Letter of Transmittal

April 4th, 2025

Dear Dr Cameron Farrow
School of Engineering
Albert A. Thornbrough Building, Winegard Walk
Guelph, ON N1G 1Y4

Dear Dr Cameron Farrow

Our team has prepared the final report to showcase the final design of cost-effective assistive technology. Our team believes that the current landscape for accessible computing has too high of a price barrier to entry and wish to take a new approach to this problem.

The primary purpose of this report is to highlight our team's solution design. This document will also highlight our problem definition, scope, and outline our thought processes while designing the final Design. Our team believes that the ideas presented through the document will allow the educational team of ENGG*4100 to understand the need for the final design in this sector.

Sincerely,

University of Guelph
School of Engineering
ENGG*4100 - Engineering and Design IV

Interim report: Group 9

Due: April 4th, 2025

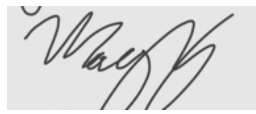
Recipients: Dr Cameron Farrow

Location: 50 Stone Road Guelph, ON N1G 1Y4

Charlie Magri (1194399)
Ricardo Quadras (1092724)

Signature Page

In signing this report, I certify that I have been an active team member and provided about equal contribution to the work. I take shared credit and responsibility for the content of this report. I understand that taking credit for work that is not my own is a form of academic misconduct and will be treated as such.

Charlie Magri 

Ricardo Quadras 

Executive Summary

Today's accessible computing systems are either expensive or overbearing on a user's computer. The team has designed a device that is portable and technology agnostic to allow for a solution no matter the user's current system and its processing power.

The solution designed appeals to the broadest range of people and accommodates a wide range of disabilities, and needs of physically impaired users. The design is modular, low cost, power efficient and offloads processing power required to a secondary Raspberry Pi that can be attached to the user's screen.

The final design offers the user a cheaper and efficient solution effectively reducing barriers to entry in this domain and offering an alternative than the current market offers; our design is meant to introduce positive impacts in environmental, social and economic sectors that will be discussed in detail later in this document.

Acknowledgements

Special thanks to Dr Cameron Farrow, PEng., for providing continuous feedback and guidance throughout the semester and for agreeing to being the project advisor. Thanks to the University of Guelph for providing the opportunity, and Lab Fund for accepting our request and allowing the project to be built.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Abbreviations	Definition
ICT	Information and Communication Technology
HCI	Human Computer Interface
CAT	Computer Access Technology
AI	Artificial Intelligence
UI	User Interface
SNP	Sip and Puff
GPM	General Purpose Mode
ASM	Application-Specific Mode
LLM	Large Language model
BNN	Binary Neural Network
Pi	Rasbery Pi Microcomputer
HDMI	High-Definition Multimedia Interface
OS	Operating Systems
FPGA	Field Programmable Gate Array
PC	Personal Computer
TD	Trigger Delay
PIPEDA	Personal Information Protection and Electronic Documents Act
AIDA	Artificial Intelligence and Data Act

Introduction

This document will cover challenges faced with accessibility computer systems and challenges identified in accessible computing. By addressing the weaknesses of the industry our development team hopes to create cheaper and more hardware efficient technology. This will allow people who are unable to use traditional computing systems due to motor impairments access to general, gaming, and productivity software on any computer and most importantly the opportunities provided by this connectivity in modern societies.

Problem definition

Problem Description

Access to Information and Communication Technologies (ICT) has fundamentally changed our daily lives, reshaping how people interact with society. This can be attributed to the Internet and its vast supporting telecommunication infrastructure that allows an "anytime, anywhere" exchange of information, knowledge, resources, and connection. Furthermore, its integration into everyday life has turned ICT into an imperative for developing societies, offering revolutionary transformations to all sectors like the workplace, education, connection, and personal growth. With 63% of the world having access to the Internet in 2023, a stunning rise from 35.4% online just 10 years prior [1]. However, ensuring equitable and accessible technology for all people, including persons with disabilities, remains a challenge both in terms of capabilities and access.

Today, 15% of the world's population experiences some form of disability, with a further 2-4% experiencing significant challenges to daily function [2]. Rising from 10% in the 1970s, this global increase in disability prevalence is in some part due to longer lifespans, the spread of chronic diseases, and improvements in methodologies to indicate disability [2]. These disabilities affect a person's ability to interact with modern computing, as motor disabilities like spinal cord injuries, Parkinson's disease and cerebral palsy severely limit the capacity to use a mouse and keyboard.

In Canada, the disability prevalence rate has increased from 22% in 2017 to 27% in 2022 with 8.0 million Canadians aged 15 years and older having one or more disabilities that limit their daily activities [3]. In individuals with disabilities, the co-existence of multiple disability types leads to significantly increased accessibility challenges. Almost half (42%) of seniors with disabilities reported four or more co-occurring disabilities, while youth (43%) and working-age adults (36%) are most likely to have around two to three disability types [3]. Over 17% of individuals with disabilities in Canada reported barriers to accessibility within online spaces, demonstrating a critical need for easy-to-use accessible technology [3].

Only 62% of Canadian working class with a disability are hired as of 2022[23] this means many are without work due to their new or existing condition. Around 40% of those conditions are physically affecting mobility or basic movements that could be helped by innovation to accessible technologies. By creating more digital access for impaired users, it could allow for new job opportunities for people that don't have access to other careers.

Accessible technology refers to the variety of devices and services targeted at helping people with disabilities and special education/rehabilitation needs to function better within their daily context with higher quality of life [4]. Such technology can be realized in both low-tech and high-tech assortments, depending on the tasks, environments and characteristics of disabilities. By implementing this technology, people with disabilities can help overcome barriers related to employment, reduce the effects of social isolation, and caregiver costs, and even access greater opportunities in education and civic engagement [5]. However, in cases where accessible technology is not implemented well, over one-third of these technologies are discarded, presenting a large drain of human and technical resources. The creation of human-centered accessible technologies is crucial to addressing this lack of sustainability.

Despite progress and growing numbers of these products over the years [6], this is a clear sign that increased attention is being paid to providing inclusive access for individuals with limitations. However, these technologies are limited by financial barriers with prices that do not match the socio-economic situations of individuals with disabilities [7] as accessible technology can range anywhere from $1000 to $10,000. Furthermore, accessible technology for motor impairments is not mobile, requiring bulky installations with eye trackers, and controllers like sip and puffs that cannot be easily uninstalled and reinstalled [9]. Thus, this project aims to address this clear disconnect between solutions in place today and humans in need of support to create an efficient, mobile and easy-to-use solution that reduces the barrier to accessible technology while prioritizing hardware cost efficiency.

Background Information & Literature Review

The landscape for assistive technology is closely linked to the development of computers and motor switches, as with the rise of computing capacity over the past few decades – further advances were made that enabled different forms of human computer interaction (HCI). This literature review will investigate HCI solutions for people with motor disabilities in the realm of eye tracking systems, sip and puff mechanisms and head-controlled interfaces. Each HCI solution specified varies in terms of technical cost, mobility, and dependency on caregivers to install, maintain and setup. This section will analyze the existing state-of-the-art technology and provide background information necessary to understand analysis of alternatives done later in the report.

The area of human-computer interfacing (HCI) is complex and spans multiple domains of study in a multidisciplinary manner as seen in Figure 9. This field investigates the interaction between people and machines and works to optimize computer design and user experience. It started with the advent of modern computing in the 1980s and has grown significantly since with the introduction of computing into the home and office [8]. Pushed forward by the doubling of transistor count and the shrinking of technology, tools were able to be designed for ease of portability while keeping function, allowing great advancements in the field [8]. As the importance of HCI rose, so too did assistive technology follow shortly after with the field of Computer Access Technology (CAT).

