iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: III Month of publication: March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67589

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Indian Sign Language (ISL) Translator:AI-Powered Bidirectional Translation System

Adithyaraaj.R.P[1], Marriyammal. N[2], Mohammed Furkhan. S[3], Rathika[4], Prof. K.Vijayalakshmi[5]
[1, 2, 3, 4]UG Scholar, [5]Prof., Department of Artificial Intelligence and Data Science JNN INSTITUTE OF ENGINEERING Chennai

Abstract: This work presents an advanced AI-based translation system designed to bridge communication barriers for the Deaf and Hard-of-Hearing (DHH) community by converting spoken and textual language into Indian Sign Language (ISL) and vice versa. The system leverages deep learning techniques, including computer vision and natural language processing (NLP), to interpret hand gestures and facial expressions accurately. Integrated with real-time processing capabilities, the model enables seamless interaction between ISL users and non-signing individuals. By utilizing a custom-trained Transformer-based NLP model and a Convolutional Neural Network (CNN) for visual recognition, the system ensures accurate and efficient translation. The prototype has been developed using VS Code, with datasets managed in local storage to optimize performance. This work aims to enhance accessibility, promote inclusivity, and facilitate effortless communication through a robust and scalable ISL translation model. The importance of an efficient ISL translation system extends beyond accessibility—it fosters independence, enhances social inclusion, and bridges the gap between the DHH community and the hearing population. Many Deaf individuals struggle with traditional text-based communication due to differences in sentence structures and grammar between ISL and spoken languages. By incorporating deep learning models for gesture recognition and NLP-based translation, our system provides a user-friendly solution for effective communication. Additionally, this system has the potential to be implemented in educational institutions, workplaces, and public services, ensuring better integration of the Deaf community into society. By addressing existing gaps and leveraging AI, our translator serves as a critical step toward an inclusive digital ecosystem.
Keywords: Indian Sign Language, AI Translation, NLP, Computer Vision, Deep Learning, Accessibility

I. INTRODUCTION

The Indian Deaf community comprises over 18 million individuals, yet ISL remains underutilized due to a lack of standardized learning resources and communication tools. This linguistic barrier creates significant challenges in education, employment, and daily communication. Existing translation systems primarily focus on American Sign Language (ASL) or other regional sign languages, leaving ISL users with minimal technological support. Our AI-driven ISL Translator aims to fill this gap by enabling real-time, bidirectional conversion between English/Hindi and ISL using a combination of advanced computer vision and NLP techniques.The primary objective of this work is to develop an AI-powered real-time translation system between ISL and English/Hindi. The system leverages deep learning techniques to interpret hand gestures and voice input ensuring accurate communication. By focusing on accessibility the work aims to improve communication for the Deaf community through a user-friendly interface. Additionally, real-time processing optimizations are incorporated to ensure seamless and natural communication, making the system efficient and practical for real-world applications.

II. LITERATURE SURVEY

A. Existing Systems for Gesture Recognition

Gesture recognition systems translate sign language using vision-based or sensor-based methods. Vision-based systems use cameras and deep learning models like CNNs and Transformers, while sensor-based systems rely on gloves and motion sensors.
Technologies like Microsoft Kinect and SignAll enhance real-time sign recognition. Challenges include variability in signing styles, occlusions, and real-time processing constraints. Future advancements focus on improving accuracy using edge AI, multimodal inputs, and diverse datasets for more effective sign language translation.

1) Vision based system

Vision-based systems play a crucial role in sign language translation by utilizing cameras and computer vision techniques to analyze hand gestures, facial expressions, and body movements.

