International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

Real-Time Indian Sign Language Translation Using Deep Learning and Multilingual Speech Technologies

[1]V. Vishnu Shankar, [2]B. Tharun Kumar, [3]N. Vignesh Kumar, [4]V. Venkat Charan Reddy

[1,2,3,4]Department of CSE (AI), Madanapalle Institute of Technology and Science, Madanapalle, India
E-mails: [1]vankadarivishnushankar@gmail.com, [2]tharunkimar10@gmail.com, [3]nvignesh123roy@gmail.com, [4]venkatcharan40@gmail.com

***Abstract* - This paper introduces a novel hybrid framework for Indian Sign Language (ISL) translation that performs real-time recognition of both static and dynamic gestures and generates multilingual outputs in both text and speech. Unlike existing systems that are limited to either static classification or single-language outputs, our approach integrates a fine-tuned ResNet50V2 model for static gesture classification (98.7% accuracy) and a YOLOv8m detector for dynamic word recognition (88.7% mAP@50). The system employs MediaPipe for efficient hand landmark extraction and incorporates frame-skipping and cooldown strategies to optimize real-time performance on CPU-based devices, achieving an average of 3.4 FPS without GPU acceleration. Recognized gestures are mapped to sequences, translated into eight Indian languages using Google Translate, and converted into synthesized speech using gTTS. Experimental results validate the system's robustness across gesture types and linguistic outputs. The proposed work is the first to offer a complete ISL-to-text-and-speech pipeline with integrated multi-language support,via a desktop User-interface. This makes it a scalable, low-cost assistive tool designed to enhance accessibility and communication for the hearing-impaired community in multilingual contexts.**

Keywords: Indian Sign Language (ISL) Recognition, Deep Learning for Gesture Recognition, ResNet50V2, YOLOv8m, Real-Time Sign Language Translation, Hand Gesture Detection with MediaPipe, Multilingual Text-to-Speech (TTS), Computer Vision.

I. INTRODUCTION

Communication barriers between the deaf-mute community and non-signers remain a critical issue, particularly in linguistically diverse regions like India. Indian Sign Language (ISL), unlike American Sign Language (ASL), predominantly relies on dual-handed gestures, posing significant challenges for computer vision due to frequent occlusion and hand overlap. While sensor-based Sign Language Recognition (SLR) systems have achieved high precision, their dependence on wearable hardware limits practicality, cost-effectiveness, and user accessibility.

Recent advances in deep learning and real-time vision frameworks have enabled robust vision-based SLR systems using standard video inputs. However, existing ISL recognition models typically focus on either static gesture classification or lack multilingual output support, restricting their real-world usability. To address these gaps, we propose a real-time, end-to-end ISL translation system that supports both static and dynamic gesture recognition with multilingual audio-visual outputs.

The framework employs ResNet50V2 for static hand sign classification, achieving 98.7% accuracy, and YOLOv8m for dynamic word detection with 88.7% mAP@50. Real-time gesture tracking is performed via MediaPipe, with optimized performance (3.4 FPS on CPU) enabled by frame-skipping and cooldown mechanisms. Recognized gestures are translated into eight Indian languages using Google Translate and vocalized using gTTS. The system is deployed through a desktop GUI with real-time feedback, making it a scalable assistive solution for inclusive, multilingual communication.

II. LITERATURE SURVEY

Sign Language Recognition (SLR) systems are generally categorized into sensor-based and vision-based approaches. Sensor-based methods, such as those using gloves or wearable motion sensors, offer high accuracy but are hindered by high costs and lack of usability in real-world environments [1], [13].

Early works using computer vision techniques relied on handcrafted features, including edge detection, contour extraction, and skin color segmentation [9], [13], but these approaches were sensitive to lighting and background noise. The introduction of deep learning significantly improved robustness and accuracy, especially through the application of Convolutional Neural Networks (CNNs) [6], [8], [14].

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

Several recent studies have applied CNN-based architectures for static gesture recognition in Indian Sign Language (ISL). For instance, Prakash et al. [14] developed a CNN-based system for ISL alphabet prediction aimed at educational applications. Similarly, Kamble [10] proposed a basic ISL-to-text pipeline using CNNs, though it lacked dynamic gesture support or multilingual outputs.

