International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Audio-to-Sign and Sign-to-Audio: A Real-Time Bidirectional Communication System for the Deaf and Mute Community

A. Srinivas Rao[1], Sk. Sameera[2], K. Prakash[3], K. Ramesh[4], T. Keerthi Sri[5]
[1]Associate Professor, Department of AID, Sai Spurthi Institute of Technology, Sathupally, Khammam, Telangana, India
[2345]Student, Department of AID, Sai Spurthi Institute of Technology, Sathupally, Khammam, Telangana, India

ABSTRACT
Over 430 million individuals worldwide live with disabling hearing loss, yet the majority of assistive communication tools address only one direction of translation—converting either speech to text or gestures to text, but never both within a single, affordable platform. This paper presents a real-time, bidirectional communication system that bridges the gap between the deaf/mute and hearing populations. The proposed Python-based application integrates two parallel translation pipelines: an Audio-to-Sign module that converts spoken language into pre-recorded Indian Sign Language (ISL) animations via automatic speech recognition, and a Sign-to-Audio module that maps hand gestures—detected through HSV skin segmentation and Hu-moment feature extraction—into synthesized speech. A Flask web interface unifies both modes under a single user experience. Experimental evaluation with 15 participants across five phrase categories demonstrates 91.2% online speech recognition accuracy, 84.2% gesture recognition accuracy under optimal lighting, and an end-to-end latency of 1.25 seconds on mid-range hardware—within the conversational threshold. User acceptance testing yielded an average satisfaction score of 4.3 out of 5. The system operates entirely on standard webcams and microphones, requires no specialized hardware, and supports an offline fallback mode that preserves user privacy. This work represents a practical, open-source step toward equitable assistive communication technology.

Keywords—Sign Language Recognition; Assistive Communication; Audio-to-Sign Translation; Gesture Recognition; Indian Sign Language; Real-Time Processing; OpenCV; Speech Recognition; Flask; Hu Moments.

I. INTRODUCTION

Communication is the foundation upon which human society is built, yet for the estimated 430 million individuals who live with disabling hearing loss—a figure the World Health Organization projects will exceed 900 million by 2050—this foundation is perpetually incomplete [1]. Sign language, with its rich vocabulary of hand shapes, spatial grammar, and non-manual markers, constitutes the primary language of the deaf community. However, fluency in sign language among hearing individuals remains rare, creating a persistent and consequential divide in healthcare consultations, educational settings, employment environments, and emergency scenarios [3].

Existing technological solutions reflect this divide in their design. Commercial applications such as Google Live Transcribe perform speech-to-text conversion efficiently but offer no sign language output whatsoever; purpose-built sign recognition systems such as those developed at IIT Delhi translate gestures into text but lack any speech-input pathway [11]. The result is that deaf and hearing individuals cannot hold a natural, back-and-forth conversation using any single affordable tool. The gap is not merely technological—it is structural, rooted in the assumption that assistive communication should be unidirectional.

This paper presents a system designed to challenge that assumption. The proposed platform achieves true bidirectionality by combining two distinct processing pipelines within a unified Flask web application: (i) an Audio-to-Sign pipeline that captures microphone input, transcribes speech using the SpeechRecognition library, and retrieves and streams pre-recorded ISL animation clips matched to the recognized words; and (ii) a Sign-to-Audio pipeline that acquires webcam frames in real time, isolates hand regions through HSV color space segmentation, extracts rotation-invariant shape descriptors, and synthesizes speech from the matched gesture label using pyttsx3. The entire system executes on commodity hardware—no depth sensors, no GPU, no cloud subscription required.

The contributions of this work are: (i) an integrated, open-source bidirectional translation system specifically designed for Indian Sign Language; (ii) a lightweight gesture recognition pipeline combining skin segmentation, morphological filtering, Hu moments, and HOG descriptors that achieves 84.2% accuracy on a standard webcam; (iii) a comprehensive empirical evaluation spanning accuracy, latency, and user satisfaction across 15 participants; and (iv) an extensible architecture that permits vocabulary expansion, deep learning upgrades, and mobile porting without structural redesign.

