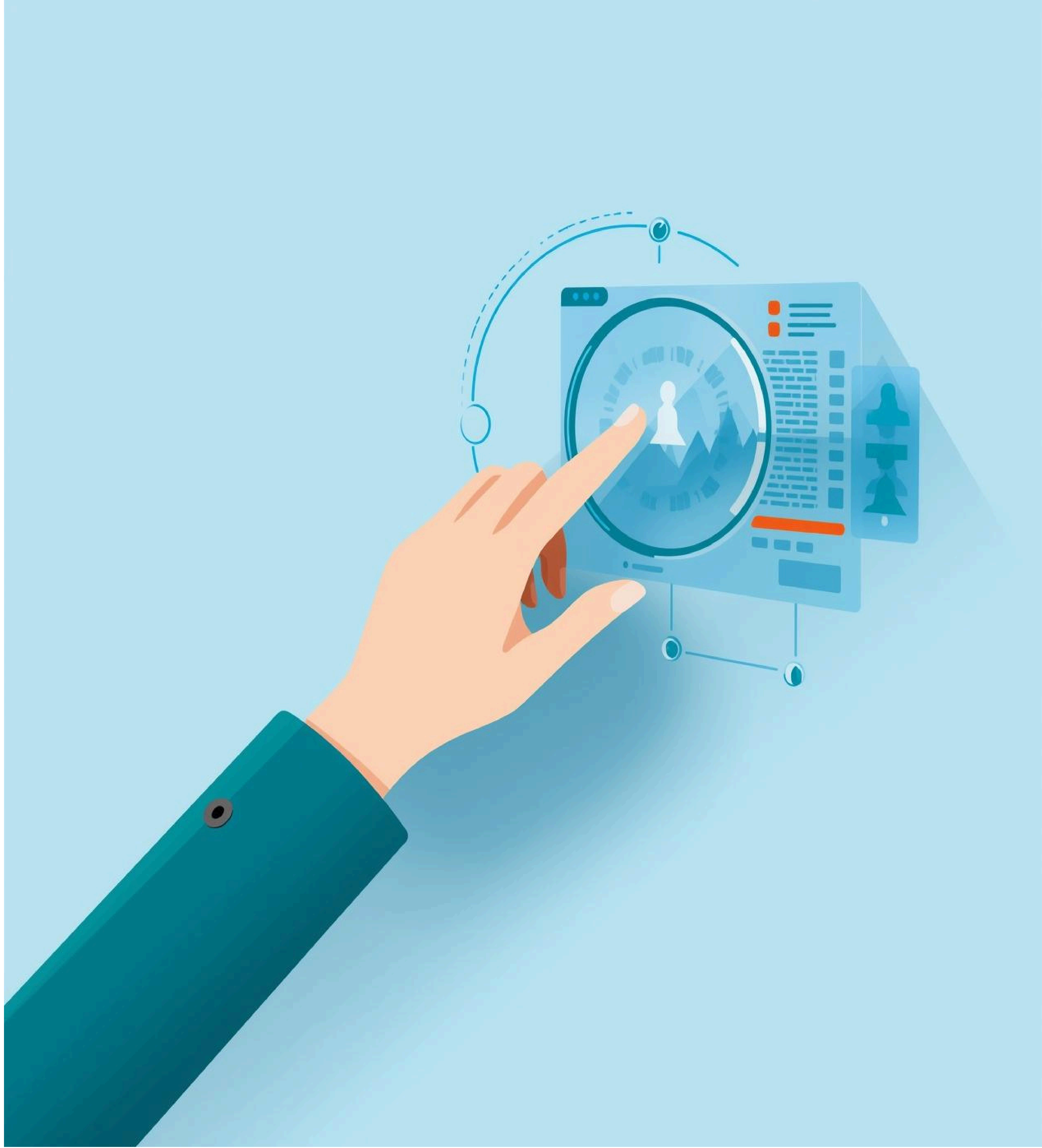
Wearable Hand Gesture Recognition for Laptop Control

Group -6
2310030157-N.Karthikeya
2310030115-A.Lokesh
2310030045-T.Sai Uday
2310030046-S.Nikhil
2310030308-V.Karthik
Guide: Dr.Shadab Siddiqi and Dr.UmaRani

Addressing Inaccessible Input Devices

Traditional input devices can often be **challenging** or inconvenient for users, especially during presentations or for those with physical disabilities, highlighting the growing need for more **natural interaction methods**.



