Ai-Based Indian Sign Language Recognition, Real-Time Speech Generation Using Deep Learning and Computer Vision.

mhc@mcehassan.ac.in | mb@mcehassan.ac.in | snehasneha5079@gmail.com | sinchanakg7@gmail.com | bindutanu9@gmail.com

Megha H C, Maseeha Banu, Sneha, Tanushree M, Sinchana K G

Malnad College of Engineering, Hassan, Karnataka, India Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning),

Megha h c and Maseeha banu (Assistant Professor); all other coauthors are students

Abstract— Communication between people who use Indian Sign Language (ISL) and people who do not understand it can be challenging. This project presents a camera-based ISL recognition system that converts hand gestures into letters and words, supports rule-based sentence formation, and converts selected sentences into speech. OpenCV processes camera frames, while MediaPipe Hands extracts hand landmarks. The letter-recognition **module** uses 21 landmarks with X, Y, and Z coordinates, producing 63 features that are classified using a Random Forest model. The dataset contains 30,884 samples, and the reported accuracy is approximately 99.97%. The word-recognition module extracts 162 features per frame and processes 16-frame sequences using a TensorFlow/Keras architecture with Conv1D, Multi-Head Attention, and Transformer-based components. It recognizes 23 predefined words and achieves 87.67% accuracy. Recognized words are stored and passed to a rule-based sentence-formation module, where the user selects a sentence for Text-to-Speech using pyttsx3. An integrated Speech-to-Text module converts spoken input into text, supporting communication in the reverse direction. A Tkinter application provides learning and recognition modes for letters and words, sentence recognition, and Speech-to-Text. Index Terms—Indian Sign Language, computer vision, MediaPipe Hands, Random Forest, deep learning, Transformer, sentence formation, Text-to-Speech, Speech-to-Text.

I. INTRODUCTION

Communication is an important part of everyday life. People who use Indian Sign Language may experience difficulty when communicating with people who do not understand sign language. This project develops a computer-based system that recognizes ISL gestures through a camera and converts them into text and speech. It also integrates Speech-to-Text to support communication in the reverse direction. The system is organized into three principal modules: ISL letter recognition, ISL word recognition, and sentence formation. The letter module recognizes A–Z using a Random Forest classifier. The word module recognizes 23 predefined words using a sequence-based deep-learning model. The sentence module stores recognized words and presents sentence options generated by rule-based logic. The final Tkinter application includes learning modes, recognition modes, sentence recognition, and Speech-to-text.

II. SYSTEM OVERVIEW

A. Sign-to-Speech Workflow

The sign-to-speech workflow is: ISL gesture → camera → OpenCV → MediaPipe Hands → landmark and feature extraction → AI model → recognized letter or word → sentence formation → Text-to-Speech. OpenCV processes the camera frames, MediaPipe Hands detects hand landmarks, and the relevant recognition model predicts a letter or word. Recognized words are then used by the sentence formation module. The user selects a sentence, which is spoken using pyttsx3. The camera provides real-time video input for gesture recognition. The extracted hand landmarks are converted into numerical features suitable for the trained AI models. The letter recognition module uses a Random Forest classifier, while the word recognition module uses a deep learning model to classify sequences of gestures. The predicted output is displayed in the graphical user interface for the user to review. The sentence formation module combines recognized words and generates meaningful sentence options using rule-based logic. Finally, the selected sentence is converted into speech, helping users communicate their intended message through audio output.

B. Speech-to-Text Workflow

The reverse workflow is speech → Speech-to-Text → text. The existing speech_to_text.py module was integrated into the final application. This feature allows spoken input to berepresented as text and provides two-way communication support. In this workflow, the user provides spoken input through a microphone. The speech-to-text module processes the input and converts the recognized speech into text. The resulting text is displayed in the application's interface so that it can be read by the user. This functionality provides an alternative way to enter information without using sign gestures. It complements the sign-to-speech workflow by supporting communication in the opposite direction. Together, these two workflows integrate gesture recognition, text processing, and speech technologies into a single application.

III. METHODOLOGY

A. Dataset Description

The proposed system uses two datasets for Indian Sign Language (ISL) recognition: the Indian Sign Language Alphabet Dataset and the ISL Isolated 40 Words dataset. These datasets are used for training the letter and word recognition modules, respectively.

