JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-POWERED SIGN LANGUAGE INTERPRETATION USING COMPUTER VISION AND ANIMATED AVATAR

A Bidirectional, Webcam-Based Approach Using MediaPipe Landmarks and a 3D WebGL Avatar

[1]Gaikwad Pooja, [2]Supekar Pallavi, [3]Handal Pooja, [4]Thawal Pratiksha, [5]Prof. Reshma Khartude

[1]B.E. Student, [2]B.E. Student, [3]B.E. Student, [4]B.E. Student, [5]Assistant Professor (Internal Guide)
[1-5]Department of Computer Engineering,
[1-5]HSBPVT's Group of Institutions, Faculty of Engineering (GOI FOE), Kashti, Maharashtra, India

Abstract : This paper reviews how computer vision and graphics are being combined to remove the communication barrier between deaf and non-signing people, and examines SignEngine, a browser-based prototype that translates in both directions. We first survey glove-based, deep-learning and landmark-based recognisers as well as avatar-based sign generation, and identify the gap that most published systems cover only one direction. We then describe how SignEngine closes this gap: Google MediaPipe extracts 21 three-dimensional hand landmarks per frame, a rotation- and scale-tolerant geometric classifier recognises a twelve-word American Sign Language (ASL) vocabulary, and the Web Speech API voices the result, while typed text is converted into signs performed by a rigged humanoid avatar rendered with Three.js. A decoupled React, Node.js/Express, FastAPI and MongoDB architecture keeps the system deployable on an ordinary laptop with a standard webcam. The prototype reports about 94% recognition accuracy, roughly 115 ms end-to-end latency and 60 FPS avatar rendering on its target vocabulary. We discuss these figures with their limitations and outline directions towards continuous, multi-dialect sign translation.

IndexTerms - Sign language recognition, MediaPipe, hand landmarks, 3D avatar, assistive technology, computer vision, human–computer interaction.

I. INTRODUCTION

Sign languages are complete visual–gestural languages and the first language of many deaf people. The World Health Organization estimates that over 5% of the global population, about 430 million people, needs rehabilitation for disabling hearing loss [1]. In hospitals, government offices and service counters, a qualified interpreter is often not available at the moment one is needed, and the resulting misunderstandings can delay care and administrative processes. Human interpreters are scarce and costly, and sensor gloves or depth cameras add expense, calibration effort and discomfort that discourage everyday use.

Advances in lightweight pose estimation have changed this picture. Frameworks such as MediaPipe [7], [8] return skeletal hand landmarks in real time on commodity hardware, which means that a gesture can be described by a few dozen numbers instead of millions of pixels. At the same time, browser graphics (WebGL) allow a rigged three-dimensional character to perform signs for a viewer who cannot hear. Despite this, most reported systems address a single direction, either recognising signs or generating them, so a conversation between a signer and a non-signer still needs two separate tools [11].

This paper makes three contributions. First, it gives a compact review of sensor-based, deep-learning and landmark-based recognition together with avatar-based generation. Second, it explains the design of SignEngine, an end-to-end bidirectional system that couples a landmark-driven recogniser with a text-to-sign avatar. Third, it consolidates the evaluation figures reported for the prototype and discusses their scope and limitations honestly, so that readers can judge how far the approach generalises.

II. Related Work

A. Sensor-based and early vision-based systems

Early recognisers relied on instrumented gloves or depth sensors. Starner et al. showed that ASL words could be recognised with hidden Markov models using only video from a desk-mounted or head-worn camera [2], establishing that wearable hardware is not strictly necessary. Glove-based solutions nevertheless remain expensive and need calibration, while consumer depth devices have largely been discontinued.

B. Deep learning for continuous recognition and translation

Deep networks moved the field from isolated words towards sentences. Koller et al. combined recurrent CNN–HMM models for continuous recognition [3], and Camgoz et al. formulated sign language translation as a sequence-to-sequence problem with attention [4]. Surveys by Rastgoo et al. [5] and Adaloglou et al. [6] compare 3D-CNN, recurrent and Transformer models and report that keypoint-based inputs are considerably more efficient than raw frames. The trade-off is that these models need large parallel corpora and substantial GPU capacity, which limits browser-side deployment.

