

Vision-Two-Voice: An AI-Powered Real-Time Two-Way Sign Language Detection and Communication System

Tanuja Balasaheb Kshirsagar
Department of Computer Science
JSPM University
Pune, India
kshirsagarb2002@gmail.com

Prof. Manisha Kshirsagar
Department of Computer Science
JSPM University
Pune, India
msk.scos@jspmuni.ac.in

ABSTRACT

Communication barriers between hearing-impaired individuals and non-sign language users often hinder effective interaction in daily life. Existing sign language translation systems primarily focus on one-way communication, limiting their practical usability. This research presents Vision-Two-Voice, an AI-powered real-time two-way sign language detection and communication system designed to facilitate seamless interaction between sign language users and spoken language users. The proposed system employs computer vision and deep learning techniques to recognize hand gestures, facial expressions, and body movements associated with sign language, converting them into text and speech in real time. Simultaneously, the system incorporates speech recognition and natural language processing to transform spoken language into corresponding sign language representations, enabling bidirectional communication. *Key words: Sign Language Recognition, Artificial Intelligence, Computer Vision, Deep Learning, Real-Time Communication, Speech Recognition, Natural Language Processing, Gesture Detection, Human-Computer Interaction, Assistive Technology, Accessibility, Bidirectional Communication, Sign Language Translation, Machine Learning, Inclusive Technology.*

I. INTRODUCTION

Sign language is the primary way of communicating for many millions of people who are deaf or have a speech disability. According to the World Health Organisation (WHO), around 5% of the world's global population is adversely affected by some form of disability (hearing loss). However, there is a significant lack of qualified and competent interpreters for sign language, which continues to present a significant barrier to effective communication for thousands of people who use this form of communication in various settings (educational, medical, banking, and emergency services).

Advances in computer vision and deep learning technology have made it possible to accurately identify gestures, based solely on the use of traditional RGB video feeds instead of the use of specialised sensors. Real-time hand and body pose

detection as well as facial landmark detection can be achieved using the MediaPipe library on-device models. The real-time detection of static and dynamic hand poses can be achieved by the integration of LSTM and CNNs, allowing gesture mappings from output data to be based on the result of using these two networks.

This paper discusses an AI-driven, bidirectional translating device that translates between spoken and signed languages (Vision to Voice). It has a novel feature that allows for palm-spreading detection to allow for verbalizing “Hello” or “Hi” through TTS (text-to-speech) when all five fingers of the user's hand have been detected and are in an open palm-spreading position. This creates the ability to provide a natural greeting for a deaf person without them having to sign a formalized word [2],[11].

The first point of contact is essential for all conversations, and automating the first warm greeting provides a reduction in awkwardness associated with the start of any combination of deaf and hearing. The paper goes on to provide the following: literature review, gaps in the research, methodology, system architecture, greeting creative pipeline, results, and resources.

II. LITERATURE REVIEW

The use of Hidden Markov Models (HMM) has historically been used for early sign language recognition based on hand-tracked video features; this was illustrated by the work of Starner et al. [4] who developed one of the first real-time American Sign Language (ASL) recognition systems. While these were significant developments, they were highly sensitive to lighting and had small vocabularies for recognition. The introduction of deep learning revolutionized sign language recognition; in particular, convolutional neural networks (CNN) became the dominant paradigm. The work of Pigou et al. [5] demonstrated that CNNs could be used to improve the level of accuracy of recognition systems by using depth and intensity video instead of handcrafted features. Bantupalli and Xie [12] developed a hybrid neural network architecture composed of CNNs for feature extraction and recurrent layers for continuous ASL recognition from RGB video.

Approaches using sensors for tracking gesture recognition, such as Leap Motion and skeletal joints have been able to achieve very high levels of precision; however, they unfortunately require additional hardware, thereby restricting access to gesture recognition systems. MediaPipe has transformed the way gesture recognition systems are created, by providing an open-source and cross-platform framework for implementing the use of media, music, and real-time perception-based applications. MediaPipe Hands detects 3D hand landmarks with high frame rate accuracy, while BlazePose detects poses of the human body accurately. The integration of these two systems with ML classifiers demonstrated high levels of accuracy while maintaining low computational demands.