1) Letter Recognition Dataset: The Indian Sign Language Alphabet Dataset contains 30,884 samples representing letters from A to Z. MediaPipe Hands detects 21 hand landmarks, each represented by X, Y, and Z coordinates, resulting in 63 features per sample. These features are used to train the Random Forest classifier for alphabet recognition.

2) Word Recognition Dataset: The ISL Isolated 40 Words dataset provides video samples of isolated sign-language words. In the proposed implementation, the word recognition model is configured to recognize 23 selected words: hello, thank_you, go, help, yes, no, please, friend, market, mother, okay, drink, father, you, today, food, teacher, sorry, sit, stop, hospital, school, and what. The model processes sequences of 16 frames and uses a deep learning architecture based on Conv1D, Multi-Head Attention, and Transformer components to classify the gestures. Both datasets support the development of the recognition modules, which are integrated into the application for camera based letter and word recognition.

A. ISL Letter Recognition

The letter-recognition module identifies letters from A to Z. MediaPipe Hands detects 21 hand landmarks, each represented by X, Y, and Z coordinates. The resulting feature count is $21 \times 3 = 63$ features. These features are passed to a Random Forest classifier, and the trained model is saved as letter_model.pkl. The dataset contains 30,884 samples, and the reported model accuracy is approximately 99.97%. The recognition steps are camera capture, OpenCV frame

processing, hand landmark detection, extraction of the 63 coordinate features, and Random Forest classification to predict the corresponding letter. The system uses the extracted landmark coordinates to represent the hand's position and shape numerically. The trained classifier compares these features with patterns learned during training to identify the most likely letter. The predicted letter can then be displayed on the application's graphical user interface. This process enables users to perform letter recognition using a camera without manually entering the alphabet. The module also supports the learning and demonstration of ISL alphabet gestures.

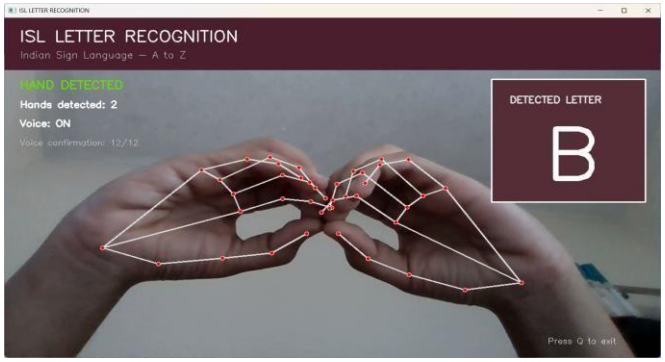


Fig. 1. Real-time ISL letter recognition with detected hand landmarks.

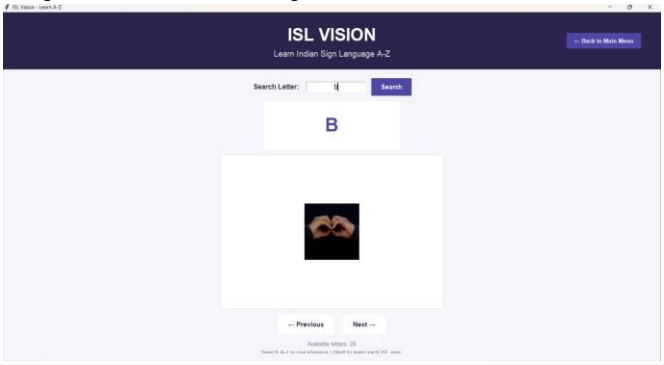


Fig. 2. ISL letter-learning interface for searching and browsing A–Z signs.

B. ISL Word Recognition

The word-recognition module identifies 23 predefined words. It uses hand landmark coordinates, angles, and hand-size features, with 162 features extracted per frame. To represent gestures over time, the model processes sequences of 16 frames. The model is implemented using TensorFlow/Keras and combines Conv1D, Multi-Head Attention, and Transformer-based components. The trained model is stored as word_model_23.keras and achieves a reported accuracy of 87.67%. The supported words are: hello, thank_you, go, help, yes, no, please, friend, market, mother, okay, drink, father, you, today, food, teacher, sorry, sit, stop, hospital, school, and what.