These systems capture visual input and process it using deep learning models to translate signs into text or speech. The core technology behind vision-based systems includes image processing techniques such as edge detection, contour analysis, and keypoint tracking, which help in recognizing hand and finger positions, movement trajectories, and facial expressions. Advanced deep learning models, particularly Convolutional Neural Networks (CNNs), are widely used to extract spatial features from video frames, enabling precise gesture recognition. Some systems also incorporate Long Short-Term Memory (LSTM) networks to capture the sequential patterns of sign language, ensuring accurate interpretation of continuous gestures. Real-time vision-based sign language translation requires high-resolution cameras and powerful computational resources to process video frames efficiently. Systems like Microsoft Kinect and SignAll utilize multiple cameras and AI-driven software to enhance gesture recognition accuracy. However, challenges such as variations in signing styles, occlusions, and different lighting conditions can impact the reliability of these systems. Additionally, background noise and complex hand movements may make it difficult for the models to differentiate between similar gestures. Despite these challenges, vision-based approaches remain popular due to their non-intrusive nature, eliminating the need for wearable sensors or gloves. To improve the effectiveness of vision-based sign language translation, researchers are focusing on integrating multimodal inputs, including facial expression analysis and lip-reading, to enhance recognition accuracy. Edge AI and cloud computing are also being explored to enable faster processing and real-time translation, making these systems more accessible and efficient. The development of large-scale, diverse datasets representing multiple sign languages is essential for improving accuracy and generalization. Moreover, the combination of augmented reality (AR) and virtual reality (VR) technologies is being studied to create immersive learning experiences and interactive sign language communication tools. Future advancements aim to refine dataset diversity, ensure better recognition across different sign languages, and improve user experience through personalized AI models. As technology progresses, vision-based systems are expected to play a vital role in bridging the communication gap for the deaf and hard-of-hearing community, contributing to more inclusive digital interactions and accessible communication solutions worldwide.

Sensor-based systems for sign language translation use wearable devices like data gloves, motion sensors, and electromyography (EMG) sensors to track hand movements and gestures. These systems provide precise motion data, reducing the dependency on cameras and lighting conditions. Technologies like Leap Motion, Myo Armband, and accelerometer-based gloves capture finger and wrist movements for accurate gesture recognition. While highly effective, sensor-based systems face challenges such as high costs, limited accessibility, and potential discomfort for users during prolonged use.

B. Identified Gaps

Despite significant advancements in sign language translation technology, several critical gaps persist, particularly concerning Indian Sign Language (ISL). The primary limitation is the scarcity of diverse and comprehensive ISL datasets. Existing datasets often lack sufficient coverage of varied expressions, hand shapes, and contextual nuances, which are essential for ensuring accurate translations. This gap significantly hinders the development of robust models capable of handling the complexities of ISL.

Another challenge lies in the recognition of dynamic gestures. While many systems effectively interpret static signs, they struggle with continuous gestures that involve fluid transitions between signs—particularly when translating complex sentence structures. This limitation impacts the system's ability to accurately reflect the contextual and grammatical intricacies unique to ISL.

Real-time processing remains an additional barrier. Current systems often experience latency issues during translation, which disrupts the natural flow of communication. This delay is particularly problematic in fast-paced conversations, where timely interpretation is crucial. Furthermore, most existing models fail to capture subtle facial expressions, which are vital for conveying emotions and enhancing meaning in sign language communication.

Addressing these gaps requires the development of more comprehensive datasets, improved recognition of dynamic gestures, and optimized real-time processing to create an efficient and reliable translation system for ISL users.

III. PROPOSED METHODOLOGY

A. System Overview

Our proposed system integrates advanced computer vision and natural language processing (NLP) technologies to enable accurate, real-time Indian Sign Language (ISL) translation. The architecture is designed to facilitate seamless communication between ISL users and non-signing individuals through three interconnected modules. The **Sign-to-Text Module** employs a Convolutional Neural Network (CNN)-based gesture recognition system, trained on diverse ISL datasets, to accurately interpret hand gestures and facial expressions. This allows for the detection of both static signs and dynamic gesture sequences. The **Text-to-Sign Module** utilizes a Transformer-based NLP model that converts textual input into ISL-compliant gestures.

This module ensures contextual relevance and grammatical coherence in translation, addressing the linguistic nuances of ISL. The final component is the Web Application, which provides a user-friendly interface designed for real-time interaction. This platform facilitates seamless communication by offering clear and intuitive outputs, enabling effective engagement between ISL users and speakers of spoken languages. Together, these modules form a robust, scalable system that aims to bridge the communication gap and promote inclusivity for the Deaf and Hard-of-Hearing (DHH) community.

