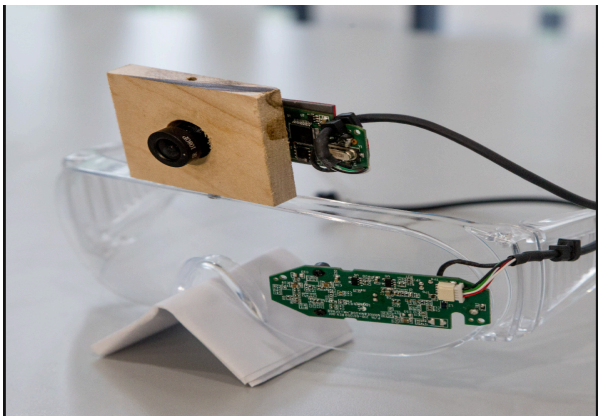
D03: V.I.S.T.A.

Vision-based Interactive System for Typing and Access

Ashmita Das, Ashmit R Sambrani, Tanneru Ranjit, Sahasra Pulumati

INTRODUCTION: -

This project introduces a new standard for keyboard and mouse functionality through a virtual concept. Hand gestures are mapped to keyboard characters, while eye-tracking enables mouse movement on the screen. The system uses two cameras mounted on a pair of transparent goggles to track eye movement, leveraging OpenCV and object detection. Additionally, a Pi Zero camera is positioned on the monitor to track hand gestures. The goggles' cameras (ESP-32) connect wirelessly via Wi-Fi or BLE to the Pi Zero camera, which communicates with the device via Bluetooth/USB.



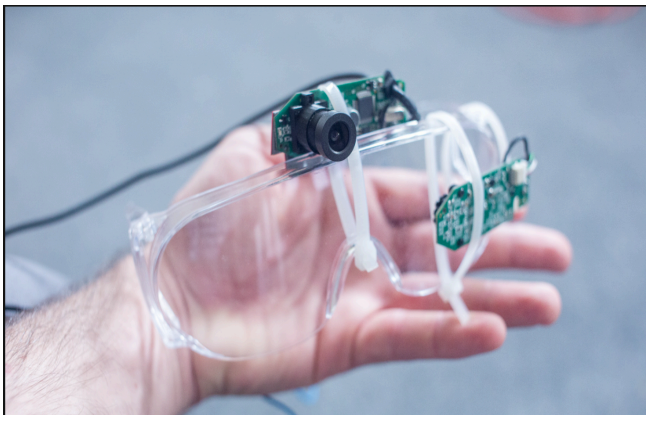


fig 1. The goggles with its 2 cameras

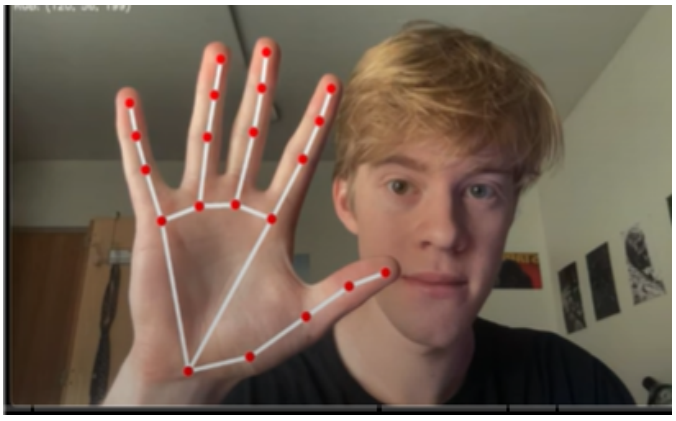


fig 2. Finger recognition using openCV

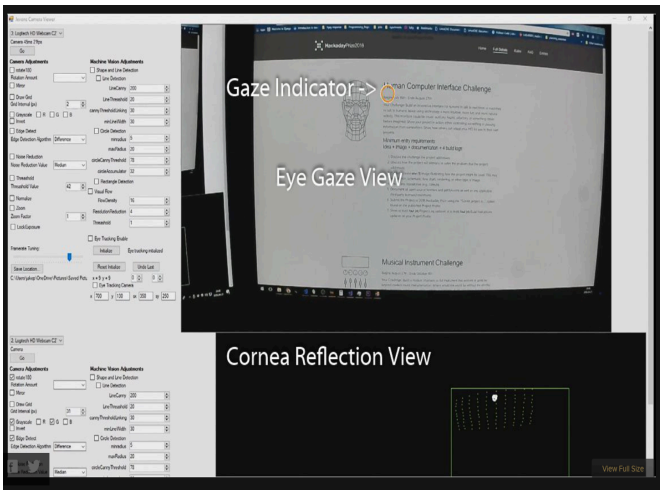


fig 3. POV of the user

fig 4. Using the Pi Cam to detect face angle

OBJECTIVE: -

The goal of this project is to rethink how we interact with computers by replacing traditional keyboards and mice with something more intuitive and natural. By combining hand gestures and eye-tracking, it offers a hands-free way to type and control a screen. This could make gaming and virtual reality more immersive, and even help in situations where physical peripherals aren't practical—like on the go or in tight spaces.

