ABSTRACT

This project proposes a camera-based mouse system to enhance computer accessibility for individuals with limited hand mobility. The system was designed to enable the empowerment of users, giving them the ability to manipulate the mouse cursor and to do common actions such as click and scroll without having to use a traditional mouse or touch-pad. Instead, it relies on head pose estimation techniques for detecting the direction and movement of the user's head and facial gestures to execute these tasks. Continuous monitoring of the user's head position and orientation is ensured by the use of real-time facial detection and tracking. This data is translated to cursor movements on the screen. To evaluate the system's performance and usability, we conducted various tests using Pygame-based simulations, custom Python scripts, and CPSTest to measure cursor responsiveness and click accuracy. Additionally, real-world testing was performed with three participants of different ages and genders, assessing the system's adaptability across users. For users who find it challenging or impossible to accomplish tasks with a traditional computer mouse or touch-pad this new type of input may offer a perfect solution which will help them to achieve their goals without depending on others.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION TO THE PROJECT

The past few years have really seen the development of better assistive technologies, which, in turn, have improved quality of life for people having limited hand mobility. As a result, this aspect has been driven by improvements in innovative alternative input methods for computers and devices. The project for ”Camera Interfaced Mouse Based on Head Pose Estimation for Individuals with Limited Hand Mobility” shall be developed in such a manner that an intuitive and very effective solution will be formulated for enabling the interaction by individuals who are physically disadvantaged to work with a computer or digital device without actually relying on the conventional, hand-operated input device, which may be that of a mouse or that of a touch-pad. This is based on the idea of merging computer vision and head pose estimation to develop a mouse interface that can understand head movements as input signals. In the system, through using a camera, it captures real-time images of a user's face, with the focus being on concentration particularly on head orientation and position. These images are assessed by pose estimation algorithms, which detect head angles and changes of direction and translate to mouse movements on the screen. Additionally, the interface adds gesture recognition for mouse clicking, so it fully renders an entirely hands-free interface. For people affected with spinal cord injuries, muscular dystrophy, or ALS, it is too hard or even impossible to make use of conventional computer input devices. While many sorts of assistive technologies are indeed available, they tend to be pricey and not simple at all, and also quite varied. Head tracking and pose estimation work based on fairly simple hardware and strong algorithms, which makes them an attractive solution for the creation of a trustworthy user-friendly interface. This project is motivated by a commitment to developing inexpensive, accessible solutions that empower individuals with limited hand mobility to navigate digital environments independently and seamlessly.

Chapter 2

LITERATURE SURVEY

A number of research studies have focused on alternative methods of interaction between humans and computers or mobile devices, with several concluding by highlighting key issues and potential areas for further research.

2.1 HUMAN INTERACTION WITH PC AND ALGORITHM

In recent research by M.N. Islam et al. (2021),[1] alternative methods of interaction between humans and smartphones were explored, focusing on hands-free systems. In this paper we got introduced a new system that uses nose and teeth movements for controlling smartphone functions. The proposed Nose-Movement Interface (NMI) system enabling individuals with hand disabilities to perform gesture and global operations without needing to use their hands or fingers on their own. In the proposed system, the movement of the cursor is provided through nose tracking. Here, the mouse cursor is moved on to the screen without using fingers. Facial feature tracking enables the nose movements of users to be tracked in real time. Nose detection of the user is achieved in this process by the Viola-Jones algorithm [2]. This is a learning machine that uses classifiers. It helps in detecting images, this time a nose, where the algorithm's efficiency during real-time facial detection takes place through it being extremely fast to pass images along due to certain features implemented, including features such as Haar-like features when classifying any area. After the nose is detected, the Lukas-Kanade method [3] is then used to track the nose. It is one of the optical flow techniques that track movement between two successive frames of a video by observing the displacement of pixels. Continuous tracking of nose movement in real-time applications like this is very important to have smooth control over the cursor. The teeth are detected with the use of the Haar Cascade classifier [4]. The pre-trained version of the algorithm of Viola-Jones is applied specifically, trained to recognize any kind of object that contains facial characteristics, including teeth. Certain actions may be executed during the recognition process of teeth, for instance clicks and other activities.

The Viola-Jones algorithm is a popular algorithm associated with real-time performance for nose detection in face-detection systems. The usage of this algorithm has led to its widespread usage: it is fast but highly prone to accuracy loss, mostly when applied to the subtle facial features, thus restricting application in the detection of nuanced movements or fine details in the face. Therefore, the Lukas-Kanade method will be used for the task of real-time optical flow tracking purposes of facial landmarks for efficiency during tracking. However, its performances may degrade when involving rapid movements or large ranges, resulting in loss and inaccuracies during tracking. The Haar Cascade Classifier is very efficient in the detection of eyes or nose like features, although it may become complex when implemented in the challenging lighting and angle conditions.

In the literature of Rahib H. Abiyev and Murat Arslan, a vision-based head mouse control system [5] has been developed specially for individuals with impaired spinal cord injuries, helping them to use a computer through head movements and eye blinks.The research so far has focused on sensor-based solutions for head mouse systems, and the drawback is that most of them have high costs, cause discomfort, and limit future applicability. By analyzing these challenges, the authors proposed a system without wearable hardware to ensure more comfort and easy accessibility for users. Using CNN [6] tracking of head direction and eye state from the images captured by the camera is ensured. It helps to reduce the complexity of preprocessing without losing high accuracy in the management of mouse pointers without extra hardware.

The system includes cost-effectiveness and comfort in its aspect, bringing out an attractiveness to users pursuing having an accessible way of inputting. It's high accuracy in the sense of detecting head movement with regard to the conversion into the actions of a cursor essentially needed for an experience no-lagging. This however might perform erratically depending on lighting conditions and different head positions, maybe making it less accurate. Further, the solution needs to give more than just high accuracy for ensuring results. In addition to the above considerations, factors like user experience and optimal positioning of the camera are of major importance to the general efficiency of the system. All these factors have to be well weighed so that the solution can be not only practical but also user-friendly in real applications.

One of the assistive technologies in the field for individuals with limited hand mobility is the Touchless Head-Control (THC) system [7] developed by Wahyu Rahmaniar, Alfian Ma'arif, and Ting-Lan Lin, through which devices are controlled by head movements. This system uses an RGB camera to track head gestures so that the users can move a mouse cursor or control robotic devices without additional hardware such as wearable sensors. It has a fast calibration procedure to set a neutral head position so users can use turn control functions on or off by making specific head movements. Experiments demonstrated that the THC system was in real-time, and head movements were converted into actions without creating unwanted movements.

However, some limitations are found that need to be overcome for greater adoption and practical use. The system uses an RGB camera for gesture tracking, but the system is prone to inconsistent lighting conditions and a complex background in the environment, which may affect the recognition of gestures. The quick calibration setup is user-friendly but creates dependency on the maintenance of consistent head position, which in case of significant shifting necessitates frequent recalibration, thereby disrupting the interaction flow. Also, its system is only lightly tested to mean that its validity, across a number of distinct user groups and various contexts of use, should have to be more ascertained in terms of widespread effectiveness. In its promised real-time capability, such detection accuracy can prove to ensure consistent performance in being able to distinguish between intended and unintended motions for the assurance of delivery in terms of reliable end-user experience. Future directions towards development shall focus on strengthening the design to weather any number of environmental conditions, refinement of gesture recognition algorithms, and testing in various settings to demonstrate its efficacy.