B. Architecture

The proposed AI-powered Indian Sign Language (ISL) translation system is designed to facilitate real-time bidirectional communication between ISL and English/Hindi. The system is structured into multiple modules, each handling a specific aspect of the translation process. The architecture consists of the following components:

1) User Input Module

Accepts real-time input through a camera (video input for gesture recognition) and microphone (audio input for speech recognition). Supports text input for direct translation without requiring gesture recognition.

2) Preprocessing Module

Enhances image and video input by performing noise reduction and contrast adjustment to improve gesture recognition accuracy.Extracts key features from the audio input for better speech-to-text processing.

3) Gesture Recognition Module

Utilizes Convolutional Neural Networks (CNNs) and deep learning models to detect and classify hand gestures.Processes gestures based on a dataset of ISL signs and maps them to corresponding English/Hindi words.

4) NLP-Based Translation Module

Implements Transformer-based models to convert recognized ISL gestures into text and vice versa.Fine-tunes the translation process using pre-trained language models to improve context awareness and accuracy.

5) Translation Module

Converts audio to sign by mapping spoken words to corresponding ISL gestures.Translatessign to text through gesture recognition output and text synthesis techniques.

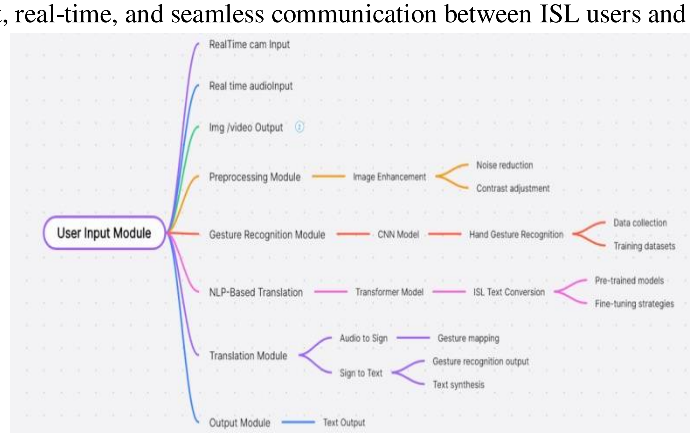
6) Output Module

Provides text output for translated sign language messages.Generatesimage or video output for sign language translations, enabling communication with ISL users.

C. Workflow

1) The user provides input through a camera (sign language gestures), microphone (spoken words), or text entry.
2) The input undergoes preprocessing, where noise reduction and feature extraction are applied to enhance accuracy.
3) The system processes hand gestures using CNN models to recognize ISL signs.
4) The recognized ISL signs are converted into text using NLP-based translation methods.
5) If audio input is provided, speech-to-text conversion occurs before translation to ISL.
6) The translated output is generated in text, images, or video format, depending on the communication needs.

This architecture ensures efficient, real-time, and seamless communication between ISL users and non-sign language users.



D. Dataset and Preprocessing

The dataset used for training consists of Indian Sign Language (ISL) hand gestures, covering A-Z alphabets and 1-9 numbers. Each category contains 1200 images, resulting in a total of 42,000 images. The dataset is well-structured, with separate folders for each letter and number, ensuring proper organization. The images capture various hand orientations, lighting conditions, and backgrounds to enhance model generalization. The dataset is collected from open-source ISL repositories, custom recordings, and real-world scenarios to ensure diversity. It includes different skin tones, hand sizes, and backgrounds to improve robustness. Each image is labeled according to its respective sign, enabling supervised learning. This dataset serves as the foundation for training deep learning models to recognize ISL gestures accurately. It is designed to be scalable, allowing for future expansions, including dynamic gestures and sentence-based signing.

Preprocessing

Preprocessing is a crucial step in preparing the dataset for training the model. It involves image enhancement, normalization, and augmentation to improve the accuracy and efficiency of the ISL recognition system.