This solution could benefit various applications, including:

1. **Accessibility**: Empowering individuals with physical disabilities to interact with computers seamlessly.
2. **Gaming**: Creating more immersive and natural control systems for virtual environments.
3. **Productivity**: Offering an efficient, hands-free interface for multitasking or environments where physical peripherals are impractical.
4. **Wearable Technology**: Pioneering advancements in smart glasses and AR/VR systems.

PREREQUISITES: -

Basic knowledge of Python Programming.

DESCRIPTION: -

Project Overview:

The project is divided into two main components: Keyboard and Mouse.

Keyboard:

1. **Hand Gesture Detection:**
    - Hand gestures are captured using a Pi Camera mounted to a Pi Zero module positioned on top of the screen.
    - The Pi Zero receives data from ESP-32 cameras, which also serve as part of the mouse system.
2. **Data Processing:**
    - The Pi camera and the ESP-32 cameras send their data to the Pi Zero for processing due to its superior processing power.
3. **Gesture Analysis:**
    - OpenCV and a lightweight learning model are used to analyze finger positions.
    - Fingers can have three states:
        - **CLOSED**
        - **HALF OPEN**
        - **OPEN**
    - This creates a 3-bit system, enabling up to 3^5 =243 possibilities, which can be mapped to macros or keyboard characters.

Mouse:

1. **Camera Setup:**
    - Two ESP-32 cameras are mounted on transparent goggles:
        - **Camera 1:** Faces the left eye and tracks pupil motion.
        - **Camera 2:** Positioned on top of the glasses to capture the observer's frame (i.e., the screen).
2. **Pupil Tracking:**
    - The second camera detects the screen area and maps pupil movements to corresponding screen locations.

3. **Data Processing and Optimization:**
    - If the ESP-32s lack sufficient processing power:
        - Data is transferred to the Pi Zero for advanced processing.
        - Filters and models, such as TensorFlow Lite, are applied for enhanced accuracy.
    - The processed data is sent to the computer via USB-serial communication.

MATERIALS:

COMPONENT	COST	LINK	STATUS
ESP32CAM * 2	-	Robocraze Link	In IEEE Inventory
FT232RL FTDI USB to TTL Adapter	-	Amazon Link	In IEEE Inventory
Micro USB cable	-	Amazon Link	In IEEE Inventory
Breadboard	-	Amazon Link	In IEEE Inventory
Jumper cables	-	Amazon Link	In IEEE Inventory
Transparent goggles	-	Amazon Link	In IEEE Inventory
PI-ZERO	-	Robocraze Link	In IEEE Inventory
PI-Cam	231	Robocraze Link	To be Procured
Li-On battery	64	Robocraze Link	To be Procured
li-ion charger	20	Robu Link	To be Procured
TOTAL	~500		

TIMELINE:

Total Allotted Time : 14 Weeks

W01 (1 Week) :: Midsem

W02 - W04 (3 Weeks) :: Learning Phase (OpenCV and basics of Embedded systems)

W05 - W08 (3 Weeks) :: Keyboard Implementation

W09 - W10 (2 Weeks) :: Mouse Implementation

W10 - W11 (2 Weeks) :: End Sem

W12 (1 Week) :: Mouse Implementation

W13 - W14 (2 Weeks) :: Integration

REFERENCES:

https://hackaday.io/project/153293-low-cost-open-source-eye-tracking

https://youtu.be/hPijy3M5150?si=AI9jLht3et2efeMC

https://youtu.be/-toNMaS4SeQ?si=3jQQ-Myi5nu6qsma

https://gazerecorder.com/

https://www.youtube.com/watch?v=a_UiYOO-Sdw/

CONTACT DETAILS

Ranjit Tanneru - 6360757505 - ranjittanneru.231ee159@nitk.edu.in

Ashmit R Sambrani - 7676394075 - ashmitrsambrani.231ec109@nitk.edu.in

Sahasra Pulumati - 8123900157 - sahasrapulumati.231ee247@nitk.edu.in

Ashmita Das - 7892120620 - ashmitadas.231ec208@nitk.edu.in