TOPIC 2

Abstract

Sign language recognition is a transformative technology designed to bridge the communication gap between the deaf community and those unfamiliar with sign language. This project aims to develop a cutting-edge web application that utilizes advanced machine learning algorithms and computer vision techniques to accurately interpret and translate sign language into text or speech in real-time. Leveraging Python for backend development with frameworks like Flask or Django, and integrating machine learning libraries such as TensorFlow, Keras, OpenCV, and MediaPipe, the system ensures robust and reliable gesture recognition. The frontend, built with modern JavaScript frameworks like React, Vue, or Angular, provides an intuitive and interactive user interface. Real-time communication is facilitated through WebSocket technology, ensuring seamless and instant feedback. The ultimate objective of this project is to enhance accessibility, promote inclusivity, and facilitate effective communication for deaf and hard-of-hearing individuals across various social, educational, and professional environments.

Key features

- Real-Time Gesture Recognition:
- Immediate Translation: The system captures and processes sign language gestures in real-time, providing instant translation to text or speech.
- High Accuracy and Reliability:
- Advanced Machine Learning Models: Utilizes TensorFlow and Keras to develop robust neural networks capable of accurately recognizing a wide range of sign language gestures.
- Web-Based Application: Accessible from any device with a web browser, including desktops, tablets, and smartphones.
- WebSocket Integration: Utilizes WebSocket technology (e.g., Socket.io) to enable continuous, low-latency communication between the frontend and backend, ensuring seamless and instant feedback.
- Scalability and Extensibility:
- Modular Architecture: Designed with a modular approach, allowing easy integration of additional features or support for other sign languages.
- Scalable Backend: Built with frameworks like Flask or Django, ensuring the backend can handle increasing loads and scale effectively.
- Customization and Personalization:
- User Profiles: Allows users to create profiles, save preferences, and customize the application according to their needs.