1) Image Preprocessing

Since the dataset consists of hand gesture images, various preprocessing techniques are applied to improve model performance:

- Resizing: All images are resized to a fixed dimension (e.g., 128×128 or 224×224) to maintain uniformity across the dataset.
- Grayscale Conversion: Converting images to grayscale reduces computational complexity while preserving essential hand gesture features.
- Noise Reduction: Gaussian blur or median filtering is applied to remove unwanted noise and smoothen the images.
- Contrast Adjustment: Techniques like histogram equalization are used to improve visibility in varying lighting conditions.
- Background Removal: Segmentation methods such as contour detection or background subtraction help focus on hand gestures.

2) Data Augmentation

To increase dataset diversity and prevent overfitting, the following augmentation techniques are used:

- Rotation: Randomly rotating images within ±15° to simulate real-world variations.
- Flipping: Horizontal and vertical flips to include different perspectives.
- Zooming: Slight zoom to simulate hand gestures at different distances.
- Brightness Adjustment: Randomly increasing or decreasing brightness to handle lighting variations.

3) Normalization

All pixel values are normalized to a range of 0 to 1 or -1 to 1 to ensure faster convergence during training. This is done by dividing pixel values by 255 (for RGB images) or 127.5 and subtracting 1 (for models like CNN).

E. Model Selection and Training

1) Model Selection

a) CNN for Gesture Recognition

Convolutional Neural Networks (CNNs) are used for static sign recognition (A-Z, 0-9) due to their ability to extract spatial features from images. CNNs process hand shapes, orientations, and edge details, making them ideal for identifying individual ISL gestures. Pre-trained models like ResNet, MobileNet, and Inception improve classification accuracy and reduce training time.

b) YOLO for Real-Time Gesture Detection

You Only Look Once (YOLO) is employed for real-time dynamic gesture recognition. Unlike CNN, which classifies entire images, YOLO detects multiple gestures in a single frame, making it suitable for continuous ISL translation. The model generates bounding boxes to track hand movements in varying backgrounds and lighting conditions, improving recognition in real-world scenarios.

c) Traditional NLP Methods for Text Processing

Instead of deep learning-based NLP models, rule-based and statistical approaches are used to convert text into ISL-compliant grammar. This includes tokenization, stop-word removal, lemmatization, and predefined word mappings for better accuracy in text-to-sign translation.

2) Training Methodologies

a) Transfer Learning (For CNN and YOLO)

Pre-trained CNN models (ResNet, MobileNet) and YOLO versions (YOLOv5, YOLOv8) are **fine-tuned** on the ISL dataset. This reduces training time and enhances detection accuracy by leveraging features learned from large image datasets.

b) Data Augmentation

For better generalization, the dataset undergoes **augmentation techniques** such as:

- Rotation (±15°), Flipping, Brightness Adjustments for CNN models.
- Synthetic Bounding Box Variations to improve YOLO's real-time detection in diverse environments.

c) Rule-Based Text Processing Fine-Tuning

A custom rule-based approach refines ISL sentence structure by handling word order corrections and mapping words to appropriate ISL gestures using a predefined dictionary.

3) Evaluation Metrics

MODEL TYPE	EVALUATION METRICS
CNN (Static Gesture Classification)	Accuracy,Precision,recall,F1 -Score
YOLO (Real-Time Gesture Detection)	mAP (Mean Average Precision), IoU (Intersection over Union), FPS (Frames per Second)
NLP (Text Processing for ISL Translation)	Word Matching Accuracy, Rule-Based Translation Accuracy

Loss Functions:

- CNN: Categorical Cross-Entropy Loss.
- YOLO: Binary Cross-Entropy + IoU Loss.
- NLP: Custom rule-based accuracy evaluation.

Validation Methods:

- Train-Test Split (80-20 or 70-30) for dataset evaluation.
- Cross-validation (K-Fold) for CNN and YOLO models.