C. Landmark-based lightweight recognition

MediaPipe Hands predicts 21 three-dimensional keypoints per hand from a single RGB image at real-time speed on mobile and desktop processors [7], inside a general pipeline framework [8]. Halder and Tayade fed such landmark vectors to a random-forest classifier for vernacular sign recognition [9]. Because a hand skeleton is naturally a graph, graph-convolutional ideas [10] are a promising route for modelling temporal sign sequences. Landmark methods discard background pixels, which reduces over-fitting to lighting and clothing, but raw coordinates do not carry linguistic meaning, so a classification stage is still required.

D. Avatar-based sign generation

Generating sign language, rather than only recognising it, is needed so that hearing users can answer a deaf signer. Bragg et al. argue that recognition, generation and translation should be studied together and with the involvement of the deaf community [11]. Rigged three-dimensional avatars are attractive because they can be driven by joint rotations, replayed at any speed and viewed from any angle. Table I summarises the approaches discussed and the gap that motivates this work.

Table I. comparison of representative approaches

Approach	Representative work	Strength	Limitation
Wearable / camera HMM	Starner et al. [2]	Showed feasibility of vision-based ASL recognition	Controlled setup; small vocabulary
CNN–HMM, seq2seq NMT	Koller et al. [3]; Camgoz et al. [4]	Continuous sentences and translation	Heavy compute; needs large corpora
Landmarks + ML classifier	Zhang et al. [7]; Halder and Tayade [9]	Real time on a CPU; compact input	Recognition only; no return channel
Avatar generation	Bragg et al. [11] (discussion)	Visual output suited to deaf users	Often isolated from live recognition
Integrated bidirectional (this study)	SignEngine	Both directions in one browser app; no extra hardware	Small vocabulary; isolated signs

III. System Architecture

SignEngine follows a decoupled three-tier microservice design (Fig. 1). The presentation tier is a React 18 client, built with Vite, that shows the camera feed, a heads-up display of the detected sign and its confidence, and the Three.js avatar. It also calls the browser Web Speech API to voice recognised sentences. The application tier is a Node.js/Express gateway that handles JWT-based sessions, sign-dictionary management, session-history logging and proxying of frames to the AI service. Passwords are stored as bcrypt hashes with ten salt rounds and tokens travel over HTTPS.

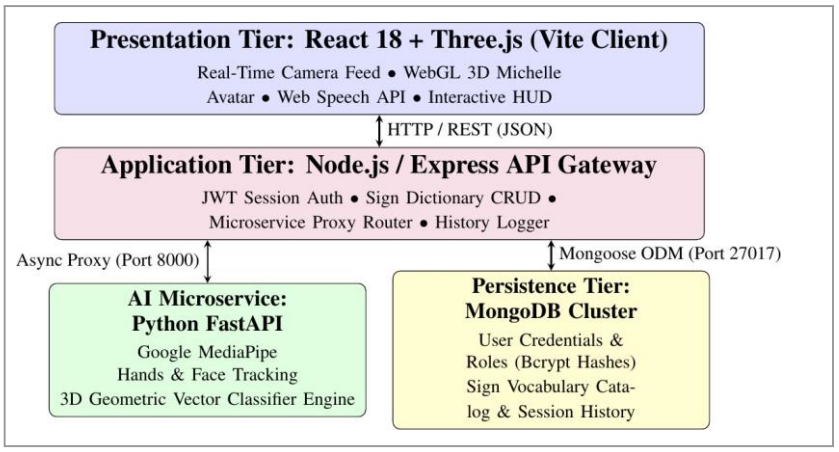

Fig. 1. three-tier decoupled architecture of SignEngine

The AI microservice is a Python FastAPI process that runs MediaPipe Hands and Face tracking and the geometric classifier, with OpenCV, NumPy and scikit-learn as supporting libraries. Running it as an asynchronous ASGI service lets several frame batches be processed concurrently. The persistence tier is a MongoDB cluster accessed through Mongoose, with three collections: users (role-based), signs (name, category, modalities, animation key) and translation history (detected signs, sentence, average confidence). Each camera frame is sent as a base64 JPEG snapshot and discarded as soon as its landmarks are computed, so no video is stored, which supports user privacy.

IV. METHODOLOGY

A. *Landmark extraction and normalisation*

For every frame, MediaPipe returns the set $L = \{p_0, \ldots, p_{20}\}$, where each $p_i = (x_i, y_i, z_i)$ is expressed in normalised camera coordinates and p_0 is the wrist. To make recognition independent of where the hand is and how far it is from the camera, the landmarks are first centred on the wrist and then divided by the largest distance from the wrist:

$$p'_i = p_i - p_0 \tag{1}$$

$$\hat{p_i} = p'_i / \max_j \|p'_j\|_2 \tag{2}$$

The result is a 63-dimensional vector (21 points × 3 coordinates) that is invariant to translation and scale. Compressing each frame in this way is what allows the classifier to run on a CPU instead of a GPU.

B. *Geometric gesture classification*

Classification is rule-based and works on relative distances rather than fixed pixel thresholds, which keeps it tolerant to hand rotation. A finger f (index, middle, ring or pinky) is considered extended when its tip lies farther from the wrist than its proximal interphalangeal joint:

$$E_f = \square[\, \|p_{\mathrm{TIP},f} - p_0\|_2 > \|p_{\mathrm{PIP},f} - p_0\|_2 \,] \tag{3}$$

The thumb folds across the palm, so its state is measured against the pinky base (landmark 17) instead:

$$E_{\mathrm{thumb}} = \square[\, \|p_4 - p_{17}\|_2 > \|p_2 - p_{17}\|_2 \,] \tag{4}$$

Thumb-up and thumb-down are separated by the vertical offset between the thumb tip and its base with a hysteresis margin $\delta = 0.04$, which prevents flicker between the two states. The five extension flags, the thumb orientation and the proximity of the hand to facial anchor points (needed for signs performed near the face or chest) are mapped to one of twelve classes: HELLO, THANK YOU, YES, NO, HELP, LOVE, STOP, WATER, GOOD, BAD, PLEASE and SORRY. A prediction is accepted only if its confidence is at least 0.70 and the pipeline latency stays within 120 ms; otherwise the interface asks the user to re-align the hand, for example when fewer than 21 landmarks are visible. Because the number of landmarks is fixed, the per-frame cost is constant, i.e. $O(K)$ with $K = 21$.

C. *Sentence aggregation and speech output*

Accepted predictions are pushed into a buffer that suppresses immediate repetitions and groups consecutive signs into a short sentence. The sentence is displayed as text and spoken through the Web Speech API, so a hearing listener receives both a transcript and audio. Each session, with its detected signs and mean confidence, can be stored in the history collection for later review.

D. *Text-to-sign avatar animation*

In the reverse direction, typed text is split into tokens and looked up in an ASL dictionary that stores, for every sign, an animation key and its modalities (hand, head or face). The humanoid avatar, named Michelle, uses a Mixamo skeleton and is drawn with Three.js through React Three Fiber. For the active sign, the system holds target Euler angles for the right arm, forearm, hand and finger bones, and on every rendered frame the current joint rotations are interpolated towards those targets. This smooth interpolation produces fluid motion at 60 FPS and also allows the avatar to mirror the sign the signer has just performed.

V. IMPLEMENTATION AND USER INTERFACE

The prototype is delivered as a single web application with five views. The landing page introduces the project and links to the interpreter (Fig. 2a). The real-time interpreter (Fig. 2b) places the webcam panel beside the avatar panel and offers a switch between “Vision to Text” and “Text to Avatar”; preset buttons let a user trigger common signs directly, and cards below the panels show the detected sign, model confidence and speech state. The dictionary (Fig. 3a) lists every supported sign with a category, a short description of the hand shape and modality badges for hand, head and face movements, and each entry can be demonstrated on the avatar. Registration and sign-in (Fig. 3b) protect the history and administration functions, and an overview page (Fig. 4) summarises the problem, the proposed solution and the processing pipeline from webcam frame to predicted sign.

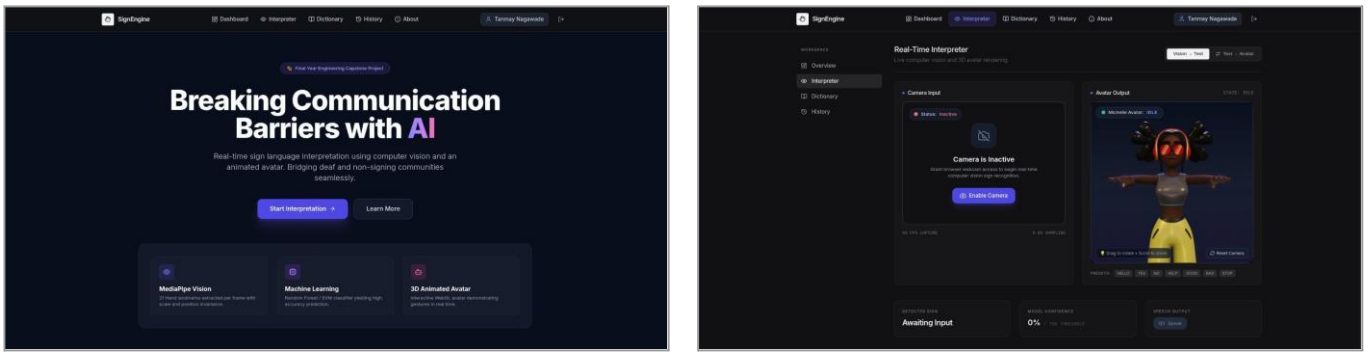

Fig. 2. (a) landing page and (b) real-time interpreter with webcam input and the 3D avatar

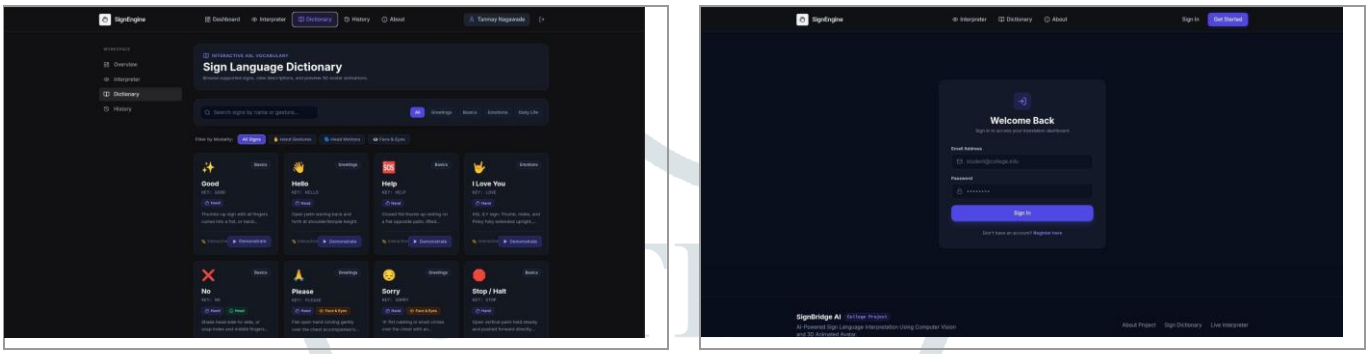

Fig. 3. (a) sign dictionary with modality badges and (b) user authentication portal

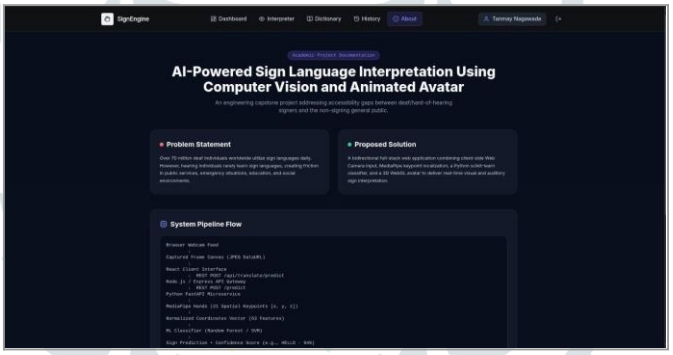

Fig. 4. overview page describing the problem, solution and system pipeline

VI. RESULTS AND DISCUSSION

Table II compares the objectives set for the prototype with the figures reported for it. All values come from the authors' own tests on the twelve-sign vocabulary under the stated operating conditions (640 × 480 frames at 30 FPS, a single person in frame, indoor lighting between 150 and 800 lux), so they describe feasibility of the approach and should not be read as a benchmark against continuous-recognition systems.

Table II. design targets and reported prototype performance

Metric	Design target	Reported for prototype
Recognition accuracy	> 90% on target vocabulary	About 94%
End-to-end latency	≤ 120 ms (limit 150 ms)	About 115 ms on average
Avatar rendering	≥ 60 FPS	Sustained 60 FPS
Hardware	Commodity laptop, 720p webcam	No gloves or depth sensor used

Functional black-box tests (Table III) exercised the main paths. Open-palm, fist and thumb-down gestures were recognised as HELLO, YES and BAD with confidences of 94%, 93% and 95%, and the avatar responded accordingly. The two failure paths, a hand leaving the frame and very low illumination (below about 50 lux), correctly produced a warning instead of a wrong prediction, which is the intended behaviour of the confidence threshold.