Virtual Mouse Operation Using Webcam [8] by Dr. K. M. Rayudu, Dr. M. Narendra, P. Anitha, Dr. T. Siva Ratna Sai, and M. Sandeep is a new approach towards controlling the computer mouse with head movements and eye blinks using the webcam. This system uses face detection and eye tracking techniques by using OpenCV and dlib for identifying critical facial landmarks, like the eyes and nose which will be captured using the webcam of the computer, to guide cursor movement. In this system, to make it totally hands-free, left click is simulated by a left eye blink and a right click by a right eye blink. Then, the system employs gaze detection for identifying what is focused by the user on the screen, enabling more precise control. To automate the movement of cursors, the system has utilized Python scripts integrated with PyAutoGUI, so that the inputs are translated into real-time cursor actions.

Despite being innovative in its approach, the system has some limitations that have to be addressed for wider usability. However, the actual performance of the system in real time is a great feature that should be accompanied by the exact movement of the cursor with the head movement and eye blinks without any delay to ensure smooth user experience. The Python scripts for personalization make the system flexible for changes in processing head movements, eye blinks, and gaze tracking, which may make it more adaptable to the needs of different users or changing environmental conditions. Still, sensitivity towards light is one major challenge because variations in ambient light conditions within the surroundings affect the accuracies of algorithms used to detect faces and track the eyes, hence feasible even in the event of failures of tracker or cursor movement without wanting. In addition to this, using PyAutoGUI may also impact the functioning of the cursor automation program as sometimes when to conduct rapid or complex mouse actions, the cursor operation becomes a bit delayed that would be irritating. In highly demanding systems where quick cursors can be needed while having less space, it can become critical issues.

This paper, ”Face as Mouse Through Visual Face Tracking” [9] by Jilin Tu, Hai Tao, and Thomas Huang, presents an innovative system of control for a computer without hands from visual face tracking. The system involves a 3D face model that learns to adapt 2D facial features and provides the use of the extended Kalman filter to extend the tracking process through non-linearity for real-time cursor manipulation in three modes-direct, joystick, and differential. Of these, joystick mode finds the middle ground for the user, optimizing navigation and readability. Beyond that, the system even contains mouth motion detection to activate mouse clicks by enabling head movements for cursor movements and facial expressions for direct user input. Standard web cameras can be used by this, doing away with the need for proprietary hardware, and a recovery mechanism makes sure it continues operating, even during brief interruptions. This system is very versatile by offering troubleshooting options and drag-and-drop functionality.

Practical usage of the system involves some limitations. It is based on standard cameras and very sensitive to lighting conditions. Therefore, if the lighting condition is poor or unstable, tracking may not go as planned. The 8.7-second reset time will be useful in the recovery process but may easily disrupt workflow and introduce observable delays. Additionally, neck strain might be an issue with new users as a consequence of the persistent movement of the head when operating with the mouse. Ambient light dependency, lacking support for features of double-click and click-and-hold makes it not too effective, and adaptation for an uncontrolled environment with regard to ambient light dependency is quite challenging. This could be an improvement about adaptive algorithms against light changes, shorter time reset, and higher features related to interaction.

The paper ”Allowing for Secure and Accessible Authentication for Individuals with Disabilities of Dexterity” [10] by Abbie Price and Fernando Loizides proposes an innovative eye-tracking solution for secure PIN entry, which is an alternative to input methods, especially people with limited dexterity. The prototype is based on the Pupil Labs eye-tracking headset, with a dual eye camera and world camera, connected to a computer via USB 3.0. It captures data on eye movement using the Pupil Capture software with some plugins, such as ’Fixation Detector’ and ’Surface Tracker,’ in order to improve data quality. The gaze data was processed using the Pupil network API with ZeroMQ messaging for real-time data transfer. Key metrics such as fixation confidence and gaze coordinates are normalized and mapped to the display to correctly detect where the user is looking on the screen. The system further secures it by using PIN handling through bcrypt hashing and verification that adds a protection layer during the authentication process.

It was a pilot study where the usability of the new input method vis-à-vis traditional was compared along with security assessment. In some cases like over the shoulder attacks wherein one person is in a covert position trying to view over the shoulder an unauthorized viewer could potentially take a video of what someone else is writing. It tracked the gaze of the user which would therefore allow the person to input hands-free his/her PIN that is a more secure option for users, who had difficulties typing conventionally because of conventional keyboard or touching screens. The security aspects in the system, with bcrypt hashing handling secure PINs, meets a significant demand to protect sensitive data during authentication. Hashing and verification mechanisms ensure that even in the event of interception of data, it is encrypted and hard to decrypt, thus adding an important security layer, especially for those with disabilities, who could be more susceptible to such breaches. The environmental factor that impacts the effectiveness of the eye-tracking solution is lighting conditions, which may cause screen glare. Ambient light variations or screen reflections may cause a failure in tracking or can fail to locate the required accuracy of the user's gaze, and therefore give incorrect input. Such a dependency on specialized hardware, like the Pupil Labs eye-tracking headset, necessitates further deployment needs that restrict accessibility to the system due to the expense or unavailability of such equipment.

”Real-Time Facial Expression Recognition Based on Edge Computing”[11] by Jiannan Yang, Tiantian Qian, Fan Zhang, and Samee U. Khan. . A significant contribution here was the demonstration of real benefits derived from the usage of such architecture and efficiency enhancements as compared to traditional computers' responses in nearly one-third of these applications, most of which would be used in real time. Edge computing also saves costs by minimizing data upload to the cloud and employs devices like Raspberry Pi that are relatively cheap, cutting down on operational costs. The architecture minimizes the use of bandwidth since exchange with cloud servers is only minimal, which is advantageous in areas with low connectivity. The use of Docker containers for packaging algorithms facilitates easy deployment and scaling.

Edge computing, however has challenges in its deployment. Limited computational resources of edge devices may limit the complexity of algorithms, and performance may vary under challenging conditions, which may lead to inconsistencies. The quality and characteristics of the edge devices employed also affect the overall system performance. Additionally, scalability issues may arise when managing a growing number of edge devices, and some users may find it difficult to obtain the technical know-how needed to configure Docker containers. In conclusion, this analysis highlights how edge computing can improve real-time applications while also pointing out the issues that need to be resolved for this technology to be properly used.

The paper by Erik Murphy-Chutorian and Mohan Manubhai Trivedi [12], ”Head Pose Estimation in Computer Vision: A Survey,” has cataloged head pose estimation techniques broadly into basic approaches rather than functional domains. This structure lends itself to more explanatory detail by methods such as Appearance Template, Detector Array, and Nonlinear Regression based upon their implementation technique and limit. One of the key issues mentioned is variation in appearance and lighting, background that greatly affects the reliability of the head pose estimation in realistic applications. The paper thus demonstrates how new datasets facilitate more accurate evaluations under dynamic conditions, and although improvements in accuracy in detection are encouraging, challenges persist, particularly at maintaining accuracy over the space of head positions and distances.