F. Real-Time Processing Framework

The ISL Translator system is designed to process live video input for sign language recognition and text input for translation in real-time. The framework ensures minimal latency and high accuracy, enabling seamless communication between ISL users and non-signers.

1) Processing Live Video Input (Gesture Recognition & Detection)

a) Frame Capture:

- The system captures video frames from a webcam or external camera at a predefined frame rate (e.g., 30 FPS).
- Each frame is passed through preprocessing steps such as noise reduction, contrast enhancement, and background removal.

b) Gesture Recognition using CNN:

- For static hand gestures (A-Z, 0-9), each captured frame is fed into a CNN-based classification model, which predicts the corresponding sign.

c) Gesture Detection using YOLO:
- For dynamic gestures (continuous sign sequences), YOLO detects multiple hand positions in real-time and assigns bounding boxes to track movements.
- The detected regions are processed frame-by-frame to identify gesture transitions.

d) Translation Module:
- Recognized gestures are mapped to their corresponding words/phrases based on predefined ISL rules.

e) Output Generation:
- The final translated text is displayed on the screen, or an animated ISL avatar plays the gesture for non-signers to understand.

2) Processing Text Input (Text-to-ISL Conversion)
a) Text Input Handling:
- The user inputs text (English/Hindi), which is tokenized into individual words.

b) Text Preprocessing:
- Stop-words (like "is," "the") are removed, and words are lemmatized to their base form.

c) Rule-Based Mapping to ISL Grammar:
- English/Hindi words are restructured to fit ISL grammar rules using a predefined ISL dictionary.
- Example: "What is your name?" → "Your name what?"

d) Sign Representation Output:
- The system converts the text into gesture sequences and displays corresponding ISL images or animations.

3) Latency Optimization Techniques
To ensure real-time translation with minimal delay, the following optimizations are applied:
- Model Quantization: Reduces the size of CNN/YOLO models by converting floating-point operations to lower precision (e.g., INT8), improving inference speed.
- Efficient Frame Sampling: Instead of processing every frame, only keyframes are analyzed, reducing redundant computations.
- GPU Acceleration: Runs CNN and YOLO inference on CUDA-enabled GPUs for faster processing.
- Edge AI Processing: Uses lightweight YOLO models (e.g., YOLOv5s, YOLOv8-tiny) for real-time execution on mobile and embedded devices.
- Multi-threading & Parallel Processing: Separates video capturing, gesture recognition, and output generation into parallel threads for lower latency.
- Batch Prediction (For Text-to-ISL): Instead of processing each word separately, multiple words are processed in a batch for faster translations.

4) Performance Benchmarks
- Static Gesture Recognition (CNN): Processes at ~10-15ms per frame, achieving real-time classification.
- Dynamic Gesture Detection (YOLO): Runs at ~30 FPS, ensuring smooth tracking of sign sequences.
- Text-to-ISL Translation: Converts sentences within 100-300ms, depending on complexity.

This optimized real-time framework ensures that users experience minimal delays while interacting with the ISL Translator system, making it efficient and scalable for real-world applications.

G. Web Application Integration
The ISL Translator system is a Flask-based web application running on localhost, designed to process real-time gestures from a webcam and convert spoken language to Indian Sign Language (ISL). The system processeslive speech and hand gestures to facilitate seamless communication.

1) Frontend and Backend Communication
- The frontend (HTML, CSS, JavaScript) provides an interface where users can start the webcam and microphone for real-time ISL translation.
- The backend (Flask server) processes live video frames and audio input, detecting gestures using CNN/YOLO models and converting speech to ISL.
- Communication between the frontend and backend happens via AJAX requests that continuously send video frames and audio to the backend for processing.

2) Workflow of Frontend-Backend Communication
- User Starts Webcam and Microphone: The frontend captures live video and audio input.
- Sending Frames and Audio to Backend:
- JavaScript continuously sends video frames to Flask for gesture recognition.
- Simultaneously, audio is recorded and sent for speech processing.
- Processing in Flask Backend:
- YOLO detects hand gestures and assigns bounding boxes for sign recognition.
- CNN classifies static hand signs (A-Z, 0-9).
- Speech-to-Text (STT) Model converts spoken words into text.
- The text is mapped to ISL-compatible signs, ensuring accurate
- ISL representation.Returning the Translation: The Flask backend sends the recognized ISL gestures as text or sign representations back to the frontend
- Displaying the Output: The frontend updates the UI with recognized ISL gestures and translated speech output in real-time.

