iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70838

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

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 V May 2025- Available at www.ijraset.com

Deep Learning - Based Real Time Sign Language Translator Using YOLO

Noorul Moufica M[1], MeenaS[2], ParkaviP[3], IndhumathiA[4], SelvashanthiS[5]
Department of Computer Science and Engineering, M.I.E.T. Engineering College, Tiruchirappalli, Tamil Nadu, India

Abstract: ***This paper presents a comprehensive Real-Time Sign Language Translator that bridges the communication gap between hearing-impaired individuals and the hearing population. The proposed system integrates two major functionalities: (1) real-time detection of alphabets and numbers using a YOLOv8-based object detection model, and (2) sentence-level gesture recognition along with speech-to-text translation using deep learning and natural language processing techniques. The system utilizes YOLOv8, OpenCV, and MediaPipefor visual detection and classification of hand signs, and *Flask for creating a responsive web-based interface. In the first module, the system supports the recognition of 36 static signs, including A–Z alphabets and digits 0–9. In the second module, 32 sentence gestures are recognized in real time and translated into meaningful sentences. Additionally, the system captures voice input using a microphone, converts it to text via speech recognition, and translates it into Tamil or Hindi. The platform includes functionality for live detection, screenshot capture, video recording, and multilingual output display, offering an inclusive and practical solution for accessible communication.***

Keywords: Sign Language Recognition, YOLOv8, Speech-to-Text, Real-Time Detection, Tamil and Hindi Translation.

I. INTRODUCTION

Communication is a vital part of human interaction, yet millions of people with hearing and speech impairments face challenges in expressing themselves in a society largely dependent on spoken language. Sign language serves as a primary means of communication for the deaf and mute community. However, the lack of understanding of sign language among the general population creates a significant communication barrier. To address this issue, we propose a Real-Time Sign Language Translator system that bridges the gap using artificial intelligence (AI) and machine learning (ML) techniques. The project is designed in two integrated modules. The first module focuses on the detection and recognition of static hand gestures representing alphabets (A–Z) and numbers (0–9) using the YOLOv8 object detection model. The second module extends the system's capabilities to recognize sentence-level gestures and supports voice-based communication using speech-to-text translation. The system is implemented using Python-based libraries such as OpenCV, MediaPipe, and Ultralytics YOLOv8, and is deployed on a web platform built with Flask, HTML, and CSS. It enables real-time webcam-based gesture detection, supports 32 predefined sentence gestures, and provides multilingual translation of detected text into Tamil and Hindi. It also integrates features such as live video feed, recording, screenshot capture, voice input, and translation output, making it suitable for practical deployment in educational, social, and assistive technology domains. This paper presents the design, implementation, and evaluation of the system as a low-cost, real-time, hardware-free solution for inclusive communication.

II. LITERATURE REVIEW

Several studies have explored various approaches to sign language recognition using deep learning and computer vision techniques. In [1], “Sign Language to Text Conversion using Transfer Learning”, the authors present a deep learning approach for American Sign Language (ASL) recognition using the VGG16 architecture with transfer learning. The study emphasizes the need for an efficient model to improve the accuracy of sign-to-text conversion. By leveraging transfer learning, the model achieved an accuracy of 98.7%, a significant improvement over the baseline CNN model’s 94%. Image pre-processing techniques such as resizing and normalization enhanced the model's performance. Furthermore, the system was integrated into a mobile application using a Django-based backend and REST API for real-time predictions. In [2], “American Sign Language Recognition with CNN”, a CNN-based model is used to recognize ASL signs captured via a webcam. The authors applied image segmentation techniques to isolate hand gestures, improving feature extraction and model training. The study used a dataset of predefined static gestures and achieved a recognition accuracy of 95.8%. Real-time recognition and robust pre-processing were key aspects of this research, enhancing model efficiency for practical applications. In [3], “Real-Time ASL Recognition using GoogLeNet”, the researchers employed the GoogLeNet architecture with pre-trained weights to classify ASL gestures (A–E).

