Real-Time Sign Language Recognition using Computer Vision

Abstract

This project implements a real-time sign language recognition system using computer vision and deep learning. By integrating OpenCV, MediaPipe, and TensorFlow, the system detects hand gestures and classifies them into corresponding signs. A dataset of hand gesture images was created, processed, and trained with a classification model. The system can recognize basic signs and display their meanings as text in real-time, with potential extensions for text-to-speech integration.

Introduction

Communication barriers between the hearing-impaired and non-signers often create challenges in daily interactions. Sign language recognition systems aim to bridge this gap by converting gestures into meaningful text or speech. This project focuses on building a real-time Indian Sign Language (ISL) recognition model. The system captures hand gestures via a webcam, processes them with image preprocessing techniques, and classifies them using a deep learning model.

Tools Used

1. Python – Programming language for implementation.
2. OpenCV – Capturing webcam input and performing image preprocessing.
3. MediaPipe & cvzone – Detecting and tracking hand landmarks.
4. TensorFlow / Keras – Training and using the gesture classification model.
5. NumPy & Math – Mathematical operations and array handling.

Steps Involved in Building the Project

1. Data Collection – Using webcam and MediaPipe, multiple hand gesture images were captured and stored in folders (DataCollection.py).
2. Preprocessing – Images were cropped, resized, and normalized into a standard format (300×300). Aspect ratio adjustments ensured gestures fit within a white canvas.
3. Model Training – A Convolutional Neural Network (CNN) was trained using the collected dataset. The trained model was saved in .h5 format along with label mappings.
4. Testing – The trained model was tested using real-time webcam input (Test.py). Predictions were displayed on screen with bounding boxes and recognized labels.
5. Final Implementation – The full pipeline integrated hand detection, preprocessing, classification, and visualization (Final.py). The recognized signs were displayed in real-time.

Conclusion

The project successfully demonstrates a real-time sign language recognition system. It shows how computer vision and deep learning can bridge communication barriers by converting hand gestures into meaningful text. While the current model is limited to a small vocabulary, it provides a scalable foundation. Future work may include expanding the dataset, supporting complete ISL/ASL alphabets, and integrating speech output for better accessibility.