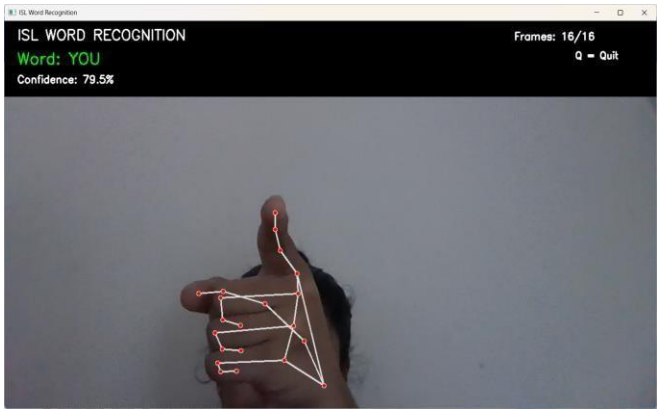


Fig.3.Real-timeISLword recognition showing a predicted word and confidence

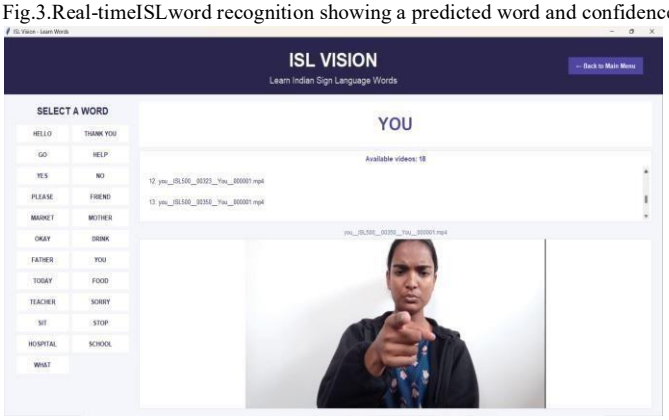


Fig. 4. ISL word-learning interface with available sign videos.

C. Sentence Formation

The sentence-formation module receives recognized words and stores them. It uses rule-based logic to generate meaningful sentence options. The user selects the required sentence from the available options. This selected sentence is passed to the Text-to-Speech component. This module helps convert individual recognized words into more understandable messages. It also allows users to choose a suitable sentence according to their intended meaning.

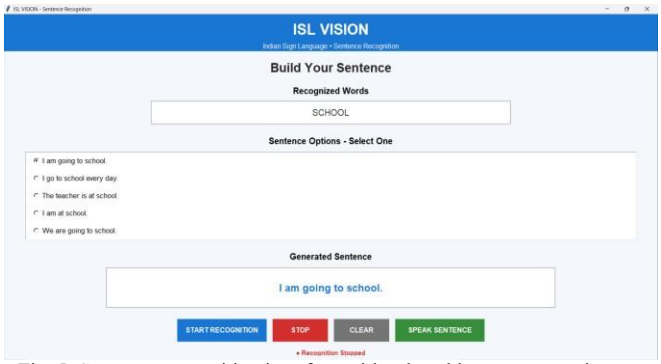


Fig. 5. Sentence-recognition interface with selectable sentence options.

D. Text-to-Speech and Speech-to-Text

The project uses pyttsx3 to convert the selected sentence into spoken output. In the reverse direction, the integrated speech_to_text.py module converts the user's speech into text. Together, these functions provide support for sign-to-speech and speech-to-text communication. The text-to-speech feature allows

the generated message to be communicated through audio output. The speech-to-text feature provides an alternative way for users to enter information using spoken language. Both functionalities are integrated into the application's graphical user interface for convenient access. These features complement the sign recognition modules and help support communication in both directions.

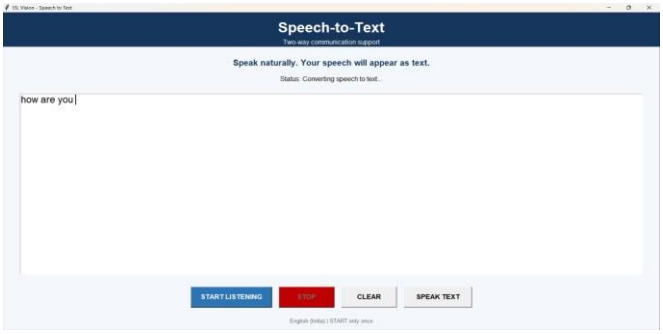


Fig. 6. Speech-to-Text interface showing transcribed spoken input.