The model provided high accuracy and reduced computational load, making it suitable for real-time applications. Feature extraction and optimized classification were crucial for performance. The approach targeted first-time users and prioritized speed and accuracy in interactive environments. In [4], "Sign Language Translation using Time-Series NeuralNetworks", the authors implemented an LSTM-based model to recognize dynamic sign gestures. Unlike traditional CNNs that process static images, this model captured temporal dependencies within gesture sequences, resulting in improved recognition accuracy for continuous sign language. The dataset included time-series data of gesture motions, and the LSTM architecture translated these into meaningful textual output. In [5], "Hidden Markov Model for Gesture Recognition", the study utilized Hidden Markov Models (HMMs) for recognizing gestures from video input. Skin-color segmentation was used to track handmovements, and the system classified symbolic and deictic gestures based on statistical features. With a large dataset of 28,000 positive and 11,100 negative samples, the model achieved higher detection accuracycompared to conventional feature-based methods, demonstrating the viability of HMMs in gesture recognition systems.

III. METHODOLOGY

The methodology involves six major components that work together to recognize gestures, process speech input, and provide translated output through a web-based interface.

A. Dataset

The two datasets were used for training the gesture recognition models, both collected and annotated using the Roboflow platform. To find over 100k other datasets and pre-trained models, visit https://universe.roboflow.com

1) *Alphabet & Number Dataset:* Contains 36 classes (A–Z, 0–9) of hand gestures in various lighting and angles. Used to train the YOLOv8 model for static gesture recognition
2) *Sentence Gesture Dataset:* Includes 32 commonly used sign language phrases (e.g., Hello, Thank you, Water, Sorry, etc.). These were also annotated and exported from Roboflow in YOLO format.

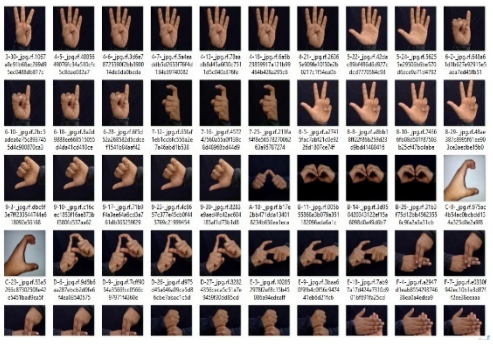


Fig.1 Datasetimage

Fig.2 Training dataset image

B. System Architecture

The system uses a modular architecture consisting of two input modes—gesture and speech. The YOLOv8 model handles gesture recognition while Google's Speech API manages voice input. Flask serves as the communication bridge between frontend and backend

1) *Client Side:* A Flask-based web interface allows users to select detection modes, view output, and interact via webcam or microphone.
2) *Server Side:* Processes video/audio, performs detection, constructs sentences, and translates the result.