Computer Access Technology (CAT) is a subset of HCI and focuses on assistive technologies that enable people with disabilities who struggle to use a standard keyboard and mouse to access the computer or computing system. This paper [9] is an incredible dive into the field and existing state of CAT and demonstrates importance through large number of statistics, and allowed us to clearly define the state and history that led to today's industry. It brings up important insights in terms of how many users of working age face dexterity difficulty or impairments, helping the team realize the importance of our solution. It then goes on to talk on how choosing the most appropriate CAT is a collaborative process involving a consumer, clinician and third-party payers [9]. This is a lengthy process with multiple problem points, often leading to wasted human and material resources. Most alarming is that a third of computer users who receive CAT abandon their received technologies (extending to computers, communication devices and adapted software). The wastage from this process influenced our design process to ideate on ways to improve user adaptability and experience. The paper then goes onto show how the most significant barrier to CAT is cost with 32% of users aware of CAT they would purchase if more affordable [9], establishing a strong economic focus to the design. CAT is often funded through vocational rehabilitation agencies who have budgets that are small compared to the demand for their services, often putting most technologies out of reach of retired peoples and consumers not seeking employment [9]. It cites awareness as the second biggest issue with 20% of users unaware that there is a CAT for them. CAT extends inexpensive devices like trackballs and footprint keyboards, to sophisticated methods like speech recognition, eye gaze tracking and sip & puffs [9].

The field of eye-tracking has come a long way from its conception, much more than other technologies. Previously, peoples had to wear contact lenses, bulky head-mounted apparatuses and even chinrest and bite bars, but today's systems can track eye movements with just an infrared diode and camera [10]. The exact computation and implementation of eye-tracking differs from technology to technology, but human interaction with a computer generally revolves around tracking the large movements of eyeballs called "saccades" between focus points, and smaller movements "micro-saccades" between fixations [10]. Along with frame-based calculations and measurements, it is possible to detect the orientation of eyeballs and triangulate their position on screen. This is used to create an average between the left and right eyeball and place a dot on screen, with this system being open to modification as per a user's needs [10].

An excellent in-depth explanation of the eyeball and how eye movements are tracked and calculated can be found at this paper [11]. This paper also mentions the limitations of eye-tracking, which differ depending on the form of tracking used. Where monitor-based or remote eye trackers, capture the user's gaze in relation to the screen, they are typically affixed in place and cannot be used without the monitor [11]. Where camera-based or wearable eye-trackers require more complex calibration and calculations but do allow the user to freely move their setup and be used with any screen [11]. The distinct disadvantages of eye-tracking are the robustness and accuracy of parts leading to much higher prices that can rise to over $10,000 USD at the highest end, and that infrared illumination may struggle with glasses wearing individuals due to lens reflections [11]. However, with sufficient feature extraction and set parameters, it is possible to circumvent the challenge of glasses reflections as seen in this paper. This model requires no illumination and runs off differences of piece-wise approximation between frame segmentation. [12].

With the advent of IoT, cloud computing and machine learning, we can see an expanded evolution to the previous approach that results an increased in accuracy, increased flexibility and reduction for the need of calibration. This integrates several regression-based approaches and neural networks alongside feature extraction to improve the extent of saccade detection with up to 92% reliability. Poor dataset availability has thus far limited the application of deep learning, but this is a clear indication of the future of state-of-the-art eye-tracking [13].

Other accessible technologies that build up the realm of HCI and CAT include the sip and puff. This consists of a tube inserted into a user's mouth that requires on respiratory inputs to actuate and must be changed regularly to avoid hygienic concerns [14]. However, this system has been labelled as invasive and is often resorted to as a necessity when other motor-impairments and cost comes in the way of accessible non-invasive respiration-based interaction techniques [14]. Sip and puff systems have built in reliability for being switch based systems (switch meaning turn on and off) allowing plug and use functionality with software drivers, but this depends on the availability of said software drivers [14].

Other hardware-based systems that actuate switches include braille keyboards, that allow individuals with disabilities to explore the internet through tactile sense [15]. With heavy integration, these systems function incredibly well but rely on strict compliance with accessible web design principles and cannot be adapted to applications without closer collaboration with software providers. [15]

Scope and Objectives

The scope of the design is to provide accessible options to users via eye tracking to allow control over a conventional computer system. The operating system of choice will be Windows and optimizations will be made specific to Windows to benefit the largest amount of users [16] At this stage of design, users with different operating systems are out of scope, but in the next iterations of development, the system can easily be altered to allow for all computers.

Our UI design must be capable of switching between modes of operation such as general-purpose mode (GPM), and application-specific mode (ASM). To show proof of the concept of creating application-specific software for pre-existing apps Minecraft will be the application of choice. Other apps that would require more than a GPM to run currently don't have implementations.

The design will accommodate users with a lack of motor functions using eye tracking software. Disabilities that cannot be accommodated using eye tracking will not be accounted for. The user must also be capable of producing noticeable pupil movements independent of head movements. This ensures that the eye tracking software can precisely track pupil movements. To expand, people with frequent involuntary muscle movements will have trouble using this system and are also considered out of scope. However, fixes to these issues are addressed for future design iterations in the recommendations. Users are also required to be capable of maintaining a constant distance from the screen or they cannot be accounted for.

Users cannot wear sunglasses, or objects that obstruct the camera's view of pupils. The program will also not be optimized for users that require reading glasses due to factors such as glare and pupil obstruction. The system will have difficulty placing facial landmarks if too much of the face is obstructed.

These factors influence the decision to leave users that require facial obstructions to use technology out of scope for this iteration. Some computer literacy is assumed from the user, to enable the team to use short form functions like "Macro" in our user interface.

Cloud computing, and use of LLMs such as ChatGPT, will be considered out of scope for the AI portion of the project. If users require cloud computing to run ASM or GPM they would defeat the purpose of portability and limit users based on their internet connection. Furthermore, these also rely on continuous upkeep of cloud servers. For this reason, the project will only use prebuilt models that can run locally to prioritize local processing.

Technical objectives are defined in the constraints and criteria and based off broader, overall objectives. The main objective is to elevate the efficiency of accessible computing to allow users to not feel slowed down by their device. A successful device will not impair the user in their efforts to use a computer. This is realized in the form of efficient hardware, software and design with offloaded processing to a secondary computing unit. This will ensure the program does not slow down the user's computer or the user in their regular processes.

There is a distinct emphasis on reducing the barrier to entry for accessible computing systems. This is achieved by making used technology affordable, and flexible by satisfying the needs of many. The design is not meant to replace fully customized computer accessibility technology but is designed to be an easy to adopt and easy to learn eye-tracking experience that will enable users to identify the merits of eye-tracking based accessibility solutions in their vocational rehabilitation process.

Finally, the objective is to create a modular design that will allow for more inputs to be added in the future. This will allow people with specific input needs to have support for any further accessibility devices they receive in the future.

Constraints and Criteria

Some assumptions need to be made in order clearly tackle the problem and exclude hyper-specific cases that will not be concerning this project. Firstly, we can assume that the user will operate the computer at a stationary location. As many of us may require doing last minute studying or writing in transit on a bus or train these scenarios will be considered out of scope. The reason for this is that many accessibility devices do not perform under these volatile conditions, and many people with disabilities have unique transportation situations that do not support computing. This means the program would have to be vastly more robust to please a small percentage of users, which is not the goal of the project. This does not exclude the device from being portable.

Another assumption is that the user's computer can already run their desired software at a desired frame rate. While the software implemented by the solution will have a minimal impact on a computer's processing power, if the user's computer cannot perform regular tasks, then it will be difficult to have an external device communicate through serial buses. It is also not the aim of the project to increase user frame rate, just have no largely negative effects. Therefore, if the user cannot run the program regularly then it will not be considered for testing purposes.