Table III. black-box test cases

ID	Scenario	Expected result	Observed result
TC-01	Open palm facing camera	HELLO, confidence ≥ 90%, avatar waves	HELLO, 94%, avatar waved
TC-02	All fingers curled into a fist	YES, confidence ≥ 90%	YES, 93%, avatar nodded

ID	Scenario	Expected result	Observed result
TC-03	Thumb extended downwards	BAD, confidence ≥ 90%	BAD, 95%, avatar mirrored
TC-04	Hand removed from camera view	"No hand detected" message	Warning banner shown
TC-05	Low ambient light (< 50 lux)	Low-confidence resample notice	"Insufficient confidence" shown

Three observations follow. First, normalising by wrist position and hand size, and using distance ratios instead of absolute thresholds, is the main reason the classifier tolerates changes in hand angle and camera distance; this matches the reduced over-fitting to background reported for skeleton-based inputs [6]. Second, the risk analysis in the project identified poor lighting and hand-angle variation as the most probable causes of error, and the confidence gate converts such cases into a request to retry rather than a silent mistake. Third, keeping computation to a few geometric operations per frame leaves enough headroom for the avatar to render at full frame rate in the same browser.

VII. Limitations and Future Directions

The present system recognises isolated signs from a closed list of twelve words, so it is a proof of concept for bidirectional interaction rather than a full interpreter. A single monocular camera gives no true depth, and depth is only approximated from relative distances; occlusion between two hands and signing at very high speed remain difficult. ASL also relies on non-manual markers such as facial expression, head movement and body posture, and on grammar that differs from English word order, so word-by-word mapping can only approximate real signing. Finally, the reported accuracy comes from a small internal test and should be confirmed with a larger group of signers of different skin tones, hand sizes and signing styles, ideally including deaf evaluators [11].

Planned work therefore includes growing the vocabulary, replacing hand-written rules for dynamic signs with spatio-temporal graph convolutional networks (ST-GCN) built on the skeleton-graph idea [10], adding regional Indian Sign Language data, enriching the avatar with facial and torso movement, and running structured usability studies with the deaf community.

VIII. Conclusion

This paper reviewed the progress of sign language technology from sensor gloves to deep sequence models and landmark-based recognisers, and showed how SignEngine joins landmark recognition and avatar generation in one browser application. By reducing each frame to a normalised 63-dimensional vector and using a geometric classifier, the prototype reports roughly 94% accuracy at about 115 ms latency on a twelve-sign vocabulary, while the Three.js avatar sustains 60 FPS. The results suggest that affordable, privacy-conscious, two-way sign communication is technically feasible on everyday hardware, and that the main remaining work lies in vocabulary size, continuous signing and evaluation with deaf users.

IX. Acknowledgment

The authors thank their internal guide Prof. R. Khartude, the Head of the Department of Computer Engineering, and the Principal of HSBPVT's Group of Institutions, Faculty of Engineering, Kashti, for their guidance, encouragement and for providing laboratory and computing facilities.

X. References

[1] World Health Organization, "Deafness and hearing loss," fact sheet. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] T. Starner, J. Weaver, and A. Pentland, "Real-time American Sign Language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1371–1375, 1998.

[3] O. Koller, S. Zargaran, and H. Ney, "Re-Sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE CVPR*, 2017, pp. 4297–4305.

[4] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. IEEE CVPR*, 2018, pp. 7784–7793.

[5] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Syst. Appl.*, vol. 164, Art. no. 113794, 2021.

[6] N. Adaloglou *et al.*, "A comprehensive study on deep learning-based methods for sign language recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 1750–1762, 2022.

[7] F. Zhang *et al.*, "MediaPipe Hands: On-device real-time hand tracking," arXiv:2006.10214, 2020.

[8] C. Lugaresi *et al.*, "MediaPipe: A framework for building perception pipelines," arXiv:1906.08172, 2019.

[9] A. Halder and A. Tayade, "Real-time vernacular sign language recognition using MediaPipe and machine learning," *Int. J. Res. Publ. Rev.*, vol. 2, no. 5, pp. 9–17, 2021.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, 2017.

[11] D. Bragg *et al.*, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *Proc. ACM ASSETS*, 2019, pp. 16–31.