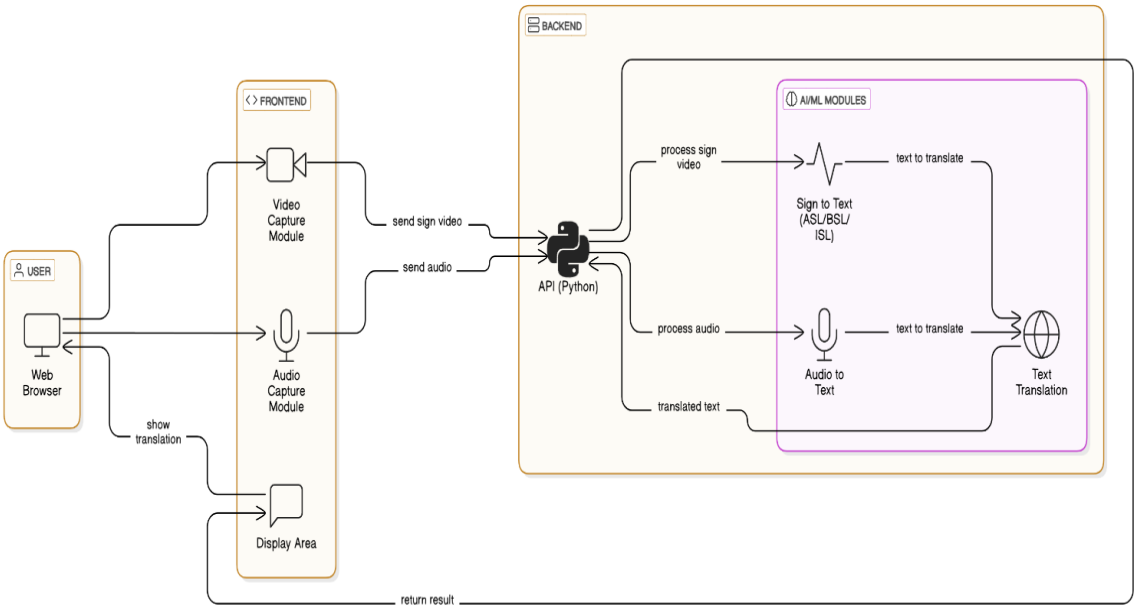


Fig.3 Architecture diagram for real time sign language translator

C. Gesture Recognition Process

1) *Input Acquisition:* Live video captured using OpenCV or uploaded image.
2) *Detection:* YOLOv8 detects hand signs and classifies them into letters, numbers, or phrases.
3) *Sentence Formation:* Detected words are accumulated and rearranged into complete sentences

D. Speech-to-Text Process

1) *Recording:* User activates microphone input through the web app.
2) *Transcription:* Audio is converted to English text using the Speech Recognition library.
3) *Filtering:* Output is cleaned for sentence consistency.

E. Translation and Output Integration

1) *Translation:* Detected or transcribed sentences are translated into Tamil or Hindi using Google Translate API.
2) *Display:* Both source and translated text are shown in real time with timestamp logs.

F. System Deployment

1) Deployment is handled locally through Flask.
2) YOLOv8 models are loaded dynamically to support webcam-based detection.
3) Users can control detection, translation, screenshots, and recordings using simple UI buttons.

IV. RESULTS AND ANALYSIS

The proposed system was evaluated through live tests using webcam input, uploaded images, and microphone-based audio input. Results show high accuracy, low latency, and smooth user interaction.

A. Experimental Setup

1) *Processor:* Intel Core i5 (8th Gen)
2) *RAM:* 8 GB
3) *Operating System:* Windows 10

4) *Tools:* Python, Flask, YOLOv8, OpenCV, MediaPipe, SpeechRecognition
5) *Input Devices:* Built-in webcam and microphone

B. Gesture Recognition Performance

1) *Accuracy:* 95% on YOLOv8 model trained with 36 classes (A–Z, 0–9) and 32 sentence gestures
2) *Inference Time:* 0.08 – 0.12 seconds per frame
3) *Confusion Matrix:* Some confusion observed between similar signs like 'M' and 'N', or 'Sorry' and 'Thank You', especially under poor lighting

C. Speech-to-Text Performance

1) *Word Error Rate (WER):* 5–10% under normal speech conditions
2) *Transcription Speed:* 1 second average
3) *Noise Impact:* Accuracy dropped to 85% in noisy environments

D. System Performance and User Interaction

1) *Real-Time Translation:* Gesture or audio input translated to Tamil/Hindi in <2 seconds
2) *User Interface:* Users found the interface intuitive with real-time feedback
3) *Features:* Buttons for start/stop detection, screenshots, audio recording, and live result display

E. Challenges and Limitations

1) *Gesture Variability:* Accuracy is affected by hand position, shape, and signing speed
2) *Environmental Factors:* Poor lighting and cluttered backgrounds reduce detection accuracy
3) *Real-Time Constraints:* Processing delay may increase on systems without GPU support