2.2 EXISTING SYSTEMS

App or Hardware Device	**Features**	**Limitations**	**Reference**
EVA Facial Mouse Pro	Face movement controls cursor movement; includes a UI menu for gestures. Cursor clicks automatically if stationary for a short period.	Time-consuming, and automatic clicking may cause accidental clicks.	[13]
GlassOuse	Bluetooth mouse worn like glasses; cursor controlled by head movements. Click performed by biting on a mouth switch.	Very expensive	[14]
QuhaZono 2	Cursor controlled by head movements; multiple wearing options for head and limb attachment. Gesture activation for cursor pausing, scrolling, etc.	Very expensive	[15]

Table 2.1: Existing systems

Chapter 3

PROBLEM STATEMENT AND OBJECTIVES

”Develop and test a camera-controlled mouse system that is easy to use and that can effectively serve as a replacement for the traditional computer mouses for individuals with limited or no use of their arms and hands”

3.1 OBJECTIVE OF THE PROJECT

To develop a camera-interfaced mouse based on Head pose estimation , enabling hands-free computer control for individuals facing hand disabilities addressing lighting issues for reliable performance.

Chapter 4

PROPOSED SOLUTION

To enable hands-free mouse control, this project integrates various computer vision and automation techniques. One of the core solutions is facial landmark detection, implemented using MediaPipe's Face Mesh model. This deep-learning-based system detects 468 facial landmarks in real time, making it highly effective for head pose estimation. Within this project, the nose tip is tracked to control cursor movement, eye blinks are detected to perform right-click actions, and mouth opening triggers left-clicks. Another key component of the system is video processing, which is handled using OpenCV. OpenCV captures the live webcam feed and processes it in real time. It is responsible for converting images to the appropriate colour format required by MediaPipe, as well as displaying useful debugging information such as frame rate, facial landmarks, and head movement boundaries. To enhance cursor stability, the system employs a smoothing technique using a deque (double-ended queue) from Python's Collections module. This method reduces jitter by maintaining a buffer of the last three detected cursor positions and averaging them, resulting in smoother movement. Without this mechanism, the cursor could become unstable due to natural variations in facial tracking. Mapping head movements to screen coordinates presents another challenge, which is solved using a custom mapping function. Since the detected range of nose movements does not directly match the screen dimensions, this function transforms the input data to fit the display properly. Additionally, sensitivity multipliers allow for fine-tuning the cursor's responsiveness based on user preferences. Optimizing performance is crucial for maintaining smooth tracking. To achieve this, the system implements frame skipping, where it processes one frame every two cycles instead of every single frame. This significantly reduces computational load while preserving an adequate tracking frame rate, ensuring an efficient balance between accuracy and performance. To maintain reliable performance under different lighting conditions, the project incorporates multiple techniques to improve robustness. Implemented IR-based tracking instead of visible light cameras to mitigate lighting variations. A light intensity sensor measures the ambient light, below a set threshold value the program switches on the IR LED arrays for smooth functioning.

Chapter 5

FEASIBILITY STUDY AND DESIGN METHODOLOGY

A feasibility study is essential for evaluating the budget, hardware, and software aspects of a project to ensure its practicality and efficiency. Budget feasibility helps determine cost-effective solutions without compromising performance. Hardware feasibility involves selecting components that meet processing power, memory, and compatibility requirements while staying within budget. Software feasibility focuses on choosing the right programming languages, libraries, and frameworks to ensure efficient development and scalability. Conducting this study enables informed decision-making, minimizes risks, and ensures the project is both technically and financially viable.

5.1 BUDGET FEASIBILITY STUDY

We conducted a budget feasibility study to evaluate the financial implications of our Camera Interfaced Mouse system, focusing on both low-cost and high-cost options. The low-cost feasibility study identified affordable alternatives that ensured performance would not be compromised while staying within budget constraints. In contrast, the high-cost option, which included premium solutions, was deemed excessive for our project's scope. Ultimately, we chose the low-cost approach, allowing us to efficiently allocate resources and achieve our project objectives without incurring significant financial burdens.

Component	Low-end Cost (₹)	High-end Cost (₹)	Notes
Single Board Computer	2,500	6,500	Radxa Zero 3 W or Raspberry Pi Zero 2 W
No-IR Camera Module	800	1,500	Raspberry Pi Camera or alternatives
Proximity Sensor	100	250	Infrared or Ultrasonic
Light Intensity Sensor	50	150	Photodiode/Phototransistor
IR Emitter Array	100	300	Simple IR LEDs
Battery	500	1,200	2000mAh - 5000mAh Li-Po/Li-Ion
Battery Management System	150	300	
Miscellaneous Components	300	500	Cables, Connectors, GPIO Pins, etc.
Enclosure/Case	200	500	3D Printed or Generic
Subtotal	4,700	11,200	
Shipping	200	500	Depends on component sources
Tools (Optional)	1,000	3,000	If not already owned
Total (without optional)	4,900	11,700	
Total (with optional)	5,900	14,700	

Figure 5.1: Budget of the Project

5.2 HARDWARE FEASIBILITY STUDY

We conducted a feasibility study on the hardware options suitable for our project, focusing on three processors: Raspberry Pi Zero 2 W, Radxa Zero 3 W, and Orange Pi Zero 2 W. One of our primary objectives was to develop a cost-effective device to make it more accessible to individuals in need. Based on this criterion, we initially selected the Raspberry Pi Zero 2 W due to its affordability and strong community support. However, due to its unavailability, we seamlessly transitioned to Radxa Zero 3 W, which offered better performance and comparable feasibility. This feasibility study proved invaluable, as it allowed us to quickly adapt to hardware constraints by having well-evaluated alternatives ready for immediate selection.

Feature	Raspberry Pi Zero 2 W	Radxa Zero 3 W	Orange Pi Zero 2 W
Processor	Quad-core ARM Cortex-A53, 1.0 GHz	Quad-core ARM Cortex-A55, 1.6 GHz	Quad-core ARM Cortex-A53, 1.5 GHz
	512 MB LPDDR2	1-8 GB LPDDR4	512 MB/1 GB LPDDR3
Storage	microSD	microSD, eMMC (up to 64 GB)	microSD
	VideoCore IV	Mali-G52	Mali-450
Connectivity	Wi-Fi 4, Bluetooth 4.2	Wi-Fi 4, Bluetooth 5.0	Wi-Fi 5, Bluetooth 4.2
	Mini HDMI, USB OTG, GPIO	Micro HDMI, USB 2.0 OTG, USB 3.0 Host, GPIO	HDMI, USB 2.0 OTG, GPIO
Power Consumption	~2.2W under load	Slightly higher due to more powerful CPU	Similar to Pi Zero 2 W, higher at peak due to Wi-Fi 5
	65 x 30 mm	65 x 30 mm	60 x 30 mm
OS Compatibility	Raspberry Pi OS, Linux distributions	Debian, Ubuntu	Linux distributions, Orange Pi OS
	Raspberry Pi camera via CSI	CSI (similar to Raspberry Pi's)	Supports cameras via GPIO (less documented)
Price	₹ 2000	₹ 3000	₹ 2800

Figure 5.2: Hardware Feasibility Study

5.3 SOFTWARE FEASIBILITY STUDY

As a part of the study a comparison between C++ and Python have been performed to determine the most suitable programming language for our project. Given the real-time processing requirements of head pose estimation, we analyzed factors such as execution speed, ease of development, library support, and hardware compatibility. While C++ offers superior performance and lower latency, Python provides extensive libraries (OpenCV, ONNX Runtime) and faster development cycles, making it ideal for implementing computer vision and machine learning tasks. Since our project prioritizes rapid prototyping and integration with deep learning