Our Solution

Wearable Hand Gesture Recognition for Laptop Control

Key Features

- Wearable Gesture Input Module: Combines ESP32 with an IMU
- Real-time Motion Capture: Utilizes accelerometer-gyroscope fusion
- BLE-based Transmission: Ensures low-latency gesture communication
- Python-driven Host-Side Processing: Implements a gesture processing engine
- Action Mapping Layer: Facilitates laptop control for mouse and keyboard functions
- Versatile Performance: Operates effectively in all lighting and environmental conditions
- User-Specific Gesture System: Offers full portability

Core Advantages

- Hardware-based Detection: Operates independently from cameras
- High Accuracy: Achieves precision with direct sensor readings
- Lightweight and Cost-effective: Designed for low power consumption
- Comfortable Wearable Design: Ensures ease of use and practicality

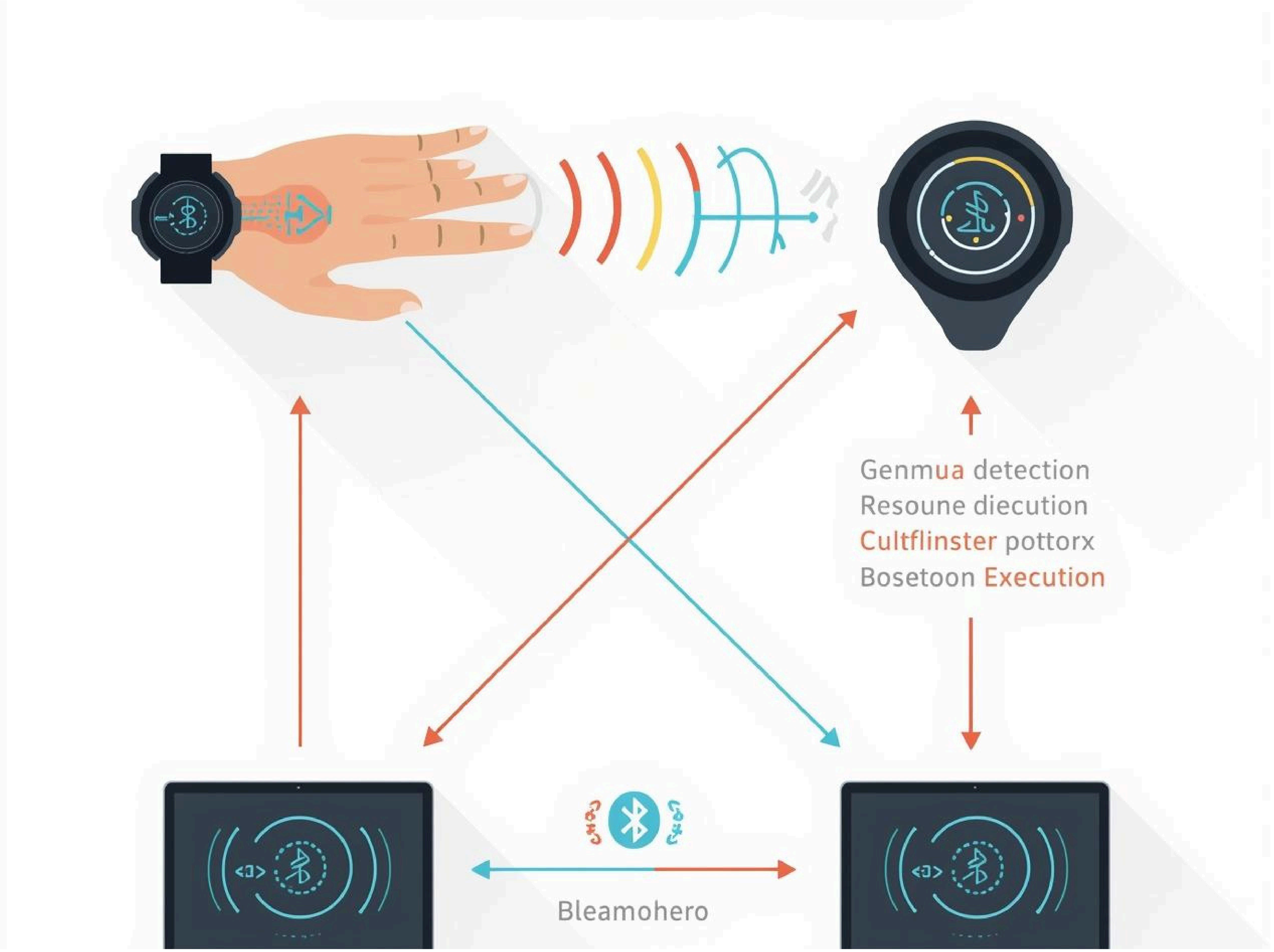
link to access:
https://www.researchgate.net/publication/391544307_AI_-Driven_Hand_Gesture_Recognition_System

