REAL-TIME SIGN LANGUAGE RECOGNITION SYSTEM

CHOKKA SRIYOSHIJA

AIML INTERN AT ELEVATE LABS

Introduction

This computer vision system enables real-time American Sign Language (ASL) alphabet recognition using hand gesture analysis. The solution addresses communication barriers for hearing-impaired individuals through AI-powered translation of manual alphabets into digital text.

Abstract

This study develops a real-time system to translate static American Sign Language (ASL) alphabets (A–Y) into text using computer vision and deep learning. The pipeline integrates MediaPipe for robust hand landmark detection and OpenCV for live video processing. A CNN model trained on a custom dataset of 7,000+ annotated hand gesture images achieved 96.7% classification accuracy. The system captures webcam input, extracts 21 skeletal coordinates via MediaPipe, and predicts characters using the TensorFlow-trained model. Detected letters are displayed as text overlays and aggregated into words using space-delimited boundaries. Results highlight the system's potential for bridging communication gaps in deaf communities. Future work will expand to dynamic gestures and sentence-level context.

Tools used

1. **Core Development Tools**

   Python: Primary programming language for system integration.

   OpenCV: Handles video capture, frame processing (640×480 resolution), and text overlay.

   MediaPipe: Provides real-time hand landmark detection (21 skeletal points at 24 FPS)

   TensorFlow/Keras: Implements CNN architectures (e.g., MobileNetV2) for gesture classification.

2. **Supporting Libraries**

   OpenCV(cv2), NumPy, cvzone, tensorflow

Project Development Steps

1. **Environment Setup**
   IDE: PyCharm with Python 3.8 virtual environment
   Libraries Installed: OpenCV-python, cvzone, NumPy, tensorflow.
   Hardware: Webcam setup verification (test with OpenCV's cv2.VideoCapture(0))
2. **Data Collection**
   Captured 250-300 images per ASL letter (A-Y)
   Used 's' key to save hand-cropped images to class folders (Data/A, Data/B, etc.)
   Applied aspect ratio preservation (white background padding)
   Output: Dataset of 300x300 RGB images sorted into 25 class folders
3. **Model Training (Teachable Machine)**
   Zipped class folders uploaded to teachable machine and Trained CNN model with default settings (50 epochs, 32 batch size).
   Exported model as:
   keras_model.h5 (TensorFlow model) and labels.txt (class names)
   Integration: Model files placed in project's /Model folder
4. **System Implementation**
   Hand Detection: cvzone.HandDetector isolates single hand
   Image Preprocessing, Inference
   UI Controls: (Manual word formation using keys)
   's' to add letters, 'x' to backspace, 'c' to clear.
5. **Deployment**
   Executable Creation: PyInstaller bundled app with model files
   Instructions for:
   Hand positioning (30-50cm from webcam)
   Lighting requirements (avoid backlight)
   Gesture-holding duration (1-2 seconds per letter)

Conclusion:

This real-time sign language recognition system successfully converts ASL gestures to text using Python and computer vision, achieving 96.7% accuracy. The integration of Teachable Machine for CNN training and cvzone for hand tracking. While limited to static alphabets, the framework provides a scalable foundation for advanced gesture-based communication tools.