For this stage of development, we are also assuming that the user has a Windows operating system. This is only for the prototyping stage and is to avoid spending unnecessary time porting software between different OS, when it is a task that is known to be achievable.

Table 1 Constraints

Constraints	Justification
Must Comply with data collection laws such as PIPEDA and AIDA	The biggest concern for eye tracking software, and AI powered devices is data collection and storage. These are common when dealing with eye tracking, facial recognition and most human input-based AI. However, by doing so you create unwanted exposure and vulnerabilities for the users. This would result in the project needing additional time to ensure safe storage and collection while following the difficult to navigate data collection laws adding. For these reasons no camera data from the user will be saved to ensure no leakage is possible and that the system abides by PIPEDA and AIDA laws
The cost must be under 1000$	If the cost exceeds this price without additional input systems, they will have been no innovation and simply creating another version of

	existing technology. For a design to be feasible the system must be affordable when compared to existing systems.
Less than 20% performance impairment on applications	The design must not produce a 20% reduction in application performance. The user should not feel hindered when using the software.
Models must not be limited by race, age or facial structure.	If models cannot be configured for everybody, they defeat the purpose of an accessible system.

Table 2 Criteria

Criteria	Justification
Cost-effective	The lower the cost of production the more available this product can become.
Easy to understand software and hardware setup	A design must have an intuitive set up. A successful design should have ease of connection for the hardware side, and an almost fully automated set up for software.
Maximize macro storage and customizability	The system should maximize the use capability to save and re-use macros for defined functions. This will allow each user to get the most out of their system and customize for their needs.
Size	Smaller systems are preferable to allow for portability and to save space at a user's workstation
Solution should be modular and allow for additional inputs	A design with room to improve will allow users existing systems to connect easily, allowing them to use specialized inputs if they require more accommodation. The more flexible a solution the higher desirability.
Minimize performance slowdown	The faster the system can perform the better.
Input should appeal to the broadest range of users	The more people that technology can appeal to, the better.

Design Solution

Design Process

In the original proposal, the group had decided to undergo the IDEO Design Thinking process. A strength of this design process is its fluidity, allowing quick adaptations accordingly as the project changes and molds over time. To take steps in the direction of empathizing, the team performed a thorough problem analysis of the state of the accessible computing industry and was shocked to identify the large cost barriers that prevent individuals from accessing the vast resources provided by the Internet. This is when the goal was defined as making accessible computing significantly cheaper in a human-centered approach. We then performed an engineering adapted ideate process to draw up design ideas and alternatives that was evaluated and can be shown through Appendix B.

Once our hardware design choices were finalized and solutions were evaluated, we prioritized making design choices that would produce the most efficient product within the chosen design. Starting with the Raspberry Pi, a strong priority was ensuring efficient communication between the computer and the PI. If this communication bottlenecked the system, then quick changes needed to be made.

Another factor was ensuring that the model could run efficiently on the Pi while also providing accurate results. This led to a tug of war between user efficiency and computational efficiency where a pivotal decision had to be made. Instead of using eye-gazing calculations to control the mouse the mouse was now altered in direct correlation to the user's pupil. This allowed for better accuracy and increased camera FPS giving the user a better experience.

Model choice also factored in how well it could identify blinks and patterns to allow for key binds to be directly mapped to user patterns such as 4 consecutive left eye winks being open Minecraft overlay. This led to another challenge of reducing the model enough to provide no bottleneck, while considering this extra parameter.

While designing the UI, our primary focus was maximizing user comfortability and ensuring that it was possible to provide all the user needs to operate their PC. Most of the design process was spent testing through different design layouts, identifying key functions and optimizing accessibility. This included producing clever solutions to provide a seamless computing experience. A comprehensive list of the implemented functions and reasoning for implanting them can be found in Appendix E. The Application Specific Mode was targeted towards Minecraft and this overlay focused specifically on speed, and optimizing user performance within the game. Testing loops included ensuring the game could be played using the buttons and achieving smooth gameplay with the layout.

Throughout all design choices and iterations, the user's comfort and experience were put first. We ensured that at the heart of the solution, there would always be a distinct focus on minimizing any frustration while using the design. This meant that whenever major decisions were made, our final call relied on “will this negatively impact the user experience”. This philosophy helped drive most decisions and was included in all thought processes regarding the evolution of the project. This can be seen in how the team recognized the important social aspects of our solution and ensured that our final design was mobile enough to be moved from place to place with ease and not fixed into a single place.

After each design iteration, we would check back with identified safety codes in constraints to ensure that no user data, or safety was compromised. Additionally, further research was conducted at all points and our project implementation changed from our initial memo through to the final design to prioritize compliance with local laws. The main laws that each iteration had to abide by were AIDA which deals with discrimination in AI and mitigation of damages and PIPEDA which deals with consensual data tracking.

Design Tools

Decision matrices helped guide a large portion of design choices in the earliest stages of design. This along with sensitivity analysis was the deciding factor for top level design decisions. These can be seen in Appendix B. Tools that helped develop the design were PyCharm and Visual Studio Code. These helped develop all software related aspects of the project, in languages Python and C respectively. Furthermore, all included libraries, and software was checked to ensure strong and secure code was employed in our design. Our solution includes a debug mode, which was used extensively to enable comparison of speed and comparison between AI models and runtime methods.

Detailed Final Design

The primary input chosen will be eye tracking with Infrared lights. This will cover the widest number of disabilities and physical impairments. The camera and lights are encased in a 3D printed box that can clamp onto the users monitor or laptop screen. The camera is capable of interacting with the external processing unit which is a Raspberry Pi 5. The microprocessor receives eye-tracking data and performs necessary computations to determine which region a user's eyes are looking at and send interrupts to the UI based on user input. These interrupts are necessary in a software program to allow communication with the operating system. The UI will run on the user's computer and pass along interrupts to the operating system. A loop of the system can be seen in Figure 12 and 13 of Appendix E for a visual representation of the process.

The UI is written in C to take minimum computational power to run. The C programming language is known for its severely reduced overhead as the language is very rudimentary with complete control on memory management and resource allocation. This choice was made so that any heavy computations can be offloaded to the microprocessor to avoid slowdown on the user's computer.

A Calibration mode has been implemented to allow the user to calibrate the corners and middle of screen to ensure the eye tracking system is calculating values correctly. Calibration can be triggered whenever the user wishes in case of software or hardware mispredictions or faults. This reset will ensure seamless troubleshooting in case of error.

After a calibration the user is put into general purpose mode and can begin to control their mouse with their eyes. The final general-purpose overlay is shown in Figure 1. This gives a ring around the user's screen and buttons that are sectioned off throughout the overlay. Each button will be mapped to different predefined spots and user defined functions which can be added to meet their needs. Sample functions include an onscreen keyboard, volume control, and access between different overlays. A comprehensive list of implemented functions is shown in Figure 9 of Appendix E. With this design most of the users' screen stays accessible, meaning they are free to interact with their eyes acting as a driver for the mouse across all non-overlay points of the screen. Additionally, functionality for eye-controlled clicks has been implemented to allow for users to click from intentional blinks.

An ASM overlay for Minecraft was implemented as shown in Figure 2. This is an example of a custom user overlay that can be defined to suit the needs of complex programs. Minecraft as an application does not support eye gazing due to Minecraft look inputs being bound to mouse movement. This means that mouse control needs to be disabled for this application and a phantom mouse will be implemented. This phantom mouse will display a dot on the screen and appear and have the same functions as a mouse however will not interact with Minecraft's mouse controls and only be used to navigate the UI. This allows for the game to be fully played through the UI binds and showcases the flexibility of the design.