F. Performance Summary

Table I. Performance metrices

Metric	Result / Observation
Gesture Detection Accuracy	95% (YOLOv8 on trained dataset)
Sentence Recognition	High accuracy for sequential gestures
Word Error Rate (WER)	5–10% (clear speech input)
Speech Recognition Accuracy	90–95% in quiet environments
Noisy Environment Accuracy	Drops to ~85% (recognizes main content)
YOLOv8 Inference Time	0.08 – 0.12 seconds per frame
Real-Time Frame Rate (CPU)	12 – 18 FPS
Translation Delay	< 1.5 seconds (gesture/speech to output)
Output Languages	English, Tamil, Hindi
User Experience	Responsive interface, easy to use

Note: Accuracy is calculated using two formulas

- If expected count ≥ detected count: Accuracy (%) = (Detected Count / Expected Count) × 10
- If expected count < detected count (overcount): Accuracy (%) = [1 - ((Detected - Expected) / Expected)] × 100

G. Attached Results

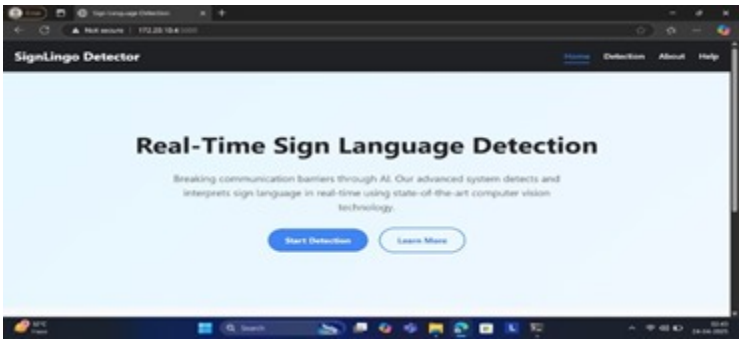


Fig.4 Home page for onboarding

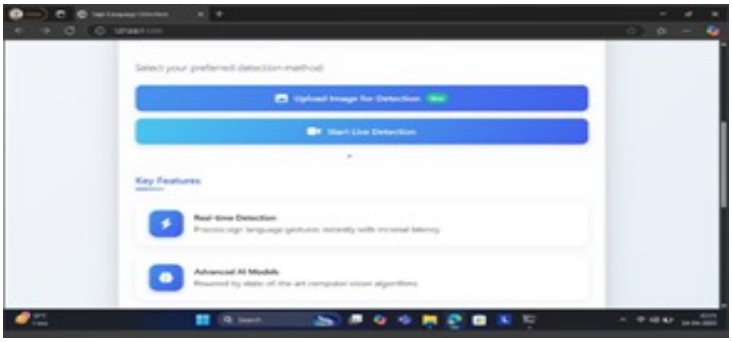


Fig.5 Preferred detection method page

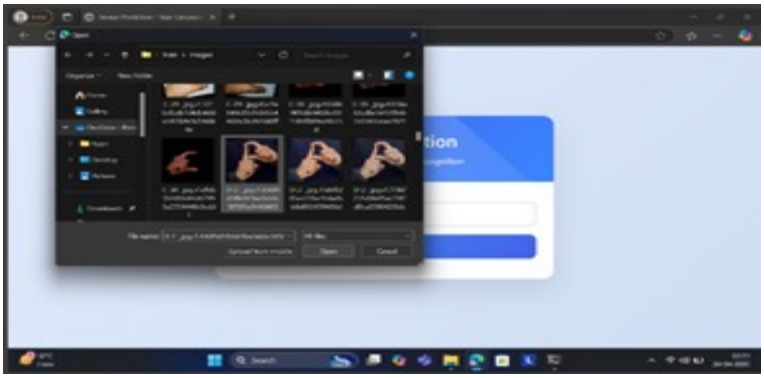


Fig.6 Upload image for detection page

Fig.7 uploaded page

Fig.8 prediction result page

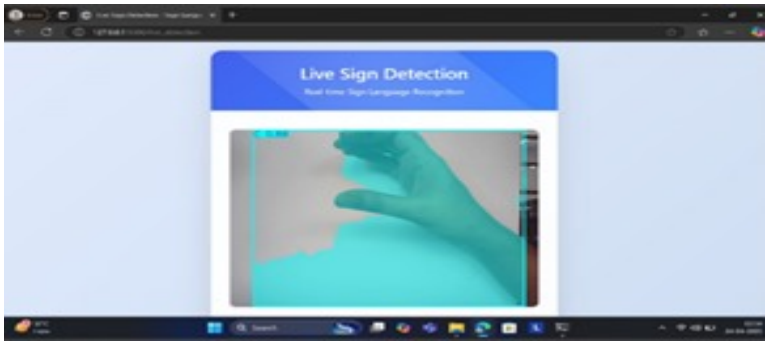


Fig.9 Live sign detection page

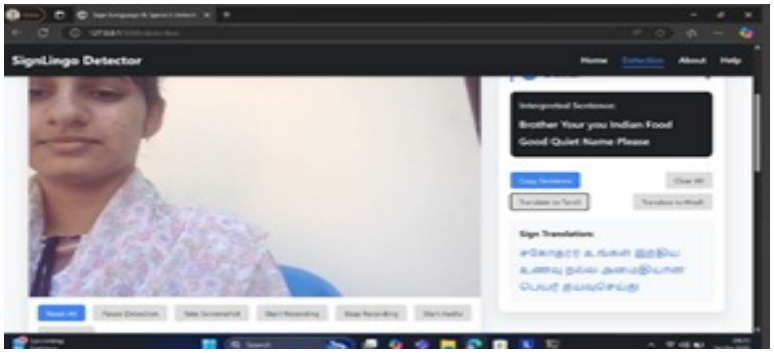


Fig 10: Sentence detection and Translate Page

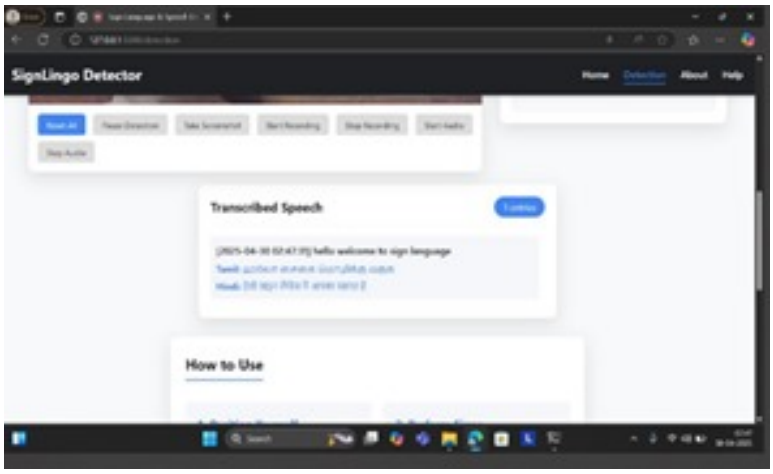


Fig.11 Audio Translate page

V. FUTURE WORK

While the current system performs well for static and sentence-level gesture recognition, several enhancements can improve its robustness, scalability, and usability:

1) *Dynamic Gesture Recognition:* The system currently supports static gestures and pre-defined sentence signs. Future versions will include dynamic gesture tracking using RNNs or LSTM models for continuous signing (e.g., Indian Sign Language grammar flow).
2) *Mobile Application Integration:*Deploying the system as a mobile app using platforms like Flutter or React Native will improve accessibility and portability, allowing real-time translation anytime, anywhere.
3) *Expanded Vocabulary:*Future datasets will include more gestures, regional sign variations, and grammar rules to better capture the complexity of real-world sign languages.
4) *Offline Mode and Edge Deployment:*Optimizing models for edge devices will enable offline detection without internet dependency, making the tool suitable for rural or low-connectivity areas.
5) *Multilingual Voice Translation:*Currently, the system supports Tamil and Hindi. Future updates will include additional regional and international languages, offering broader multilingual communication support.
6) *Feedback-Based Learning:*Adding a feedback loop where users can correct wrong predictions will allow the system to learn and adapt over time, improving accuracy and personalization.