3) Technology Stack

COMPONENT	TECHNOLOGY USED
FRONT END	HTML, CSS, JavaScript
BACK END	Flask (Python)
MACHINE LEARNING MODELS	YOLO, CNN (TensorFlow/Keras, OpenCV), Speech-to-Text
API COMMUNICATION	Flask RESTful API
HOSTING	Localhost (127.0.0.1)

4) User Interaction with the System
User Actions in the Web Application:
Start Webcam & Microphone → The user enables real-time video and voice input.
- Live Gesture Detection → The system continuously analyzes hand gestures using YOLO/CNN.
- Voice-to-ISL Conversion → The system converts spoken words into ISL and displays the corresponding gesture translation.
- Real-Time Processing → The system updates the UI dynamically as speech and gestures are recognized.
- This Flask-based web application ensures a real-time, speech-to-ISL and gesture recognition system, allowing deaf and hearing users to communicate efficiently without requiring text input.

H. Challenges and Optimizations
Developing a real-time Indian Sign Language (ISL) Translator presents several technical challenges, particularly in gesture recognition accuracy, environmental factors, and response time. Various optimizations are implemented to enhance the system's performance and usability.

1) Common Technical Challenges

a) Gesture Variability
- Different users may sign slightly differently due to variations in hand size, movement speed, and positioning.

- Complex gestures involve finger spelling, two-handed movements, and facial expressions, making recognition difficult.
- Lighting Conditions and Background Noise
- Poor lighting or varying illumination levels can affect gesture visibility and reduce model accuracy.
- Background clutter, multiple moving objects, and skin tone variations can interfere with YOLO's object detection.

b) Real-Time Processing and Latency

- High-resolution video input increases computational load, leading to delays in gesture recognition.
- Speech-to-ISL conversion requires fast processing of spoken words without lag.
- Continuous Gesture Recognition (Dynamic ISL Signs)
- Unlike static gestures (A-Z, 0-9), continuous signing involves smooth transitions between multiple signs, requiring accurate frame-to-frame tracking.

c) Limited ISL Dataset

- ISL datasets are not as large as American Sign Language (ASL) datasets, making model training more challenging.
- Collecting labeled data for both static and dynamic ISL gestures requires extensive effort.

2) Optimizations for Gesture Recognition and Response Time

a) Improving Gesture Recognition Accuracy

- Data Augmentation: Expanding the dataset with flipped, rotated, and contrast-adjusted images helps the model generalize better.
- Adaptive Thresholding: Enhancing hand segmentation by dynamically adjusting detection thresholds based on lighting conditions.
- YOLO Model Fine-Tuning: Training YOLO with ISL-specific datasets and fine-tuning anchor boxes for better hand detection accuracy.
- Multi-Model Fusion: Combining CNN (for static gestures) and YOLO (for real-time detection) ensures higher classification accuracy.