E. Learning Modes

The application includes Learn ISL Letters and Learn ISL Words. These are available alongside letter recognition, word recognition, sentence recognition, and Speech-to-Text in the Tkinter-based main application. The Learn ISL Letters mode helps users become familiar with the hand gestures associated with the English alphabet from A to Z. The Learn ISL Words mode provides a way to explore the selected ISL word gestures supported by the application. These learning features complement the recognition modules by helping users understand the gestures before trying real-time recognition. The Tkinter-based graphical user interface brings all these functions together in one application, making the system easier to navigate and use.

IV. SYSTEM ARCHITECTURE

The architecture consists of input, processing, recognition, sentence formation, and output stages. Camera input is processed by OpenCV. MediaPipe Hands extracts hand landmarks, and feature extraction prepares the input for either the letter or word model. The prediction is passed to the sentence formation module, where applicable, and the selected sentence is converted into speech. The separate reverse path accepts speech and produces text through Speech-to-Text.

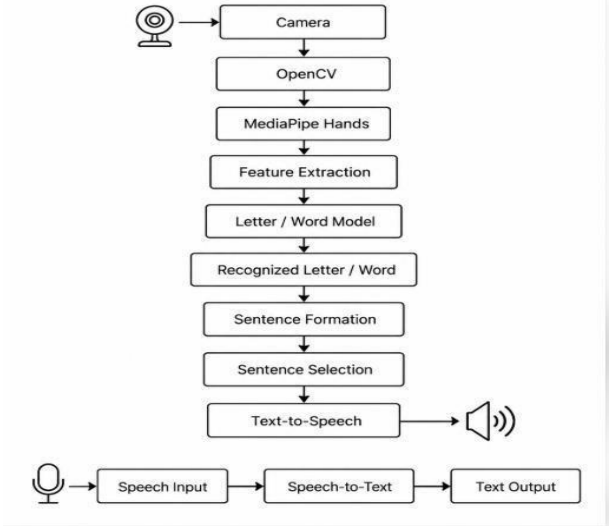


Fig. 1. Overall workflow of the proposed ISL communication system.

V. IMPLEMENTATION

The application is developed in Python using Visual Studio Code. OpenCV is used for camera and video processing, and MediaPipe Hands is used for hand landmark detection. NumPy and Pandas support numerical and dataset processing. Scikit learn provides the Random Forest implementation, while TensorFlow/Keras supports the word-recognition model. Tkinter provides the graphical user interface, pyttsx3 provides Text-to-Speech, and Joblib/Pickle supports model storage.

Technology	Purpose
Python	Main programming language
OpenCV	Camera and video processing
MediaPipe Hands	Hand landmark detection
NumPy / Pandas	Numerical and dataset processing
Scikit-learn / Random Forest	Letter classification
TensorFlow/Keras	Word-recognition model
Conv1D, Multi-Head, Attention, Transformer	Sequence-base word recognition
Tkinter	Graphical user interface
pyttsx3	Text-to-Speech
Joblib / Pickle	Model storage
VS Code	Development environment

The final Tkinter interface provides six main options: Learn ISL Letters, Recognize ISL Letters, Learn ISL Words, Recognize ISL Words, Sentence Recognition, and Speech-to-Text.

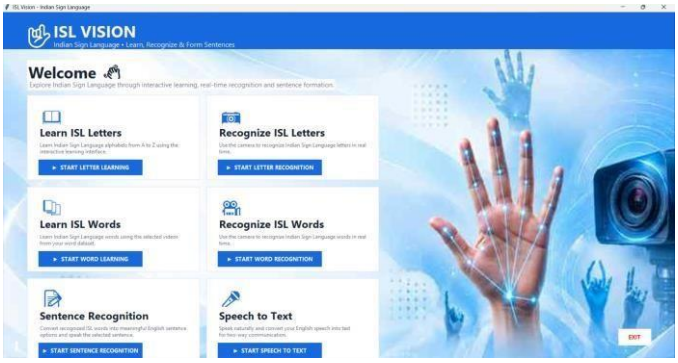


Fig. 7. Main ISL Vision dashboard showing learning and recognition modules.

VI. RESULTS

A. Letter Recognition

The letter model uses 30,884 samples and 63 coordinate features from 21 hand landmarks. The reported accuracy is approximately 99.97%. The model file is letter_model.pkl.

B. Word Recognition

The word model processes 16-frame sequences with 162 features per frame. It recognizes 23 predefined words and achieves a reported accuracy of 87.67%. The model file is word_model_23. keras.

Module	Input Features	Model	Reported Accuracy
Letter recognition	63 features	Random Forest	≈99.97%

Word recognition	162 features/frame; 16 frames	Conv1D + multi-head Attention	87.67%

VII. DISCUSSION

The letter and word modules use different models because their input representations differ. Letter recognition uses a single set of hand landmark coordinates represented by 63 features and classified by Random Forest. Word recognition uses sequences of 16 frames and 162 features per frame, processed by a deep-learning model. Sentence formation is handled separately using rule-based logic, while speech output and reverse text input are provided by pyttsx3 and Speech-to Text, respectively. The letter recognition model achieved a reported accuracy of 99.97%, while the word recognition model achieved 87.67%. These results indicate that both models can recognize their respective classes, although word recognition may require further improvement. The system integrates these modules into a single Tkinter-based application, making its features accessible through one interface. However, recognition performance may vary depending on hand movements, lighting conditions, camera quality, and the gestures performed by the user.

Table I. Comparative Analysis of Existing ISL Models and the ProposedModel.

Research Model	Recognition Task	Technique	Reported Accuracy
Three-layer CNN [1]	ISL alphabet	Convolutional Neural Network (CNN)	97.6%
ResNet152V2 [1]	ISL alphabet	Transfer Learning	90.8%
DeepSign [2]	11 isolated ISL signs	LSTM-GRU	Approximately 97%
MediaPipe + LSTM-GRU [3]	ISL words	Landmark Tracking and Recurrent Networks	89.50%
Proposed Letter Model	26 letters (A-Z)	MediaPipe Hands + Random Forest	99.97%
Proposed Word Model	23 selected ISL words	Conv1D + Multi-Head Attention + Transformer-based Model	87.67%

Table II compares the proposed letter and word recognition models with existing approaches to Indian Sign Language recognition. For alphabet recognition, the proposed MediaPipe Hands and Random Forest model reports an accuracy of 99.97%, compared with 97.6% for a three-layer CNN and 90.8% for ResNet152V2 in a published study. For word recognition, the proposed deep learning model achieves 87.67%, while a MediaPipe-based LSTM-GRU approach reports 89.50%. The results indicate that the proposed letter model achieves a high reported accuracy, while the word model has scope for further improvement. The proposed system also integrates sentence formation, text-to-speech, speech-to-text, and learning modes in one application, extending its functionality beyond gesture classification alone. Important limitation: These accuracy values were obtained using different datasets, class counts, and experimental setups, so they do not establish that one model is definitively better than another. The comparison is based on reported results, not a controlled experiment using the same dataset.

VIII. ADVANTAGES

The system supports camera-based recognition, separate letter and word models, sentence formation from recognized words, Text-to-Speech, Speech-to-Text, and learning modes. The Tkinter interface brings these functions together in one application. The reported results are approximately 99.97% accuracy for letter recognition and 87.67% for word recognition.

IX. CONCLUSION

This project combines computer vision, machine learning, deep learning, rule-based sentence formation, and speech technologies to support two-way communication using Indian Sign Language (ISL). The letter-recognition module uses MediaPipe hand landmarks and a Random Forest classifier, while the word-recognition module processes 16-frame sequences using a Conv1D, Multi-Head Attention, and Transformer-based architecture. The reported accuracies are approximately 99.97% for letter recognition and 87.67% for the 23-word recognition model.

Recognized letters and words are displayed through the application's interface, and recognized words can be used to generate meaningful sentence options through rule-based logic. The selected sentence is converted into spoken output using the pyttsx3 library. In the reverse direction, the integrated Speech to-Text module converts spoken input into text, providing an additional communication method. The Tkinter-based application brings together letter recognition, word recognition, sentence formation, learning modes, and speech-related features in a unified interface.