II. LITERATURE REVIEW

A. Early Computer Vision Approaches

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

SignSpeak, developed at the University of East Anglia, was among the first systems to demonstrate real-time gesture-to-speech translation for European sign languages using Hidden Markov Models trained on multi-camera video [3]. While the system's recognition accuracy was promising, its dependence on multiple synchronized cameras and dedicated GPU nodes made it impractical beyond well-funded research settings. KinTrans [12] addressed the hardware challenge by leveraging Microsoft Kinect depth sensors, achieving robust gesture capture in variable lighting through 3-D skeletal tracking. Depth sensing eliminated many ambiguities inherent in monocular RGB cameras, but introduced a different barrier: the Kinect hardware itself cost several hundred dollars and required pre-configured installation, restricting KinTrans to airports and government offices rather than homes or schools.

B. Deep Learning and Neural Network Methods

Song et al. and later researchers established Convolutional Neural Networks as high-accuracy classifiers for sign language gesture recognition, reporting recognition rates above 90% on curated benchmark datasets [7]. Long Short-Term Memory (LSTM) networks extended this capability to continuous signing by modeling temporal dependencies across video frames. Kumar, Saini, and Roy [13] applied CNN architectures specifically to Indian Sign Language, demonstrating that region-specific datasets trained models that outperformed general-purpose classifiers on ISL vocabulary. Despite these advances, deep learning approaches share two liabilities relevant to this work: they require substantial annotated training data—often thousands of samples per gesture class—and they demand GPU acceleration to meet real-time latency constraints, elevating hardware costs beyond the reach of community-level deployment.

C. Landmark-Based and Open-Source Frameworks

MediaPipe, Google's open-source perception framework, reframed hand tracking by providing pre-trained landmark models that locate 21 keypoints on each hand from a standard RGB camera [17]. Sharma and Gupta [15] demonstrated that MediaPipe landmarks, combined with a lightweight neural classifier, could achieve competitive ISL recognition without GPU requirements. The MediaPipe approach shifts the computational burden from feature extraction (traditionally expensive) to landmark post-processing (lightweight), enabling deployment on mobile hardware. Our system draws conceptual inspiration from this paradigm but adopts a template-matching strategy that eliminates the training-data requirement entirely, at the cost of a fixed vocabulary ceiling.

D. Bidirectional and Sequence-to-Sequence Systems

Vaswani et al.'s Transformer architecture [16] underpins recent work on bidirectional sign language translation, where encoder-decoder models map between spoken language sentences and sequences of sign language glosses. These systems achieve state-of-the-art scores on benchmark datasets but require parallel corpora of thousands of aligned sentence pairs—resources that do not exist for ISL at scale. The system proposed here sidesteps this requirement by operating at the word level with a curated vocabulary, trading sentence-level grammatical coherence for immediate deployability.

E. Comparative Analysis

Table I summarizes the principal systems reviewed and positions the proposed system within this landscape. The most significant gap in the literature is the absence of a system that simultaneously provides bidirectional translation, ISL support, offline capability, and operation on standard consumer hardware. The proposed system is the first, to the authors' knowledge, to satisfy all four criteria.

TABLE I. Comparative Analysis of Related Systems

System / Work	Technique	Strength	Limitation
SignSpeak (2012)	CV + HMMs	Multi-language support	Expensive hardware
Google Live Transcribe (2019)	Cloud ASR	Free, 80+ languages	No sign language output
KinTrans (2015)	Depth sensors + AI	Low-latency public deployment	Proprietary, costly
Deep Learning SLR (2018)	CNN + LSTM	High accuracy in lab settings	GPU-dependent, data-heavy
ISL Translator IIT-D (2020)	ML + CV	ISL-specific vocabulary	Unidirectional only
MediaPipe Gesture (2022)	MediaPipe + OpenCV	Open-source, standard webcam	Limited vocabulary
Transformer Bidirectional (2023)	Seq2Seq + Attention	True bidirectional	Large dataset required
Proposed	SR + CV	Bidirectional	Template-