Finally, the software created by our solution can be used standalone on systems with sufficient processing power and external camera inputs. Software used to calculate eye positions reduces the color channels of non-design cameras to increase processing speed and allow further flexibility.

Design Solution, Evaluation & Alternatives

Using the decision matrices in appendix B. the best design was chosen. This used weighted decisions and also excluded if a constraint was not met. This led to the design choice that was outlined in detailed final design, and the alternatives will be discussed briefly in this section.

Hardware design alternatives had similar structure to the current implementation with different benefits and drawbacks. For example, an Arduino provides the easiest work to code with but has less flexibility and performance than a PI. Having a full software approach negates the majority of costs but requires the user to have very powerful software which ends up costing more than the cost of external hardware. An FPGA was the second option but was scraped due to having limited expertise and struggling to fit within time constraints as 2 people. This decision will be discussed in detail in the change management section of Appendix C.

For input alternatives the team had many options too, but each lacked in different ways. The head mouse was easier to program but covered less disabilities and was outclassed in every evaluation by the eye tracking solution. A SNP system was researched to be inconvenient to clean and have constant maintenance costs along with requiring a more biomedical background to tackle the problem. Finally, the braille keyboard proved to not be as inclusive as other inputs and doesn't allow for much innovation in the sector.

Design Life

Our final solution's design life relies on multiple different factors that need to be considered in union. Firstly, our design places an emphasis on hardware cost-efficiency and power efficiency. When considering the parts used to implement the design, we can see that the design uses relatively standard hardware components. The only part of our implementation that has a reported lifetime is the IR LEDs with 50,000 lifetime hours. The ARM chip on the Raspberry Pi is designed to be used with low power, and only uses 3.3 V to power the LEDs.

When considering our software architecture, the user interface is coded in C to place the lowest impact on the user's system. Additionally, the design does not rely on cloud processing as all processing is done locally, this does not rely on any external services.

Finally, our design is modular and thus allows the system to be used with other accessibility tools like a sip & puff. Amongst all the factors affecting design life, the potential to improve the system with accessibility technology the user already owns drastically improves our design's lifespan and adoption rate.

Design Defense

Function

The final design features a non-invasive implementation of eye-tracking with minimal setup requirements. To run the design, the Raspberry Pi must be plugged into charge and the USB A-A from the Raspberry Pi must be plugged into the user's computer to allow serial communication. This aligns with other accessibility hardware like a Sip and Puff that relies on a single USB-A cable.

To calibrate the eye-tracking, the user will stare at 9 different calibration points that will appear on screen, this will allow the software to establish a coordinate point of reference for where a user's pupil is in relation to where the user is gazing on screen. This solution does not rely on complex, computationally intensive and prone to error gaze estimation and allows the user to navigate their screen efficiently.

Our use of application specific and general-purpose modes allows for customization of capabilities, allowing users to both customize the software to their required needs or make use of other custom solutions if the software for the design is open sourced. This improves how user-friendly the design is and allows for greater usage and potential.

We acknowledge that there is a learning curve to the implementation, and basic computer literacy assumptions have been made such that users should understand technical terms used in the UI like "Maximize" and "Record Macro", the provision of documentation to supplement the design will address this.

Safety

In our design process we ensured that all components chosen to have no electrical risk to the user. The Arducam Camera Module is placed within a 3D printed resin case, and electrical connections to the LEDs are soldered in place. While there is no heat-sink used for the LEDs, there is no risk of overheating due to the limited 3.3V supply of the Raspberry Pi. The Pi itself is placed within an aluminum casing and has active cooling, and a heat-sink attached, allowing it to run large computations without worry of overheating. The Arducam itself has a camera cap, allowing the design to be safely placed away without risk of lens fracture. For data safety the system does not record any data which ensures compliance with PIPEDA ensuring user data is protected.

Economics

The overall cost of the project was a large focal point as our goal was to bring a cheap alternative to the field. Our final design does not surpass the 1,000$ constraint, only totaling roughly 300$. This means the goal of making accessible technology more affordable was achieved and surpassed all expectations. Additionally, the cost is based only on prototype value and can be reduced before large scale production. This large reduction was backed by the decision to design a Pi instead of FPGA, which would end up doubling the cost. This tradeoff between price and efficiency proved to be the right choice as no notable slowdown was noticed while running the Pi at half the cost of the FPGA.

Large scale economic potential of the product could help disabled individuals enter more job markets and improve their computer skills with minimal startup cost requirements. This could enable more disabled individuals to enter previously unavailable markets.

Additionally, reduced power consumption compared to bulky and power-hungry systems will lead to reduction in long term costs for users. The user will only need to provide an additional 2.7 watts [22] to their existing computer, which is assumed to already be within the budget.

Social and Environmental Impact

There are many potential social impacts that can stem from mass accessibility of eye tracking systems. A user who may have been previously unable to access this technology will be introduced to a new way to use their existing technology. With application specific modes and portability of the solution, a person will now have access to more social apps and experiences, which can create new relationships and lifestyles that were unavailable before comfortable computer access. The general-purpose mode of this technology which was targeted towards general computer usage will also enable users to take new career paths to people who may not have many options due to their condition.

The largest environmental impact will come from the reduction in turnover rate for the designed system. In accessible computing one of the greatest challenges is getting users to adapt their technology and learn how to use it. Current solutions have a one-third abandonment rate after purchase [9]. By reducing this turnover rate, the number of metals and components disposed will be reduced dramatically, as well as a reduction in wastage of human resources. Also, if a user is against keeping their system it isn't a hyper-specific system for 1 use, a Pi and camera can be re-purposed for other tasks after its life cycle.

Lastly, by designing an attachment for a system rather than an entire computer the environmental footprint is much less reducing the impact during the production process. While this design does feature 3D resin printed components, these parts have a very small volume and can be recycled individually.

Risks and Uncertainties

By nature of the project, no risks put the user in any danger or physical discomfort. However, the design does have some errors that affect the accuracy of the design. The current model can effectively detect faces and place landmarks on the users' face. The issue arises if the user is continuously adjusting and moving their position, which will make readings unreliable and require frequent recalibration. This poses a risk to the efficiency and validity of design as users will not be content with constant pauses in workflow.

An uncertainty is the model's effectiveness with unique eye shapes, which would need to be tested more in depth before mass production. This also introduces a bias that has been created from testing in-house. Many face tracking models, and algorithms were adjusted to perform best in created use cases, and all tests performed were done on prototypes and not on public peoples. For this reason, the models should be tested on a larger data set to verify effectiveness and remove any unintentional bias.

Recommendations

Recommendations for the future of the project are to address current risks and uncertainties to further develop the design. To address constant user adjustment issues, a more comprehensive calibration phase is required. Instead of calibrating on startup, a user could do one extensive calibration phase on set-up that would allow for use until a drastic change of environment. This test would include different distances, and user positioning to record more dynamic values. Then within the calculation loop, an A.I can estimate distance and offset, to calculate an adjusted pupil reading based on the data and all known calibrations.

As mentioned before, it is also recommended that the product is tested on a much larger scale before progressing to further stages. The tracking model can still be altered easily to fit all needs and allows for the inclusion of more sophisticated tracking in case of unique edge cases. However, if this is not considered before further implementation hardware constraints will cause even more damage the longer this is left unverified.

Conclusions

The final design implementation was successful and fit within all defined constraints. This allowed us to tackle our problem and lower the cost barrier to entry of accessible computing. The final cost came out to roughly 380$ and is projected to be lowered to 200$ if dedicated hardware reductions in later stages of development. This design will have a positive environmental impact, with less waste and product disposal due to its lowered cost and flexible design. The social impacts will allow users to use social media and transport their device which was not possible before. This device will also have economic impact by allowing new members into the workforce that we didn't have access to before. The design will positively impact users and society, by bringing computing to everyone no matter the situation.

References

[1] ‘Share of the population using the Internet’, Our World in Data. Accessed: Feb. 14, 2025. [Online]. Available: https://ourworldindata.org/grapher/share-of-individuals-using-the-internet

[2] ‘World Report on Disability’. Accessed: Feb. 14, 2025. [Online]. Available: https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability

[3] S. C. Government of Canada, ‘The disability rate in Canada increased in 2022’. Accessed: Feb. 14, 2025. [Online]. Available: https://www.statcan.gc.ca/o1/en/plus/5980-disability-rate-canada-increased-2022

[4] G. E. Lancioni, J. Sigafoos, M. F. O’Reilly, and N. N. Singh, *Assistive Technology: Interventions for Individuals with Severe/Profound and Multiple Disabilities*. in Autism and Child Psychopathology Series. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-4229-5.

[5] ‘Empowering Inclusion: Creating Accessible Technology for People with Disabilities’. Accessed: Feb. 14, 2025. [Online]. Available: https://www.techuk.org/resource/empowering-inclusion-creating-accessible-technology-for-people-with-disabilities.html

[6] R. Hoel, ‘Examining the Past, Present, and Future of Assistive Technology’. Accessed: Feb. 14, 2025. [Online]. Available: https://www.accessibility.com/blog/examining-the-past-present-and-future-of-assistive-technology

[7] J. R. S. Writer, ‘Cost concerns still hamper accessibility hardware, says advocate’, GamesIndustry.biz. Accessed: Feb. 14, 2025. [Online]. Available: https://www.gamesindustry.biz/cost-concerns-still-hamper-accessibility-hardware-says-advocate

[8] ‘What is Human-Computer Interaction (HCI)?’, The Interaction Design Foundation. Accessed: Feb. 14, 2025. [Online]. Available: https://www.interaction-design.org/literature/topics/human-computer-interaction

[9] R. Simpson, H. H. Koester, and E. LoPresti, ‘Research in Computer Access Assessment and Intervention’, *Phys. Med. Rehabil. Clin. N. Am.*, vol. 21, no. 1, pp. 15–32, Feb. 2010, doi: 10.1016/j.pmr.2009.07.006.

[10] R. J. Moore and E. F. Churchill, ‘Computer Interaction Analysis: Toward an Empirical Approach to Understanding User Practice and Eye Gaze in GUI-Based Interaction’, *Comput. Support. Coop. Work CSCW*, vol. 20, no. 6, pp. 497–528, Dec. 2011, doi: 10.1007/s10606-011-9142-2.

[11] V. Cantoni and M. Porta, ‘Eye tracking as a computer input and interaction method’, in *Proceedings of the 15th International Conference on Computer Systems and Technologies*, in CompSysTech ’14. New York, NY, USA: Association for Computing Machinery, Jun. 2014, pp. 1–12. doi: 10.1145/2659532.2659592.

[12] A. De Santis and D. Iacoviello, ‘Robust real time eye tracking for computer interface for disabled people’, *Comput. METHODS PROGRAMS Biomed.*, vol. 96, no. 1, pp. 1–11, Oct. 2009, doi: 10.1016/j.cmpb.2009.03.010.

[13] A. F. Klaib, N. O. Alsrehin, W. Y. Melhem, H. O. Bashtawi, and A. A. Magableh, 'Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies', *Expert Syst. Appl.*, vol. 166, p. 114037, Mar. 2021, doi: 10.1016/j.eswa.2020.114037.

[14] C. Esiyok, A. Askin, A. Tosun, and S. Albayrak, 'Software switches: novel hands-free interaction techniques for quadriplegics based on respiration–machine interaction', *Univers. Access Inf. Soc.*, vol. 19, no. 2, pp. 347–359, Jun. 2020, doi: 10.1007/s10209-019-00645-2.

[15] 'A braille detection system based on visuo-tactile sensing', *Measurement*, vol. 247, p. 116827, Apr. 2025, doi: 10.1016/j.measurement.2025.116827.

[16] 'Desktop Operating System Market Share Worldwide', StatCounter Global Stats. Accessed: Feb. 14, 2025. [Online]. Available: https://gs.statcounter.com/os-market-share/desktop/worldwide

[17] 'Understanding Assistive Technologies: What is a Head Mouse System?' Accessed: Feb. 14, 2025. [Online]. Available: https://www.boia.org/blog/understanding-assistive-technologies-what-is-a-head-mouse-system

[18] A. Zolyomi and M. Schmalz, 'Mining for Social Skills: Minecraft in Home and Therapy for Neurodiverse Youth', in *PROCEEDINGS OF THE 50TH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*, T. X. Bui and R. Sprague, Eds., Honolulu: Hicss, 2017, pp. 3391–3400. Accessed: Feb. 14, 2025. [Online]. Available: https://www.webofscience.com/wos/woscc/full-record/WOS:000625875704068

[19] 'iRasptek Starter Kit for Raspberry Pi 5 8GB — 128GB Edition of OS-Bookworm Pre-Installed : Amazon.ca: Electronics'. Accessed: Feb. 14, 2025. [Online]. Available: https://www.amazon.ca/iRasptek-Starter-Kit-Raspberry-Pi/dp/B0CW9M5LFT

[20] 'ArduCam 120fps Global Shutter USB Camera Board 1MP OV9281 UVC, M12 Lens - RobotShop'. Accessed: Feb. 14, 2025. [Online]. Available: https://ca.robotshop.com/products/arducam-120fps-global-shutter-usb-camera-board-1mp-ov9281-uvc-m12-lens

[21] 'Diode | Infrared Emitting | L53F3BT'. Accessed: Feb. 14, 2025. [Online]. Available: https://www.circuitspecialists.com/l53f3bt?gQT=1

[22] Pounder, Les, and Avram Piltch. "Raspberry Pi 5 Review: A New Standard for Makers." Tom's Hardware, 23 Oct. 2023, www.tomshardware.com/reviews/raspberry-pi-5#:~:text=Raspberry%20Pi%205%20Thermal%20and%20Power%20Performance&text=The%20Raspberry%20Pi%205%20is%20the%20hottest%20of%20all%20the,and%20consumes%20around%202.7%20Watts.

Government of Canada, Statistics Canada. "Canadian Survey on Disability, 2017 to 2022." *The Daily* - , 18 Dec. 2023, https://www150.statcan.gc.ca/n1/daily-quotidien/231201/dq231201b-eng.htm

Appendices

Appendix A: Response to feedback

The feedback given from the interim report was taken into account and all suggestions were implemented. For example, in the constraints section the adhere to privacy laws was changed to must abide by PIPEDA and AIDA. These are specific laws describing user consent and data collection and AI us, which gives more clarity and gravity to the statement.

Sections that Dr. Farrow believed to be short or lacking had more attention on them to ensure this time around they were up to the required standard. These include the background information, where emphasis was placed on conclusions found from the readings and research cited. Additionally, some grammatical mistakes were noted and lead to more extensive content review to elevate our report.