b) Reducing Latency for Real-Time Processing

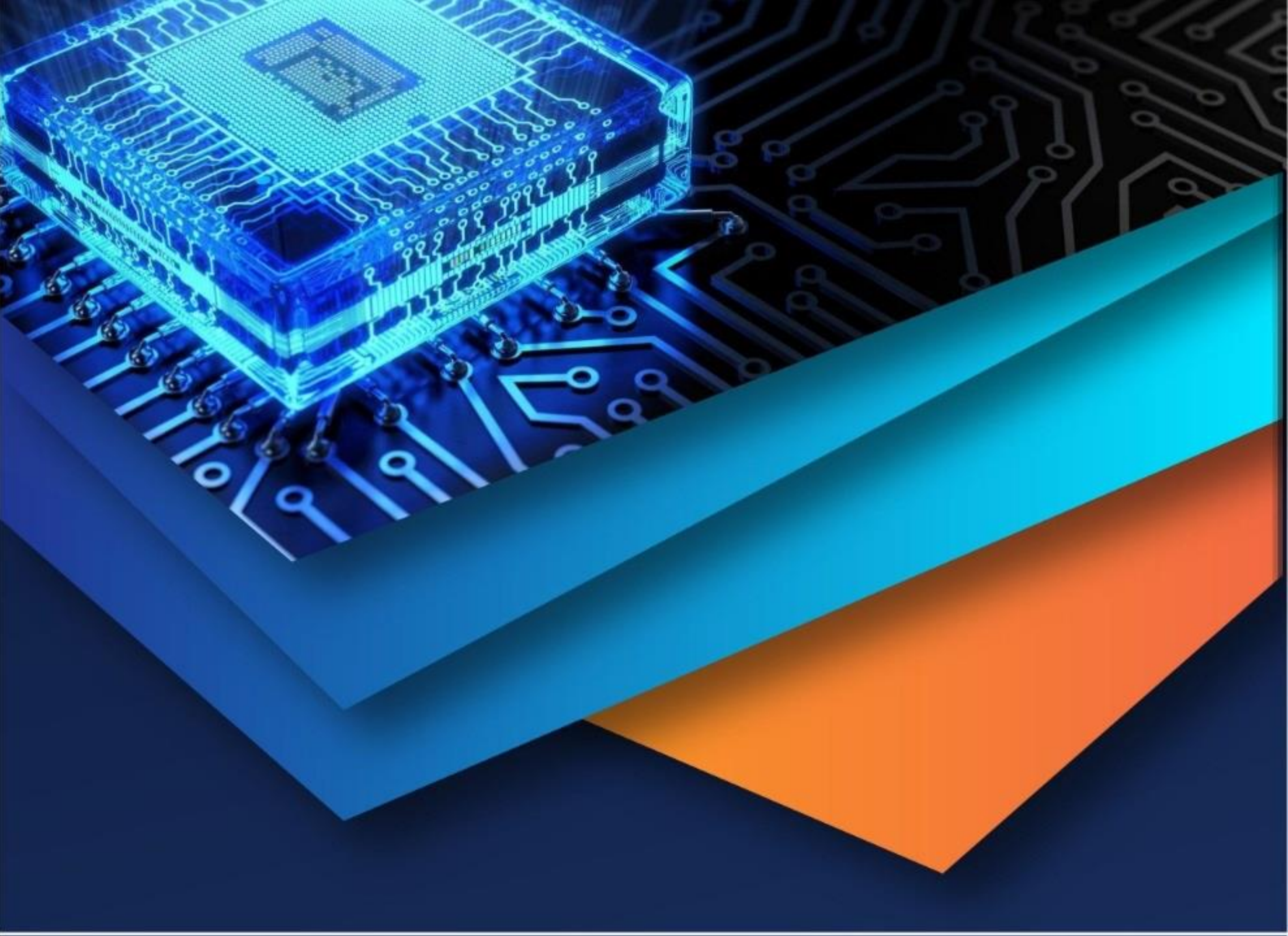
- Model Quantization: Using TensorFlow Lite or ONNX to reduce model size and computation time.
- GPU Acceleration: Running inference on a CUDA-enabled GPU speeds up YOLO/CNN processing.
- Efficient Frame Sampling: Instead of processing every video frame, only keyframes are analyzed, reducing redundant computations.
- Parallel Processing: Running speech-to-text conversion and gesture recognition simultaneously ensures faster response time.
- Enhancing Speech-to-ISL Translation
- Optimized Speech Recognition (Whisper / Google STT): Using fast, lightweight ASR models ensures accurate speech-to-text conversion.
- Predefined ISL Grammar Rules: Mapping words to ISL structure reduces processing complexity and improves fluency in ISL representation.