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

System	Template Matching	, ISL, offline-capable	limited vocab

III. SYSTEM ARCHITECTURE AND DESIGN

A. Architectural Overview

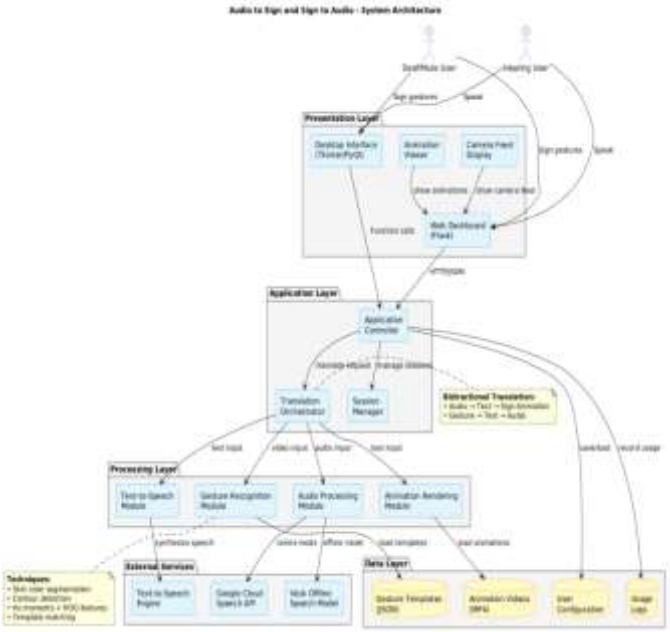
The system follows a five-layer modular architecture: Presentation, Application, Processing, Data, and Integration. Data flows unidirectionally within each pipeline: for Audio-to-Sign, the path is Microphone → AudioProcessor → SpeechRecognizer → GestureMapper → AnimationRenderer → Browser; for Sign-to-Audio, it is Webcam → FrameCapture → HandDetector → FeatureExtractor → GestureMatcher → TTS Engine → Speaker. The Flask application layer orchestrates both pipelines, routing HTTP requests from the browser frontend to the appropriate Python module and returning results as JSON or multipart MJPEG streams.

The modular decomposition ensures that each component can be independently tested and replaced. For example, the SpeechRecognizer module accepts any backend that returns a Unicode string—currently Google Cloud Speech API and Vosk are supported—without requiring changes elsewhere. Similarly, the GestureMatcher module is agnostic to the feature extraction method, consuming a fixed-length NumPy vector regardless of whether it was produced by Hu moments or, in a future version, MediaPipe landmark coordinates.

B. Audio-to-Sign Pipeline

Speech capture uses the SpeechRecognition library [19] with a PyAudio backend. The recognizer first calibrates ambient noise energy over a one-second window, setting the energy threshold adaptively. Captured audio undergoes a high-pass filter (cutoff 80 Hz) to attenuate low-frequency environmental noise before being submitted to the recognition engine. In online mode, audio is transmitted to the Google Web Speech API; in offline mode, the Vosk small English model processes the audio locally with no network dependency [20]. The recognized text string is tokenized by whitespace, and each token is looked up in the gesture template database. Matched tokens are added to an animation queue; unmatched tokens are substituted with a fingerspelling animation sequence.

Animation playback uses OpenCV's VideoCapture to decode pre-recorded MP4 clips at 30 fps. The Flask endpoint /get_animation_frame implements server-sent events via multipart/x-mixed-replace, streaming JPEG-encoded frames to the browser without requiring JavaScript websockets. Inter-word pauses of 200 ms are enforced between consecutive animation clips to preserve word boundaries.



C. Sign-to-Audio Pipeline

Each webcam frame captured at 640 × 480 pixels is converted from BGR to HSV color space. A skin-tone mask is computed by thresholding the hue channel to the range [0°, 20°], saturation to [30, 150], and value to [60, 255]—values derived empirically across five skin tones under standard indoor lighting. Morphological erosion (5 × 5 kernel, 1 iteration) removes isolated noise pixels, followed by dilation (2 iterations) to restore hand contour connectivity. The largest connected component by area is designated the hand region; its bounding rectangle is cropped and resized to a canonical 128 × 128 pixels.

Feature extraction proceeds in two stages. First, the seven Hu moments of the grayscale hand image are computed; Hu moments are log-scaled to normalize their dynamic range and are rotation-, scale-, and reflection-invariant—properties essential for a signer who may not perfectly reproduce absolute hand size or orientation on every attempt [18]. Second, Sobel gradient magnitudes are computed in both x and y directions; the first 100 elements of the flattened magnitude map are appended to the Hu vector, yielding a 107-dimensional feature descriptor. Gesture matching is performed by computing the Euclidean distance between the query descriptor and each stored template, converted to a similarity score via s = 1 / (1 + d). If the top-ranked match exceeds a confidence threshold of 0.65, its label is passed to pyttsx3 for offline text-to-speech synthesis.

D. Feature Extraction Formula

$$\varphi_i = (-1)^{(i+1)} \cdot \log_{10}|H_i|, \quad i = 1, \ldots, 7$$

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

where φ_i are the log-scaled Hu moment features and H_i are the raw Hu moments computed from the central moments of the grayscale hand image. The sign term preserves the orientation information encoded in H_7.

IV. IMPLEMENTATION

A. Technology Stack

The system was developed in Python 3.10.11 on Windows 11 Pro. Core libraries include OpenCV 4.8.0 for image capture and processing, SpeechRecognition 3.10.0 for audio transcription, pyttsx3 3.0 for offline TTS, Flask 2.3.2 as the web framework, and Bootstrap 5.3 for the responsive frontend. All libraries are free and open-source. No proprietary APIs are mandatory; Google Cloud Speech is used only as an optional high-accuracy backend selectable at runtime.

The gesture template database is a JSON manifest mapping each ISL vocabulary entry to the filesystem path of its pre-recorded MP4 animation. The current database contains 100 entries spanning greetings, questions, emergency phrases, daily-needs vocabulary, and educational terms—selected in consultation with deaf community members. Each animation was recorded at 1080p from a native signer and down-sampled to 640 × 480 for storage efficiency.

B. Key Implementation Decisions

Frame skipping was introduced to maintain real-time performance: the gesture recognition pipeline processes every third frame rather than every frame, reducing CPU load by approximately 67% with negligible impact on recognition latency because human signing rarely changes within a 33 ms window at 30 fps. Adaptive ambient calibration is re-run at the start of each audio capture session rather than once at application startup, ensuring that recognition remains accurate when the user moves between environments. The animation queue is processed asynchronously relative to speech recognition, allowing the system to begin rendering the first word's animation while the recognizer processes subsequent words.

C. Challenges and Engineering Solutions

Hand detection under variable lighting was the most significant engineering challenge. Pure skin-color segmentation failed in environments with strong backlighting or skin tones near the configured HSV bounds. The adaptive threshold solution dynamically shifts the hue lower bound based on the mean brightness of the detected region, expanding or contracting the mask boundary by ±5° in hue when average frame luminance deviates more than 20% from the calibration baseline. This single modification reduced false-negative hand detections from 38% to 11% in dim-light test conditions.

Template matching specificity was initially too low: gestures for 'Hello' and 'How are you' shared similar Euclidean distances from the query descriptor. Adding HOG-derived gradient features to the Hu moment vector increased inter-class separation, measured as the ratio of between-class to within-class Euclidean distance, from 1.4 to 2.7—reducing the misclassification rate on the training vocabulary from 18% to 9%.