Rastgoo et al., Cheok et al., and Garg and Naik provide detailed reviews of gesture recognition systems, as well as documenting the most significant barriers remaining in gesture recognition technology: lack of diversity in datasets, real-time performance, and the inability to interpret non-manually marked gestures. One consistent observation among the reviewed works is that the majority of systems only allow for one directional conversion (e.g., sign to text) to occur, therefore, the majority of the methods do not allow for the conversion of spoken language into signed language back to the deaf user — this is the primary motivation behind the Vision-Two-Voice.

The use of recurrent network architectures was essential for recognizing continuous sentences. Cui et al. [8] proposed a recurrent convolutional architecture to enable continuous sign language recognition. Koller et al. [9] introduced large-vocabulary continuous recognition benchmarks based on the RWTH-PHOENIX corpus. Camgoz et al. [6] expanded this idea to develop a complete neural network for sign language translation based on encoder-decoder architectures, known as transformers. [17] has since been widely adopted due to its strength in modeling long-range temporal dependencies.

III. OBJECTIVES AND SCOPE

Objectiv

- The project objectives include: 1) designing a real-time method of detecting Sign Language using Mediaipe for capturing hand, pose, and facial landmark data from a standard webcam; 2) creating a hybrid model (CNN-LSTM) that classifies temporal sequences of landmarks (hands and faces) into words/phrases; 3) incorporating Text-To-Speech (TTS) conversion of recognized sign language into speech for hearing users; 4) creating an alternate means of communication from spoken text to signed gesture/avatar animation; 5) implementing palm spread detection to auto-generate greetings such as 'Hi!' / 'Hello!' when two people are trying to get each other's attention; and 6) to collect/preprocess a data set of gestures (including the aforementioned), determine the accuracy, latency, and robustness of the data set against the evaluation criteria of other publicly available signed communication datasets.

Key Limitations

- Accuracy decreased by lack of illumination, disarrayed background, and obstructive visual obstruction from people or objects. Non-manual grammatical markers (facial expressions and head tilt) only detected to a limited extent. There are variations in sign language use from individual to individual; therefore they may require tuning. Cloud-based STT and text-to-speech services require a reliable internet connection to function optimally [25].

IV. METHODOLOGY

A. Data Collection

A systematic signing dataset was created from many volunteers signing various sign phrases (Hello, Thank you, I'm Sorry, Yes and No, Please), across a variety of lighting, multiple backgrounds, and varied distances from the camera. Each of the classified sign phrases were captured (recorded) several times, by multiple signers, as to provide a true diversity of signing type. Each signing area was recorded at frame extraction rates of 15 to 30 frames per second.

The normalisation of the 0 coordinate landmarks (based upon the wrist) and the use of a bounding box for scaling the remaining landmarks resulted in the landmarks being placed relative to each other and therefore invariant to the camera's distance from the signer's body and the position of the camera in regard to the signer. Data augmentation was applied including random rotation, horizontal flip, and brightness/contrast jitter to ensure that the system can perform well under realistic conditions.

B. Data Preprocessing

The labelled signing data was split into three sets: a training, validation and testing set with each set made up of separate recording sessions (i.e., there can be no recordings from the same session across all three combination of datasets). By separating recording sessions this way, we remove any chance of data leakages occurring. In order to create a fixed-length sequence of landmark vectors, for dynamic signing we concatenate the landmark vectors of the previous N frames in order to create a fixed length sequence of normalised landmark vectors (i.e., padded or truncated) that can then be used as input into an LSTM network. To create designs with synthetic variability, brightness/contrast means were applied to the original video frames before they were used to extract the landmarks.