Dynamic gesture recognition remains a critical challenge due to temporal dependencies and hand articulation complexity. LSTM-based architectures have been used to model temporal dynamics effectively. Abraham et al. [1] and Tharsan et al. [15] proposed LSTM-driven ISL translation systems, but these were limited to a narrow vocabulary and single-language output. Sabharwal and Singla [12] introduced a machine learning-based approach with limited scalability and lacking sentence construction.

MediaPipe has also emerged as a valuable tool for extracting hand landmarks in real time. Goyal et al. [4] integrated MediaPipe Holistic with ISL recognition, although their model was constrained to static gestures. Meanwhile, YOLO-based models have shown promise in dynamic gesture detection due to their real-time object detection capabilities, as demonstrated in DeepSign by Kothadiya et al. [6], which used YOLO in conjunction with CNNs for static-to-dynamic integration.

Some hybrid systems exist, such as Thakar et al. [3], who used transfer learning with pre-trained CNNs, and Punekar [5], who proposed a rule-based translator with limited dynamic gesture capabilities. However, few systems address real-time sentence formation, multilingual translation, and full pipeline integration.

Unlike previous approaches that either focus solely on static gestures or lack real-time multilingual support, our proposed system integrates ResNet50V2 for static classification and YOLOv8m for dynamic recognition, coupled with MediaPipe for efficient hand tracking. The framework supports sentence-level construction and translates outputs into eight Indian languages using Google Translate and gTTS, offering a unified, deployable pipeline for real-time ISL recognition and communication assistance.

III. PROPOSED METHODOLOGY

A. Dataset Summary

The system is trained using two publicly available ISL datasets. The first, sourced from Kaggle, contains 42,000 labeled images across 35 classes (A–Z and digits 1–9). Each image is 64×64 pixels, and due to visual similarity, '0' is excluded to avoid confusion with the letter 'O'. This dataset supports static gesture classification.

Figure 1: Indian Sign Language Alphabets and Digits

The second dataset, obtained from Roboflow, contains 221 annotated images for 18 dynamic ISL signs representing common words and phrases such as "I", "thank-you", "want", and "your". Together, these datasets enable the system to recognize both alphabets and high-frequency ISL vocabulary.

B. System Overview

The video stream is acquired using OpenCV, and hand landmarks are extracted using MediaPipe. Each hand is represented as a set of 3D keypoints:

$$H = \{(x_i, y_i, z_i) \mid i = 1, 2, \ldots, N\}$$

Where (x_i, y_i, z_i) are the 3D coordinates of the i_{th} keypoint.

A fine-tuned ResNet50V2 model is used for classifying static signs (A–Z, 1–9). It is pre-trained on ImageNet and retrained on ISL data.The model's original layers are removed, and new ones are added to improve classification performance. Feature extraction is defined as:

$$F = \text{ResNet50V2}(X; \theta)$$

Where X is the input image (hand gesture),θ are the weights, and F is the output feature map.

Post-processing includes:

Flatten Layer:

$$\text{Flatten}(F)=\{f1,f2,\ldots,fN\}$$

Dense layer (512 neurons, ReLU):

$$Z_j = \sum_{i=1}^{N} w_{ji} f_i + b_j \;,\; a_j = \max(0, z_j)$$

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

Dropout Layer (20% rate):

$$\tilde{a}_j = \begin{cases} a_j, & with\ Probability\ p = 0.2 \\ x, & with\ probability\ 1-p \end{cases}$$

Softmax Output:

$$P(y = j \mid X) = \frac{\exp(z_j)}{\sum_{k=1}^{35} \exp(z_k)}$$

YOLOv8m is deployed to detect dynamic word-level gestures in real-time. The model is optimized using a composite loss function that integrates three components:

$$L_{total} = \lambda_{box} \cdot L_{box} + \lambda_{obj} \cdot L_{obj} + \lambda_{cls} \cdot L_{cls}$$

Where L_{box} represents the bounding box regression loss (CIoU), L_{obj} denotes objectness loss, and L_{cls} corresponds to classification error".