Feature	C++	Python
Performance	Compiled language, faster execution.	Interpreted language, slower execution.
	Ideal for real-time and resource-constrained environments.	Better for prototyping, less suitable for performance-critical tasks.
Memory Management	Manual memory management, which can optimize performance.	Automatic garbage collection (simpler but less efficient).
	Can be complex and error-prone (e.g., memory leaks).	More memory overhead.
Control over Hardware	Provides direct access to hardware and system resources.	Limited direct hardware control.
	Better suited for low-level optimizations.	Usually relies on high-level libraries like OpenCV, making it less efficient.
Library Support	Extensive libraries like OpenCV and CUDA for high-performance applications.	OpenCV is widely used in Python but may not be as fast as in C++.
	Strong integration with hardware-accelerated libraries.	More flexible for beginners and rapid development.
Development Speed	Slower to develop due to complexity of C++ syntax and lower-level programming.	Fast development and prototyping due to Python's simplicity and high-level abstractions.
Learning Curve	Steeper learning curve.	Easier to learn and code.
	Requires understanding of pointers, memory management, and compilation.	Suitable for beginners and non-experts.
Portability	C++ code can be more complex to port across different platforms.	Python is highly portable and cross-platform.
Community and Ecosystem	Strong community, especially in performance-critical domains like gaming, embedded systems, and real-time applications.	Large and active community in machine learning and AI, with abundant resources for rapid development.
Use Cases	High-performance computer vision tasks.	Prototyping, education, and non-real-time applications.
	Robotics, embedded systems, real-time applications.	Ideal for data processing, analysis, and quick iterations.

Figure 5.3: Software Feasibility Study

models, we chose Python as the primary language. However, in the future, C++ can be utilized for optimized performance, lower resource consumption, and improved real-time processing, especially when deploying on embedded hardware with limited computational power. This study was crucial in ensuring an efficient development process while balancing performance and ease of implementation.

5.4 SYSTEM DESIGN

This project involves designing a camera-interfaced mouse system that uses head pose estimation techniques to enable hands-free interaction for individuals with limited hand mobility. It integrates multiple advanced computer vision frameworks and development tools to achieve real-time tracking of head movements.One of the core components of the system is MediaPipe Head Pose Detection, a powerful and lightweight deep-learning-based solution for extracting facial landmarks and estimating head orientation. MediaPipe provides a computationally efficient pipeline that enables robust performance, making it well-suited for real-time applications, even on low-power hardware. By leveraging MediaPipe, the system can accurately track key facial features such as the nose, eyes, and mouth, which are crucial for determining head pose angles (yaw, pitch, and roll). This data is then used to translate natural head movements into cursor control, offering an intuitive and hands-free alternative for individuals with limited hand mobility.In addition to head movement-based cursor control, the system implements gesture-based mouse actions for seamless interaction. Opening the mouth is detected as a left-click, while opening the mouth along with moving the head enables a drag function, allowing users to perform click-and-drag operations effortlessly. Similarly, closing the eyes for approximately one second triggers a right-click, ensuring a natural and accessible method for performing various computer interactions.

To facilitate video input processing and real-time image analysis, OpenCV is integrated into the system. OpenCV provides a wide range of image processing functions, including video capture, frame manipulation, edge detection, and feature extraction, all of which are essential for stabilizing and refining head pose estimation. The combination of MediaPipe and OpenCV ensures smooth tracking with minimal latency, allowing for an efficient and responsive user experience. Additionally, the system is developed and implemented using Visual Studio Code (VSCode), a widely used integrated development environment (IDE) that supports efficient debugging, real-time code execution, and seamless integration with Python libraries. VSCode's versatility enables streamlined development, making it easier to optimize the system's performance and ensure compatibility across different platforms. By integrating these technologies, the system effectively transforms head movements into precise cursor control, enhancing accessibility and human-computer interaction for users with disabilities.

The system is structured as follows:

5.4.1 Block Diagram

The block diagram illustrates the hardware architecture of the Camera Interfaced Mouse Based on Head Pose Estimation, detailing the key components and their interconnections. The system is built around the Radxa Zero 3W, a powerful and energy-efficient Single Board Computer (SBC) responsible for processing head movements and executing machine learning algorithms for cursor control.

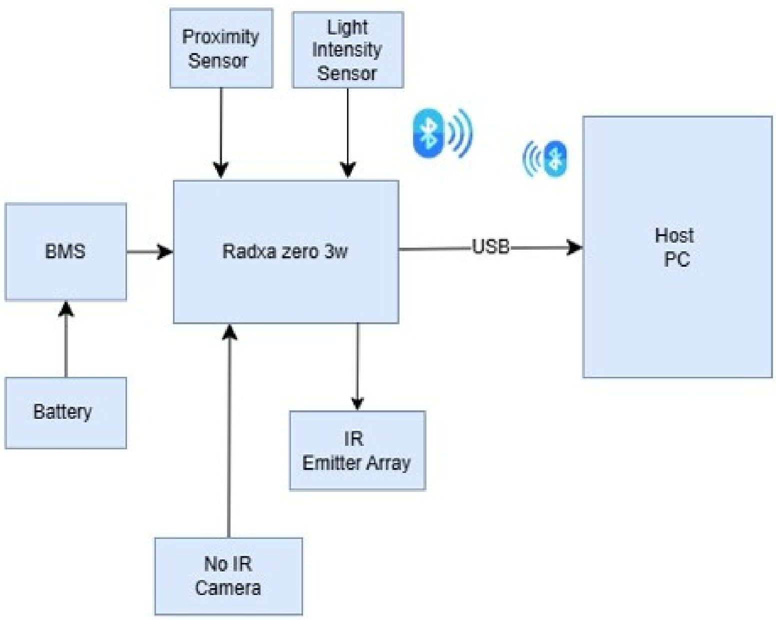

Figure 5.4: Block diagram.

The No-IR Camera captures real-time facial movements and sends data to the Radxa Zero 3W for head pose estimation. An IR Emitter Array is included to enhance visibility in low-light conditions, ensuring reliable face tracking regardless of ambient lighting. Two additional sensors, a Proximity Sensor and a Light Intensity Sensor, help in power management and adaptive camera exposure adjustment, optimizing system performance in varying environments.

Power management is handled by a Battery Management System (BMS), which regulates the battery to ensure stable power delivery to the SBC and peripheral components. The processed head movement data is transmitted to the Host PC via USB or Bluetooth, enabling hands-free cursor control. This architecture ensures seamless operation, efficient power utilization, and enhanced accessibility for users with limited hand mobility.

