Real-Time Sign Language Recognition System

By Priyanshu Walia and Sanskriti Mahore

1. Project Overview

This project aims to develop a deep learning-based real-time sign language recognition system to facilitate communication between deaf individuals and others. The system uses computer vision techniques and deep learning models to interpret sign language gestures from video input and translate them into text or speech in real time.

Key Objectives:

- Capture hand gestures and movements from video feeds.
- Build a deep learning model that can classify different signs.
- Develop a real-time system for continuous gesture recognition and translation.

2. Problem Statement

Sign language is a vital mode of communication for the deaf community. However, the lack of knowledge of sign language among the general population creates a communication barrier. This project aims to bridge that gap by building a real-time system that can automatically recognize and translate sign language gestures into text or speech, helping facilitate communication between deaf and hearing individuals.

3. Dataset Description

3.1 Sign Language Dataset:

- **Source:** Publicly available datasets like the American Sign Language (ASL) Dataset or custom-collected video data.
- **Contents:**
    - Video clips of hand gestures representing different signs.
    - Labels corresponding to the gesture meaning.

Features:

- **Hand landmarks:** Key points representing the position of fingers, wrists, and joints.
- **Gesture movement:** Tracking of movement across frames to capture continuous gestures.
- **Orientation and speed:** Variations in the speed and orientation of gestures.

4. Model Architecture

4.1 Preprocessing for Gesture Recognition:

- **Objective:** Extract relevant features from video input for gesture recognition.
- **Architecture:**
    - **Preprocessing:** Extract hand landmarks using models like OpenPose or MediaPipe. Normalize hand positions and preprocess video frames.
    - **Embedding:** Convert video sequences into feature vectors using Convolutional Neural Networks (CNNs) to capture spatial information.
    - **Temporal Modeling:** Use Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks to capture temporal relationships in gestures.

4.2 Gesture Recognition Model:

- **Objective:** Classify hand gestures into corresponding sign language words or phrases.
- **Architecture:**
    - **Input Layer:** Preprocessed hand landmarks and video frames.
    - **CNN Layers:** To extract spatial features from the video frames.
    - **LSTM Layers:** To capture temporal dependencies across frames.
    - **Dense Layers:** Fully connected layers for gesture classification.
    - **Output Layer:** Softmax activation for multi-class gesture classification.

5. Data Preprocessing and Feature Engineering

5.1 Data Preprocessing:

- **Video Preprocessing:** Convert videos into frames, detect and segment hands from the background.
- **Hand Landmark Extraction:** Use MediaPipe to extract 2D/3D coordinates of the hand's key points (fingers, wrist).
- **Normalization:** Normalize hand positions to handle variability in hand size, orientation, and camera angles.

5.2 Feature Engineering:

- **Velocity and Trajectory:** Compute the velocity of hand movements and their trajectory.
- **Shape and Finger Positioning:** Extract features based on finger positions and distances between hand key points to define static signs.

6. Model Training

6.1 CNN + LSTM for Gesture Recognition:

- **Loss Function:** Categorical cross-entropy for multi-class classification of signs.
- **Training Process:** Train the CNN to capture spatial information, and use LSTM for temporal sequencing of hand movements across frames. Fine-tune the model to minimize classification errors.

7. Deployment Strategy

7.1 Real-Time Gesture Recognition System:

- Develop a system that:
    1. Captures video input from a webcam.
    2. Processes the video frames in real-time using the CNN-LSTM model.
    3. Displays the recognized sign language gestures as text or converts them to speech.

7.2 User Interface:

- A mobile or web-based application that:
    - Takes real-time video input.
    - Displays recognized gestures or provides a speech output.

8. Evaluation Metrics

- **Accuracy:** Overall performance in classifying the gestures correctly.
- **Precision & Recall:** Measure how well the system recognizes each gesture and minimizes false positives.
- **AUC-ROC:** To measure the model's ability to discriminate between different gestures.

9. Results and Future Improvements

Results:

- Report accuracy, precision, recall, and AUC-ROC on the test dataset.
- Display examples of real-time sign language recognition.

Future Improvements:

- Expand the system to recognize continuous gestures, representing entire phrases or sentences.
- Support for multiple sign languages like ASL, BSL, etc.
- Improve robustness by incorporating additional datasets and environmental conditions.

10. Conclusion

This project demonstrates the application of deep learning in real-time sign language recognition. By using CNN-LSTM networks, the system can accurately interpret hand gestures and translate them into text or speech, enabling better communication for deaf individuals.