C. System Architecture (Fig. 1)

The system architecture for Vision-Two-Voice is a bidirectional architecture as shown in Fig. 1. There are two avenues of communication centred on a shared MediaPipe landmark-extraction core. Both communication pathways utilize the same underlying MediaPipe landmark extraction core.

In the first pathway (i.e., sign-to-speech), the camera captures the sign language gesture(s) made by the person; the gestures' landmarks are extracted using MediaPipe; the CNN-LSTM model classifies the gesture that was signed; and the TTS engine produces speech output for a hearing person corresponding to the recognized printed text from the classifier model output [25].

In the second pathway (i.e., speech-to-sign), when a spoken word is heard via STT, STT converts it to textual printed form, which is then mapped onto the sign vocabulary (via an NLP module), which will then generate a printed image of the signed word(s), or, alternatively generate an avatar animation showing how the signed word(s) will look when signed. In addition, a bidirectional communication connection between both pathways allows for genuine two-way communication rather than one-way translation of sign to speech or speech to sign.

Fig. 1: System Architecture

SIGN → SPEECH	SHARED CORE	SPEECH → SIGN
Webcam / Camera	MediaPipe Landmarks	Microphone
Landmark Norm.	Feature Vector	STT API
CNN-LSTM Classifier	Hybrid Model	NLP Token Mapper
TTS Engine → Audio	Bidir. Bridge	Avatar → Visual

Fig. 1. Bidirectional sign-to-speech and speech-to-sign architecture of Vision-Two-Voice [1],[2],[25].

D. MediaPipe Pipeline & Greeting Detection

The process, illustrated in Figure 2, consists of 8 consecutive stages involving the conversion of a raw image taken from a camera to create an interpretation as to what type of gesture or greeting is present in the current frame. In regard to the mechanism used by a creative greeting detector at Stage 5 of this process; after performing landmark normalizations on the landmarks associated with the gestures, two consecutive fingertip pair landmarks are compared against each other (landmark pairs 4-8, 8-12, 12-16 and 16-20) in order to

measure the normalized distance between these pairs, then calculates a score for each pair. If all five fingertips have a score greater than a defined threshold (0.25 normalized units), AND the normalized vector of the palm points away from the wrist (defined by the z-coordinate of landmark 0 relative to landmarks 5 and 17), then the system sends a command immediately to initiate a creative greeting (e.g., Text-to-Speech would say “Hi!” on first detection and “Hello!” on all subsequent detections in the same session). The creativeness of this method comes from providing users with a natural, personalized greeting experience without prior explicit training for the gesture vocabulary [2],[11],[25].

Fig. 2: CV Pipeline & Greeting Detection

#	Stage	Description & Output
1	Frame Capture	OpenCV reads webcam frame BGR→RGB. <u>25–30 FPS</u> on standard CPU.
2	Palm Detection	Lightweight CNN detects hand bounding boxes. Robust to scale & occlusion.
3	Landmark Reg.	Regression model predicts <u>21 3D keypoints</u> (wrist, knuckles, tips) per hand.
4	Normalization	(x,y) normalized to wrist (pt 0), scaled by bounding-box. Distance-invariant.
5	Greeting Detect	ALL 5 tips spread > <u>0.25 threshold</u> + palm facing outward → “Hi!” / “Hello!” via TTS.
6	Seq. Assembly	30-frame normalized vectors stacked into spatio-temporal tensor for LSTM.
7	Classification	CNN-LSTM outputs class probabilities; top-1 selected (conf ≥ <u>0.80</u>).
8	TTS Output	Recognized text → pyttsx3/Google TTS → spoken audio output in real time.

Fig. 2. 8-stage CV pipeline from camera frame to gesture output, including palm-spread greeting detector at Stage 5 [1],[2],[11].

The “Hello” gesture is an exception to the rule of classification as it will additionally trigger the greeting detection logic prescribed in Section IV-C. The gesture recognition system does not only classify a static hand sign, it also tracks the rotation and position of the palm to respond with a verbally contextualized greeting [2],[11] to the user. This creative combination of rule-based landmark geometry and model-based classification provides a more fluid interaction with the user.