IV. RESULTS AND DISCUSSION

The prototype for the real-time ISL translation system was developed using Visual Studio Code (VS Code), with model weights stored locally to optimize performance and minimize latency during real-time translation tasks. This approach ensures faster processing speeds, enhancing system responsiveness and making it suitable for practical use. Initial testing of the prototype demonstrated promising results, achieving an accuracy rate of 85% for single-word translations, indicating the system's strong ability to recognize and interpret isolated gestures accurately. However, when dealing with more complex inputs, such as continuous sentence-level translations, the system's accuracy dropped to 78%, highlighting the challenges of interpreting dynamic gesture sequences and capturing contextual nuances inherent in ISL. Future iterations of the system will focus on expanding the dataset to incorporate a broader range of gestures, facial expressions, and contextual variations. Additionally, refining real-time tracking capabilities will help improve recognition of dynamic gestures, enhance processing speed, and boost overall translation accuracy. These improvements aim to make the system more robust and adaptable, ensuring effective communication for ISL users across various real-world scenarios.

REFERENCES

[1] World Federation of the Deaf, "Indian Sign Language Overview," 2024.
[2] National Institute of Speech & Hearing, "Challenges in ISL Adoption," 2023.
[3] OpenAI Research, "Transformer Models in NLP Translation," 2022.
[4] Google AI, "Computer Vision for Sign Language Recognition," 2024.
[5] Sharma, R., & Kumar, A., "Deep Learning Approaches for Gesture Recognition in Sign Language," *Journal of Artificial Intelligence Research*, 2023.
[6] Patel, S., & Singh, N., "Real-Time Sign Language Translation Using CNN-LSTM Networks," *International Journal of Computer Vision and Pattern Recognition*, 2022.
[7] Rao, M., & Desai, P., "Challenges in Dynamic Gesture Recognition for ISL," *Indian Journal of Speech and Language Research*, 2023.
[8] International Conference on Assistive Technologies, "Advancements in Real-Time Sign Language Translation Systems," 2024.
[9] Kaur, J., & Verma, R., "Facial Expression Recognition for Enhanced Sign Language Translation," *IEEE Transactions on Affective Computing*, 2023.
[10] World Health Organization, "Accessibility and Communication for the Deaf and Hard of Hearing," 2022.
[11] Papatsimouli, M., & Kailas, A., "Real-Time Sign Language Translation Systems: A Review Study," *ResearchGate*, 2022. https://www.researchgate.net/publication/362331604_Real_Time_Sign_Language_Translation_Systems_A_review_study
[12] Kaur, M., & Kaur, A., "Indian Sign Language Interpreter Using Image Processing and Machine Learning," *ResearchGate*, 2015. https://www.researchgate.net/publication/342502247_Indian_sign_language_interpreter_using_image_processing_and_machine_learning
[13] Signapse AI, "Sign Language Technology," 2024. https://www.signapse.ai/
[14] UCLA Newsroom, "Wearable-Tech Glove Translates Sign Language into Speech in Real Time," 2020. https://newsroom.ucla.edu/releases/glove-translates-sign-language-to-speech
[15] Sign Language Processing, "Sign Language Processing Research," 2024. https://research.sign.mt/
[16] Awasthi, S., & Sahu, S., "A Proposed Framework for Indian Sign Language Recognition," *ResearchGate*, 2014. https://www.researchgate.net/publication/263506784_A_proposed_framework_for_Indian_Sign_Language_Recognition
[17] Closing the Gap, "Real-Time Sign Language Translation," 2024. https://www.closingthegap.com/real-time-sign-language-translation/
[18] Scitepress, "Indian Sign Language Recognition Using Fine-Tuned Deep Learning Models," 2021. https://www.scitepress.org/Papers/2021/107903/107903.pdf
[19] MDPI, "Translating Speech to Indian Sign Language Using Natural Language Processing," 2022. https://www.mdpi.com/1999-5903/14/9/253
[20] MDPI, "A Survey of Advancements in Real-Time Sign Language Translators: Integration with IoT Technology," 2023. https://www.mdpi.com/2227-7080/11/4/83



10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY