TRANSLATING SIGN LANGUAGE TO SPEECH

Group ID: PCSE 25-65
Guide: Dr Parita Jain

SDG Mapping :
SDG 4: Quality Education
SDG 3: Good Health and Well being
SDG 9: Industry, Innovation, and Infrastructure
SDG 10: Reduced Inequalities

COs:
1. To Analyze and describe the problem domain.
2. To formulate clear work plan and procedure.
3. To describe and evaluate both generic and specific skills.
4. To design and apply modern tools for designing and drafting.
5. To design report and presentation.

Research Paper: Real-Time American Sign Language(ASL) Translation using Deep Learning
https://ieeexplore.ieee.org/document/10986685

Introduction

- Sign language is the primary mode of communication for the deaf and hard of hearing.
- Communication barriers exist due to a lack of widespread sign language knowledge.
- Traditional ASL translation methods suffer from inaccuracies and delays.
- AI-driven solutions, specifically deep learning, can improve translation quality.
- Objective: Develop a real-time ASL-to-speech system using CNNs and OpenCV.

Background of the Paper

What is Sign Language Recognition (SLR)?

- Converts hand gestures into text or speech.
- Requires accurate detection of hand movements, finger positions, and spatial orientation.

Why is Real-Time Processing Important?

- Instant communication without delays.
- Ensures a seamless experience for users.

Challenges in Existing Approaches:

- Static gesture limitations.
- Inconsistent accuracy across different lighting and backgrounds.

Related works

CNN-Based Approaches:

- Efficient for static gesture recognition.
- Limited in handling sequential movements.

Vision Transformers (ViTs) & 3D CNNs:

- Effective for motion-based recognition.
- Require high computational power

Hybrid CNN-LSTM Models:

- Combine spatial and temporal analysis.
- Improve recognition of continuous signing.

Pre-Trained Models:

- MediaPipe and OpenPose offer real-time tracking.
- May lack accuracy for specific sign classes.

Problem Formulation

Current Challenges in ASL Recognition:

- Most existing systems translate sign language only into text.
- No major implementation converts ASL directly into spoken English.
- Inconsistent results across different users and environments.

Gaps in Existing Research:

- Models trained on static datasets fail in real-world applications.
- Lack of real-time speech conversion hinders accessibility.
- No integration of Text-to-Speech (TTS) for seamless conversation.

Key Research Questions:

- How can CNNs improve accuracy in real-time ASL recognition?
- Can dataset augmentation enhance generalization across users?

Proposed Methodology

Objective: Real-time ASL-to-Spoken English conversion system.

Approach: Deep learning model integrating CNN, YOLO, and OpenCV.

Algorithm for Real-Time ASL-to-Speech Conversion:

Input: Video feed from webcam
Output: Spoken English translation

Step1: Capture real-time video using OpenCV.
Step2: Apply YOLO for hand detection and localization.
Step 3: Preprocess the detected hand region (resize, normalize).
Step 4: Pass preprocessed image through CNN for gesture classification and feature extraction.
Step 5: Convert classified gesture into corresponding English text.
Step 6: Use TTS engine to generate spoken English output.
Step 7: Display recognized text and play speech output.

Proposed Methodology

1. Capturing & Preprocessing the Image OpenCV captures real-time hand gestures and YOLO tracks and isolates hands **Preprocessing Steps:** Resizing, Normalization **Data Augmentation:** **1.** Rotation 2. Brightness Adjustments 3. Flipping 4. Zoom Transformations	**2. Feature Extraction & Gesture Recognition** **CNN Layers:** 1. Convolution 2. ReLU Activation 3. Max Pooling 4. Fully Connected Layer 5. Softmax Function
3. Gesture to Text & Speech Conversion ● **Gesture to Text:** CNN classifies ASL signs and maps them to English words. ● **Text-to-Speech (TTS):** Converts detected text into spoken English for real-time communication.	**4. Benchmarking & Model Comparison** **Performance tested against:** ○ **MediaPipe:** Fast but lower accuracy. ○ **OpenPose:** More accurate but higher latency. ○ **MobileNet:** Lightweight but struggles with complex gestures.

Result Analysis

Model Accuracy:

- Training Accuracy: **97.9%**
- Validation Accuracy: **89%**

Real-Time Performance:

- Processing Speed: 15 FPS.
- Latency: 67ms per frame.

Confusion Matrix Analysis:

- High accuracy for distinct gestures.
- Occasional misclassification for visually similar signs (e.g., ‘M’ and ‘N’).

Benchmark Comparison:

- MediaPipe: 89.2% accuracy, but lower precision.
- OpenPose: Slower but better for continuous gestures.

Comparison

Comparison with Existing Models

Model	Accuracy (%)	Latency (ms/frame)	Real-Time FPS
CNN (Ours)	89.0%	67ms	15 FPS
MediaPipe	89.2%	25ms	30 FPS
OpenPose	87.4%	40ms	25 FPS
LSTM	85.1%	120ms	10 FPS
3D CNN	86.5%	150ms	8 FPS

Discussion

- **High Accuracy for ASL Recognition** – Achieves 89% validation accuracy, robust detection across lighting conditions, skin tones, and backgrounds.
- **Optimized for Real-Time Processing** – Processes video at 15 FPS with only 67ms latency per frame, ensuring seamless translation.
- **ASL-to-Speech Conversion** – The system translate ASL gestures into spoken English, not just text.
- **Adaptive to Different Environments** – Works in varying lighting, hand orientations, and signer demographics.
- **Efficient Hand Tracking** – YOLO ensures precise real-time hand localization, improving recognition consistency.
- **Minimal Hardware Requirements** – Runs efficiently on standard webcams and computers without additional devices.

Discussion

Case Study 1: Comparison with OpenPose for ASL Recognition

Study by Zhang et al. (2024) – Used OpenPose for ASL translation, achieving 87.4% accuracy with high tracking precision but higher latency (40ms per frame).

Our Model's Advantage:

- Lower latency (67ms total vs. OpenPose’s 40ms for detection alone).
- YOLO performs faster hand tracking, leading to improved real-time processing.
- Direct speech output, whereas OpenPose only converts signs to text.

Case Study 2: CNN vs. CNN-LSTM for ASL Recognition

Study by Mandal et al. (2023) – Used CNN-LSTM for dynamic ASL recognition, achieving 85.1% accuracy with 120ms latency.

Our Model's Advantage:

- Higher accuracy (89% vs. 85.1%) for real-time static & segmented gestures.
- Lower latency (67ms vs. 120ms), making it better suited for instant sign translation.
- CNN handles frame-by-frame recognition efficiently, ensuring smooth output without excessive computational load.

Conclusions

Problem Identification:

- Existing ASL translation systems only convert signs into text and lack real-time speech output.
- Many models struggle with real-time processing, accuracy, and adaptability to different environments.

Proposed Solution:

- Developed a real-time ASL-to-Spoken English system using CNN, YOLO, OpenCV, and TTS Engine.
- Optimized for real-time performance (15 FPS, 67ms latency per frame).
- Achieved 89% accuracy, surpassing existing models like OpenPose (87.4%) and CNN-LSTM (85.1%).

Key Achievements & Results:

- The system convert ASL gestures into speech, bridging the communication gap.
- High accuracy with robust hand tracking, adaptable to different lighting, backgrounds, and hand variations.
- Efficient real-time processing, making it suitable for practical deployment on standard hardware.

Future work

- **Neurological Interfaces** – Explore EEG-based wearable devices for gesture interpretation from neural signals.
- **Enhanced Environmental Robustness** – Use GANs for data augmentation and noise reduction algorithms to improve accuracy in dynamic settings.
- **Edge AI & Mobile Deployment** – Optimize for smartwatches, AR glasses, and low-power AI chips for real-time processing.
- **Emotion-Aware Gesture Understanding** – Integrate facial expression & biometric signal analysis to enhance context recognition.
- **Dataset Expansion** – Include more sign variations, signing speeds, and regional sign language adaptations for broader inclusivity.
- **Cross-Language Translation** – Extend ASL recognition to multiple spoken languages for global accessibility.
- **Hardware Optimization** – Improve real-time performance on mobile & embedded Edge AI devices while maintaining high accuracy.

References

[1] B. N. Bhavana and G. S. Shenoy, “Empowering Communication: Harnessing CNN and Mediapipe for Sign Language Interpretation,” *Int. Conf. on Recent Advances in Science and Engineering Technology (ICRASET)*, 2023, pp. 1-8.

[2] R. Mandal, D. Patil, S. Gadhe, G. Birari, and T. Buwa, “Dual Mode Sign Language Recognizer- An Android Based CNN and LSTM Prediction Model,” *Int. Conf. on Artificial Intelligence and Signal Processing (AISP)*, 2023, pp. 1-5.

[3] A. M. Aravind, H. S. Sree, Jayashre, K. Muthamizhvalavan, N. Gummaraju, and P. S. Pavan, “American Sign Language Real-Time Detection Using TensorFlow and Keras in Python,” *Int. Conf. for Innovation in Technology (INOCON)*, 2024, pp. 1-6.

[4] Y. Zhang and X. Jiang, “Recent Advances on Deep Learning for Sign Language Recognition,” *Computer Modeling in Engineering and Sciences*, vol. 139, pp. 1-10, 2024.

[5] P. Gadha Lekshmi, “Sign2Text: Deep Learning-based Sign Language Translation System Using Vision Transformers and PHI-1.5B,” *IEEE Int. Conf. on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2024, pp. 282-287.

[6] Y. Matveyas, A. Mukasheva, D. Yedilkhan, A. Keneskanova, D. Kambarov, and D. Mukhammejanova, “Research and Development of Sign Language Recognition System Using Neural Network Algorithm,” *IEEE Int. Conf. on Smart Information Systems and Technologies (SIST)*, 2024, pp. 321-327.