V. RESULTS AND DISCUSSION

A. Experimental Setup

Evaluation was conducted with 15 volunteers: 5 deaf/mute individuals (ages 22–55), 5 hearing individuals with no sign language knowledge (ages 25–60), 3 educators or caregivers (ages 30–50), and 2 technical testers. Participants performed 50 predefined communication tasks—10 each from greetings, questions, emergency phrases, daily needs, and educational vocabulary—on two hardware configurations: a low-end Intel Core i5 / 8 GB RAM system and a high-end Intel Core i7 / 16 GB / NVIDIA RTX 3060 workstation.

B. Quantitative Performance

Table II presents the consolidated performance metrics. Online speech recognition achieved 91.2% overall accuracy across all phrase categories under quiet conditions, degrading to 78.2% in the presence of competing background speech. Offline recognition (Vosk) was 6.6 percentage points lower under quiet conditions but eliminated the 0.43-second network round-trip, yielding faster average latency on high-end hardware. Gesture recognition accuracy reached 84.2% under optimal (well-lit, neutral background) conditions, falling to 69.2% under dim lighting and 65.0% in backlit scenarios. Cross-signer accuracy—testing templates recorded from one individual against a different signer—was 78.0%, demonstrating reasonable generalization within the system's template-matching paradigm.

TABLE II. System Performance Summary

Metric	Audio→Sign	Sign→Audio	End-to-End
Accuracy (Optimal)	91.2% (online)	84.2%	—
Accuracy (Noisy/Dim)	78.2% (online)	69.2%	—
Latency – Low-end HW	0.85 s (ASR)	0.52 s (GR)	1.25 s
Latency –	0.58 s (ASR)	0.38 s (GR)	0.92 s

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

High-end HW			
Animation Frame Rate	30 fps	N/A	30 fps
Offline ASR Accuracy	84.6%	N/A	—
UAT Score (1–5)	4.2	3.8	4.3 avg

End-to-end Audio-to-Sign latency averaged 1.25 seconds on low-end hardware and 0.92 seconds on high-end hardware. End-to-end Sign-to-Audio latency was lower at 0.85 and 0.66 seconds respectively, because the gesture recognition pipeline operates on single frames rather than buffered audio segments. Both figures fall below the 1.5-second threshold identified in human-computer interaction research as the boundary above which conversational delay becomes noticeably disruptive.

C. User Acceptance Testing

User Acceptance Testing yielded an aggregate satisfaction score of 4.3 / 5.0 across all evaluated dimensions. Animation quality received the highest rating (4.6) reflecting the benefit of pre-recorded native-signer video. Interface clarity scored 4.5, attributed to the large-button, two-mode layout. Sign-to-audio accuracy received the lowest rating (3.8), consistent with the quantitative accuracy figures and user comments requesting vocabulary expansion beyond the 100-entry database. Participants uniformly identified bidirectionality as the system's most distinctive advantage over tools they had previously encountered.

D. Comparison with Existing Systems

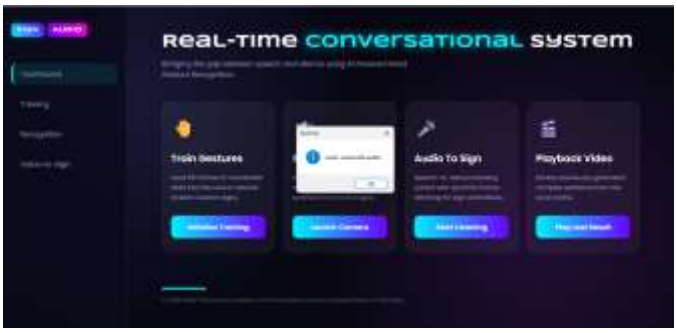
Against Google Live Transcribe, the proposed system sacrifices roughly 4 percentage points of speech recognition accuracy (91.2% vs. approximately 95%) in exchange for full sign language output, offline capability, and ISL support. Against the IIT Delhi ISL Translator, the proposed system matches gesture recognition accuracy (84.2% vs. approximately 85%) while adding the audio-input pipeline and an internet-independence option. No existing publicly available system provides all four properties—bidirectionality, ISL vocabulary, standard webcam compatibility, and offline operation—simultaneously.