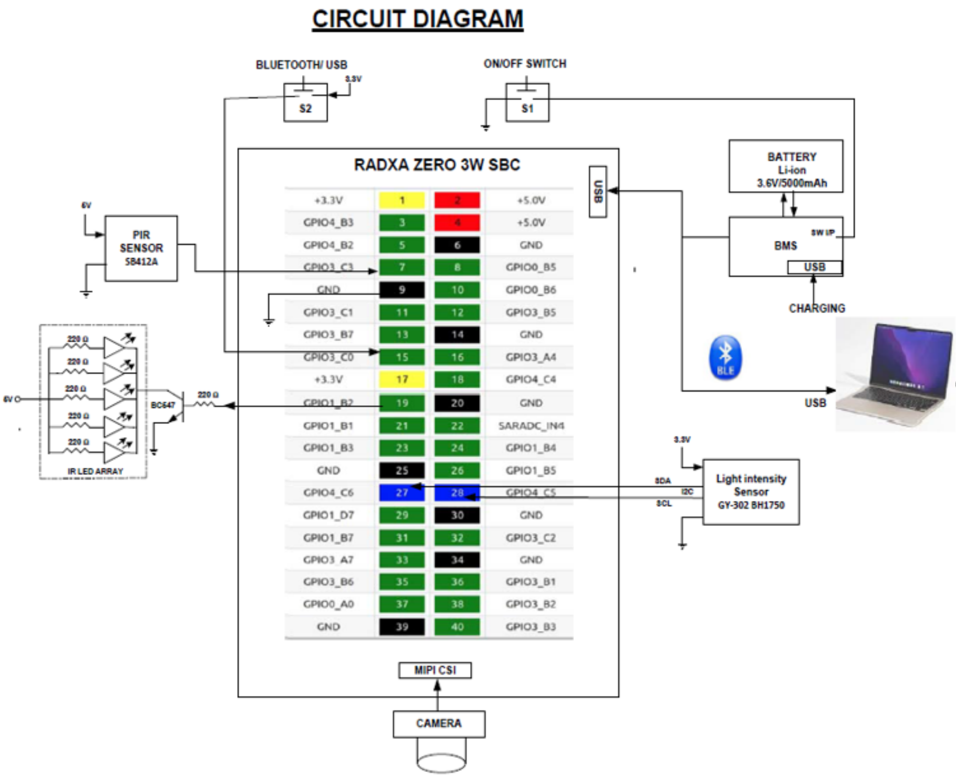

Figure 5.5: Circuit diagram

5.4.2 3D Printing

In this project, a custom enclosure was designed and fabricated using 3D printing to house the hardware components efficiently. The design was created using Blender 4.3.2,

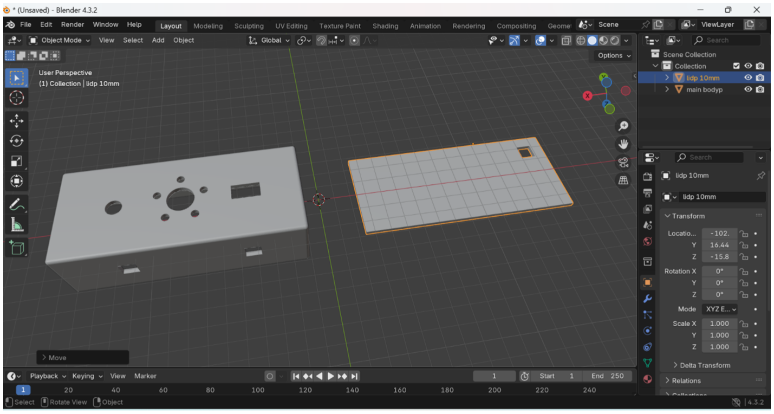

Figure 5.6: 3D Printing - Design

where the cutouts and structural elements were modeled to ensure proper placement of sensors, cameras, and other electronic components. The finalized 3D

Figure 5.7: 3D Printing

model was then exported and printed using a Creality 3D printer, ensuring a sturdy and well-fitted enclosure. The printing process utilized high-quality PLA filament, providing durability while keeping the enclosure lightweight.

Figure 5.8: 3D Printing - Case

Chapter 6

TESTING AND RESULTS

The project works were divided into two phases to ensure the systematic development and implementation of the camera-interfaced mouse based on head pose estimation. In Phase 1, cursor movements were achieved based on head pose using the laptop webcam. This involved tasks like implementing head tracking algorithms, mapping head movements to cursor positions. Phase 2 focused on expanding functionality by integrating click and drag actions, improving the current system, and porting the software to a hardware. Additionally, hardware design tasks were carried out, including developing setups for camera mounting and board integration. The final step involved testing the system on the selected hardware board, validating its performance by testing it in various conditions like different light, different users and different devices, in order to ensuring reliable operation under different conditions.

6.1 TESTING

A custom testing tool was developed to evaluate the performance and functionality of the camera-interfaced mouse based on head pose estimation. This tool was designed to systematically assess cursor accuracy, responsiveness, etc under different conditions. It facilitated the validation of head movement-based cursor control, click and drag actions, and overall system performance on both standard webcams and the hardware. Using the tool the Range, Clicks and Drag function can be tested.It has 4 windows for Range Test, Click Test, Drag Test and Test Results.

Range test measures the cursor's movement across the screen based on head pose, ensuring that tracking is accurate within the designated operating area. This helps in verifying the system's sensitivity, boundary limits, and smoothness of movement.

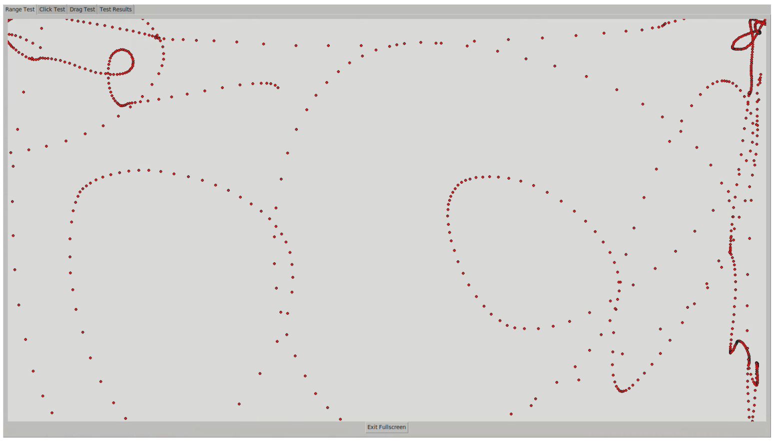

Figure 6.1: Range Test

Click Test evaluates the effectiveness of click actions triggered by head movements. It assesses the accuracy, latency, and consistency of the clicks, ensuring that the system registers user input reliably under different conditions.

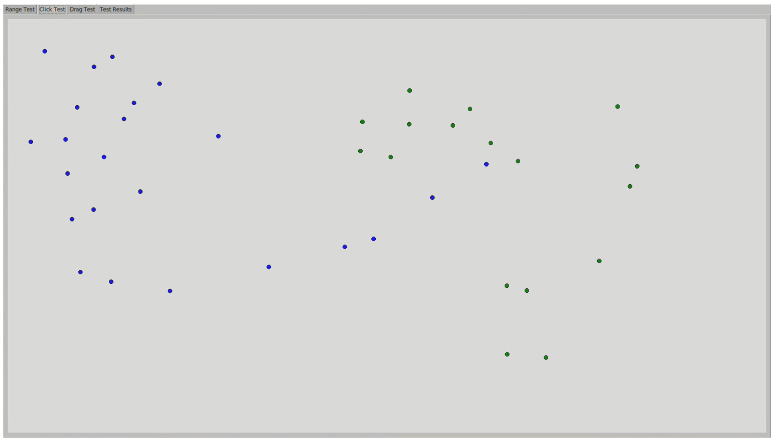

Figure 6.2: Click Test

Drag Test tests the system's ability to maintain a stable and continuous drag action. It verifies whether the cursor can smoothly follow head movements while holding a selection, ensuring object manipulation and usability.