VI. CONCLUSION

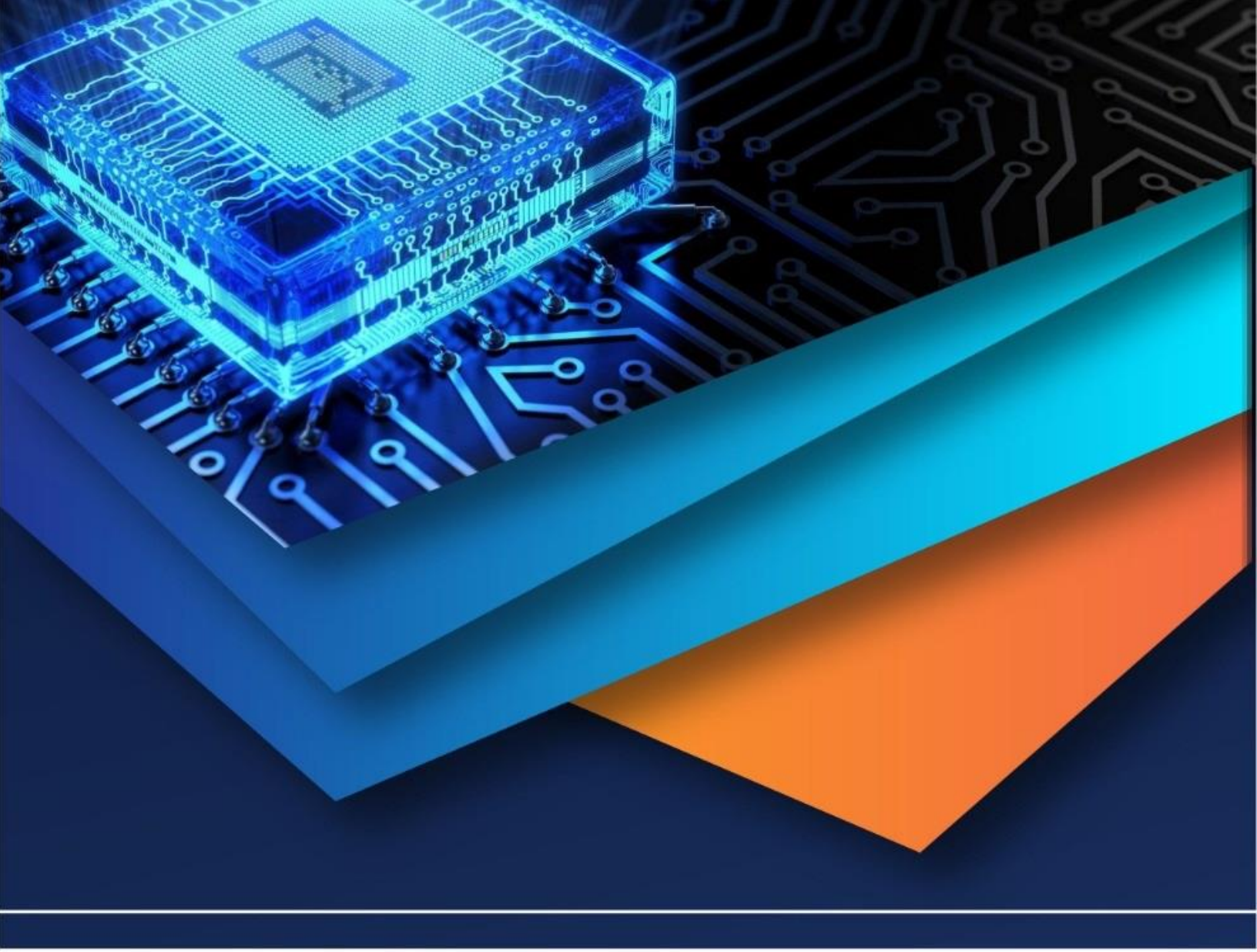
This paper proposed a real-time sign language translator combining gesture and speech input to bridge communication barriers. Using YOLOv8 for gesture detection and speech recognition for voice input, the system delivers accurate, multilingual translation through a web-based interface. It achieves real-time performance without external hardware and supports alphabets, numbers, and sentence gestures. With high usability and promising results, the system offers an effective step toward inclusive communication for the hearing-impaired.