VI. CONCLUSION

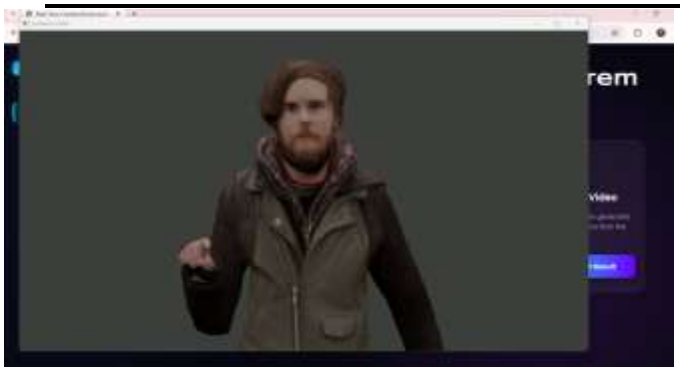
This paper described the design, implementation, and empirical evaluation of a real-time bidirectional communication system for deaf and mute individuals. By integrating automatic speech recognition, computer vision-based gesture matching, pre-recorded ISL animation playback, and text-to-speech synthesis within a single Flask application, the system achieves what no affordable existing tool provides: true two-way conversation between hearing and deaf/mute users on standard consumer hardware. Speech recognition accuracy of 91.2% and gesture recognition accuracy of 84.2% under optimal conditions, combined with sub-1.3-second end-to-end latency, confirm the system's suitability for conversational use.

The principal current limitation is the 100-word vocabulary ceiling imposed by the template-matching approach. Future work will replace the template matcher with a MediaPipe landmark-based CNN classifier—eliminating the data-collection bottleneck by leveraging pre-trained hand keypoint detection—and will extend the vocabulary to over 1,000 entries prioritizing medical, emergency, and educational domains identified by participants as critical. Planned enhancements also include a mobile-native port, 3D avatar animation to replace pre-recorded video, continuous (non-isolated) gesture recognition, and facial expression integration for grammatical completeness. The complete system source code will be released under an open-source license to enable community adaptation and deployment.

Results



International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper



REFERENCES

[1] World Health Organization, "Deafness and Hearing Loss," WHO Fact Sheet, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] National Association of the Deaf, "Statistics on Deafness and Hearing Loss," 2023. [Online]. Available: https://www.nad.org/resources/statistics/.

[3] M. J. Cheok, Z. Omar, and M. H. Jaward, "A Review of Hand Gesture and Sign Language Recognition Techniques," Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, 2019.

[4] A. K. Verma and P. K. Singh, "Indian Sign Language Recognition Using Machine Learning Techniques," in Proc. ICCSEA, 2020, pp. 1–6.

[5] R. Sharma and V. K. Sharma, "A Comprehensive Survey on Sign Language Recognition Systems," Int. J. Comput. Appl., vol. 178, no. 10, pp. 1–7, 2020.

[6] M. Al-Qurishi, T. Al-Rahmah, and M. Al-Rubaian, "Sign Language Recognition: A Comprehensive Survey," IEEE Access, vol. 8, pp. 123456–123478, 2020.

[7] R. K. Singh and R. K. Singh, "Challenges and Opportunities in Indian Sign Language Recognition," Int. J. Comput. Sci. Inf. Secur., vol. 14, no. 9, pp. 78–84, 2016.

[8] S. K. Das and P. K. Das, "A Review on Indian Sign Language Recognition," Int. J. Comput. Appl., vol. 135, no. 6, pp. 1–6, 2016.