Regarding gestures that depict a motion such as “Thank You” or “Please”, the LSTM layers model how a gesture develops over time and across each frame in the 30-frame input window; thus, allowing the system to distinguish between a static hand shape and the same motion path to produce two different meanings [16].

E. Gesture Classes (Table II)

The information found in Table II contains a summary of gesture vocabulary. The following information is provided for

each item in the gesture vocabulary: (1) the configuration of the landmarks; (2) the type of motion; (3) how the gesture will be triggered; (4) what output is produced by the gesture recognition system.

TABLE II: Gesture Classes & System Outputs

Gesture	Key Landmarks	Motion	Trigger	Output
Hello	All 5 tips extended Palm outward	Static + wrist wave	Tips spread ≥ 0.25 norm.	Hi!/Hello!
Thank You	Flat hand, tips near chin	Dynamic – fwd arc	Hand-to-chin 15–20 frames	“Thank You”
Yes	Closed fist, thumb up	Static / nod	Thumb-up landmarks	“Yes”
No	Index ext., wrist lateral	Dynamic – lateral	Wrist angle $\geq 30^\circ$	“No”
Please	Flat hand, chest circle	Dynamic – circular	Circular traj. 20+ frames	“Please”
Sorry	Closed fist, chest circle	Dynamic – circular	Fist + circular chest region	“Sorry”

Table II. Gesture vocabulary with landmark configurations, motion types, detection triggers, and system outputs [2],[7],[11],[22].

F. Model Design

The hybrid CNN-LSTM architecture is used for gesture classification [15],[16]. The normalized landmark vector created from each frame is first processed by a small block of 1D convolutional layers that learn compact spatial representations for each frame. After processing as above, the per frame embeddings are sent into LSTM layers which model how the gesture evolves over time within the 30-frame input window.

The model's last layer produces class probabilities using the softmax function over a vocabulary of different gestures. The model is trained on categorical cross-entropy, using the Adam optimizer, and dropout was applied between LSTM

and dense layers to reduce possible overfitting caused by having a relatively small custom dataset [7].

G. Model Evaluation

Standard classification metrics are used for evaluation – overall accuracy, per-class precision/recall/F1 score, confusion matrix (to show which gesture pairs are most commonly confused). In addition to classifying results, the system will also be evaluated on real-time performance by measuring latency from frame capture to spoken output [1],[2].

Robustness will be evaluated utilizing varying lighting, backgrounds and signers, as defined by the data augmentation strategy. Convergence will be observed through validation

accuracy reaching a plateau while training accuracy is still increasing, indicating overfitting has started and providing a basis for early stopping [15],[16].

V. RESULTS AND DISCUSSION

Preliminary evaluation of the dataset based on custom landmarks indicates that the landmark-based classification system converges quickly via CNN-LSTM. Validation accuracy will parallel training accuracy after approximately 15–20 epochs and will stabilize at about 99.0% (training) and 97.0% (validation) through low overfitting. The results on the held-out test set indicate the system achieves an overall classification accuracy of 90.4% (as shown in Table III; with a per-class F1-score average between 87–94%) using MediaPipe landmark extraction [1],[2]. MediaPipe landmark extraction is optimised to run on a CPU and achieves very high performance operating in real-time within a standard laptop without using a GPU. The palm greeting detection system performed well in isolated tests, achieving 96.2% precision and 94.8% recall rates, thereby providing reliable results for this particular automation use case [2],[11].

From examining the confusion matrix, it appears that most misclassifications happen between gesture pairs with common or visually similar hand shapes or have intersecting motion trajectories (e.g., ISL “Thank You” and ISL “Sorry” gestures both follow a hand-to-chin arc). Adding more facial-

expressions landmarks could increase the overall detection accuracy by over 93% [3],[7].

TABLE III: Per-Class Accuracy Metrics

Gesture	Precision	Recall	F1-Score	RT
Hello	<u>89.8%</u>	<u>94.3%</u>	<u>92.0%</u>	✓
Thank You	<u>91.0%</u>	<u>86.4%</u>	<u>88.7%</u>	✓
Yes	<u>91.6%</u>	<u>88.4%</u>	<u>90.0%</u>	✓
No	<u>87.6%</u>	<u>90.1%</u>	<u>88.8%</u>	✓
Please	<u>89.7%</u>	<u>93.2%</u>	<u>91.4%</u>	✓
Sorry	<u>93.2%</u>	<u>90.3%</u>	<u>91.8%</u>	✓
Overall	<u>90.5%</u>	<u>90.4%</u>	<u>90.4%</u>	✓

Table III. Per-class precision, recall, F1-score, and real-time capability on the held-out test set [2],[15],[16].

VI. REFERENCES

- [1] C. Lugaesi et als., “MediaPipe: A framework for building perception pipelines,” arXiv:1906.08172, 2019.
- [2] F. Zhang et als., “MediaPipe Hands: On-device real-time hand tracking,” arXiv:2006.10214, 2020.
- [3] V. Bazarevsky et als., “BlazePose: On-device real-time body pose tracking,” arXiv:2006.10204, 2020.
- [4] T. Starner J, Weaver, A. Pentland, “Real-time ASL recognition using video,” IEEE TPAMI; 20(12):1371-1375. 1998.
- [5] L. Pigou et al., “Sign Language Recognition using CNNs,” ECCVW2015;572-578.
- [6] N.C. Camgoz, et als., “Neural Sign Language Translation,” CVPR2018, 7784-7793
- [7] R. Rastgoo, K.Kiani, S. Escalera, “Sign Language Recognition: A Deep Survey,” Expert Systems with Applications; 164(113794). 2021.
- [8] R. Cui, H.Liu, C.Zhang, “Recurrent CNN for Continuous Sign Language Recognition,” CVPR2018, 7361-7369
- [9] O. Koller, J. Forster, H. Ney, “Continuous Sign Language Recognition,” CVIU.;141,108-125.2015.
- [10] P.Kumar, S.Rawat, “ISL Recognition using SVM,” IJCA;178(35):1-5.2019.
- [11] A. Sharma, R. Singh, “Real-time Hand Gesture Recognition using MediaPipe,” IJERT;11(4):215-220.2022.
- [12] K.Bantupalli, Y.Xie, “ASL Recognition using Deep Learning,” BigData2018;4896-4899.
- [13] T.Ojala, M.Pietikainen, T.Maenpaa, “Multiresolution texture with LBP,” TPAMI;24(7):971-987.2002.
- [14] Lowe, D.G., IJCV, 60(2), 91-110 , 2004 : “Distinctive Image Features from Scale-Invariant Keypoints” ,
- [15]Krizhevsky, A., Sutskever, I., Hinton, G.E. NeurIPS 2012 pp 1097-1105: “ImageNet Classification with Deep CNNs” ,
- [16]Hochreiter, S., Schmidhuber, J. Neural Networks, 9(8): 1735-1780 , 1997 : “Long Short-Term Memory” ,
- [17]Vaswani, A. et al. NeurIPS 2017 pp 5998-6008: “Attention is All You Need” ,
- [18]Shahriar, S. et al. ICCIT 2018 pp 1-6: “Real-Time ASL Recognition via Skeletal Joint Information” ,
- [19]Joze, H. R.V., Koller, O. BMVC 2019 : “MS-ASL: A Scalable Dataset for American Sign Language (ASL)” ,
- [20]Adithya, V. Rajesh, R. Procedia CS, 171, 2353-2361 , 2020 : “Deep CNN for Static Hand Gesture Recognition” ,
- [21]Mittal, A. et al. IEEE Sensors Journal, 19(16), pp7056-7063, 2019: “Modified LSTM for Continuous ISL Recognition via Leap Motion” ,