VII. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their faculty guides and project coordinators for their continuous support, guidance, and valuable feedback throughout the development of this project. Special thanks to the creators of Roboflow, Ultralytics YOLOv8, and open-source tools such as OpenCV, Flask, and MediaPipe, which made the implementation of this real-time sign language translator system possible. The authors also thank their peers and volunteers who contributed during the dataset collection and testing phases.

REFERENCES

[1] A. I. Singh, B. Mathai, S. Silas, and J. B. Princess, “Real-Time Sign Language Translator for Deaf and Mute,” in Proc. Int. Conf. on Electronics, Robotics and Computer Science (ICERCS), 2023, doi: [10.1109/ICERCS57948.2023.10433971] (https://doi.org/10.1109/ICERCS57948.2023.10433971).
[2] S. Thakar, S. Shah, B. Shah, and A. V. Nimkar, “Sign Language to Text Conversion in Real Time using Transfer Learning,” arXiv preprint, arXiv:2211.14446v1 \[cs.CV], 2022. \[Online]. Available: [https://arxiv.org/abs/2211.14446] (https://arxiv.org/abs/2211.14446)
[3] P. K. Saw, N. Nancy, S. Gupta, A. Raj, S. Chauhan, and K. Agrawal, “Gesture Recognition in Sign Language Translation: A Deep Learning Approach,” in Proc. ICIC3S, 2024, doi: [10.1109/ICIC3S61846.2024.10603225] (https://doi.org/10.1109/ICIC3S61846.2024.10603225).
[4] E. B. Setiawan, A. Darmawan, and B. Herdiana, “Static Sign Language Translator Using Hand Gesture and Speech Recognition,” JMSI, vol. 10, no. 2, 2024, doi: [10.46754/jmsi.2024.10.002] (https://doi.org/10.46754/jmsi.2024.10.002).
[5] M. Papatsimouli et al., “Real Time Sign Language Translation Systems: A Review Study,” in Proc. MOCAST, 2022, doi: [10.1109/MOCAST54814.2022.9837666] (https://doi.org/10.1109/MOCAST54814.2022.9837666).
[6] S. Dhulipala, F. F. Adedoyin, and A. Bruno, “Sign and Human Action Detection Using Deep Learning,” J. Imaging, vol. 8, no. 7, p. 192, 2022, doi: [10.3390/jimaging8070192] (https://doi.org/10.3390/jimaging8070192).
[7] S. Mhatre, S. Joshi, and H. B. Kulkarni, “Sign Language Detection using LSTM,” in Proc. IEEE CCET, Bhopal, India, 2022, pp. 1–6, doi: [10.1109/CCET56606.2022.10080705] (https://doi.org/10.1109/CCET56606.2022.10080705).
[8] M. S. Amin and S. T. H. Rizvi, “Sign Gesture Classification and Recognition Using Machine Learning,” Cybernetics and Systems, 2023, doi: [10.1080/01969722.2022.2067634] (https://doi.org/10.1080/01969722.2022.2067634).
[9] J. Gangrade and J. Bharti, “Vision-based Hand Gesture Recognition for Indian Sign Language Using CNN,” IETE J. of Research, 2023, doi: [10.1080/03772063.2020.1838342] (https://doi.org/10.1080/03772063.2020.1838342).
[10] M. Al-Hammadi et al., “Spatial Attention-Based 3D Graph Convolutional Neural Network for Sign Language Recognition,” Sensors, vol. 22, no. 12, p. 4558, 2022, doi: [10.3390/s22124558] (https://doi.org/10.3390/s22124558).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY