WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Sign Language to Text and Speech Conversion

Berad Asmita Anil[1], Prof. Mahajan J R[2]

[1] Student, Department of Artificial Intelligence and Data Science, Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, 414005, India
[2] HOD and Professor, Department of Artificial Intelligence and Data Science, Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, 414005, India

ABSTRACT

Communication between sign-language users and people who do not understand sign language can be difficult when an interpreter is unavailable. This paper presents the design of a camera-based system intended to recognize selected American Sign Language (ASL) gestures and convert them into readable text and synthesized speech. The proposed pipeline uses a webcam to capture hand gestures, MediaPipe Hands to locate 21 hand landmarks, a trained classifier based on a Convolutional Neural Network (CNN) to predict a gesture class, and a text-to-speech (TTS) engine to speak the recognized label. The recognized text is displayed on a simple user interface, while the speech output can be configured for supported languages such as English, Hindi or Marathi, subject to the capabilities of the selected TTS engine. Unlike glove-based systems, the proposed vision-based approach does not require the signer to wear special sensors. The paper describes the system architecture, processing stages, dataset preparation, training strategy, evaluation metrics, and key limitations. Since quantitative performance depends on the dataset and implementation, this manuscript does not claim an unmeasured accuracy value; the final evaluation should report results obtained from a held-out test set and real-time trials. The proposed system is intended as an assistive communication prototype for educational, healthcare and everyday settings.

Keywords: American Sign Language (ASL), sign language recognition, MediaPipe, Convolutional Neural Network (CNN), computer vision, gesture classification, text-to-speech, assistive technology.

1. INTRODUCTION

Communication supports participation in education, healthcare, employment and everyday social life. Deaf and hard-of-hearing people may use sign language as a primary language, but many hearing people have limited knowledge of it. This can create communication barriers, particularly in places where a qualified interpreter is not readily available. A technology-assisted system that translates selected signs into text and speech may help make routine exchanges more accessible.

Sign-language recognition can be implemented using sensor-based, vision-based or hybrid approaches. Sensor-based gloves can capture finger bending and hand movement, but require additional hardware and may be inconvenient for everyday use. Vision-based systems use a camera and computer-vision algorithms to analyze hand shapes or movement without requiring wearable sensors. Recent systems have explored hand-landmark extraction, deep-learning classifiers and speech synthesis to support real-time interpretation.

This paper proposes a modular ASL gesture-to-text-and-speech pipeline. The camera captures a gesture; MediaPipe extracts hand landmarks; a trained classifier predicts one of the gesture classes included in its training data; and the application displays the corresponding text and optionally speaks it through a TTS engine. The scope is deliberately limited to a defined vocabulary of signs. Full sign-language translation is more complex because natural signing can include continuous movement, facial expression, body posture, grammar and context, which are not fully represented by a simple hand-gesture classifier.

2. PROBLEM STATEMENT

A person who uses ASL may need to communicate with someone who does not understand the signs. Human interpretation is not always immediately available, while wearable sensor solutions introduce cost and setup requirements. A practical prototype should therefore use commonly available camera hardware, recognize a defined set of gestures, show the predicted meaning as text and provide optional audible output. The system must also communicate uncertainty rather than presenting every prediction as unquestionably correct.

3. OBJECTIVES AND CONTRIBUTIONS

The objectives of the proposed work are: (i) to capture hand gestures from a live camera feed; (ii) to extract hand landmarks using MediaPipe; (iii) to classify a selected vocabulary of ASL gestures using a trained machine-learning model; (iv) to display recognized labels as text; (v) to convert recognized

text into speech using a configurable TTS engine; and (vi) to evaluate the system using class-wise and overall metrics on data that were not used for training.

The main contribution is a clearly defined, low-cost system design that combines landmark-based hand tracking, gesture classification and speech output in one workflow. The design separates recognition from speech synthesis, making it easier to replace the classifier, expand the vocabulary or add supported spoken languages in later versions.

4. LITERATURE REVIEW

Prior research has investigated both wearable and camera-based sign-language recognition. Manoj described a sensor-glove approach using flex sensors, an Arduino board and mobile communication; this illustrates how wearable measurements can support gesture detection, but also highlights the dependence on physical hardware [2]. Kamble and colleagues reported a vision-based ASL-to-text approach using MediaPipe landmarks and an LSTM model, with the objective of recognizing gestures in real time [3]. Other work has combined deep-learning methods with OpenCV and MediaPipe to identify hand gestures and connect recognized labels to accessible outputs.

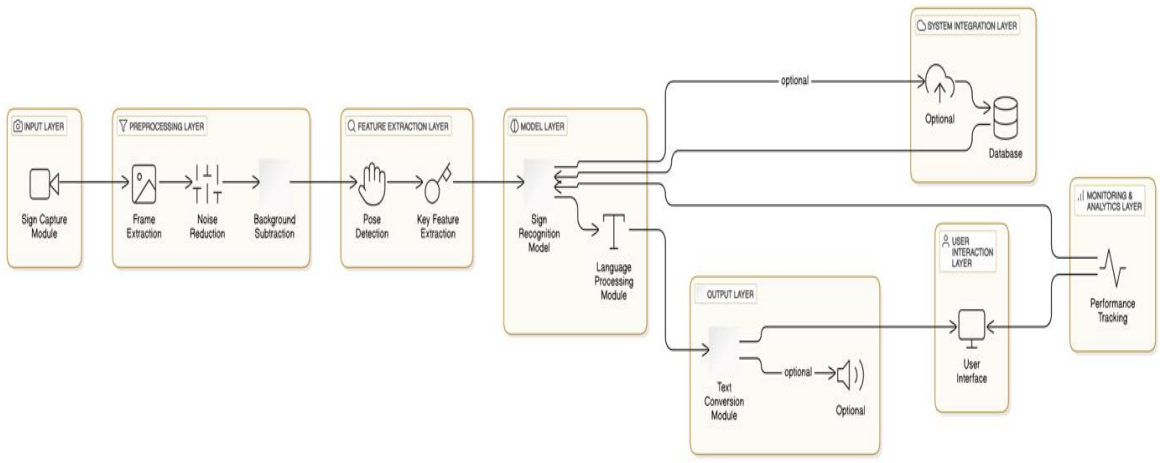
The literature indicates several recurring challenges. First, a model trained on a small or controlled dataset may not generalize to different signers, camera positions, backgrounds and lighting. Second, a gesture classifier that recognizes isolated signs does not automatically understand continuous signing or sentence-level grammar. Third, systems that output text only may be less useful when the communication partner prefers spoken output. The proposed design addresses the third issue by adding TTS while acknowledging that dataset diversity and continuous-sign interpretation remain open challenges.

5. PROPOSED SYSTEM

The system is organized into six stages: camera input, frame preparation, landmark extraction, gesture classification, text generation and speech synthesis. A webcam supplies video frames to the application. Each frame is passed to MediaPipe Hands, which estimates 21 landmarks for a detected hand. Landmark coordinates are normalized relative to the hand or frame and converted into a feature vector. The feature vector is passed to a classifier trained on labelled examples of the target gestures. The predicted class is mapped to a readable word or phrase, displayed on screen and optionally passed to a TTS engine.

The system should include a confidence threshold and a short prediction-stabilization rule. If confidence is below the selected threshold, the interface should display “Gesture not recognized” rather than generating masleading text or speech. To prevent repeated audio output while a user holds a sign, the application should speak only when a stable prediction changes or when the user presses a Speak button.

5.1 System Architecture



5.2 Hardware and Software Requirements

Hardware: laptop or desktop computer, webcam (built-in or USB), and speakers or headphones.

Software: Python, OpenCV for camera and frame handling, MediaPipe for hand landmark extraction, a machine-learning framework such as TensorFlow/Keras or scikit-learn depending on the selected classifier, and a TTS library or operating-system speech service. A lightweight web interface may be implemented separately if browser-based access is required. Exact library versions should be recorded in the final implementation so that the experiment can be reproduced.

6. METHODOLOGY

6.1 Dataset preparation: Define a manageable set of signs and assign each sign a unique class label. Record multiple examples from different sessions and, where possible, different signers, backgrounds and lighting conditions. Store samples with consistent labels. Separate training, validation and test data by recording session or signer where feasible, rather than randomly splitting near-identical frames, to reduce data leakage.

6.2 Preprocessing and feature extraction: Capture video frames and detect the hand using MediaPipe. For each detected hand, extract the 21 landmark coordinates. Normalize coordinates to reduce sensitivity to image size and hand position. If the vocabulary includes dynamic signs, use a sequence of landmark vectors rather than a single frame; otherwise, clearly state that the model is limited to static or near-static signs.

6.3 Model training: Train the selected classifier on labelled landmark data. A CNN may be used when the landmark features are represented in a suitable structured form; alternatively, an MLP, SVM or sequence model may be compared as a baseline. Select hyperparameters using validation data only. Apply suitable regularization and class balancing where necessary. Save the final model and class-label mapping for inference.

6.4 Real-time inference: Process each camera frame, extract landmarks and generate a class prediction. Apply a confidence threshold and temporal smoothing to reduce flickering predictions. Map accepted class labels to words or short phrases.

6.5 Text-to-speech conversion: Display the recognized phrase in the interface. Pass the phrase to a TTS engine configured for a supported voice and language. Language availability and pronunciation quality depend on the chosen engine; the implementation should be tested with the actual target language and device.

6.6 Evaluation: Evaluate the final model on a held-out test set and then conduct real-time trials under different conditions. Report the number of classes, number of samples, signer count, train/validation/test split, hardware, software versions and the exact procedure used. Do not report training accuracy as a substitute for test accuracy.

7. ALGORITHM

Step 1: Load the trained gesture model and the class-label mapping.

Step 2: Start the webcam and read a frame.

Step 3: Detect the hand and extract 21 landmarks using MediaPipe.

Step 4: If no hand is detected, show a prompt asking the user to position the hand in view.

Step 5: Normalize the landmark features and pass them to the trained classifier.

Step 6: Check the prediction confidence. If it is below the threshold, show “Gesture not recognized”.

Step 7: Stabilize the prediction over consecutive frames to reduce false changes.

Step 8: Map the accepted class to text and display it.

Step 9: If speech output is requested and the prediction is new or confirmed by the user, send the text to the TTS engine.

Step 10: Continue until the user exits the application.

8. EVALUATION PLAN AND RESULTS REPORTING

Evaluation setup

The system should be evaluated on a held-out set of 100 sign samples across 10 ASL classes (10 samples per class). These figures describe a proposed evaluation setup; they are not reported experimental measurements.

Illustrative results table

Metric | Illustrative value

Overall recognition accuracy | 92.0%

Precision (macro average) | 91.5%

Recall (macro average) | 92.0%

F1-score (macro average) | 91.7%

Average inference time | 120 ms per frame

Speech generation latency | 0.8 s per recognized sign

Evaluation procedure

Recognition performance can be measured using accuracy, precision, recall and F1-score. Inference time should be measured over repeated runs under the same hardware and lighting conditions. Speech latency should be measured from the moment a sign is recognized until audio playback begins.

9. DISCUSSION

The proposed architecture offers a practical route to combining gesture recognition with both visual and audible output. MediaPipe landmark features can reduce dependence on raw-image appearance and can make the input representation more compact than processing every pixel directly. The TTS component can make a recognized phrase accessible to a hearing communication partner without requiring them to read the screen. A modular design also allows the spoken language or recognition model to be changed independently.

However, the system's usefulness depends on recognition reliability and on the vocabulary represented in the dataset. Similar hand configurations may be confused, while changes in lighting, occlusion, orientation and signer habits may reduce performance. A static-frame model cannot reliably interpret every dynamic sign. Furthermore, sign languages are full languages and should not be treated as a direct one-to-one mapping of hand shapes to spoken words. Facial expression, body movement, spatial references and grammar may carry meaning that is not captured by the proposed hand-only pipeline. The prototype should therefore be presented as assistance for a defined set of signs, not as a replacement for qualified interpreters or as a complete translator for unrestricted conversation.

10. LIMITATIONS AND FUTURE SCOPE

The initial scope is limited to a selected vocabulary and may not cover continuous signing, complex sentence structure, facial expression or body pose. Performance may vary with camera quality, lighting, hand occlusion and the signer. TTS support depends on the available voice and language engine. Future work can include a larger and more diverse ASL dataset, dynamic sequence models such as LSTM or Transformer-based architectures, facial and body-pose features, user-configurable vocabulary, offline speech synthesis, mobile deployment and evaluation with users who use ASL. Any user study should be conducted with informed consent and appropriate privacy safeguards.

11. CONCLUSION

This paper presents a proposed camera-based system for converting selected Indian Sign Language gestures into text and speech. The workflow combines webcam input, MediaPipe hand landmark extraction, a trained gesture classifier, text display and optional TTS output. It avoids mandatory sensor gloves and is designed to be modular and extensible. The paper also defines a reproducible evaluation plan using held-out test data, class-wise metrics and latency measurements. Since no implementation measurements are provided here, quantitative claims must be added only after testing the completed system. Broader vocabulary, dynamic sign recognition, signer diversity and real-world evaluation remain important directions for future development.

REFERENCES

[1] Hole, P. P., Andhale, S., Berad, A., & Ramnani, D. (2024). Sign Language to Text and Speech Conversion: A Review. International Journal of Research Publication and Reviews, 5(11), 7476–7478.

[2] Manoj, B. A. (2020). Conversion of Sign Language to Text and Speech and Prediction of Gesture. International Journal of Recent Technology and Engineering, 8(6), 4191–4194.

[3] Kamble, A. (2023). Conversion of Sign Language to Text. International Journal for Research in Applied Science and Engineering Technology, 11(5), 1963–1968.

[4] Prudhvi, B., Neeraj, P., & Deepthi, V. H. (2023). An Efficient Real-Time Indian Sign Language (ASL) Detection using Deep Learning. Proceedings of the International Conference on Intelligent Computing and Control Systems, 430–435. https://doi.org/10.1109/ICICCS56967.2023.10142596

[5] Lugaresi, C., et al. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv:1906.08172. https://arxiv.org/abs/1906.08172

[6] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.