System Workflow

The system detects **hand gestures** through sensors, sending processed data via Bluetooth to the laptop, which executes commands based on user-defined mappings, enhancing intuitive control without physical interaction.
ser performs hand gestures which are detected by sensors

Microcontroller processes the sensor data and sends it via Bluetooth

Laptop receives the gesture data and performs the required action



PRE Existing technologies

<u>Overview of Gesture and Interface Systems</u>

1. Vision-Based Gesture System:-

- Utilizes cameras and image processing technology
- Provides high accuracy, but can be affected by lighting conditions
- Demands considerable computational resources

2. Sensor-Based Wearable Systems

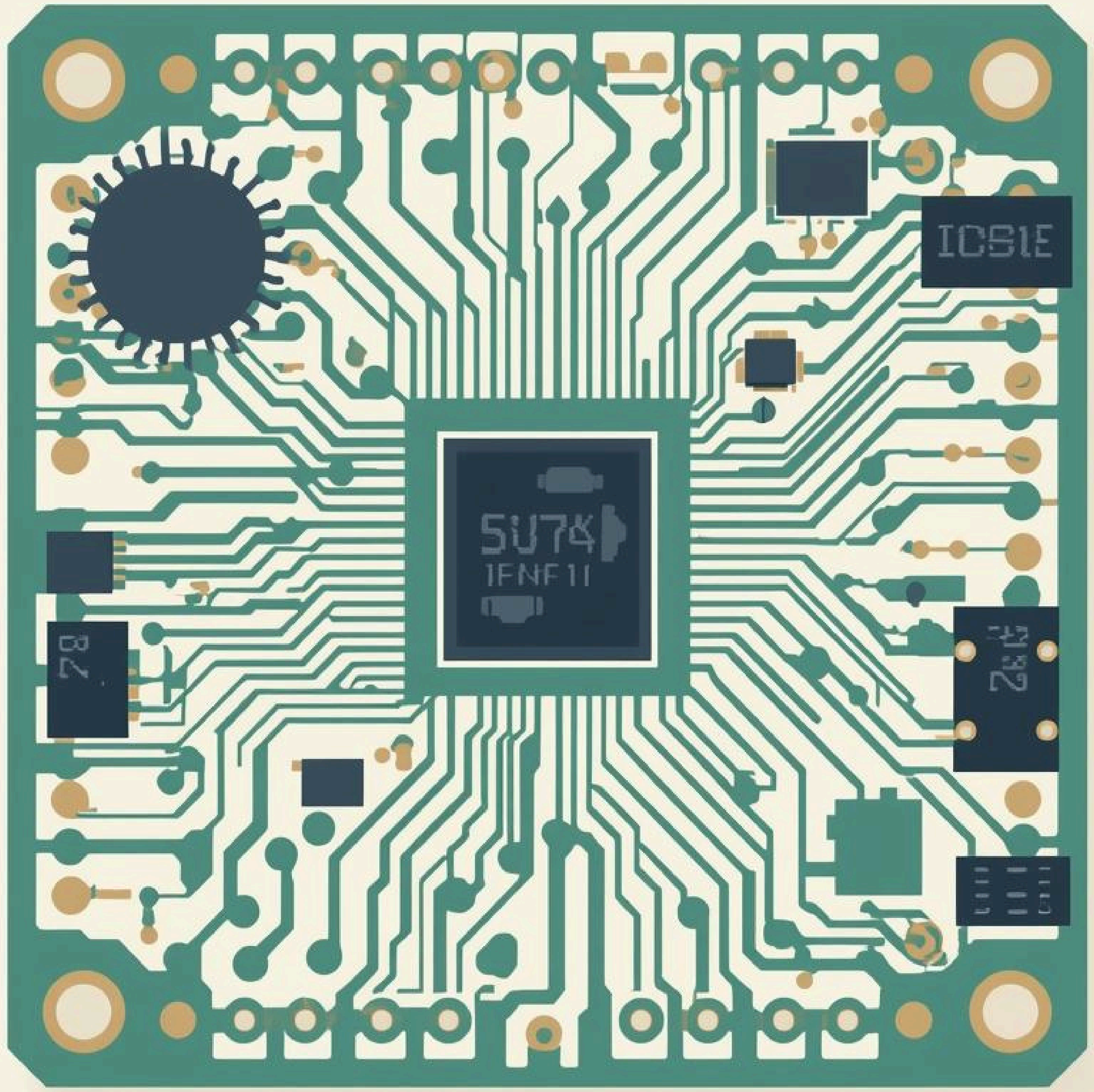
- Incorporates accelerometers and gyroscopes
- Operates independently of lighting and background influences
- Lightweight design, making it perfect for real-time control