To optimize performance on CPU systems, the detection module processes **every 3rd frame**, achieving ~3.4 FPS.

Detected gestures are assembled into a text sequence:

$$S_t = f(S_{t-1}, G_t)$$

Where S_t is the sentence generated at time t, S_{t-1} is the previous sentence, and G_t represents the detected gesture at time t.

The sequence is translated into the target language using the Google Translate API. For example, the English phrase "Thank you" is converted to "धन्यवाद" in Hindi.

Google Text-to-Speech (gTTS) converts translated text into audio in one of eight supported Indian languages. Audio playback is handled via asynchronous threads to avoid blocking detection.

IV. IMPLEMENTATION

A. Dataset Preparation and Model Training

Step 1: Dataset collection: The system utilizes two datasets: (i) the Indian Sign Language (ISL) dataset with 42,000 images across 35 classes (A–Z, 1–9) for static gestures, and (ii) the Roboflow ISL Hand Recognition dataset with 221 labeled samples spanning 18 dynamic ISL words.

Step 2: Data Preprocessing: Images are resized (64×64 for static; 800×800 for dynamic), normalized to [0,1], and split into 80% training and 20% validation sets. Static image loading and augmentation are handled using ImageDataGenerator, while mosaic augmentation is applied during YOLOv8m training.

Step 3: ResNet50V2 Training: The ResNet50V2 model is initialized with ImageNet weights, with the top layers frozen. Custom dense layers with ReLU activation and dropout (20%) are appended. Training uses categorical cross-entropy loss with the Adam optimizer at a learning rate of 0.001 over 10 epochs.

Step 4: YOLOV8m Training YOLOv8m is trained using 800×800 resolution inputs over 25 epochs with multi-threaded data loaders and caching. Confidence and IoU thresholds are fine-tuned for optimal detection. The composite loss function includes CIoU (L_box), objectness (L_obj), and class prediction (L_cls) components.

Step 5: Model Evaluation: ResNet50V2 is evaluated using accuracy, precision, recall, and F1-score. YOLOv8m performance is measured using mean Average Precision at 0.5 IoU (mAP@50).

Table 1: Training Configuration

Model	Loss	Optimizer	Epochs
ResNet50V2	Categorical Cross-Entropy	Adam	10
YOLOV8m	YOLOV8 Composite Loss	SGD	25

B. Real-Time Gesture Recognition Pipeline

Step 1: Frame Capture: OpenCV captures live frames at 640×480 resolution. Frames are horizontally flipped to mirror the signer's orientation.

Step 2: Landmark Extraction: MediaPipe Hands detects 21 3D hand keypoints. Cropped bounding boxes with padding are resized based on model input requirements.

Step 3: Gesture Classification: Alphabet classification is performed by ResNet50V2 via softmax, while YOLOv8m detects dynamic gestures. A frame-skipping mechanism (1-in-3) maintains ~3.4 FPS on CPU.

Step 4: Sequence Processing: A temporal buffer accumulates detected gestures, forming complete phrases. Cooldown logic prevents duplicate entries and enhances contextual continuity.

Step 5: Multilingual Translation and Audio Output: The final sentence is translated using Google Translate API and synthesized into speech using gTTS. Playback is handled asynchronously to avoid blocking real-time detection.

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

The overall system architecture for real-time ISL recognition is illustrated in Fig. 2, demonstrating the processing pipeline from input video frames to text and speech output.

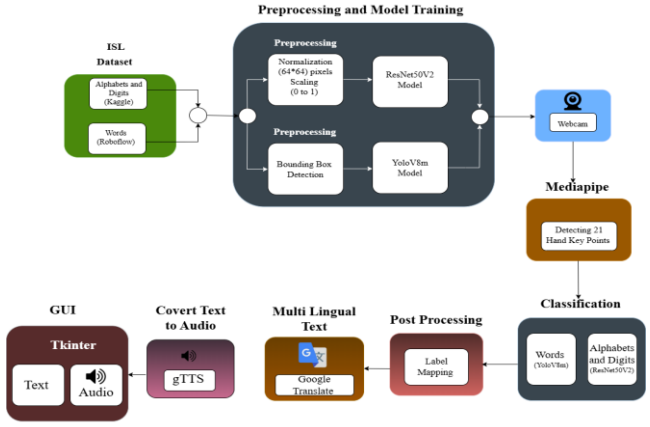

Figure 2: System Architecture for Real-Time ISL Recognition

V. EXPERIMENTAL RESULTS

This section outlines the hardware configuration, training results, and the criteria employed to evaluate the performance of the proposed Indian Sign Language recognition system, which utilizes ResNet50V2 and YOLOv8m. Both models have been fine-tuned to enable real-time gesture recognition on CPU-based systems..

A. Experimental Setup

All experiments were conducted on a system equipped with an Intel Core i7-12700H CPU (2.3 GHz), 32 GB RAM, and an NVIDIA RTX 3080 GPU (8 GB VRAM). The framework was implemented using Python with TensorFlow/Keras for ResNet50V2 and PyTorch (Ultralytics API) for YOLOv8m. OpenCV and MediaPipe were used for real-time video and landmark extraction, while Pillow handled multilingual text rendering and augmentation.

B. Train and Validating Results

Both datasets were split into 80% training and 20% testing sets. The ResNet50V2 model achieved a validation accuracy of 98.7% for static ISL classification. Accuracy and loss curves over training epochs are shown in Figs. 3 and 4, respectively.

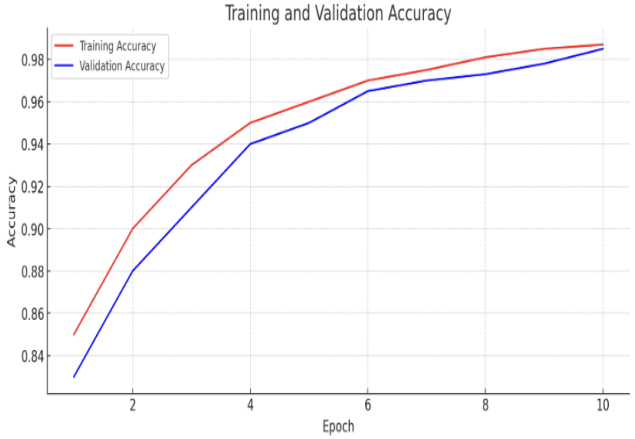

Figure 3: ISL ResNet50V2 Training Accuracy

Figure 4: ISL ResNet50V2 Training Loss

The YOLOv8m model, trained on 18 dynamic ISL gestures, achieved a mean Average Precision (mAP@50) of 88.7%, as illustrated in Fig. 5.

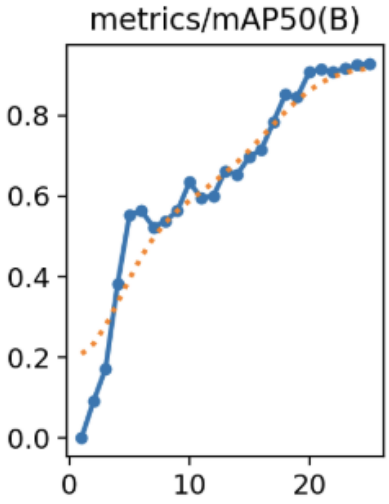

Figure 5: mAP@50 Curve of Yolov8m

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

C. Real-Time Performance Analysis

ResNet50V2 is optimized for event-driven inference on cropped static regions, incorporating a 1-second cooldown buffer to suppress redundant detections and support coherent sequence formation. YOLOv8m processes live video at approximately **3.4 FPS** using frame skipping (every 3rd frame), enabling real-time dynamic gesture detection on CPU.

Audio feedback is handled asynchronously to avoid blocking, and Pillow-based rendering allows real-time multilingual text overlays. Despite CPU-only deployment, the system maintains stable performance for both static and dynamic ISL signs.

A detailed summary of model performance, including task type, accuracy, and frame handling strategies, is presented in Table 2.

Table 2: Model Performance Summary

Model Used	Task Type	Accuracy (%)	FPS (Device)	Frame Handling
Resnet50V2	Alphabet/ Numbers	98.7	N/A (CPU)	Event Driven (1s Cooldown)
YOLOV8m	Phrase Detection	88.7	3.4 (CPU)	Frame Skipping (1 in 3)

VI. DISCUSSIONS

To assess the distinctiveness of the proposed system, a focused comparison was conducted across five recent ISL recognition frameworks selected from a broader literature pool of fifteen. Selection criteria included architectural diversity, year of publication, and output capabilities. Evaluation focused on three core metrics critical for practical deployment: **text output**, **multilingual speech synthesis**, and **availability of real-time graphical user interface (GUI)**.

While most existing systems support basic text output, they typically lack either multilingual speech capabilities or deployable user interfaces. For example, Punekar [5] integrates Hindi-only TTS but lacks a GUI. Others, such as DeepSign [6] and CNN-based transfer learning approaches [3], support text output but offer no multilingual feedback or GUI interactivity.

In contrast, the proposed hybrid framework combines ResNet50V2 for static gesture recognition with YOLOv8m for real-time dynamic detection, integrating multilingual speech synthesis via gTTS across **eight Indian languages**, and a desktop GUI built using OpenCV and Tkinter. This configuration delivers end-to-end translation, voice feedback, and user interactivity — bridging critical gaps in accessibility and real-time usability.

Table 2: Comparative Evaluation of Existing ISL Models Based on Real-Time Output Capabilities

Ref	Approach	Text Output	Speech & Multilingual	Real-Time GUI
[1]	LSTM based Translating Systems	✓	✗	✗
[3]	CNN + Transfer Learning	✓	✗	✓
[5]	Region Mapping + Hindi TTS	✓	✓ (Hindi Only)	✗
[6]	CNN + deepsign Architecture	✓	✗	✗
[8]	Deep Learning + Gesture Detection	✓	✗	✓
Ours	ResNet50V2 + YOLOV8m Hybrid	✓	✓ (8 langs via gTTS)	✓

VII. INTERFACE DEVELOPMENT

The user interface is built using Python's **Tkinter** library, designed for desktop platforms with a focus on real-time ISL gesture interaction. It features a live webcam feed where hand gestures are processed and recognized in real time. **Figure 6** shows the core layout of the ISL interface.

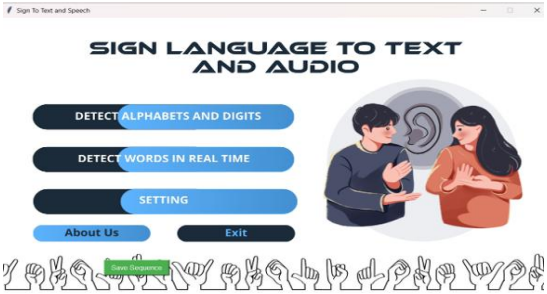

Figure 6: Interface of ISL

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

A language selection dropdown enables users to choose from eight supported languages, including English and regional options. Upon selection, the system converts recognized signs into translated text and generates corresponding speech output using **Google Text-to-Speech (gTTS)**. This feature is illustrated in **Figure 7**, which includes the dropdown and audio toggle.

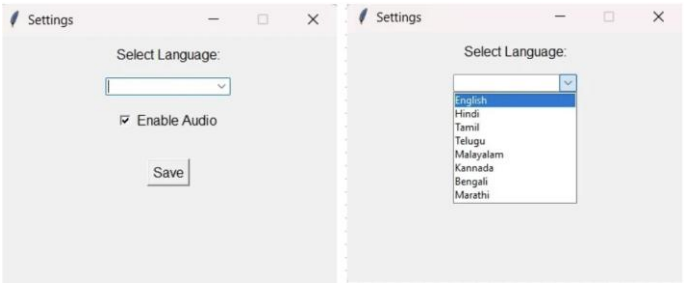

Figure 7: Language Selection and Enabling Audio

User preferences for language and audio are stored persistently, and confirmation dialogs provide feedback upon saving. An example of the "Settings Saved" prompt is shown in **Figure 8**.

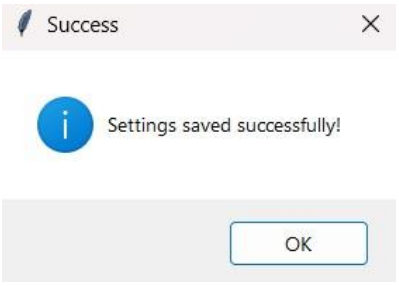

Figure 8: Settings Saved Confirmation

When operating in English mode, the system detects both letters and predefined words, rendering results as text and speech output. This functionality is depicted in **Figure 9**, where gesture recognition results are displayed on-screen alongside spoken feedback.

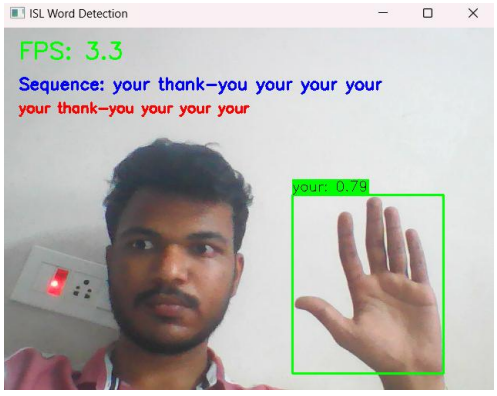

Figure 9: Displaying English Signs with Speech Output

For local languages, the same pipeline is followed post-translation. Once a regional language is selected, gesture recognition output is rendered in the target language, both visually and via speech, as shown in **Figure 10**.

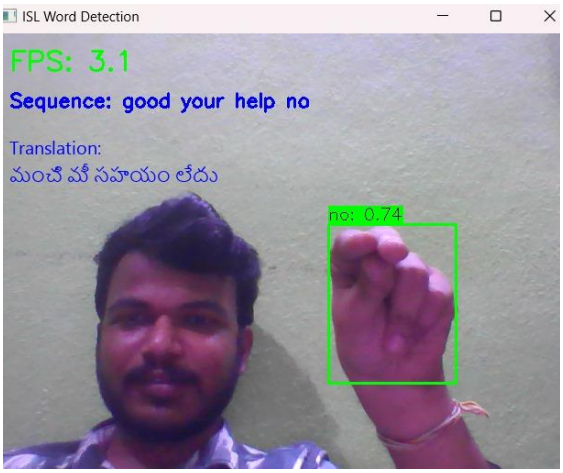

Figure 10: Displaying Local Language Signs with Speech Output

VIII. CONCLUSION

The proposed ISL recognition system delivers real-time translation of static and dynamic gestures into both text and speech via a desktop application. It integrates ResNet50V2 for static classification and YOLOv8m for dynamic detection, with hand landmark extraction enabled by MediaPipe for robust tracking.

Multilingual support is provided through Google Translate and gTTS, allowing voice output in eight Indian languages. The system interface, developed using Tkinter and OpenCV, offers real-time gesture capture, language selection, and audio-visual feedback.

Optimized for CPU-based execution, the application achieves an average inference speed of **~3.4 FPS** using frame skipping. The static gesture classifier attains **98.7% accuracy**, while the dynamic detector reaches **88.7% mAP@50**. Unlike existing systems, this implementation combines all critical functionalities — real-time inference, multilingual speech generation, and a deployable GUI — in a single pipeline optimized for accessible hardware.

Future Work

Future enhancements of the system will focus on expanding the dataset to include a broader range of ISL gestures, thereby improving model generalization and classification accuracy. To support more natural, sentence-level interactions, continuous sign recognition will be explored using temporal modeling techniques such as LSTMs or Transformer-based architectures. In parallel, efforts will be

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 9, Special Issue ICCIS-2025, pp 155-161, May-2025
https://doi.org/10.47001/IRJIET/2025.ICCIS-202525
Second International Conference on Computing and Intelligent Systems (ICCIS-2025)

directed toward developing a lightweight mobile version of the application to extend accessibility in real-world and resource-constrained environments. Optimization for low-power edge devices will further enable deployment on platforms such as smartphones and embedded systems. Additionally, integrating Augmented Reality (AR) to animate sign gestures from speech or text input presents a promising direction for enabling immersive, bidirectional communication between non-signers and the hearing-impaired community.

REFERENCES

[1] E. Abraham, A. Nayak, and A. Iqbal, "Real-time translation of Indian Sign Language using LSTM," *in Proc. of the International Conference on Machine Learning*, 2019.

[2] M. J. C. Samonte, C. J. M. Guingab, R. A. Relayo, M. J. C. Sheng, and J. R. D. Tamayo, "Using deep learning in sign language translation to text," *Proc. Int. Conf. Ind. Eng. Oper. Manag. *, 2022.

[3] S. Thakar, S. Shah, B. Shah, and A. V. Nimkar, "Sign language to text conversion in real time using transfer learning," *arXiv preprint arXiv:2211.14446v2*, 2022. [Online]. Available: https://arxiv.org/abs/2211.14446v2.

[4] K. Goyal, G. V., and Vellore Institute of Technology, "Indian sign language recognition using MediaPipe holistic," *in Proc. of the International Conference on Artificial Intelligence*, 2021.

[5] A.C. Punekar, "A translator for Indian sign language to text and speech," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 6, pp. 1640–1646, 2020. doi: 10.22214/ijraset.2020.6267.

[6] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A. Gil-González, and J. M. Corchado, "DeepSign: Sign language detection and recognition using deep learning," *Electronics*, vol. 11, no. 11, p. 1780, 2022. doi: 10.3390/electronics11111780.

[7] P. C. Badhe and V. Kulkarni, "Indian sign language translator using gesture recognition algorithm," *in Proc. 2015 IEEE Int. Conf. Computer Graphics, Vision and Information Security (CGVIS)*, 2015.

[8] H. K. Vashisth, T. Tarafder, R. Aziz, M. Arora, and A. Alpana, "Hand gesture recognition in Indian Sign Language using deep learning," *Engineering Proceedings*, vol. 96, 2023. doi: 10.3390/engproc2023059096.

[9] R. Janani, R. Harini, S. Keerthana, S. Madhubala, and S. Venkatasubramanian, "Sign language recognition system using computer vision," *Engineering Proceedings*, 2019.

[10] A.Kamble, "Conversion of sign language to text," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 5, pp. 1963–1968, 2023. doi: 10.22214/ijraset.2023.51981.

[11] "Sign Language-to-Text Dictionary with Lightweight Transformer Models," *in Proc. 32nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2023)*, 2023.

[12] S. Sabharwal and P. Singla, "Optimized machine learning-based translation of Indian sign language to text," *Int. J. Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 398–408, 2023. doi: 10.22266/ijies2023.0831.32.

[13] A.S. Ghotkar, R. Khatal, S. Khupase, S. Asati, and M. Hadap, "Hand gesture recognition for Indian sign language," *in Proc. 2012 Int. Conf. on Computer Communication and Informatics (ICCCI)*, 2012.

[14] S. S. Prakash, B. M. Devi, P. Arulprakash, M. Bandlamudi, and R. Radhakrishnan, "Educating and communicating with deaf learners using CNN-based sign language prediction system," *Int. J. Early Child. Spec. Educ. (INT-JECSE)*, vol. 14, no. 2, 2022. doi: 10.9756/INT-JECSE/V14I2.245.

[15] S. Ezhil Tharsan, S. Dharshan, G. Dinesh, and S. Saraswathi, "Real-time Indian sign language detection using LSTM and keypoint extraction," *Int. J. Computer Applications*, 2022.

Citation of this Article:

V. Vishnu Shankar, B. Tharun Kumar, N. Vignesh Kumar, & V. Venkat Charan Reddy. (2025). Real-Time Indian Sign Language Translation Using Deep Learning and Multilingual Speech Technologies. In proceeding of Second International Conference on Computing and Intelligent Systems (ICCIS-2025), published in *IRJIET*, Volume 9, Special Issue ICCIS-2025, pp 155-161. Article DOI https://doi.org/10.47001/IRJIET/2025.ICCIS-202525

****