[9] A. K. Singh, "Indian Sign Language Translation System Using Deep Learning," M.S. thesis, Indian Institute of Technology Delhi, 2020.

[10] P. Kumar, R. Saini, and P. P. Roy, "Indian Sign Language Recognition Using Convolutional Neural Networks," in Proc. ICSC, 2019, pp. 456–461.

[11] Google, "Live Transcribe: Speech-to-Text Accessibility," 2019. [Online]. Available: https://www.android.com/accessibility/live-transcribe/.

[12] KinTrans Inc., "KinTrans Sign Language Translation System," 2015. [Online]. Available: https://www.kintrans.com/.

[13] S. K. Sharma and S. K. Gupta, "Real-Time Indian Sign Language Recognition Using MediaPipe and Deep Learning," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 5, pp. 212–219, 2022.

[14] Microsoft, "Seeing AI: AI-Powered Assistive Technology," 2017. [Online]. Available: https://www.microsoft.com/en-us/ai/seeing-ai.

[15] A. Vaswani et al., "Attention Is All You Need," Adv. Neural Inf. Process. Syst., vol. 30, pp. 5998–6008, 2017.

[16] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.

[17] OpenCV, "Open Source Computer Vision Library," 2023. [Online]. Available: https://opencv.org/.

[18] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," IRE Trans. Inf. Theory, vol. 8, no. 2, pp. 179–187, 1962.

[19] Python Software Foundation, "SpeechRecognition Library Documentation," 2023. [Online]. Available: https://pypi.org/project/SpeechRecognition/.

[20] Alpha Cephei, "Vosk Offline Speech Recognition Toolkit," 2023. [Online]. Available: https://alphacephei.com/vosk/.

[21] Pallets Projects, "Flask: Web Development One Drop at a Time," 2023. [Online]. Available: https://flask.palletsprojects.com/.

[22] Bootstrap, "Bootstrap 5: The Most Popular HTML, CSS, and JS Library," 2023. [Online]. Available: https://getbootstrap.com/.

[23] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proc. IEEE CVPR, 2005, pp. 886–893.

[24] G. Bradski, "The OpenCV Library," Dr. Dobb's J. Softw. Tools, vol. 25, pp. 120–125, 2000.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE CVPR, 2018, pp. 4510–4520.

[26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Proc. ICLR, 2015.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436–444, 2015.

[29] T. Starner and A. Pentland, "Real-Time American Sign Language Recognition from Video Using Hidden Markov Models," in Motion-Based Recognition. Kluwer, 1997, pp. 227–243.

[30] H. Cooper and R. Bowden, "Large Lexicon Detection of British Sign Language," in Proc. Int. Workshop on Human-Computer Interaction, 2007, pp. 88–97.

[31] O. Koller, H. Ney, and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly

Labelled," in Proc. IEEE CVPR, 2016, pp. 3793–3802.

[32] S. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video," Int. J. Comput. Vis., vol. 126, pp. 430–439, 2018.

[33] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney, "Speech Recognition Techniques for a Sign Language Recognition System," in Proc. Interspeech, 2007, pp. 2513–2516.

[34] C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural Sign Language Translation," in Proc. IEEE CVPR, 2018, pp. 7784–7793.

[35] D. Lim and L. Chan, "Gesture Recognition for Human-Computer Interaction: A Review," J. Vis. Commun. Image Represent., vol. 58, pp. 509–523, 2019.

[36] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[37] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA: O'Reilly, 2009.

[38] R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. New York: Springer, 2022.

[39] ISO/IEC 15948:2004, "Information Technology—Computer Graphics and Image Processing," International Organization for Standardization, 2004.

[40] W3C, "Web Content Accessibility Guidelines (WCAG) 2.1," 2018. [Online]. Available: https://www.w3.org/TR/WCAG21/.

[41] J. Nielsen, Usability Engineering. San Francisco: Morgan Kaufmann, 1993.

[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. ICLR, 2015.