Figure 6.3: Drag Test

6.2 RESULTS

Various tests were performed to assess detection performance and multi-user compatibility. The results provide insights into the system's effectiveness and areas for improvement. The camera module plays a crucial role in accurately capturing

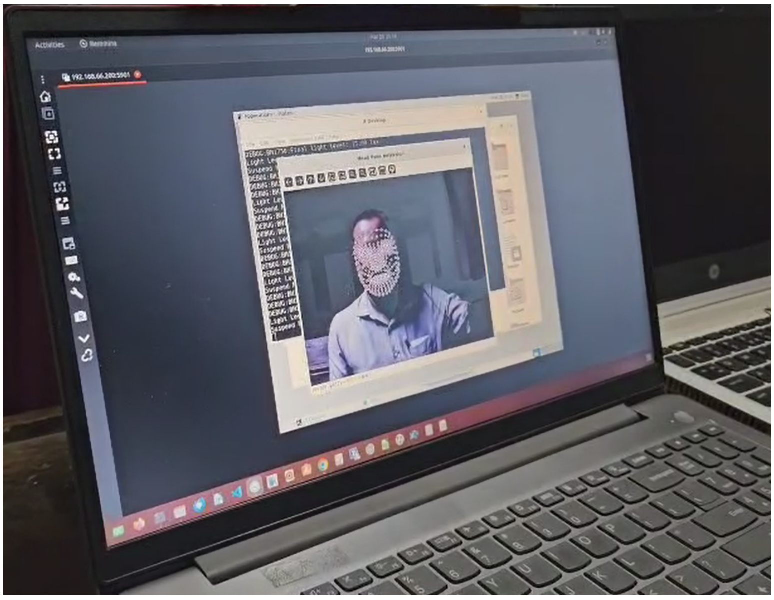

Figure 6.4: Test With light

head movements under different lighting conditions. These tests evaluate the performance of the infrared (IR) camera in daylight, dim light, and complete darkness, as well as its field of view adequacy. The goal is to ensure clear image capture, consistent detection, and reliable tracking across various environments.

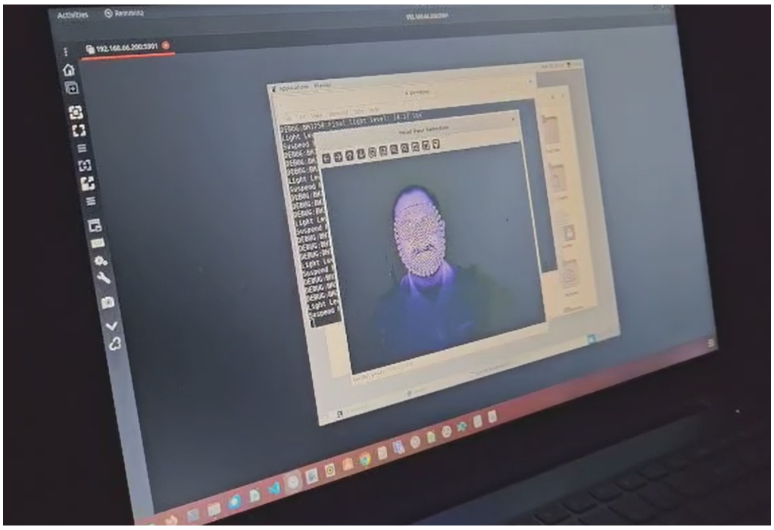

Figure 6.5: Test without light

Test ID	Description	Criteria	Method	Result	Observations
HW-CAM-01	Daylight IR performance	Clear image, stable tracking	Test in 500+ lux	Pass	Clear image, slight IR washout. Consider IR cut-off filter.
HW-CAM-02	Dim light IR performance	Clear image, stable tracking	Test in 50-100 lux	Pass	Stable, minor shadow noise. Noise reduction may help.
HW-CAM-03	Darkness IR performance	Clear image with IR LEDs	Test in $<$10 lux	Pass	Good visibility, but motion blur on fast head movements.
HW-CAM-04	Field of View	User remains in frame	Test head movement limits	Pass	Works well within $\pm 30°$ yaw, $\pm 20°$ pitch. Beyond this, occlusion issues.

Table 6.1: Camera Functionality Testing

Head Pose Detection Testing evaluates the system's ability to accurately track head movements under different lighting conditions and assess detection latency. The following tests measure the detection rate in low-light, and variable lighting environments, along with latency performance. The methodology includes running multiple detection cycles, comparing results against ground truth data, and measuring real-time system response.

Test ID	Description	Criteria	Method	Result	Observations
SW-HPD-01	Detection rate - best conditions	>98% successful detection	Run 100 detection cycles in ideal lighting conditions	Pass (99%)	Head pose tracking is smooth and stable in well-lit environments.
SW-HPD-02	Detection rate - low light	>75% successful detection	Run 100 detection cycles in low light conditions	Pass (76%)	Minor tracking instability in very dim settings, but within acceptable range.
SW-HPD-03	Detection rate - variable lighting	>80% successful detection	Run detection while changing lighting conditions	Pass (85%)	Brief drops in detection accuracy during abrupt lighting changes. Adaptive exposure helps.
SW-HPD-04	Detection latency	<100ms from movement to detection	Measure time between physical movement and software detection	Pass (85-95ms)	Real-time performance achieved. Minor delays in low-light conditions.

Table 6.2: Head Pose Detection Testing Results

Figure 6.6: Test - Mouse pointer movement

Mouse movement translation testing assesses the responsiveness and smoothness of the cursor controlled via head movements. The evaluation includes measuring cursor positioning, input-to-output latency, movement smoothness, and scaling appropriateness.

Multi-user testing was conducted to evaluate the robustness and fairness of the system across different user characteristics. The evaluation focused on face shape, skin tone, eye shape, eyewear compatibility, hairstyle variation, age, and gender to ensure unbiased performance.The results indicate high detection success

Figure 6.7: Test - Mouse pointer click by opening mouth

Figure 6.8: Test - Mouse pointer click

rates across all categories, with minor performance deviations in cases of thick-framed glasses and older participants.

Test ID	Description	Criteria	Method	Result	Observations
SW-MSE-01	Cursor position accuracy	$\pm$20 pixels from intended target	Test against reference grid of target points	Pass ($\pm$17 pixels)	Cursor tracking is correct, slight deviation at screen edges.
SW-MSE-02	Input-to-output latency	<200ms end-to-end latency	Measure time between head movement and cursor movement	Pass ($\sim$150ms)	System is responsive, minor delay in low light.
SW-MSE-03	Movement smoothness	No visible cursor jitter during slow movements	Qualitative assessment and jitter measurement	Pass	Jitter-free for slow to medium speeds, slight overshoot at fast movements.
SW-MSE-04	Scaling appropriateness	Full screen coverage with comfortable head movement range	Test screen edge accessibility with various users	Pass	Covers full screen; slight discomfort at extreme angles.

Table 6.3: Mouse Movement Translation Testing Results

Test ID	Description	Criteria	Method	Result	Observations
USR-PHY-01	Face shape variation	>95% detection success	Test with 3 users of different face shapes	Pass (100%)	No issues in face tracking.
USR-PHY-02	Skin tone variation	>95% detection success	Test with different skin tones	Pass (100%)	No bias detected.
USR-PHY-03	Eye shape variation	>95% detection success	Test with diverse eye shapes and sizes	Pass (100%)	All users detected reliably.
USR-PHY-04	Eyewear compatibility	>90% detection success	Test with one participant using glasses	Pass (90%)	Minor tracking delay with thick frames.
USR-PHY-05	Hairstyle variation	>90% detection success	Test with different hairstyles	Pass (100%)	No impact from different hairstyles.
USR-PHY-06	Age variation	>90% detection success	Test with 20s vs 40s participants	Pass (94%)	40s participant showed slightly slower response time.
USR-PHY-07	Gender variation	Equal performance across genders	Test with one male and one female in 20s	Pass	No performance difference.

Table 6.4: Multi-User Testing Results

Figure 6.9: Test - Multiple users

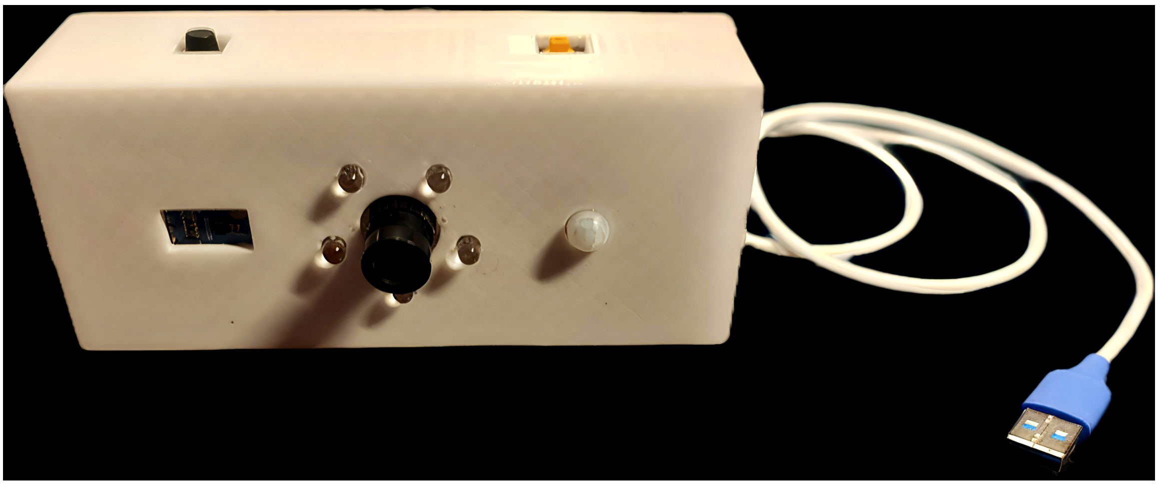

Figure 6.10: Camera Interfaced Mouse Based on Head Pose Estimation

Chapter 7

CONCLUSION

This project successfully designed and implemented a camera-interfaced mouse system utilizing head pose estimation to enable hands-free interaction for individuals with limited hand mobility. By integrating MediaPipe Head Pose Detection, the system achieved real-time and computationally efficient tracking of facial landmarks, allowing accurate estimation of head orientation. The use of OpenCV further enhanced video processing and image stabilization, ensuring smooth and responsive cursor movement.

The system's gesture-based interaction, including mouth opening for left-click, mouth movement for drag, and eye closure for right-click, provided an intuitive and accessible alternative to traditional input methods. Development and implementation using Visual Studio Code (VSCode) enabled seamless debugging and optimization, ensuring compatibility and efficient execution.

Through a combination of advanced computer vision techniques and real-time processing, the project demonstrated the feasibility of head movement-based cursor control as a practical accessibility solution. The successful implementation of this system highlights its potential to improve digital accessibility and human-computer interaction for users with motor impairments. Future enhancements could focus on further optimizing tracking algorithms, expanding gesture recognition, and adapting the system for broader hardware compatibility, making it an even more effective and widely usable assistive technology.

A literature review revealed the shortcomings of the present assistive technology, mainly in the following: very low latency is difficult and stability without controlled environments. Thus, in the preliminary rounds of testing, the objectives of this project were met, with an indication of a system whose performance is consistent in diversified settings. This makes up for a successful first phase as well as a strong ground for further development in the next stage.

7.1 SCOPE OF FURTHER WORK

7.1.1 What is future directions?

The development of a camera-interfaced mouse based on head pose estimation provides a significant advancement in accessibility and human-computer interaction. However, there are several areas for future enhancement and expansion. The following key points outline potential future developments:

1. **Machine Learning-Based Adaptive Calibration** Implementing AI-driven models that learn user-specific movement patterns over time. This adaptation will improve accuracy, reduce false detections, and ensure a more personalized experience for users with different mobility needs.

2. **Performance Optimization using C++** Implementing performance-critical components in C++ instead of Python can significantly reduce latency and improve real-time processing. This optimization will ensure smoother interactions and enhanced system responsiveness.

3. **Wireless and Wearable Adaptation** Developing a lightweight, wireless, wearable version, such as smart glasses or headbands with integrated sensors. This will enhance usability by providing a more natural and comfortable user experience.

Chapter 8

APPENDIX

8.1 COMPONENTS

8.1.1 Radxa Zero 3W

The project requires a robust and energy-efficient SBC. The Radxa Zero 3W is powered by the 64-bit ARM Cortex-A55 quad-core CPU, delivering a notable performance improvement over its predecessors. It provides ample computational power for real-time image processing and machine learning tasks related to head-pose estimation. It features up to 4GB LPDDR4 RAM, ensuring smooth multi-tasking and efficient execution of applications involving computer vision and AI inference.

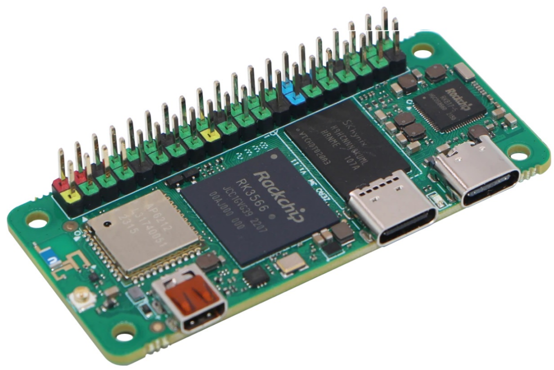

Figure 8.1: Radxa Zero 3W

For connectivity, it includes Wi-Fi 5 (802.11ac) for high-speed wireless networking and Bluetooth 5.2 with BLE, enabling stable and low-power communication with peripherals, sensors, and mobile devices. The board is equipped with multiple interfaces, including USB-C for power and data transfer, micro HDMI for display output, and GPIO pins for hardware expansion, enhancing its adaptability for various applications.

8.1.2 Battery

The battery is the primary power source for the entire system. It provides the required voltage and current to drive the SBC and all the other components connected to it. Typically, a rechargeable lithium-ion or lithium-polymer battery is used; it offers a compact, high-energy-density solution suitable for portable electronics. Proper sizing of the battery is critical to allow the system to operate for a specified duration without recharging too often. **Samsung INR21700-50S**

Figure 8.2: Battery

3.6V 5000mAh 9C Li-ion Battery have been used for this device.The Samsung INR21700-50S is a high-performance 3.6V lithium-ion battery with a capacity of 5000mAh and a discharge rate of 9C, ideal for power-hungry devices requiring sustained high current output.

8.1.3 BMS (Battery Management System)

The BMS monitors and takes care of the health performance of the battery, putting the battery in a protected position from possible risks or threats such as overcharge, over-discharge, and overheating. For example, it monitors any changes in the battery parameters relating to voltage, current, or temperature; balances the cells and extends their life; and acts as a protector by power cutting under unsafe conditions, hence allowing safe and stable supply of power for reliable and efficient energy provision without possible thermal runaway or other faults.**Type-C USB 5V 2A Step-Up Boost Converter with USB Charger** is the BMS we have used for the project.This is a Type-C USB 5V 2A Step-Up Boost Converter with a USB Charger.Great Step-up Module with 5V input, output 5V for many digital devices, so this is really a great module for designing a portable charger. Useful for your USB charger projects and onboard USB device supply solutions.

8.1.4 Proximity Sensor

This proximity sensor is used to detect the presence and distance of objects in a given range. It can be an infrared (IR) proximity sensor or ultrasonic sensor that sends out a signal and measures the reflection time to determine object distance. Proper positioning of the sensor is required so that accurate readings can be taken; otherwise, it may trigger specific actions or may provide information to the SBC for adaptive system behavior based on proximity data. **SB412A PIR Sensor Module (based on the AM412 PIR sensor)** is used here.SB412A is the latest digital sensor module that is based on AM412 digital sensor. It only contains a Fresnel lens, PCB, and a few resistance.

Figure 8.3: PIR Sensor

8.1.5 Light Intensity Sensor

The light intensity sensor determines the ambient light levels within the surrounding environment, thereby enabling an automatic adjustment in operations by lighting conditions. It's a photodiode or a light-dependent resistor known by the abbreviation LDR; this sensor is either giving out an analog or a digital signal that is proportional to the amount of surrounding light.**GY-302 BH1750 Light Intensity Module** is used for the device.This is a Light Intensity Detection Module based on the BH1750 chip. BH1750FVI is a digital Ambient Light Sensor IC for the I2C bus interface. These readings need to be very accurate if a system were to run

Figure 8.4: Light Intensity Sensor

with changing light conditions, the sensitivity and the range to suit the environment whether indoor or outdoor for proper operation.

8.1.6 IR (Infrared) Emitter Array

An array of IR emitters allows infrared illumination and especially at night or even in darkness, the camera can capture an image or even locate objects by the reflected infrared light coming off surrounding objects. A set of IR LEDs comprise

Figure 8.5: IR Emitter Array

the IR emitter array and send out invisible infrared light that bounces off any surrounding objects to be received by the IR camera and thus applied for night-vision applications. Correct positioning and intensity of the IR emitters must be obtained to successfully capture an image in dark environments for uniform illumination.

8.1.7 No-IR Camera

We have used the **Raspberry PI Infrared IR Night Vision Surveillance Camera Module** in this device.It features 5MP with OmniVision 5647 sensor which is in fixed focus mode. The No IR camera captures images or video in the infrared spectrum, so that there will be a possibility to acquire visual data in very dim light or even complete darkness. Without an IR-cut filter, it is capable of capturing infrared light; hence, it will work perfectly in complete darkness when mated with the IR emitter array. The essential aspects that will ensure high quality of images are camera resolution, frame rate, and infrared sensitivity, as well as correct alignment with the IR emitters for maximum visibility.

Figure 8.6: Raspberry PI Infrared IR Night Vision Surveillance Camera Module

8.1.8 Bluetooth

Bluetooth module is a wireless communication module that enables data transfer between the SBC and other compatible devices controlled or monitored remotely. The key considerations include managing the range of Bluetooth bandwidths and pairing security to ensure stable and secure connections where wireless connections are ideal for applications that require portability, meaning they cannot be implemented using a wired connection.

8.1.9 USB

USB provides a wired link from the SBC to the Host PC, so it supports high-speed transfer of data and can provide power to the SBC when needed. This means that the data can be transferred directly from the Host PC for processing or for storage purposes, and that firmware updates to the SBC can also be conducted through this connection. In any case, the actual speed of data transfer depends on the version of the USB, and cable length adversely affects the signal quality; hence, it is preferable for applications where sustained communication at high bandwidth is critical.

8.1.10 Host PC

The Host PC is the primary computation or storage node where all the data received from SBC is processed and viewed information or further analysis executed. Generally, a desk or laptop computer accepts the data received from SBC and processes it by further analysis or visualization of image or sensor reading, by recording all the collected data, and provides an easier way for the user for viewing and controlling the remote SBC. The incoming data requirements must be supported by the Host PC for real-time processing and smooth communication with the SBC.

REFERENCES

[1] **M. N. Islam, M. S. Aadeeb, R. R. Khan, M. M. H. Munna, M. Sarwar, and S. Nasrin** , Developing a Novel Hands-Free Interaction Technique Based on Nose and Teeth Movements for Using Mobile Devices, Publisher: IEEE, Vol.9, 2021, pp. [58127 - 58141].

[2] **P. Viola and M. J. Jones**, Robust real-time face detection, Int. J. Comput.Vis., vol. 57, no. 2, pp. 137154, May 2004.

[3] **B. D. Lucas and T. Kanade**, An iterative image registration technique with an application to stereo vision," in Proc. 7th Int. Joint Conf. Artif. Intell., vol. 2, 1981, pp. 674679.

[4] **P. Viola and M. Jones**, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Dec. 2001, p. 1.

[5] **Rahib H. Abiyev, M. Arslan**, "Head mouse control system for people with disabilities," Applied Artificial Intelligence Research Centre, Dept. of Computer Engineering, Near East University, North Cyprus, Mersin-10, Turkey. [Email: murat.arslan@neu.edu.tr]

[6] **Baldominos, A., Saez, Y., & Isasi, P**,Evolutionary convolutional neural networks: An application to handwriting recognition, Neurocomputing, vol.283, 29 March 2018, Pages 38-52.

[7] **W. Rahmaniar, A. Ma'arif, and T.-L. Lin**, "Touchless Head-Control (THC): Head Gesture Recognition for Cursor and Orientation Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, 2022.

[8] **K. M. Rayudu, M. Narendra, P. Anitha, T. S. Ratna Sai, and M. Sandeep**, "Virtual Mouse Operation Using Webcam," *Mathematical Statistician and Engineering Applications*, vol. 70, no. 2, pp. 536–549, 2021. doi: https://doi.org/10.17762/msea.v70i2.1829

[9] **J. Tu, H. Tao, and T. Huang**, "Face as mouse through visual face tracking," *Computer Vision and Image Understanding*, vol. 108, no. 1–2, pp. 35–40, Oct.–Nov. 2007.

[10] **A. Price and F. Loizides**, "Allowing for Secure and Accessible Authentication for Individuals with Disabilities of Dexterity," in *Human-Centered Software Engineering (HCSE 2024)*, Lecture Notes in Computer Science, vol. 14793, pp. 133–146, First Online: 01 July 2024.

[11] **J. Yang, T. Qian, F. Zhang, and S. U. Khan**, "Real-Time Facial Expression Recognition Based on Edge Computing," *IEEE Access*, vol. 9, pp. 76178–76190, May 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3082641.

[12] **E. Murphy-Chutorian and M. M. Trivedi**, "Head Pose Estimation in Computer Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607-626, Apr. 2009, doi: https://doi.org/10.1109/TPAMI.2008.106.

[13] CREA. (2020) EVA Facial Mouse-Apps on Google Play. Accessed: Jun. 22, 2020. [Online]. Available: https://bit.ly/3hSlzje

[14] G. A. Devices. GlassOuse Assistive Device. Accessed: Jun. 22, 2020. [Online]. Available: https://glassouse.com

[15] Quha. Quha Zono. Accessed: Jun. 22, 2020. [Online]. Available: https://www.quha.com/products-2/zono/