Real-Time Sign Language Recognition - Project Report

1. Introduction

In an increasingly inclusive digital world, bridging communication gaps through technology is vital. Sign language is the primary means of communication for millions of deaf and hard-of-hearing individuals. This project presents a real-time sign language recognition system designed to interpret static American Sign Language (ASL) hand gestures using a standard webcam. The aim is to convert recognized hand gestures into text in real time using computer vision and deep learning techniques.

2. Abstract

This project builds a real-time sign language recognition system using a Convolutional Neural Network (CNN) trained on the ASL Alphabet dataset. The system uses OpenCV and MediaPipe for hand detection and gesture cropping, and TensorFlow for classification. It classifies static gestures into 29 categories (A-Z, 'space', 'del', 'nothing') and displays the prediction on the screen. Optional features include forming complete words and converting them to voice using text-to-speech (TTS).

3. Tools Used

- Python 3.9
- OpenCV
- MediaPipe
- TensorFlow + Keras
- NumPy, Scikit-learn
- Matplotlib (optional)
- ASL Alphabet Dataset

4. Steps Involved in Building the Project

Dataset Preparation:

- Used `asl_alphabet_train` for training and `asl_alphabet_test` for validation

- Resized all images to 64-64 pixels and normalized them

Model Building & Training:

- Built a CNN with Conv2D, MaxPooling2D, Flatten, and Dense layers

- Used categorical cross-entropy loss and Adam optimizer

- Achieved ~99.8% training and ~98.8% validation accuracy

Saving Model:

- Trained model saved as `sign_language_model.h5`

- Class labels saved in `classes.npy`

Real-Time Recognition:

- Captured frames using webcam (OpenCV)

- Detected hand region using MediaPipe and cropped the ROI

- Resized ROI, passed it to the model, and predicted the sign

- Displayed predicted letter on-screen in real time

Optional Features:

- Combined letters into words using buffering

- Voice output using `pyttsx3` for recognized text

5. Results

- Training Accuracy: ~99.8%

- Validation Accuracy: ~98.8%

- System Speed: Real-time recognition (<50ms/frame)

- Usability: Smooth UI using webcam and label overlay

- Robustness: Detects well under varied lighting if hand is clearly shown

6. Deliverables

- train_model.py - Trains CNN model
- realtime_predict.py - Uses webcam for real-time recognition
- sign_language_model.h5 - Trained CNN model
- classes.npy - Class label mapping
- README.md - GitHub-ready project documentation
- PDF Report - This detailed document
- (Optional) Demo video of working model

7. Conclusion

This project successfully implements a deep learning-based system to recognize static sign language alphabets in real time. By combining TensorFlow, MediaPipe, and OpenCV, it creates a reliable solution to help break communication barriers. With enhancements like dynamic gesture support, sentence formation, and multilingual datasets, it can evolve into a fully functional sign language interpreter system.