3. Touch & Voice-Based Interfaces

- Includes touchpads and voice recognition assistants
- Limited performance in noisy settings or when hands are occupied
- Less accommodating for individuals with physical disabilities

<u>Limitations of Current Systems</u>

- Often expensive or require complicated setups
- Lack portability and flexibility
- Not entirely suited for all user scenarios.



TECH STACK

1. Hardware Layer (Wearable Device)
ESP32 Control Unit
MPU6050 / MPU9250 IMU Sensor Module
On-board BLE Transmitter
Wearable Glove / Strap Platform
Lithium-ion Power Module
2. Firmware Layer (Embedded Programming)
Arduino C / ESP-IDF (for ESP32)
Firmware Gesture Encoder
Sensor Fusion Algorithms (Complementary Filter / Kalman Filter)
BLE Data Packet Transmission Layer
Gesture Opcode Generator
3. Communication Layer
Bluetooth Low Energy (BLE)
Low-Latency Device-to-Host Data Channel
Custom Gesture Data Packet (GDP) Protocol
Host-Side Command Listener (Python)
4. Software Layer (Laptop / Application Side)
Language:
Python
Core Libraries:
pyserial / bleak – BLE communication
numpy – sensor data handling
scikit-learn (optional) – gesture classification ML
pyautogui – laptop input automation
pynput – mouse/keyboard control
matplotlib – gesture debugging (optional)
Modules:
Host-Side Gesture Processing Engine
Gesture Decoding Module
Action Mapping Layer (AML)
Virtual Input Controller
Purpose:
Receives BLE packets → decodes gesture → executes laptop actions.
5. **System Integration Layer**
Motion-to-Action Pipeline (MAP)
Laptop Control Abstraction Layer (LCAL)
Real-time Human-Computer Interaction Layer (HCI Layer)
6. Tools & Platforms
Arduino IDE / PlatformIO – ESP32 programming
VS Code – Application development
Jupyter Notebook (optional) – ML model training
Git/GitHub – version control
Figma / Canva – UI & diagrams

Societal Impact and Applications

Assistive technology for physically challenged users

Touchless control in healthcare environments

Smart classrooms & presentations

Virtual Reality (VR) and Gaming

Industrial automation control

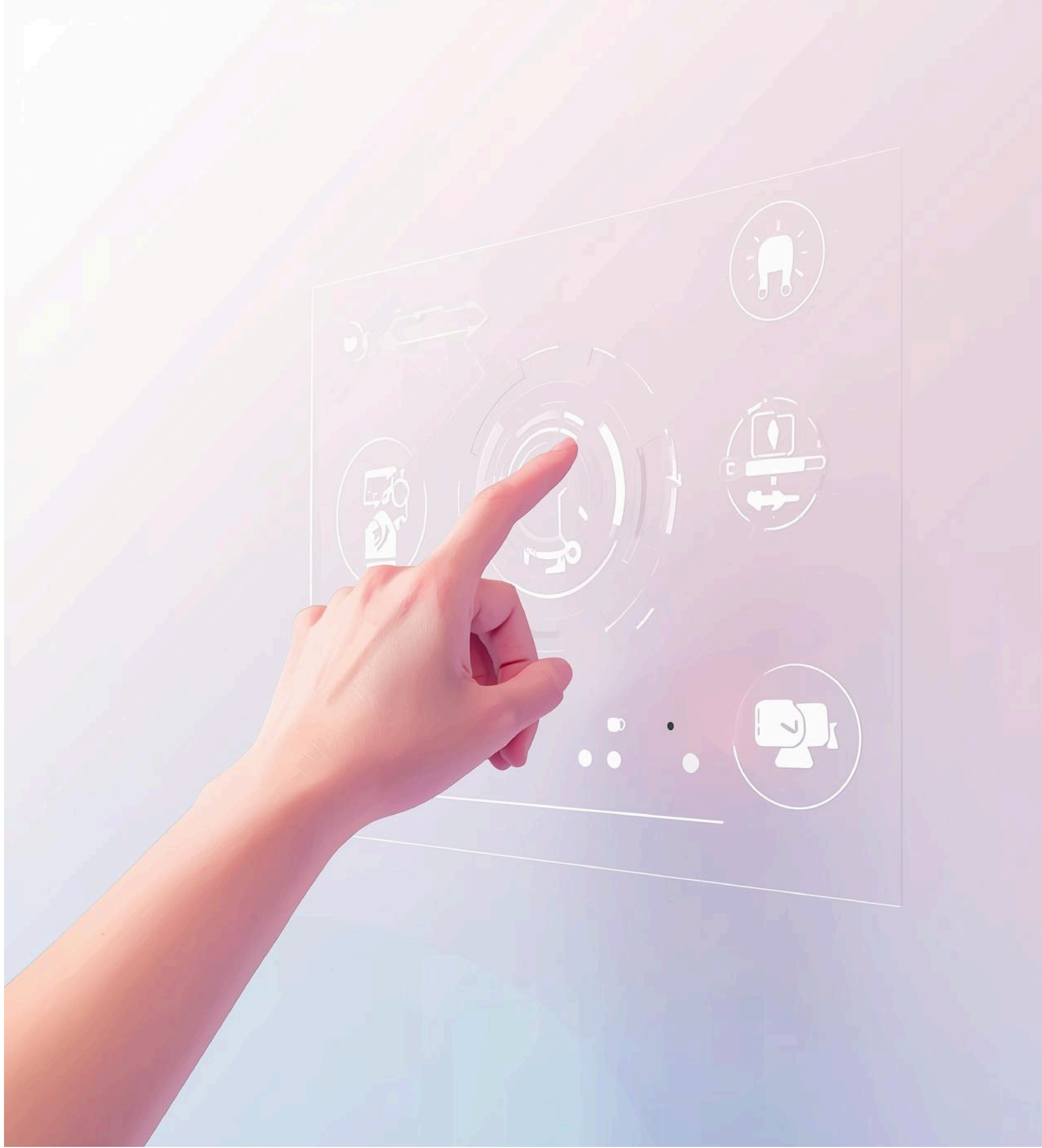
Benefits to Society

Improves digital accessibility

Reduces dependency on physical devices

Promotes inclusive technology

Enhances productivity and comfort



Novelty and Uniqueness

- Interaction through wearable devices rather than relying on camera-based systems
- A portable and user-specific control mechanism
- A low-power, cost-effective solution
- Customizable gestures tailored for different users
- Real-time recognition of gestures
- Distinctive features
- Independence from external cameras
- Functionality in low-light environments
- Personalized mapping of gestures

Conclusion

The project offers a smart and easy way to control a laptop using simple gestures. It shows strong use of wearable sensors, improves the way people interact with computers, and sets the stage for future upgrades like AI-driven learning, richer gesture sets, smart home links, mobile control, and better accuracy.

Thank You