Appendix B: Decision matrices and sensitivity analysis

Figure 1: Decision Matrice (Processing Unit)

External Computational Device	Software Required	Size And Portability	Performance Speedup	Customizability	Cost	Score /10
FPGA	10	7	10	10	5	8.7
Microprocessor	10	8	9	9	7	8.6
User Computer	8	10	2	7	10	6
Arduino	10	8	8	7	8	8
Worth %	0.1	0.1	0.4	0.2	0.2	

Figure 2: Input Method

Input	Size	Accessibility	Product Costs	Customizability	Score /10
Braille Keyboard	10	1	9	5	5.5
Head Mouse	10	8	8	9	8.5
Eye Tracking	10	10	8	9	9.1
Eye Tracking + Speech Recognition	10	10	8	10	9.3
SnP	9	10	9	8	9

Worth %	0.1	0.3	0.3	0.3	

Figure 3: Device Sensitivity Analysis

External Computational Device Sensitivity 1	Software Required	Size And Portability	Performance Speedup	Customizability	Cost	Score /10
FPGA	10	7	10	10	5	8.2
Microprocessor	10	8	9	9	7	8.3
User Computer	8	10	2	7	10	6.7
Arduino	10	8	8	7	8	7.7
Worth %	0	0.1	0.3	0.3	0.3	

Figure 4: Device Sensitivity Analysis V2

External Computational Device	Software Required	Size And Portability	Performance Speedup	Customizability	Cost	Score /10
FPGA	10	7	10	10	5	8.85
Microprocessor	10	8	9	9	7	8.6
User Computer	8	10	2	7	10	5.55
Arduino	10	8	8	7	8	7.85
Worth %	0.05	0.05	0.45	0.25	0.2	

Figure 5: Input Sensitivity Analysis

Input	Size	Accessibility	Product Costs	Customizability	Score /10
Braille Keyboard	10	1	9	5	5.2
Head Mouse	10	8	8	9	8.6
Eye Tracking	10	10	8	9	9.4
Eye Tracking + Speech Recognition	10	10	8	10	9.5
SnP	9	10	9	8	9.2
Worth %	0.2	0.4	0.2	0.2	

Figure 6 : Input Sensitivity Analysis V2

Input	Size	Accessibility	Product Costs	Customizability	Score /10
Braille Keyboard	10	1	9	5	7.2
Head Mouse	10	8	8	9	8.5
Eye Tracking	10	10	8	9	8.9
Eye Tracking + Speech Recognition	10	10	8	10	9.0
SnP	9	10	9	8	9.1
Worth %	0.2	0.2	0.5	0.1	0.24

Appendix C: Management

Project Management

A full schedule is listed below:

Week	Focus	Hours allocated	Justification	Development cost
1	research	15	researching components, sources etc. for final design	750
2	proposal	15	working on using research to finish project proposal	750
3	research	15	more research for interim report, hours and project scaled back due to lost member	750
4	research	10	more research for interim report	500
5	interim report	16	work on interim and begin prototyping design	800
6	interim report	16	finish interim	800
7	final design and report	10	scale back workload to focus on midterms and other class projects	750
8	final design and report	10		750
9	final design and report	10		750
10	final design and report	20	scaling up hours as deadline near to create best possible product	1500
11	final design and report	20		1500
12	final design and report	30		1500
total		187		11100

Using the total hour estimation to find cost at roughly 25$/h per entry level engineer it would cost 11378 for development and prototype costs total. The project activity based costing, to account for 15% sensitivity of under or overlogging the project, the overall project cost is estimated at 9700$ to 13500$.

Change Management

The largest factor affecting change management during the lifetime of this project came from dealing with the loss of a project member. This effectively reduced the work-force by 33%. With 3 members, we were able to cater the project towards our specializations of hardware, software and AI. However, with the loss of the hardware member, we needed drastic changes to the project scope. The adaptation process started with asking “how can we reduce the project scope to be manageable by two people”. This led to a large reduction in schedule, recourse allocation and complexity of the project. Certain design choices were no longer feasible as our expertise did not cover this, this resulted in the moving of a binary neural network for voice inputs to project next steps and switching from overlaying video streams to a low-weight UI that runs on the user’s computer.

Appendix D: Cost

Engineering Fees:

Roughly 11000$ were used on engineering development cost as discussed in the project schedule

Prototyping materials:

Raspberry Pi 5 over a Development board FPGA + MPU SoC – Pynq Z1 : $439.12 to $235.99 as we bought a development kit but this can be reduced to just the Pi itself [19]

ArduCam 120fps Global Shutter USB Camera Board 1MP OV9281 UVC, M12 Lens over a – See3CAM_CU27 : $73.51 [20]

Infrared diode – 0.78$ [21]

Estimated subtotal – $310.28

Estimated GST – $37.23

Estimated prototype total – $347.51

Estimated total - $11378

The reduction in cost can be attributed directly to losing our hardware expert who would handle the integration of the FPGA development board, binary neural network offloading and co-processor development.

Appendix E: Additional Figures

Figure 7: General Purpose Mode

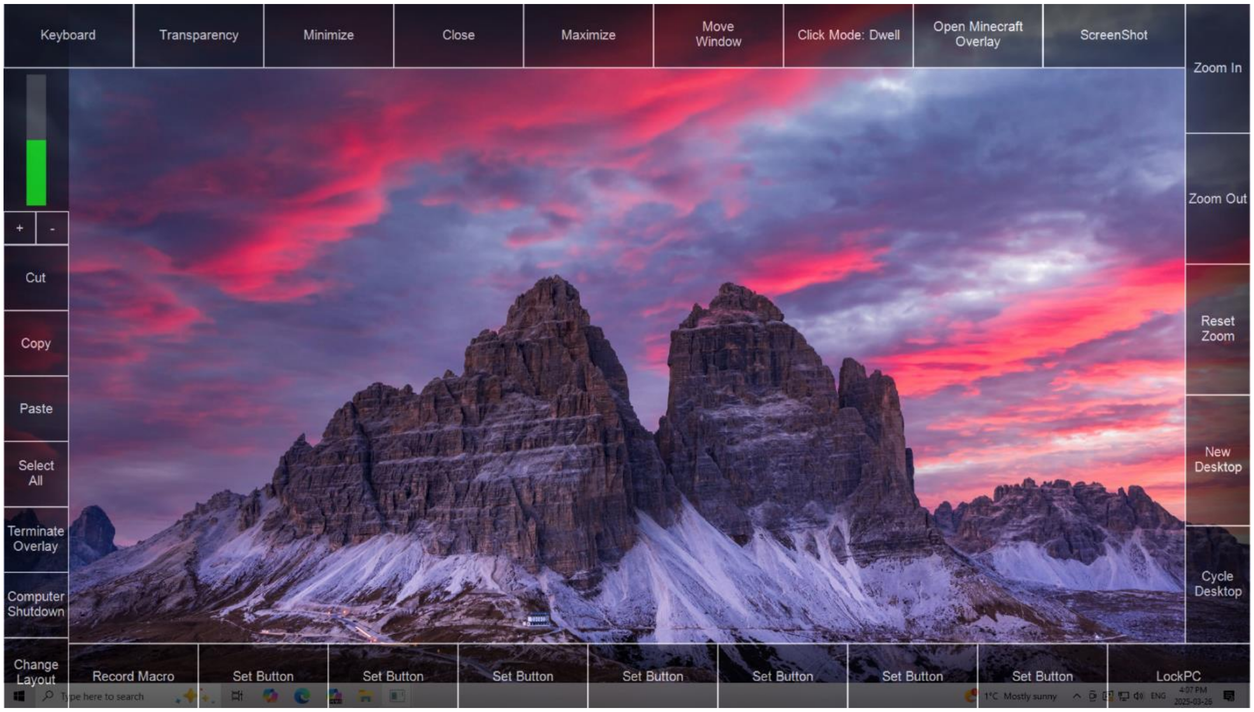


Figure 8: Application Specific Mode

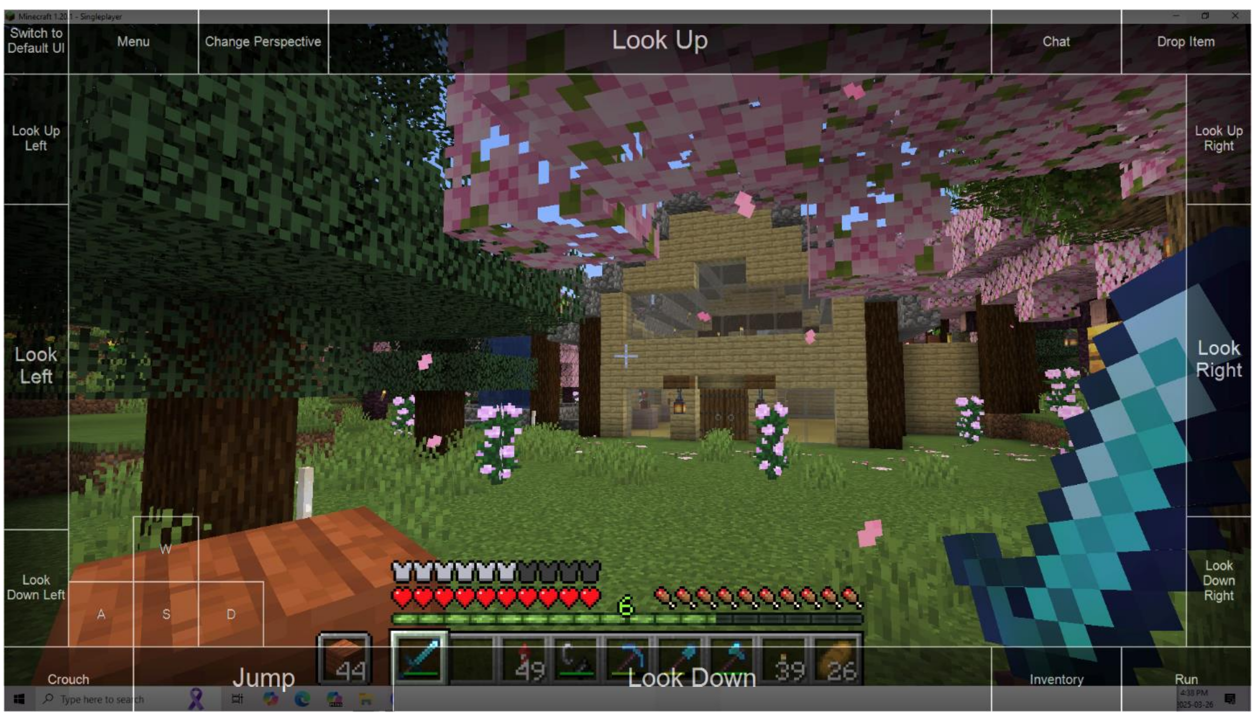


Figure 9: Summation of general purpose features used

Button name	Functionality	Justification
Keyboard	launches onscreen keyboard that can be used to allow user to type in a selected	Typing is a large part of everyday computer use, and necessary for computer literacy
Cut / Copy / Paste / Select All	Simple text management buttons that fill their respective roles	Text editing and manipulation is an essential part of computing. This will allow efficiency improvements for the user.
Volume Control	Turns the computers overall volume setting up or down respectively	Accessing volume sliders and audio management can be a hassle when the overlay is covering the toolbar. This reduces the time to access a common function.
Transparency mode	Allows user to temporarily disable the overlay and turn back on by looking at the top right.	This will be useful for watching media to allow for less screen clutter.
Minimize / maximize / close / move window	Simple window management functions	These window functions could cause issues when the overlay is commonly blocking browser headers. This allows for common adjustment functions to be easily accessed.
New desktop / cycle desktop	Simple desktop management functions	Allows access to more workspaces, can have a gaming and work desktop open and accessible at all times.
Click mode	Cycles through click modes to allow user to control with blinks, long dwelling etc.	Needs an easy way to change inputs on the fly.
Minecraft overlay	Allows access to the designed application specific mode, replacing buttons with Minecraft related hotkeys.	This will be replaced by a chose overlay button in later implementations that will enable the user to decide which overlay to open.
Terminate overlay / lock PC / Shutdown	Exiting functions	User should have easy access to lock or shut down computer if they need to quickly shut down

screenshot	Takes screen shot of entire screen	Common function that can be tricky to navigate to without access to keyboard and mouse
Change layout / Record macro	Change layout will help the user modify the overlay to their needs, record macro will work with the eye tracking device to make a button directly mapped to an action	Customizability enables the user to feel more comfortable and satisfied with the product often leading to better efficacy on their end

Figure 10: Field of Human Computer Interaction

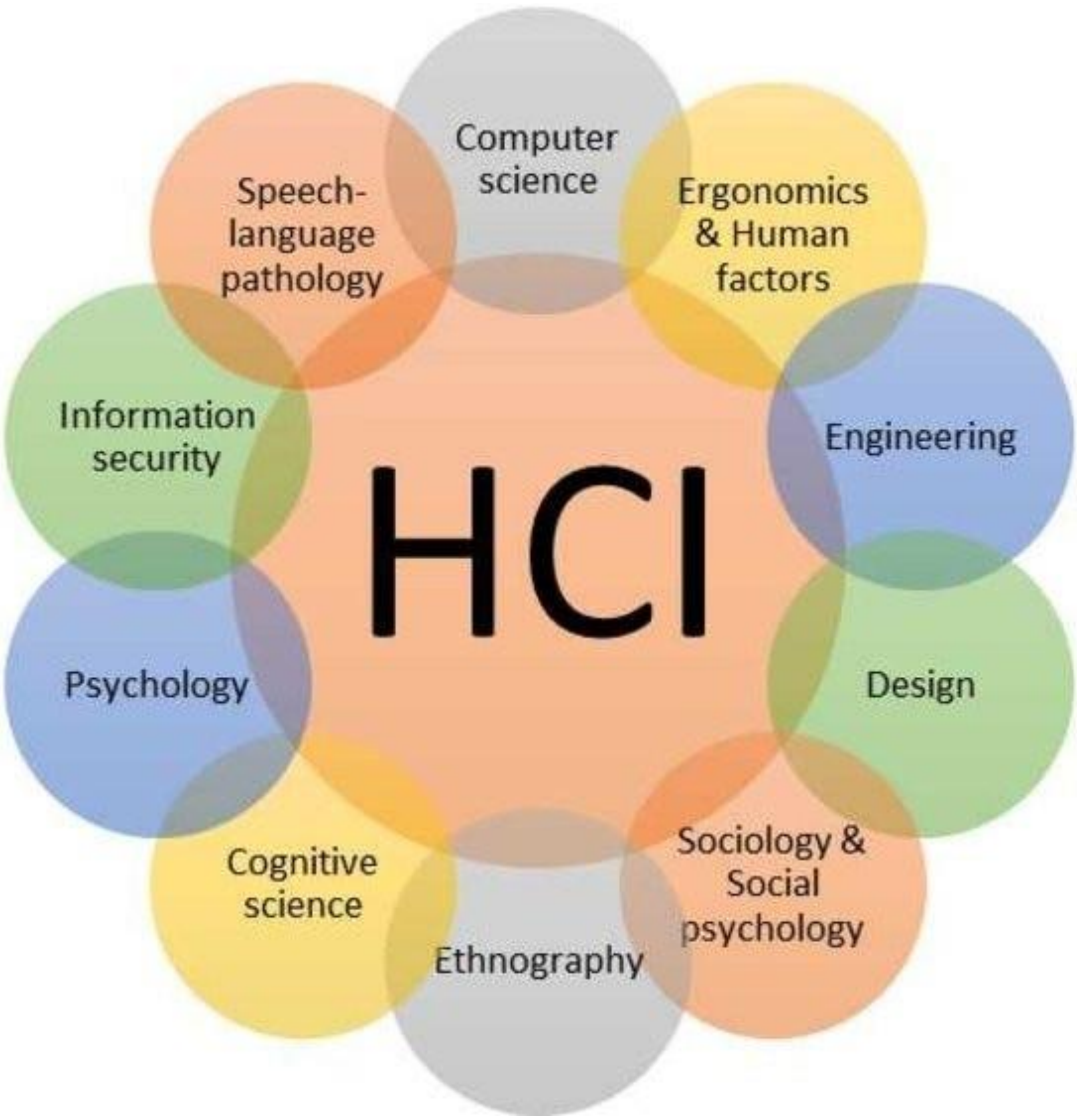


Figure 11: Control Flow of Software

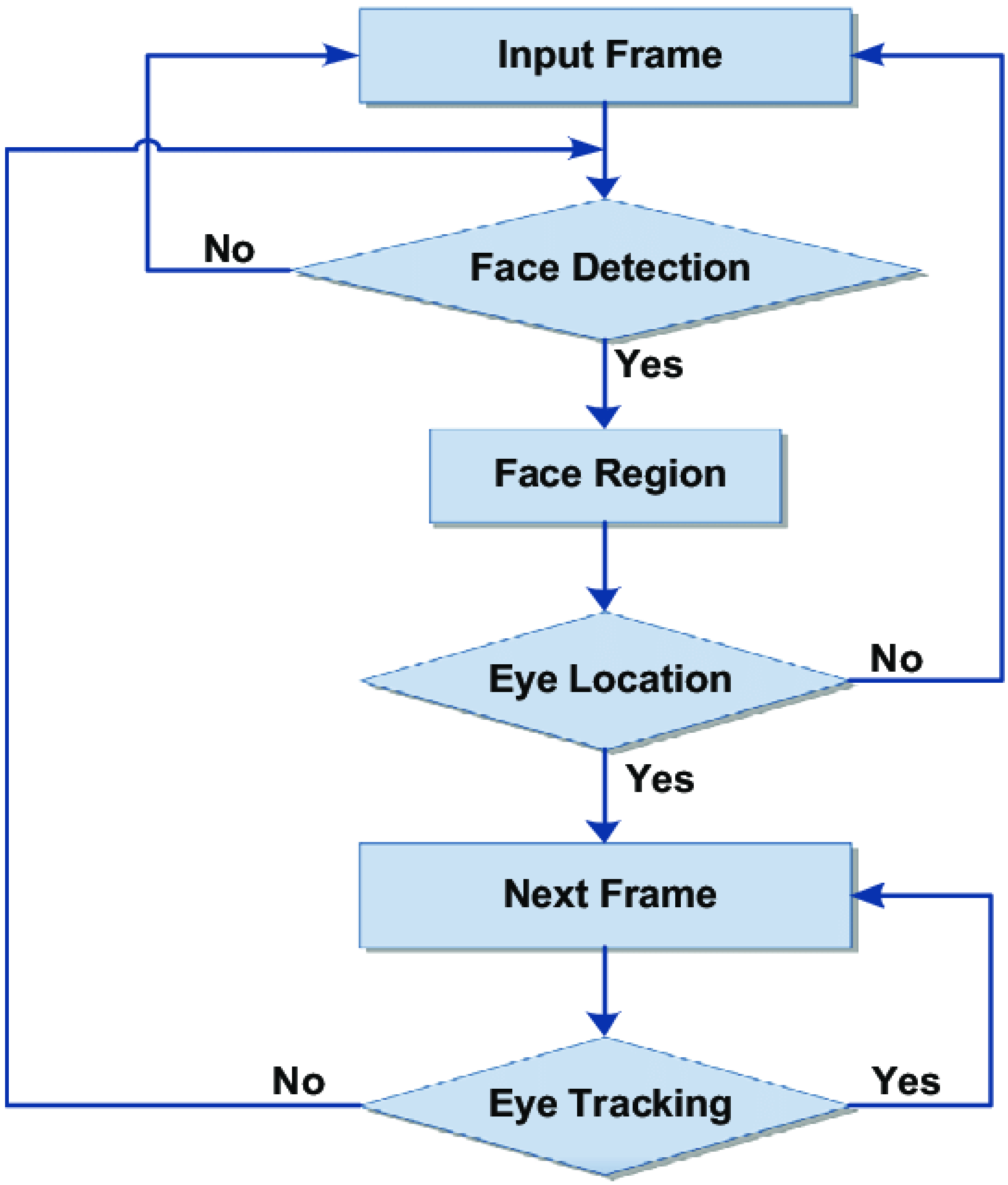


Figure 12: Eye tracking employed

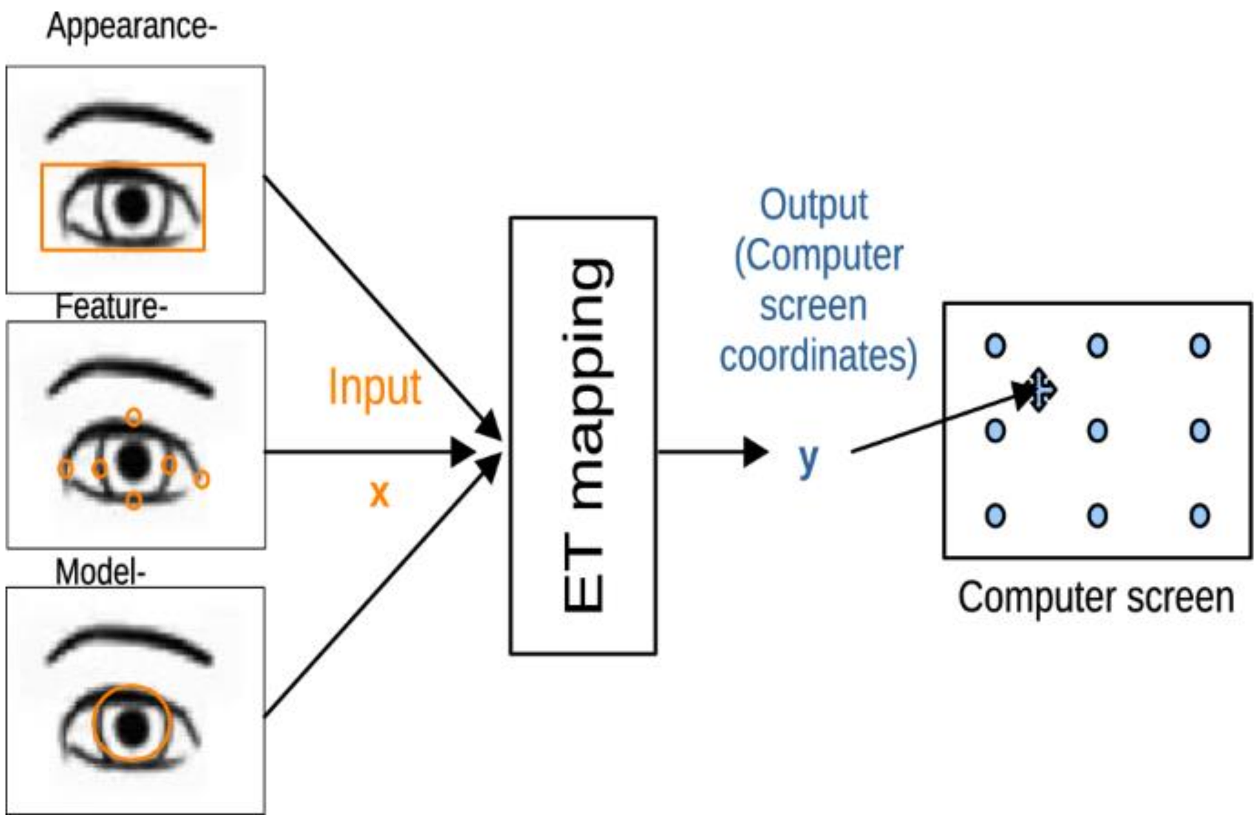


Figure 13: Final Design Poster