International Research Journal of Education and Technology
Peer Reviewed Journal, **ISSN: 2581-7795**

SIGN SPEAK SIGN LANGUAGE TRANSLATOR

Dr. Antony Cynthia, M.Sc., M.Phil., Ph.D. Assistant Professor,

Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

Renjith K.S, Department of Computer Applications,

Sri Krishna Arts and Science College, Coimbatore-641 008

renjithks23bcc141@skasc.ac.in

Abstract

Communication barriers between deaf or hard-of-hearing individuals and the hearing population remain a persistent global challenge. Although sign language serves as a primary communication medium for millions of individuals, limited societal proficiency in sign language creates a divide that impacts education, employment, healthcare, and daily social interaction. This research presents the design, development, and evaluation of a real-time American Sign Language (ASL) recognition system powered by computer vision and artificial intelligence. The system integrates MediaPipe-based hand landmark extraction, geometric feature engineering, and TensorFlow-based classification models within a modern full-stack architecture consisting of a Python FastAPI backend and a React frontend.

The proposed solution recognizes 26 ASL alphabets (A–Z), 10 numerical digits (0–9), and over 80 common words and phrases including greetings, emotions, actions, and questions. The system operates at approximately 30 frames per second (FPS) with latency under 100 milliseconds under optimal hardware conditions. It incorporates a 2-second debounce mechanism to reduce duplicate detections and supports sentence formation with automatic spacing and text-to-speech (TTS) functionality for enhanced accessibility. The model achieves 85–95% accuracy in controlled lighting conditions and demonstrates resilience against minor variations in hand pose and environmental noise.

The system emphasizes single-handed ASL gestures for practical deployment feasibility and highlights future directions including multi-language support, dynamic continuous signing recognition, and mobile deployment optimization. This research contributes to assistive AI technologies by offering a scalable, real-time, production-oriented framework for accessible communication.

Keywords

American Sign Language (ASL), Sign Language Recognition, Computer Vision, MediaPipe, TensorFlow, Real-Time Detection, Hand Landmark Detection, Assistive Technology, Deep Learning, Accessibility AI.

1. Introduction

1.1 Background

According to the World Health Organization (WHO), over 430 million people worldwide experience disabling hearing loss. Sign language is the primary communication method for a substantial portion of this population. However, a majority of the hearing community lacks fluency in sign language, creating communication challenges across education, employment, public services, and healthcare.

Traditional solutions involve human interpreters, but interpreter availability is limited and costly. Furthermore, real-time assistance may not always be available in informal or emergency situations. Advances in computer vision and artificial intelligence present an opportunity to develop automated sign language recognition systems capable of bridging this communication gap.

1.2 Problem Statement

Despite significant research in gesture recognition and sign language detection, practical real-time applications face several limitations:

- High latency processing unsuitable for conversation.
- Requirement of specialized hardware (depth sensors, motion gloves).
- Low robustness to lighting variations.
- Poor sentence formation and repetition handling.
- Limited real-world deployment readiness.

There is a need for a lightweight, real-time, scalable, and user-friendly system that:

1. Operates using standard webcam input.
2. Detects ASL gestures with high accuracy.
3. Enables sentence construction.
4. Supports speech output.
5. Maintains low computational overhead.

1.3 Objectives

The primary objectives of this research are:

- To develop a real-time ASL detection system using MediaPipe and TensorFlow.
- To recognize 26 alphabets, 10 numbers, and 80+ common words.

- To maintain latency below 100ms.
- To achieve accuracy above 85% under optimal conditions.
- To implement sentence formation and text-to-speech functionality.
- To design a scalable, full-stack web architecture.

2. Literature Review

2.1 Traditional Sign Language Recognition Approaches

Early systems relied on:

- Data gloves with sensors
- Depth cameras (e.g., Microsoft Kinect)
- Colored markers for hand segmentation

While accurate, these methods required expensive hardware and limited user accessibility.

2.2 Computer Vision-Based Methods

With advancements in OpenCV and convolutional neural networks (CNNs), camera-based recognition became feasible. Researchers employed:

- Image classification models (e.g., ResNet, VGG)
- Hand segmentation via background subtraction
- Pixel-based classification

Limitations included:

- Sensitivity to lighting
- Background interference
- Large computational cost

2.3 MediaPipe-Based Landmark Detection

Google's MediaPipe provides robust real-time hand tracking with 21 3D landmarks per hand. Landmark-based recognition offers several advantages:

- Invariance to background clutter
- Lower computational cost
- Reduced data dimensionality
- Improved generalization

2.4 Gaps in Existing Systems

Most systems focus on:

- Static gesture classification only
- Offline dataset evaluation
- No sentence-building logic
- No debounce mechanisms
- No speech output

This project addresses these practical gaps.

3. System Architecture

3.1 Overall Architecture

The system follows a full-stack architecture:

Frontend: React.js
Backend: FastAPI (Python)
AI Model: TensorFlow
Computer Vision: OpenCV + MediaPipe
Speech Module: Text-to-Speech (TTS)

The architecture pipeline:

1. Webcam frame capture
2. Frame preprocessing
3. Hand landmark extraction
4. Feature vector creation
5. Model prediction

6. Confidence scoring
7. Debounce logic
8. Sentence building
9. Text-to-speech conversion

4. Methodology

4.1 Data Collection

The dataset includes:

- 26 alphabet classes
- 10 number classes
- 80+ word/phrase classes

Data sources:

- Self-recorded webcam samples
- Augmented landmark variations
- Public ASL datasets (filtered)

Each class contains approximately 500–1000 samples.

4.2 Hand Landmark Extraction

MediaPipe detects 21 hand landmarks:

- Wrist
- 4 joints per finger
- Fingertip positions

Each landmark provides:

- x-coordinate
- y-coordinate
- z-coordinate

Total features per frame:
$21 \times 3 = 63$ features

These normalized coordinates are used instead of raw pixel data.

4.3 Feature Engineering

To improve robustness:

- Relative distances between key joints
- Finger angle calculations
- Normalization against wrist position
- Scale invariance adjustments

This reduces noise and improves model consistency.

4.4 Model Architecture

A TensorFlow-based neural network classifier was implemented.

Model Layers:

- Input Layer: 63 features
- Dense Layer (128 neurons, ReLU)
- Dropout (0.3)
- Dense Layer (64 neurons, ReLU)
- Dense Output Layer (Softmax activation)

Loss Function: Categorical Crossentropy
Optimizer: Adam
Learning Rate: 0.001
Epochs: 50
Batch Size: 32

4.5 Training and Validation

Training split:

- 80% training
- 20% validation

Performance metrics:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion matrix analysis

5. Implementation

5.1 Backend (FastAPI)

FastAPI handles:

- Frame reception
- Landmark processing
- Model inference
- JSON response

Advantages:

- Asynchronous performance
- Lightweight
- Easy deployment

5.2 Frontend (React)

Features:

- Webcam integration
- Start/Stop detection
- Clear sentence button
- Confidence display
- Speaker icon

5.3 Debounce Mechanism

Problem:
Repeated predictions cause duplicated letters.

Solution:
2-second debounce window.

Logic:

- Store last detected sign
- Accept new sign only if:
  - Confidence > threshold (e.g., 80%)
  - Time gap > 2 seconds

5.4 Text-to-Speech Integration

After sentence construction:

- TTS engine converts text to speech
- Enhances usability in public settings

6. Performance Evaluation

6.1 Accuracy

Under optimal lighting:

- Alphabets: 92–95%
- Numbers: 90–94%
- Words: 85–92%

6.2 Latency

Measured end-to-end:

- Average latency: 85–100ms
- Frame rate: ~30 FPS

6.3 Robustness Testing

Tested across:

- Different lighting conditions
- Skin tones
- Background variations
- Slight hand rotations

Performance drop observed under:

- Extremely low lighting
- Severe motion blur

7. Challenges and Solutions

7.1 Lighting Variations

Solution:

- Normalized landmark coordinates
- Increased training diversity

7.2 Hand Steadiness

Solution:

- Debounce logic
- Confidence threshold filtering

7.3 Class Similarity

Certain letters (e.g., M and N) visually similar.

Solution:

- Increased dataset
- Added angular feature engineering

8. Applications

8.1 Education

- Classroom communication aid
- Learning ASL interactively

8.2 Healthcare

- Patient-doctor communication
- Emergency assistance

8.3 Public Services

- Government offices
- Banking systems

9. Ethical Considerations

- No biometric storage
- Real-time processing only
- No facial recognition
- No personal data retention

Bias mitigation:

- Diverse training dataset
- Multi-skin tone sampling

10. Limitations

Be realistic:

1. Single-hand detection only.
2. Static gestures only.
3. No continuous sentence prediction.
4. Accuracy decreases in poor lighting.
5. Not a replacement for human interpreters.

11. Future Scope

- Two-hand gesture recognition
- Continuous sign recognition using LSTMs

- Mobile deployment (TensorFlow Lite)
- Multi-language support (ISL, BSL)
- Cloud-based scalable deployment

12. Conclusion

This research presents a real-time, AI-driven ASL detection system integrating MediaPipe and TensorFlow within a modern web-based architecture. By combining landmark-based feature extraction, neural network classification, debounce control, and text-to-speech synthesis, the system achieves practical real-time usability with strong accuracy and low latency.

While limitations exist—particularly regarding continuous and two-handed signing—the system demonstrates the feasibility of deploying scalable, accessible AI-driven communication tools. With further refinement and expansion, such systems can play a significant role in reducing communication barriers for the global deaf community.