The proposed system demonstrates how different artificial intelligence techniques can be combined to develop an assistive communication application for selected ISL gestures. The separate learning modes also help users become familiar with alphabet and word gestures. However, the current implementation is limited to the selected vocabulary and depends on the quality of camera input and the ability of the trained models to recognize different hand gestures accurately. Future work can focus on expanding the vocabulary, improving recognition under varying lighting conditions and backgrounds, supporting continuous sign language sentence recognition, and evaluating the system with more users and diverse signing styles. Further improvements to model accuracy, response time, and recognition robustness could help make the application more suitable for practical communication environments. Overall, the project provides a foundation for developing more comprehensive ISL recognition and speech-support systems.

References

[1] B. Alsharif, E. Alalwany, A. Ibrahim, I. Mahgoub, and M. Ilyas, “Real-time American Sign Language interpretation using deep learning and keypoint tracking,” Sensors, vol. 25, no. 7, Art. no. 2138, 2025.

[2] “Real-time American Sign Language recognition using 3D convolutional neural networks and LSTM: Architecture, training, and deployment,” arXiv preprint arXiv:2512.22177, 2025.

[3] B. Fonya and C. Worrell, “Real-time sign language gestures to speech transcription using deep learning,” arXiv preprint arXiv:2508.12713, 2025.

[4] P. Sheth, R. Dedhia, A. Chheda, and V. Sawant, “American Sign Language recognition and generation: A CNN-based approach,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 9, no. 4, pp. 241–250, 2023.

[5] A. Joshi, S. Sierra, and S. Arora, “CISLR: Corpus for Indian Sign Language recognition,” in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), 2022.

[6] “Isolated Indian Sign Language recognition using MediaPipe and Multihead Attention Transformer,” Elsevier book chapter, 2023.

[7] “YOLOv10-ST: Advanced gesture recognition for Indian Sign Language using Swin Transformer,” Scientific Reports, vol. 15, 2025.

[8] T. Starner and A. Pentland, “Real-time American Sign Language recognition from video using hidden Markov models,” in Proc. Int. Symp. Computer Vision (ISCV), 1995, pp. 265–270.

[9] O. Koller, S. Zargaran, and H. Ney, “Re-Sign: Re-aligned end-to end sequence modelling with deep recurrent CNN-HMMs,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4297–4305.

[10] O. Koller, S. Zargaran, H. Ney, and R. Bowden, “Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs,” International Journal of Computer Vision, vol. 126, no. 12, pp. 1311–1325, 2018.

[11] S. Lang, M. Block, and R. Rojas, “Sign language recognition using Kinect,” in Proc. Int. Conf. Artificial Intelligence and Soft Computing, 2012, pp. 394–402.

[12] Q. Munib, M. Habeeb, B. Takruri, and H. A. Al-Malik, “American Sign Language (ASL) recognition based on Hough transform and neural networks,” Expert Systems with Applications, vol. 32, no. 1, pp. 24–37, 2007.

[13] V. Ranga, N. Yadav, and P. Garg, “American Sign Language fingerspelling using hybrid discrete wavelet transform-Gabor filter and convolutional neural network,” Journal of Engineering Science and Technology, vol. 13, no. 9, pp. 2655–2669, 2018.

[14] A. Kasapbaşi, A. E. A. Elbushra, O. Al-Hardanee, and A. Yilmaz, “DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals,” Computer Methods and Programs in Biomedicine Update, vol. 2, Art. no. 100048, 2022.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998 6008.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[20] Z. Liu et al., “Swin Transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2021, pp. 10012–10022.

[21] C. Lugaresi et al., “MediaPipe: A framework for building perception pipelines,” arXiv preprint arXiv:1906.08172, 2019.

[22] F. Zhang et al., “MediaPipe Hands: On-device real-time hand tracking,” arXiv preprint arXiv:2006.10214, 2020.

[23] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, “Neural sign language translation,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7784–7793.

[24] D. Bragg et al., “Sign language recognition, generation, and translation: An interdisciplinary perspective,” in Proc. ACM Conf. Computers and Accessibility (ASSETS), 2019.