aspects, with the aim of exploring its potential as an assistive communication tool for completely paralyzed patients, such as those in CLIS due to the progression of ALS. The system leverages the PAR to establish a human-computer interaction pathway. Unlike previous approaches that relied on binary communication schemes, this work aimed to investigate the feasibility of enabling multi-symbolic communication, allowing for a broader set of commands and more expressive interaction. To this end, a complete communication interface and a GUI for application selection were developed. The system achieved excellent performance in pupil segmentation, showed good reliability in detecting PAR events, and yielded acceptable results in classifying these events based on their type and duration. This work represents the first actual implementation of multi-symbolic interaction using the PAR mechanism, laying the foundation for future advancements in this domain. In particular, this thesis demonstrates that a major limitation of the binary communication, namely its slowness, can be effectively solved by exploiting a multi-symbolic PAR. However, several limitations remain. Most notably, the system has not yet been tested on actual CLIS patients, and some technical constraints were observed, particularly related to the robustness of the PAR event recognition algorithm, especially concerning temporal encoding strategies. Future developments should therefore include clinical validation of the device with CLIS patients and an enhancement of the event recognition performance through more advanced and reliable signal analysis techniques. Furthermore, the system architecture has been designed to support the development of additional applications beyond the communicator already implemented, potentially expanding the device's capabilities towards areas such as video games or smart home control to improve autonomy and quality of life. In conclusion, this work demonstrates the viability of pupillometry-based multi-symbolic interaction through the PAR and provides a solid groundwork for the development of new assistive technologies aimed at enhancing communication, interaction, and independence for individuals affected by severe paralysis.

Bibliography

[1] L. P. Rowland e N. A. Shneider. «Amyotrophic lateral sclerosis». In: *N Engl J Med* 344 (2001), pp. 1688–1700.

[2] S. Laureys et al. «The locked-in syndrome: what is it like to be conscious but paralyzed and voiceless?» In: *Progress in Brain Research*. Ed. by S. Laureys. Vol. 150. The Boundaries of Consciousness: Neurobiology and Neuropathology. Elsevier, 2005, pp. 495–611.

[3] M. Fried-Oken, A. Mooney, and B. Peters. «Supporting communication for patients with neurodegenerative disease». In: *NeuroRehabilitation* 37.1 (2015), pp. 69–87.

[4] F. Lotte et al. «A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update». In: *J. Neural Eng.* 15.3 (Apr. 2018), p. 031005.

[5] S. B. Borgheai et al. «Enhancing Communication for People in Late-Stage ALS Using an fNIRS-Based BCI System». In: *IEEE Trans. Neural Syst. Rehabil. Eng.* 28.5 (May 2020), pp. 1198–1207.

[6] F. Ponzio, A. E. L. Villalobos, L. Mesin, C. de'Sperati, and S. Roatta. «A human-computer interface based on the "voluntary" pupil accommodative response». In: *Int. J. Hum.-Comput. Stud.* 126 (June 2019), pp. 53–63.

[7] A. Bogucki and R. Salvesen. «Sympathetic iris function in amyotrophic lateral sclerosis». In: *J. Neurol.* 234.3 (Apr. 1987), pp. 185–186.

[8] J. Stoll, C. Chatelle, O. Carter, C. Koch, S. Laureys, and W. Einhäuser. «Pupil responses allow communication in locked-in syndrome patients». In: *Curr. Biol.* 23.15 (Aug. 2013), R647–R648.

[9] A. E. Lorenzo Villalobos et al. «When assistive eye tracking fails: Communicating with a brainstem-stroke patient through the pupillary accommodative response – A case study». In: *Biomed. Signal Process. Control* 67 (May 2021), p. 102515.

[10] S. Kasthurirangan and A. Glasser. «Characteristics of pupil responses during far-to-near and near-to-far accommodation». In: *Ophthalmic Physiol. Opt.* 25.4 (July 2005), pp. 328–339.

[11] J. Barberio, C. Lally, V. Kupelian, O. Hardiman, and W. D. Flanders. «Estimated Familial Amyotrophic Lateral Sclerosis Proportion». In: *Neurol. Genet.* 9.6 (Nov. 2023), e200109.

[12] M. Nijs and P. Van Damme. «The genetics of amyotrophic lateral sclerosis». In: *Curr. Opin. Neurol.* 37.5 (Oct. 2024), pp. 560–569.

[13] A. E. Renton et al. «A hexanucleotide repeat expansion in C9ORF72 is the cause of chromosome 9p21-linked ALS-FTD». In: *Neuron* 72.2 (Oct. 2011), pp. 257–268.

[14] P. M. Andersen. «Amyotrophic lateral sclerosis associated with mutations in the CuZn superoxide dismutase gene». In: *Curr. Neurol. Neurosci. Rep.* 6.1 (Feb. 2006), pp. 37–46.

[15] S. Lattante, G. A. Rouleau, and E. Kabashi. «TARDBP and FUS mutations associated with amyotrophic lateral sclerosis: summary and update». In: *Hum. Mutat.* 34.6 (June 2013), pp. 812–826.

[16] W. van Rheenen et al. «Common and rare variant association analyses in amyotrophic lateral sclerosis identify 15 risk loci with distinct genetic architectures and neuron-specific biology». In: *Nat. Genet.* 53.12 (2021), pp. 1636–1648.

[17] S. H. Van Daele et al. «Genetic variability in sporadic amyotrophic lateral sclerosis». In: *Brain* 146.9 (Apr. 2023), pp. 3760–3769.

[18] M. R. Turner et al. «Genetic screening in sporadic ALS and FTD». In: *J. Neurol. Neurosurg. Psychiatry* 88.12 (Dec. 2017), pp. 1042–1044.

[19] F. Trojsi, M. R. Monsurrò, and G. Tedeschi. «Exposure to Environmental Toxicants and Pathogenesis of Amyotrophic Lateral Sclerosis: State of the Art and Research Perspectives». In: *Int. J. Mol. Sci.* 14.8 (July 2013), pp. 15286–15311.

[20] M. R. Langley et al. «High fat diet consumption results in mitochondrial dysfunction, oxidative stress, and oligodendrocyte loss in the central nervous system». In: *Biochim. Biophys. Acta BBA Mol. Basis Dis.* 1866.3 (Mar. 2020), p. 165630.

[21] B. Marin et al. «Age-specific ALS incidence: a dose-response meta-analysis». In: *Eur. J. Epidemiol.* 33.7 (July 2018), pp. 621–634.

[22] J. M. Aragones et al. «Amyotrophic lateral sclerosis: A higher than expected incidence in people over 80 years of age». In: *Amyotroph. Lateral Scler. Frontotemporal Degener.* 17.7-8 (Nov. 2016), pp. 522–527.

[23] C. J. Jagaraj, S. Shadfar, S. A. Kashani, S. Saravanabavan, F. Farzana, and J. D. Atkin. «Molecular hallmarks of ageing in amyotrophic lateral sclerosis». In: *Cell Mol Life Sci* 81.1 (Mar. 2024), p. 111.

[24] C. Cui et al. «Associations between autoimmune diseases and amyotrophic lateral sclerosis: a register-based study». In: *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration* (Apr. 2021).

[25] I. P. Sousa and T. C. R. G. Vieira. «Enterovirus infection and its relationship with neurodegenerative diseases». In: *Mem Inst Oswaldo Cruz* 118 (Mar. 2023), e220252.

[26] J. M. Al-Khayri et al. «Amyotrophic Lateral Sclerosis: Insights and New Prospects in Disease Pathophysiology, Biomarkers and Therapies». In: *Pharmaceuticals (Basel)* 17.10 (Oct. 2024), p. 1391.

[27] E. B. Moloney, F. de Winter, and J. Verhaagen. «ALS as a distal axonopathy: molecular mechanisms affecting neuromuscular junction stability in the presymptomatic stages of the disease». In: *Front. Neurosci.* 8 (Aug. 2014).

[28] E. Longinetti and F. Fang. «Epidemiology of amyotrophic lateral sclerosis: an update of recent literature». In: *Curr Opin Neurol* 32.5 (Oct. 2019), pp. 771–776.

[29] B. K. Vanselow and B. U. Keller. «Calcium dynamics and buffering in oculomotor neurones from mouse that are particularly resistant during amyotrophic lateral sclerosis (ALS)-related motoneurone disease». In: *J Physiol* 525.Pt 2 (June 2000), pp. 433–445.

[30] S. Silva-Hucha, A. M. Pastor, and S. Morcuende. «Neuroprotective Effect of Vascular Endothelial Growth Factor on Motoneurons of the Oculomotor System». In: *Int J Mol Sci* 22.2 (Jan. 2021).

[31] G. Maugeri, A. G. D'Amico, G. Morello, D. Reglodi, S. Cavallaro, and V. D'Agata. «Differential Vulnerability of Oculomotor Versus Hypoglossal Nucleus During ALS: Involvement of PACAP». In: *Front. Neurosci.* 14 (Aug. 2020).

[32] J. M. Das, K. Anosike, and R. M. D. Asuncion. «Locked-in Syndrome». In: *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2025.

[33] L. Xu et al. «Global variation in prevalence and incidence of amyotrophic lateral sclerosis: a systematic review and meta-analysis». In: *J Neurol* 267.4 (Apr. 2020), pp. 944–953.

[34] C. Wolfson, D. E. Gauvin, F. Ishola, and M. Oskoui. «Global Prevalence and Incidence of Amyotrophic Lateral Sclerosis: A Systematic Review». In: *Neurology* 101.6 (Aug. 2023), e613–e623.

[35] Global Burden of Disease Study 2016. «Global, regional, and national burden of motor neuron diseases 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016». In: *Lancet Neurol* 17.12 (Dec. 2018), pp. 1083–1097.

[36] K. C. Arthur, A. Calvo, T. R. Price, J. T. Geiger, A. Chiò, and B. J. Traynor. «Projected increase in amyotrophic lateral sclerosis from 2015 to 2040». In: *Nat Commun* 7 (Aug. 2016), p. 12408.

[37] P. Mehta et al. «Amyotrophic lateral sclerosis estimated prevalence cases from 2022 to 2030, data from the national ALS Registry». In: *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration* 26.3–4 (Apr. 2025), pp. 290–295.

[38] E. Aust et al. «Impairment of oculomotor functions in patients with early to advanced amyotrophic lateral sclerosis». In: *Journal of Neurology* 271.1 (2024), pp. 325–339.

[39] B. Peters et al. «SSVEP BCI and Eye Tracking Use by Individuals With Late-Stage ALS and Visual Impairments». In: *Frontiers in Human Neuroscience* 14 (Nov. 2020).

[40] H. O. Edughele, Y. Zhang, F. Muhammad-Sukki, Q.-T. Vien, H. Morris-Cafiero, and M. Opoku Agyeman. «Eye-Tracking Assistive Technologies for Individuals With Amyotrophic Lateral Sclerosis». In: *IEEE Access* 10 (2022), pp. 41952–41972.

[41] S. Mahajan, P. Leslie, C. Spani, W. Hook, N. Livingston, and C. Harris. «Facial EMG signal analysis method and its implementation as a stand-alone system». In: *CMBES Proceedings*. Vol. 33. June 2010.

[42] C. L. Mitchell et al. «Ability-based Keyboards for Augmentative and Alternative Communication: Understanding How Individuals' Movement Patterns Translate to More Efficient Keyboards». In: *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. 2022, p. 412.

[43] N. S. Card et al. «An Accurate and Rapidly Calibrating Speech Neuroprosthesis». In: *New England Journal of Medicine* 391.7 (Aug. 2024), pp. 609–618.

[44] M. J. Vansteensel et al. «Fully Implanted Brain–Computer Interface in a Locked-In Patient with ALS». In: *New England Journal of Medicine* 375.21 (Nov. 2016), pp. 2060–2066.

[45] S. Khan et al. «Invasive Brain–Computer Interface for Communication: A Scoping Review». In: *Brain Sciences* 15.4 (Apr. 2025).

[46] B. J. Edelman et al. «Non-Invasive Brain-Computer Interfaces: State of the Art and Trends». In: *IEEE Reviews in Biomedical Engineering* 18 (2025), pp. 26–49.

[47] S. Fazli et al. «Enhanced performance by a hybrid NIRS-EEG brain computer interface». In: *NeuroImage* 59.1 (Jan. 2012), pp. 519–529.

[48] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch. «EMG and EOG artifacts in brain computer interface systems: A survey». In: *Clinical Neurophysiology* 118.3 (Mar. 2007), pp. 480–494.

[49] M. Motlagh and R. Geetha. «Physiology, Accommodation». In: *StatPearls [Internet]*. `https://www.ncbi.nlm.nih.gov/sites/books/NBK542189/`. StatPearls Publishing, 2022.

[50] G. Chiarion et al. «e-Pupil: IoT-Based Augmentative and Alternative Communication Device Exploiting the Pupillary Near-Reflex». In: *IEEE Access* 10 (2022), pp. 130078–130088.

[51] N. J. Volpe, J. Simonett, A. A. Fawzi, and T. Siddique. «Ophthalmic Manifestations of Amyotrophic Lateral Sclerosis (An American Ophthalmological Society Thesis)». In: *Transactions of the American Ophthalmological Society* 113 (Sept. 2015), T12.

[52] F. E. M. Al-Obaidi, A. J. M. A. Al-Saeed, and A. J. M. Al-Zuhairi. «Self-Automatic Threshold Technique For Eye Pupil Detection». In: *Journal of Optics* (Feb. 2025).

[53] N. Min-Allah, F. Jan, and S. Alrashed. «Pupil detection schemes in human eye: a review». In: *Multimedia Systems* 27.4 (Aug. 2021), pp. 753–777.

[54] D. Iacoviello, M. Lucchetti, G. Calcagnini, and F. Censi. «Pupil edge detection and morphological identification from blurred noisy images». In: *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. unspecified.

[55] W. Fuhl, T. Kübler, K. Sippel, W. Rosenstiel, and E. Kasneci. «ExCuSe: Robust Pupil Detection in Real-World Scenarios». In: *Computer Analysis of Images and Patterns*. Ed. by G. Azzopardi and N. Petkov. Cham: Springer International Publishing, 2015, pp. 39–51.

[56] M. T. Setiawan, S. Wibirama, and N. A. Setiawan. «Robust Pupil Localization Algorithm Based on Circular Hough Transform for Extreme Pupil Occlusion». In: *2018 4th International Conference on Science and Technology (ICST)*. Aug. 2018, pp. 1–5.

[57] A. De Santis and D. Iacoviello. «Optimal segmentation of pupillometric images for estimating pupil shape parameters». In: *Computer Methods and Programs in Biomedicine* 84.2 (Dec. 2006), pp. 174–187.

[58] A. A. Jarjes, K. Wang, and G. J. Mohammed. «Improved greedy snake model for detecting accurate pupil contour». In: *2011 3rd International Conference on Advanced Computer Control*. Jan. 2011, pp. 515–519.

[59] D. Li, D. Winfield, and D. J. Parkhurst. «Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches». In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. Sept. 2005, pp. 79–79.

[60] R. Rathnayake et al. «Current Trends in Human Pupil Localization: A Review». In: *IEEE Access* 11 (2023), pp. 115836–115853.

[61] W. Fuhl, T. C. Santini, T. Kuebler, and E. Kasneci. *ElSe: Ellipse Selection for Robust Pupil Detection in Real-World Environments*. arXiv preprint arXiv:1511.06575. Nov. 2015.

[62] A.-H. Javadi, Z. Hakimi, M. Barati, V. Walsh, and L. Tcheang. «SET: a pupil detection method using sinusoidal approximation». In: *Frontiers in Neuroengineering* 8 (Apr. 2015).

[63] S. Rabba, Y. He, M. Kyan, and L. Guan. «Pupil localization for gaze estimation using unsupervised graph-based model». In: *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2017, pp. 1–4.

[64] M. Mottalli, M. Mejail, and J. Jacobo-Berlles. «Flexible image segmentation and quality assessment for real-time iris recognition». In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. Nov. 2009, pp. 1941–1944.

[65] S. Zhao and R.-R. Grigat. «Robust Eye Detection under Active Infrared Illumination». In: *18th International Conference on Pattern Recognition (ICPR'06)*. Aug. 2006, pp. 481–484.

[66] K. Bai, J. Wang, and H. Wang. «A Pupil Segmentation Algorithm Based on Fuzzy Clustering of Distributed Information». In: *Sensors* 21.12 (2021).

[67] G. Chen, Z. Dong, J. Wang, and L. Xia. «Pupil Localization Algorithm Based on Improved U-Net Network». In: *Electronics* 12.12 (2023).

[68] Y.-H. Yiu et al. «DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning». In: *Journal of Neuroscience Methods* 324 (Aug. 2019), p. 108307.

[69] V. Maquiling, S. A. Byrne, D. C. Niehorster, M. Carminati, and E. Kasneci. *Zero-Shot Pupil Segmentation with SAM 2: A Case Study of Over 14 Million Images*. arXiv preprint arXiv:2410.08926. Jan. 2025.

[70] Y. Takaoka. «[A study on measurement of auditory reaction time]». In: *Nihon Jibiinkoka Gakkai Kaiho* 93.5 (May 1990), pp. 746–755.

Appendix A

Device prototype

The following images show the prototype of the device from the right and left sides. These views provide a clearer understanding of its physical structure and component placement.

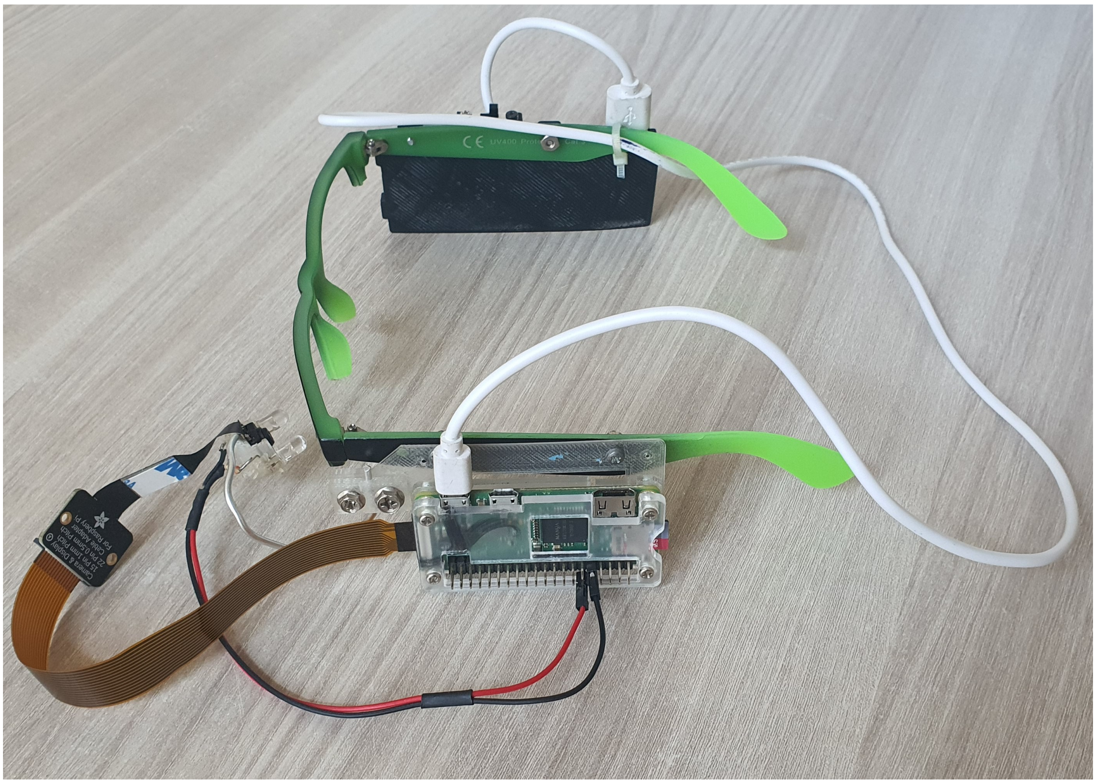


Figure A.1: The device prototype seen from raspberry side (left side)

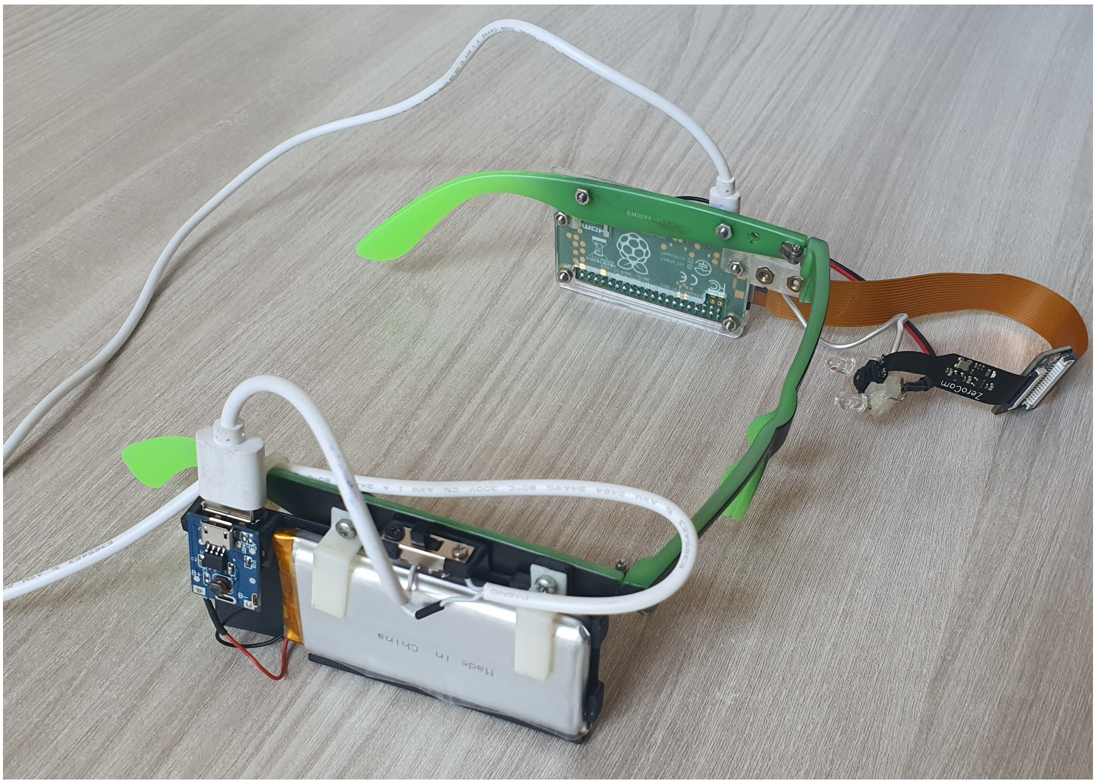


Figure A.2: The device prototype seen from battery module side (right side)